

UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”  
FACULDADE DE ENGENHARIA  
CAMPUS DE ILHA SOLTEIRA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

MAICON APARECIDO SARTIN

PROJETO E IMPLEMENTAÇÃO DE REDES NEURAIS ARTIFICIAIS EM  
DISTINTOS NÍVEIS DE ABSTRAÇÕES PARA O RECONHECIMENTO DE  
DEFICIÊNCIAS DE DIVERSOS MACRONUTRIENTES E CULTIVARES

Ilha Solteira  
2014

MAICON APARECIDO SARTIN

PROJETO E IMPLEMENTAÇÃO DE REDES NEURAIAS ARTIFICIAIS EM  
DISTINTOS NÍVEIS DE ABSTRAÇÕES PARA O RECONHECIMENTO DE  
DEFICIÊNCIAS DE DIVERSOS MACRONUTRIENTES E CULTIVARES

Tese apresentada à Faculdade de Engenharia -  
UNESP - Campus de Ilha Solteira, para obtenção do  
título de Doutor em Engenharia Elétrica.

Área de Conhecimento: Automação.

Prof. Dr. Alexandre César Rodrigues da Silva

**Orientador**

Ilha Solteira

2014

FICHA CATALOGRÁFICA

Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

S249p Sartin, Maicon Aparecido.  
Projeto e implementação de redes neurais artificiais em distintos níveis de abstrações para o reconhecimento de deficiências de diversos macronutrientes e cultivares / Maicon Aparecido Sartin. -- Ilha Solteira: [s.n.], 2014  
251 f. : il.

Tese (doutorado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira. Área de conhecimento: Automação, 2014

Orientador: Alexandre César Rodrigues da Silva  
Inclui bibliografia

1. Computação reconfigurável. 2. Fpga. 3. Agricultura de precisão.  
4. Deficiência de nutrientes. 5. Redes neurais artificiais. 6. Processamento digital de imagens.

CERTIFICADO DE APROVAÇÃO

**TÍTULO:** Projeto e implementação de redes neurais artificiais em distintos níveis de abstrações para o reconhecimento de deficiências de diversos macronutrientes e cultivares

**AUTOR:** MAICON APARECIDO SARTIN

**ORIENTADOR:** Prof. Dr. ALEXANDRE CESAR RODRIGUES DA SILVA

Aprovado como parte das exigências para obtenção do Título de DOUTOR EM ENGENHARIA ELÉTRICA, Área: AUTOMAÇÃO, pela Comissão Examinadora:

  
Prof. Dr. ALEXANDRE CESAR RODRIGUES DA SILVA  
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira

  
Prof. Dr. ALEDIR SILVEIRA PEREIRA  
Departamento de Ciência de Computação e Estatística / Instituto de Biociências, Letras e Ciências Exatas de São José do Rio Preto

  
Prof. Dr. NORIAN MARRANGHELLO  
Departamento de Ciência de Computação e Estatística / Instituto de Biociências, Letras e Ciências Exatas de São José do Rio Preto

  
Prof. Dr. TÉRCIO ALBERTO DOS SANTOS FILHO  
Departamento de Ciências Da Computação / Universidade Federal de Goiás

  
Prof. Dr. IVAN NUNES DA SILVA  
Departamento de Engenharia Elétrica / Escola de Engenharia de São Carlos-Usp

Data da realização: 01 de agosto de 2014.

A MINHA ESPOSA ELIANA E A  
MINHA FILHA YASMIN,  
DEDICO ESTE TRABALHO.

## **AGRADECIMENTOS**

Agradeço primeiramente a toda a minha família pelo exemplo de força, determinação, respeito e apoio incondicional. Gostaria de agradecer a compreensão, o carinho e a paciência da minha esposa Eliana e da minha filha Yasmin, que sempre estiveram ao meu lado para oferecer o apoio e a motivação tornando-se possível a conclusão desse trabalho. Aos meus pais, Mário e Cleusa, a minha irmã Sâmela, a minha sogra Maria e os meus cunhados, João e Edimar, pela ajuda em vários momentos difíceis em todos estes anos.

Ao meu orientador, Alexandre César Rodrigues da Silva, pelos ensinamentos, sugestões, incentivo e a atenção dedicada em todo o doutorado. Aos amigos do LPSSD, Tiago, Tércio, Welington, Alex, por não medirem esforços e atenção nos momentos necessários durante a elaboração deste trabalho. Aos amigos de outros laboratórios, Marcos Estremote, Sanches, Silvio, Francisco, Mateus, os dois Marcos Vinicius, pelo companheirismo e descontração. Aos vários colegas do DINTER, principalmente na fase de conclusão dos créditos, nos períodos de estadia em Barra do Bugres e nas intensas semanas de estudo. Aos funcionários, técnicos e professores da UNEMAT e da UNESP pela colaboração em momentos essenciais. Ao amigo Claudinei Kappes e a FMT (Fundação de Mato Grosso), pelo auxílio na área de agronomia e pela receptividade nas estações experimentais.

Por fim, gostaria de agradecer o suporte e apoio financeiro da Universidade do Estado de Mato Grosso (UNEMAT), a Universidade Estadual Paulista (UNESP), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e a Fundação de Amparo à Pesquisa do Estado de Mato Grosso (FAPEMAT).

A todas as pessoas e instituições, muito obrigado.

“As far as the propositions of mathematics refer to reality,  
they are not certain;  
and as far as they are certain, they do not refer to  
reality.”(EINSTEIN, 1954)

## RESUMO

Com o uso da agricultura de precisão procura-se melhorar o sistema de produção agrícola com o objetivo de reduzir os custos, aumentar a produtividade e minimizar a degradação do meio ambiente. Para isso, faz-se necessário o monitoramento de recursos essenciais às plantas com o intuito de reduzir o uso de insumos agrícolas. No monitoramento da folha da planta pode-se identificar doenças ou a falta de nutrientes. Neste trabalho desenvolveu-se um sistema que identifica a deficiência de nutrientes pela análise das folhas de diversas culturas. O sistema foi desenvolvido em diferentes níveis de abstrações para consolidar os resultados do sistema e facilitar o projeto em baixo nível. A principal contribuição do trabalho está no desenvolvimento de um sistema com uma rede neural artificial multicamadas em dispositivo reconfigurável, com a função de identificar a deficiência do macronutriente de Potássio pela folha da soja. O sistema faz uso de arquitetura parcialmente paralela para computação dos neurônios em ponto flutuante, com precisão de 32 bits padronizada. A aproximação da função de ativação foi investigada com diversos métodos, dois principais métodos híbridos foram desenvolvidos: HPR - Híbrido com relação entre divisão em partes lineares (PWL) e múltiplo endereçamento de entradas (RALUT), e HPC - Híbrido entre PWL e expressões booleanas simplificadas. O sistema desenvolvido em hardware foi aplicado na segmentação de imagens pelas folhas da soja e comparado ao sistema de alto nível de abstração. Nos resultados do dispositivo reconfigurável, a média do percentual de acertos na folha é de 92%, no trifólio é de 96% e no ambiente externo é de 95%. O erro médio quadrático alcançou valores em  $10^{-2}$  e o fator de qualidade entre 8,5 e 9,0. Além disso, diversas outras contribuições foram realizadas no trabalho, para tornar possível o desenvolvimento do sistema em baixo nível de abstração.

**Palavras-chave:** Computação reconfigurável. FPGA. Agricultura de precisão. Deficiência de nutrientes. Redes neurais artificiais. Processamento digital de imagens.

## ABSTRACT

The precision agriculture seeks improve the agricultural production system with aim of reduce costs, increase productivity and minimize environment degradation. Thus, the monitoring of essential resources plants is necessary for reducing the use of inputs agricultural. In the monitoring of the plant leaf can be identified disease or nutrients deficiency. This research work was made a system that identifies the nutrient deficiency by leaf of the several cultivars. The system was developed in different levels of abstractions for consolidate the results of the system and facilitate low-level design. The main contribution of the work is in the development of a multilayer artificial neural network system in reconfigurable device, with the function of identify deficiency of the Potassium macronutrient by soybean leaf. The system makes use of partially parallel architecture for computing of the neuron in floating point, with precision 32 bits standardized. The approximation of the activation function was investigated with methods distinct, two main hybrids methods were developed: HPR - Hybrid with relation between piecewise linear(PWL) and multiple addressing of inputs (RALUT), and HPC - Hybrid between PWL and the simplified booleans expressions. The system developed in hardware was applied in the images segmentation by soybean leaves and was compared to high-level system. In the results of the reconfigurable device the mean of the hit percentage by leaf is 92%, in the trefoil is 96% and in external environment is 95%. The mean square error achieved values in  $10^{-2}$  and the quality factor between 8.5 and 9.0. Furthermore, several others contributions were made in the work for make possible the development of the system in abstraction low-level.

**Keywords:** Reconfigurable computing. FPGA. Precision agriculture. Nutrient deficiency. Artificial neural network. Digital image processing.

## LISTA DE FIGURAS

Figura 1 – Modelo conceitual do sistema embarcado desenvolvido.	37
Figura 2 – Arquitetura básica de um CPLD.	45
Figura 3 – Arquitetura FPGA baseada em matriz lógica.	46
Figura 4 – Estrutura de um CLB.	47
Figura 5 – Componentes embarcados típicos em uma FPGA.	49
Figura 6 – Fluxo de desenvolvimento de um dispositivo reconfigurável (FPGA).	50
Figura 7 – Forma de representação de números racionais na base binária.	53
Figura 8 – Modelo geral neurônio artificial.	55
Figura 9 – Neurônio biológico.	55
Figura 10 – Principais funções de ativação parcialmente diferenciável: (a) Degrau; (b) Degrau Bipolar; (c) Rampa.	56
Figura 11 – Principais funções de ativação totalmente diferenciável: (a) Gaussiana; (b) Logística; (c) Tangente Hiperbólica.	56
Figura 12 – Rede neural perceptron de múltiplas camadas (MLP).	57
Figura 13 – Faixa de comprimentos de onda e o espectro visível	61
Figura 14 – Etapas do processamento digital de imagens.	62
Figura 15 – Tipos de Vizinhança: (a) vizinhança-4; (b) vizinhança diagonal; (c) vizinhança-8.	64
Figura 16 – Limiarização por histogramas: (a) Imagem original; (b) Histograma.	66
Figura 17 – Organização do projeto.	87
Figura 18 – Estrutura RTL do nível topo do projeto.	89
Figura 19 – Diagrama de tempo.	93
Figura 20 – Comparação da área utilizada ( <i>Slices</i> ) na FPGA	95

Figura 21 – Fluxo de execução no funcionamento e comparação dos resultados do sistema.	100
Figura 22 – Árvore com a hierarquia de níveis do sistema.	102
Figura 23 – Arquitetura de hardware para a rede neural artificial.	103
Figura 24 – Hardware do sistema com as três principais unidades: UC, UA e UP.	104
Figura 25 – Máquina de estados finitos do controle geral da rede neural artificial.	105
Figura 26 – Elemento de processamento do neurônio.	107
Figura 27 – Arquitetura do elemento de processamento.	108
Figura 28 – Máquina de estados para o controle das operações em ponto flutuante no neurônio.	109
Figura 29 – Resposta da função de ativação Tangente Hiperbólica com 512 pontos.	110
Figura 30 – Arquitetura destinada aos experimentos realizados com AF independente da RNA.	111
Figura 31 – Máquina de estados para o controle da função de ativação e testes iniciais do subsistema.	112
Figura 32 – Máquina de estados para o controle adequado da execução da função de ativação na E/S.	112
Figura 33 – Implementação correspondente ao 1º cenário (comportamental).	115
Figura 34 – Implementação correspondente aos cenários 2 e 3 (comportamental).	116
Figura 35 – Implementação correspondente aos cenários 4 e 5 (PWL e RALUT).	117
Figura 36 – Implementação correspondente aos cenários 6 a 9 (PWL e combinacional).	118
Figura 37 – Simulação da máquina de estados da RNA.	120
Figura 38 – Erro na simulação com IP core Havoc.	120
Figura 39 – Aproximação da AF Tangente Hiperbólica com 500 pontos em todas as implementações.	121
Figura 40 – Aproximação da AF Tangente Hiperbólica: (a) 80 pontos; (b) 15 pontos; (c) 80 pontos; (d) 15 pontos; (e) 80 pontos; (f) 15 pontos.	122

Figura 41 – Aproximação da função não linear pela RNA em todas as implementações.	123
Figura 42 – Aproximação da função não linear pela RNA: (a) 45 pontos; (b) 10 pontos; (c) 45 pontos; (d) 10 pontos; (e) 45 pontos; (f) 10 pontos.	124
Figura 43 – Comparação da área utilizada (Slices) na FPGA com a implementação da AF e RNA.	128
Figura 44 – Aproximação da AF Tangente Hiperbólica com 500 pontos em todas as implementações.	129
Figura 45 – Aproximação da AF Tangente Hiperbólica: (a) 85 pontos; (b) 15 pontos; (c) 85 pontos; (d) 15 pontos; (e) 80 pontos; (f) 15 pontos.	130
Figura 46 – Aproximação da função não linear pela RNA em todas as implementações com 128 pontos.	131
Figura 47 – Aproximação da função não linear pela RNA: (a) 45 pontos; (b) 10 pontos; (c) 45 pontos; (d) 10 pontos; (e) 40 pontos; (f) 10 pontos.	132
Figura 48 – Erro relativo dos cenários HPR para a AF: (a) 500 pontos; (b) 250 pontos.	134
Figura 49 – Erro relativo dos cenários HPC para a AF: (a) 500 pontos; (b) 250 pontos.	134
Figura 50 – Erro acumulado para todos os cenários com 500 pontos.	135
Figura 51 – Erro relativo com 128 pontos para o sistema da RNA: (a) Cenários HPR; (b) Cenários HPC.	136
Figura 52 – Erro acumulado para todos os cenários do sistema da RNA com 128 pontos.	137
Figura 53 – Utilização de área na FPGA na AF e sistema da RNA.	138
Figura 54 – Imagens originais do Mamão e Caferana: (a) Imagem (Caferana); (b) Imagem (Mamão); (c) Imagem (Caferana c/ deficiência); (d) Imagem (Mamão c/ deficiência); (e) Imagem (Mamão c/ distinta iluminação); (f) Imagem (Mamão c/ alta iluminação e fundo escuro).	141

- Figura 55 – Limiarização de imagens das folhas com aparência normal: (a) Imagem original (Mamão); (b) Imagem original (Caferana); (c) Histograma Mamão; (d) Histograma Caferana; (e) Segmentação Mamão; (f) Segmentação Caferana. 143
- Figura 56 – Limiarização de imagens das folhas com deficiência de nutrientes: (a) Imagem original (Mamão); (b) Imagem original (Caferana); (c) Histograma Mamão; (d) Histograma Caferana; (e) Segmentação Mamão; (f) Segmentação Caferana. 144
- Figura 57 – Limiarização de imagens das folhas de Mamão com deficiência de nutrientes e iluminação distinta: (a) Baixa diferença na iluminação; (b) Alta diferença na iluminação; (c) Histograma (baixa); (d) Histograma (alta); (e) Segmentação (baixa); (f) Segmentação (alta). 146
- Figura 58 – Limiarização múltipla de imagens das folhas com aparência normal: (a) Original (Mamão); (b) SMO (Mamão); (c) SMOOE (Mamão); (d) Original (Caferana); (e) SMO (Caferana); (f) SMOOE (Caferana). 149
- Figura 59 – Limiarização de imagens das folhas de Mamão com deficiência de nutrientes e iluminação distinta: (a) Original (Mamão); (b) SMO (Mamão); (c) SMOOE (Mamão); (d) Original (Caferana); (e) SMO (Caferana); (f) SMOOE (Caferana). 150
- Figura 60 – Limiarização de imagens das folhas de Mamão com deficiência de nutrientes e iluminação distinta: (a) Original com baixa diferença na iluminação; (b) SMO (baixa); (c) SMOOE (baixa); (d) Original com alta diferença na iluminação; (e) SMO (alta); (f) SMOOE (alta). 150
- Figura 61 – Organização das principais etapas de execução do sistema. 152
- Figura 62 – Sequência de execução dos métodos utilizados nas etapas e tarefas. 153
- Figura 63 – Organização geral do sistema em alto nível de abstração. 155
- Figura 64 – As imagens geradas com as seis melhores segmentações com seleção em baixa (BI) e alta intensidade (AI): (a) Desejada - BI; (b) Desejada - AI; (c) N° 1 - BI; (d) N° 1 - AI; (e) N° 2 - BI; (f) N° 2 - AI; (g) N° 5 - BI; (h) N° 5 - AI; (i) N° 6 - BI; (j) N° 6 - AI; (k) N° 9 - BI; (l) N° 9 - AI; (m) N° 13 - BI; (n) N° 13 - AI; 158

Figura 65 – Curva de aprendizagem das melhores e piores simulações com 100 épocas: (a) 6 melhores. (b) 5 piores.	159
Figura 66 – Médias das seis melhores simulações em porcentagens de acertos.	160
Figura 67 – Máscaras dos filtros de média ponderada: (a) 3x3; (b) 5x5; (c) 7x7; (d) 9x9; (e) 11x11.	162
Figura 68 – As 13 melhores simulações de imagens geradas: (a) M3-V8-E5 <sup>1</sup> ; (b) M5-SV-E11; (c) M7-SV-E11; (d) M7-V4-E11; (e) M9-SV-E11; (f) M9-V4-E11; (g) M9-V8-E11; (h) MP3-V8-E5; (i) MP9-SV-E11 <sup>2</sup> ; (j) MP9-V4-E11; (k) MP11-SV-E11; (l) MP11-V4-E11; (m) MP11-V8-E11.	164
Figura 69 – Organização das principais etapas de execução do sistema com a RNA utilizada para filtro.	172
Figura 70 – Resultados da segmentação de imagens com a RNA: (a) Imagem original sem ruído; (b) Imagem desejada (alvo); (c) Segmentação de (a) com RNA e treinamento sem ruído (Tabela 20 número 13); (d) Segmentação de (a) e treinamento com ruído; (e) Imagem original com ruído de 5%; (f) Segmentação de (e) com RNA; (g) Ruído de 10%; (h) Segmentação de (g) com RNA; (i) Ruído de 15%; (j) Segmentação de (i) com RNA; (k) Ruído de 20%; (l) Segmentação de (k) com RNA; (m) Ruído de 25%; (n) Segmentação de (m) com RNA; (o) Ruído de 30%; (p) Segmentação de (o) com RNA.	178
Figura 71 – Imagens sem sintomas de deficiência de fósforo (P) e com plano de fundo.	189
Figura 72 – Imagens sem sintomas de deficiência de fósforo (P) e sem plano de fundo.	189
Figura 73 – Imagens com sintomas de deficiência de fósforo (P) e com plano de fundo.	190
Figura 74 – Imagens com sintomas de deficiência de fósforo (P) e sem plano de fundo.	191

Figura 75 – Imagens sem deficiência do macronutriente K, com baixa luminosidade e com plano de fundo. Em cada conjunto contém quatro tipos de imagens em sequência: original, desejada, segmentação pela RNA (SRNA) e segmentação pelo método de Otsu (SMO).	195
Figura 76 – Imagens sem deficiência do macronutriente K, com baixa luminosidade e sem plano de fundo. Em cada conjunto contém quatro tipos de imagens em sequência: original, desejada, segmentação pela RNA (SRNA) e segmentação pelo método de Otsu (SMO).	196
Figura 77 – Imagens sem deficiência do macronutriente K, com alta luminosidade e com plano de fundo. Em cada conjunto contém quatro tipos de imagens em sequência: original, desejada, SRNA e SMO.	198
Figura 78 – Imagens sem deficiência do macronutriente K, com alta luminosidade e sem plano de fundo. Em cada conjunto contém quatro tipos de imagens em sequência: original, desejada, SRNA e SMO.	198
Figura 79 – Imagens com deficiência do macronutriente K, com distintos planos de fundo e luminosidade. Em cada conjunto contém quatro tipos de imagens em sequência: original, desejada, SRN e SMO.	199
Figura 80 – Imagens com deficiência do macronutriente K, com distintos planos de fundo e luminosidade. Em cada conjunto contém quatro tipos de imagens em sequência: original, desejada, SRN e SMO.	200
Figura 81 – Imagens com deficiência do macronutriente K, com distintos planos de fundo e luminosidade. Em cada conjunto contém quatro tipos de imagens em sequência: original, desejada, SRN e SMO.	200
Figura 82 – Organização geral do sistema em baixo nível de abstração.	205
Figura 83 – Máquina de estados finitos do controle geral da rede neural artificial.	207
Figura 84 – Máquina de estados finitos do estado Load.	208
Figura 85 – Resultados visuais das imagens das folhas independentes adquiridas em laboratório sem deficiência de K, com agrupamentos de três imagens: original, segmentação com RNA no Matlab e segmentação com RNA no FPGA. As imagens não têm deficiência de K, a segmentação no editor gráfico é uma imagem totalmente branca.	211

- Figura 86 – Resultados visuais das imagens das folhas independentes adquiridas em laboratório com deficiência de K, com agrupamentos de quatro imagens: original, segmentação com editor gráfico, segmentação com RNA no Matlab e segmentação com RNA no FPGA. 212
- Figura 87 – Resultados visuais das imagens das folhas independentes adquiridas em laboratório sem deficiência de K, com agrupamentos de quatro imagens: original, segmentação com editor gráfico, segmentação com RNA no Matlab e segmentação com RNA no FPGA. 213
- Figura 88 – Resultados visuais das imagens dos trifólios adquiridas em laboratório sem deficiência de K, com agrupamentos de três imagens: original, segmentação com RNA no Matlab e segmentação com RNA no FPGA. As imagens não têm deficiência de K, a segmentação no editor gráfico é uma imagem totalmente branca. 214
- Figura 89 – Resultados visuais das imagens dos trifólios adquiridas em laboratório sem deficiência de K, com agrupamentos de quatro imagens: original, segmentação com editor gráfico, segmentação com RNA no Matlab e segmentação com RNA no FPGA. Apenas o primeiro agrupamento contém três imagens, porque não tem deficiência de K e a imagem da segmentação com editor gráfico foi retirada. 215
- Figura 90 – Resultados visuais das imagens dos trifólios adquiridas em laboratório sem deficiência de K, com agrupamentos de quatro imagens: original, segmentação com editor gráfico, segmentação com RNA no Matlab e segmentação com RNA no FPGA. 216
- Figura 91 – Resultados visuais das imagens externas com e sem sintomas de deficiência de K na soja, com agrupamentos de três imagens: original, segmentação com RNA no Matlab e segmentação com RNA no FPGA. As imagens não têm deficiência de K, a segmentação no editor gráfico é uma imagem totalmente branca. 217

- Figura 92 – Resultados visuais das imagens externas com e sem sintomas de deficiência de K na soja, com agrupamentos de quatro imagens: original, segmentação com editor gráfico, segmentação com RNA no Matlab e segmentação com RNA no FPGA. Apenas os dois últimos agrupamentos contém as quatro imagens, os outros foram retiradas a segmentação com editor gráfico por ser uma imagem sem deficiência, totalmente em branco. 218
- Figura 93 – Resultados visuais das imagens externas com e sem deficiência de K, com agrupamentos de duas imagens: Original e segmentação com RNA no Matlab. 219
- Figura 94 – Resultados visuais das imagens externas com e sem deficiência de K, com agrupamentos de duas imagens: Original e segmentação com RNA no Matlab. 220
- Figura 95 – Resultados visuais das imagens externas com e sem deficiência de K, com agrupamentos de duas imagens: Original e segmentação com RNA no Matlab. 220
- Figura 96 – Avaliação dos erros absolutos nas implementações da AF para o primeiro grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6. 235
- Figura 97 – Avaliação dos erros relativos nas implementações da AF para o primeiro grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6. 236
- Figura 98 – Avaliação dos erros médios quadráticos nas implementações da AF para o primeiro grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6. 237
- Figura 99 – Avaliação dos erros absolutos nas implementações da RNA para o primeiro grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6. 238
- Figura 100 – Avaliação dos erros relativos nas implementações da RNA para o primeiro grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6. 239

- Figura 101 – Avaliação dos erros médios quadráticos nas implementações da RNA para o primeiro grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6. 240
- Figura 102 – Avaliação dos erros absolutos nas implementações da AF para o segundo grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6. 241
- Figura 103 – Avaliação dos erros absolutos nas implementações da AF para o segundo grupo de cenários: (a) Cenário 7; (b) Cenário 8; (c) Cenário 9. 242
- Figura 104 – Avaliação dos erros relativos nas implementações da AF para o segundo grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6. 243
- Figura 105 – Avaliação dos erros relativos nas implementações da AF para o segundo grupo de cenários: (a) Cenário 7; (b) Cenário 8; (c) Cenário 9. 244
- Figura 106 – Avaliação dos erros médios quadráticos nas implementações da AF para o segundo grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6. 245
- Figura 107 – Avaliação dos erros médios quadráticos nas implementações da AF para o segundo grupo de cenários: (a) Cenário 7; (b) Cenário 8; (c) Cenário 9. 246
- Figura 108 – Avaliação dos erros absolutos nas implementações da RNA para o segundo grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6. 247
- Figura 109 – Avaliação dos erros absolutos nas implementações da RNA para o segundo grupo de cenários: (a) Cenário 7; (b) Cenário 8; (c) Cenário 9. 248
- Figura 110 – Avaliação dos erros relativos nas implementações da RNA para o segundo grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6. 249
- Figura 111 – Avaliação dos erros relativos nas implementações da RNA para o segundo grupo de cenários: (a) Cenário 7; (b) Cenário 8; (c) Cenário 9. 250
- Figura 112 – Avaliação dos erros médios quadráticos nas implementações da RNA para o segundo grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6. 251

- Figura 113 Avaliação dos erros médios quadráticos nas implementações da RNA para o segundo grupo de cenários: (a) Cenário 7; (b) Cenário 8; (c) Cenário 9. 252
- Figura 114 Relatório de análise foliar da soja. 253

## LISTA DE TABELAS

Tabela 1 – Comparação de características em distintos fabricantes de FPGA.	48
Tabela 2 – Formatos de representações em ponto flutuante.	53
Tabela 3 – Comparação entre os trabalhos na implementação da RNA em hardware	75
Tabela 4 – Comparação entre os trabalhos na implementação da AF em hardware	79
Tabela 5 – Cenários analisados.	92
Tabela 6 – Cenários analisados - Spartan 3-E.	94
Tabela 7 – Cenários analisados - Spartan 3-E.	94
Tabela 8 – Cenários analisados - Cyclone II.	94
Tabela 9 – Grupo 1 - cenários analisados para AF Tangente Hiperbólica.	114
Tabela 10 – Grupo 2 - cenários analisados para AF Tangente Hiperbólica.	114
Tabela 11 – Número de elementos nos intervalos da função de ativação Tangente Hiperbólica.	117
Tabela 12 – Erros na implementação do hardware para função de ativação.	126
Tabela 13 – Erros na implementação do hardware para o sistema da RNA.	126
Tabela 14 – Utilização de área na FPGA para a função de ativação.	127
Tabela 15 – Utilização de área na FPGA para o sistema da RNA.	127
Tabela 16 – Erros absolutos e relativos encontrados no hardware para a aproximação da AF.	133
Tabela 17 – Erros absolutos e relativos encontrados na implementação do hardware para o sistema da RNA.	133
Tabela 18 – Área utilizada na FPGA para AF TH.	138
Tabela 19 – Utilização de área na FPGA para o sistema da RNA	138

Tabela 20 – Resultados da primeira etapa e primeira tarefa com 32 simulações. As colunas estão organizadas com o Tipo de Imagem (TI), a Quantidade de Neurônios na Camada Escondida (QNCE), a Taxa de Aprendizagem (TA), a Constante Momento (CM), a Porcentagem de Acertos (PA), o tempo de execução em segundos e o Erro Médio Quadrático (EMQ).	157
Tabela 21 – Resultados da segunda tarefa da primeira etapa com as seis melhores simulações e alteração da quantidade de neurônios na camada escondida.	160
Tabela 22 – Combinação dos filtros aplicados na preparação da imagem desejada para a segmentação, com dois tipos de filtros (média e média ponderada) para ruído impulsivo, máscara ou janela de vários tamanhos ou ordens, a vizinhança com 3 tipos, Sem-Vizinhança (SV), Vizinhança-4 (V4) e Vizinhança-8 (V8) e a erosão em três tamanhos de máscara e Sem Erosão(SE).	163
Tabela 23 – Resultados da segunda tarefa da segunda etapa com simulações de 1 a 13, a segunda coluna corresponde ao Tipo de Filtro (TF) utilizado.	165
Tabela 24 – Resultados da terceira tarefa da segunda etapa com simulações de 1 a 8, a segunda coluna corresponde ao Tipo da Imagem Desejada e a Original (TIDO) utilizada.	165
Tabela 25 – Resultados da primeira tarefa da terceira etapa com 16 simulações e RNA de 4 entradas.	167
Tabela 26 – Resultados da segunda tarefa da terceira etapa com 16 simulações e RNA de 3 entradas.	168
Tabela 27 – Resultados da terceira tarefa da terceira etapa com 16 simulações e RNA de 2 entradas.	169
Tabela 28 – Resultados da quarta tarefa da terceira etapa com 16 simulações e RNA de 1 entrada.	169
Tabela 29 – Resultados do filtro da imagem RGB para o componente R.	173
Tabela 30 – Resultados do filtro da imagem RGB para o componente G.	174
Tabela 31 – Resultados do filtro da imagem RGB para o componente B.	174

Tabela 32 – Resultados do filtro da imagem RGB para os componentes RGB.	174
Tabela 33 – Média dos resultados do filtro das imagens.	174
Tabela 34 – Resultados da segmentação da imagem RGB para o componente R.	175
Tabela 35 – Resultados da segmentação da imagem RGB para o componente G.	175
Tabela 36 – Resultados da segmentação da imagem RGB para o componente B.	175
Tabela 37 – Resultados da segmentação da imagem RGB para o componente RGB.	176
Tabela 38 – Média dos resultados da segmentação da imagem RGB.	176
Tabela 39 – Ensaio com 9 tipos de tratamentos e média entre 4 amostras de 4 blocos diferentes.	181
Tabela 40 – Primeiro estágio para encontrar valores adequados de TA e CM.	185
Tabela 41 – Segundo estágio com alteração na quantidade de neurônios na camada escondida.	186
Tabela 42 – Terceiro estágio com alteração na porcentagem de imagem utilizada para o treinamento.	187
Tabela 43 – Quarto estágio, comparação dos resultados com todas as imagens entre a RNA e o método de Otsu.	187
Tabela 44 – Ensaio com 20 tratamentos e média entre 4 amostras de 4 blocos diferentes.	192
Tabela 45 – Comparação da RNA e o método de Otsu, nos resultados com as 10 primeiras imagens sem deficiência do macronutriente K.	194
Tabela 46 – Comparação da RNA e o método de Otsu, nos resultados com as imagens (11 a 18) sem deficiência do macronutriente K.	197
Tabela 47 – Comparação da RNA e o método de Otsu, nos resultados com todas as imagens com deficiência do macronutriente K.	199
Tabela 48 – Análise foliar da soja de macronutrientes nos três blocos.	203
Tabela 49 – Utilização de área na FPGA para o sistema da RNA.	209
Tabela 50 – Resultados encontrados nas imagens de folhas independentes adquiridas em laboratório para implementação em níveis de abstrações, alto (M-Matlab) e baixo (F-FPGA).	210

Tabela 51 – Resultados encontrados nas imagens dos trifólios adquiridas em laboratório para implementação em níveis de abstrações, alto (M-Matlab) e baixo (F-FPGA).	211
Tabela 52 – Resultados encontrados nas imagens externas em níveis de abstrações, alto (M-Matlab) e baixo (F-FPGA).	216
Tabela 53 – Média dos Resultados com os três tipos de imagens em níveis de abstrações, alto (M-Matlab) e baixo (F-FPGA).	216

## LISTA DE ABREVIATURAS E SIGLAS

AF	Activation Function
ALUT	Adaptive Look-Up Table
AP	Agricultura de Precisão
ASIC	Application-Specific Integrated Circuit
BRAM	Block Random-Access Memory
CAD	Computer-Aided Design
CCD	Charge-Coupled Device
CE	Chip Enable
CI	Circuito Integrado
CLB	Configurable Logic Block
CM	Constante de Momento
CMOS	Complementary Metal-Oxide-Semiconductor
CMYK	Cyan Magenta Yellow blackK
CORDIC	COordinate Rotation DIgital Computer
CPLD	Complex Programmable Logic Devices
CS	Ciclo Simples
DA	Direct Approximation
DCM	Digital Clock Manager
DFS	Digital Frequency Synthesizer
DLL	Delay-Locked Loop
DSP	Digital Signal Processor
ECU	Electronic Control Unit
EMQ	Erro Médio Quadrático
EPP	Entradas por Parâmetros probabilísticos
FIFO	First In, First Out
FLP	Floating Point
FNL	Função Não Linear
FPGA	Field Programmable Gate Arrays
FPU	Floating Point Unit
FQ	Fator de Qualidade
FXP	Fixed Point
GCC	GNU Compiler Collection
GPS	Global Positioning System
GTP	Low-Power Gigabit Transceivers

HDL	Hardware Description Language
HSI	Hue Saturation Intensity
HDMI	High-Definition Multimedia Interface
HPC	Híbrido - PWL com Combinacional
HPR	Híbrido - PWL com RALUT
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
IP	Intellectual Property
LAI	Leaf Area Index
LE	Logic Elements
LM	Levenberg-Marquardt
LSB	Least Significant Bit
LUT	Look-Up Table
MAC	Multiply and ACcumulate
MCT	Manage Clock Tile
MLP	MultiLayer Perceptron
MSB	Most Significant Bit
MSI	Medium Scale Integration
MSE	Medium Square Error
NaN	Not a Number
NPI	Número de Píxeis da Imagem
NTSC	National Television Standards Committee
OSPF	Original Sem Plano de Fundo
PA	Porcentagem de Acertos
PAL	Programmable Array Logic
PCA	Principal Component Analysis
PDI	Processamento Digital de Imagens
PE	Processing Element
PHM	Percentage of Hit-or-Miss
PIB	Produto Interno Bruto
PLA	Programmable Logic Array
PLAN	Piecewise Linear Approximation of a Nonlinear Function
PLD	Programmable Logic Devices
PLL	Phase Locked Level
POTB	Power-Of-Two-Based
PS	Phase Shift
PSNR	Peak Signal-to-Noise Ratio
PWL	PieceWise Linear
QI	Quality Index

QNCE	Quantidade de Neurônios na Camada Escondida
RALUT	Range Addressable Look-Up Table
RAM	Random-Access Memory
RBPNN	Radial Basis Probabilistic Neural Network
RFS	RNA-Filtro-Segmentação
RSFS	RNA-SemFiltro-Segmentação
RGB	Red Green Blue
RNA	Rede Neural Artificial
RNN	Rede Neural Natural
ROM	Read-Only Memory
RPROP	Resilient Backpropagation
RTL	Register-Transfer Level
SDC	Synopsys Design Compiler
SDRAM	Synchronous Dynamic Random-Access Memory
SE	Sem-Erosão
SIG	Sistema de Informação Geográfica
SM	Switch Matrix
SMO	Segmentação pelo Método de Otsu
SMOOE	Segmentação pelo Método de Otsu com Operador de Erosão
SoC	System on Chip
SOP	Sum-Of-Product
SP	Saídas Parciais
SPLD	Simple Programmable Logic Devices
SRAM	Static Random-Access Memory
SRNA	Segmentação pela RNA
SV	Sem-Vizinhança
SVD	Singular Value Decomposition
TA	Taxa de Aprendizagem
TH	Tangent Hyperbolic
TI	Tecnologias da Informação
TI	Tipo de Imagem
TSMC	Taiwan Semiconductor Manufacturing Company
UA	Unidade de Armazenamento
UC	Unidade de Controle
UCP	Unidade Central de Processamento
UP	Unidade de Processamento
USB	Universal Serial Bus
VANT	Veículo Aéreo Não Tripulado
Verilog	Verifying Logic

VGA Video Graphics Array  
VHDL Very High Speed Integrated Circuits Hardware Description Language  
YIQ Y (perceived luminance), I, Q (color/luminance information)

## LISTA DE SÍMBOLOS

$\eta_j^k$	Símbolo para Computação dos Dados no Neurônio $j$ da Camada $k$
$G(\eta_j^k)_j^k$	Símbolo para Função de Ativação da RNA no Neurônio $j$ da Camada $k$
$Y_j^k$	Saída Parcial no Neurônio $j$ da Camada $k$
$\hat{Y}_j^{k_n}$	Saída Desejada da RNA no Neurônio $j$ da Última Camada $k_n$
$W_{qj}^{(m+1)k_n}$	Símbolo da Atualização da Matriz de Pesos da RNA no Neurônio $j$ e Entrada $i$ da Última Camada $k_n$ com a Época Seguinte
$\sigma$	Parâmetro de Variação para atualização da matriz de pesos da RNA
$\alpha$	Taxa de Treinamento da RNA
$\theta$	Derivada da Função Sigmoidal
$\delta$	Parâmetro Parcial para Aquisição do Erro nas Camadas Intermediárias
$E_j^{k_n-1}$	Símbolo para Aquisição do Erro nas Camadas Intermediárias (Entrada e Escondida) da RNA
$f(x, y)$	Função para Representar uma Imagem em Coordenadas Cartesianas $x$ e $y$
$i(x, y)$	Símbolo para Representar a Iluminância Incidente em uma Imagem
$r(x, y)$	Símbolo para Representar a Refletância Emitida pelos Objetos e Cena da Imagem
$g(x, y)$	Símbolo para Representar a Transformação em uma Imagem
$Th(x)$	Função de Ativação Tangente Hiperbólica
$E_{abs}$	Símbolo para o Módulo do Erro Absoluto
$E_{rel}$	Símbolo para o Erro Relativo
$E_{quad}$	Símbolo para o Erro Quadrático
$\lambda$	Símbolo que representa a inclinação da tangente hiperbólica
$PO_k$	Probabilidade de Ocorrência
$S_a$	Soma Acumulada da Imagem
$M_g$	Intensidade Média Global da imagem
$I_k$	Intensidade Acumulada
$P1_k$	Soma Cumulativa das Probabilidades da Classe 1
$P2_k$	Soma Cumulativa das Probabilidades da Classe 2
$\sigma_g$	Variância Global
$\sigma_c$	Variância entre as Classes
$MS_c$	Medida de Separabilidade entre as Classes
$E_{truncMax}$	Limite e Truncamento do Erro
$A \ominus B$	Operador de Erosão
$f''(x, y)$	Normalização da Imagem $f(x, y)$

$ft(x, y)$	Aplicação de Filtro Linear na Imagem $f(x, y)$
$\sigma_Y$	Covariância da imagem de saída da RNA
$\sigma_D$	Covariância da imagem desejada ou alvo da RNA

## SUMÁRIO

1	INTRODUÇÃO	33
1.1	Justificativa	37
1.2	Objetivo	39
2	REFERENCIAL TEÓRICO	42
2.1	Computação Reconfigurável	42
2.1.1	<i>FPGA</i>	44
2.1.1.1	<i>Arquitetura de FPGAs</i>	45
2.1.1.2	<i>Recursos Dedicados</i>	48
2.1.2	<i>Projeto de Hardware Reconfigurável</i>	49
2.2	O Padrão IEEE Std 754	52
2.3	Redes Neurais Artificiais	54
2.3.1	<i>Neurônio Artificial</i>	55
2.3.2	<i>Arquitetura</i>	57
2.3.3	<i>Treinamento</i>	57
2.4	Processamento Digital de Imagens	61
2.4.1	<i>Etapas do PDI</i>	62
2.4.2	<i>Características, Vizinhança e Conectividade</i>	63
2.4.3	<i>Histogramas</i>	65
2.4.4	<i>Método de Otsu</i>	67
3	MÉTODOS E FERRAMENTAS DE APOIO	70
3.1	Redes Neurais Artificiais em Hardware	70

3.1.1	<i>Arquitetura e Computação dos Dados no Neurônio</i>	71
3.1.2	<i>Função de Ativação em Hardware</i>	74
3.2	Métodos Computacionais para Agricultura de Precisão	80
3.2.1	<i>Monitoramento</i>	80
3.2.2	<i>Identificação, Reconhecimento e Classificação de Padrões</i>	81
4	RNA EM HARDWARE PARA FUNÇÃO XNOR	86
4.1	Estrutura Geral	88
4.2	Cenários	89
4.3	Resultados	92
4.4	Conclusão deste Capítulo	97
5	RNA MLP NA APROXIMAÇÃO DE FUNÇÃO NÃO LINEAR	98
5.1	Arquitetura do Hardware	101
5.1.1	<i>Unidade de Controle</i>	104
5.1.2	<i>Unidade de Armazenamento</i>	106
5.1.3	<i>Unidade de Processamento</i>	106
5.1.3.1	<i>Elemento de Processamento</i>	107
5.1.3.2	<i>Função de Ativação</i>	109
5.2	Métodos Propostos na Aproximação da Tangente Hiperbólica	110
5.2.1	<i>Arquitetura de Hardware para Função de Ativação</i>	110
5.2.2	<i>Cenários e Intervalos</i>	112
5.2.3	<i>Mapeamento dos Métodos</i>	115
5.3	Resultados	119
5.3.1	<i>Cenários do Grupo 1</i>	121
5.3.1.1	<i>Aproximação</i>	121

5.3.1.2	<i>Análise de Erros e Área</i>	125
5.3.1.3	<i>Área Utilizada na FPGA</i>	126
<b>5.3.2</b>	<b><i>Cenários do Grupo 2</i></b>	<b>127</b>
5.3.2.1	<i>Aproximação</i>	128
5.3.2.2	<i>Análise de Erros</i>	129
5.3.2.3	<i>Área Utilizada na FPGA</i>	136
<b>5.4</b>	<b>Conclusão deste Capítulo</b>	<b>137</b>
<b>6</b>	<b>MÉTODOS PARA SEGMENTAÇÃO DE IMAGENS</b>	<b>140</b>
<b>6.1</b>	<b>Limiarização Global por Histograma</b>	<b>140</b>
<b>6.2</b>	<b>Limiarização Múltipla pelo Método de Otsu</b>	<b>147</b>
<b>6.3</b>	<b>Segmentação com Redes Neurais Artificiais</b>	<b>151</b>
<b>6.3.1</b>	<b><i>Primeira Etapa - Avaliação de Parâmetros</i></b>	<b>155</b>
<b>6.3.2</b>	<b><i>Segunda Etapa - Aplicação de Filtros</i></b>	<b>161</b>
<b>6.3.3</b>	<b><i>Terceira Etapa - Alteração de Entradas</i></b>	<b>166</b>
<b>6.3.4</b>	<b><i>Quarta Etapa - Análise de Ruídos</i></b>	<b>170</b>
<b>6.4</b>	<b>Conclusão deste Capítulo</b>	<b>177</b>
<b>7</b>	<b>ANÁLISE NA CULTURA DO ALGODÃO</b>	<b>180</b>
<b>7.1</b>	<b>Experimento 1: Análise de Deficiência do Fósforo</b>	<b>180</b>
<b>7.1.1</b>	<b><i>Segmentação com RNA e deficiência de fósforo (P)</i></b>	<b>182</b>
7.1.1.1	<i>Parâmetros de Qualidade</i>	183
7.1.1.2	<i>Análise na Qualidade das Segmentações</i>	185
7.1.1.3	<i>Análise Visual da Segmentação de imagens</i>	187
<b>7.2</b>	<b>Experimento 2: Análise de Deficiência do Potássio</b>	<b>192</b>
<b>7.2.1</b>	<b><i>Segmentação com RNA e deficiência de potássio (K)</i></b>	<b>193</b>

7.2.1.1	<i>Resultados</i>	193
7.3	Conclusão deste Capítulo	201
8	SISTEMA RNA PARA CULTURA DA SOJA	202
8.1	Análise de Deficiência na Soja	202
8.2	Implementação em Diferentes Níveis de Abstrações	204
8.2.1	<i>Arquitetura do Sistema</i>	204
8.2.1.1	<i>Unidade de Controle</i>	206
8.3	Resultados	208
8.3.1	<i>Utilização de área na FPGA</i>	208
8.3.2	<i>Resultados Visuais e Parâmetros de Qualidade</i>	209
8.4	Conclusão deste Capítulo	221
9	CONCLUSÃO E TRABALHOS FUTUROS	222
9.1	Sugestões para Trabalhos Futuros	224
	REFERÊNCIAS	225
	APÊNDICE A - RESULTADOS ADJACENTES DE ERROS	234
	ANEXO A - RELATÓRIO DE ANÁLISE FOLIAR DA SOJA	253

## 1 INTRODUÇÃO

O Brasil dispõe de áreas agricultáveis em torno de 180 milhões de hectares do total de 880 milhões de hectares de seu território e mais de 1/3 da balança comercial brasileira está relacionado ao agronegócio (CRUVINEL, 2009). O PIB (Produto Interno Bruto) nacional está diretamente relacionado a três setores da economia, a agropecuária, a indústria e os serviços. A produção agrícola (grãos) é um dos principais responsáveis pelo setor da agropecuária e a crescente produção nacional nas últimas safras, confirmam o país como um dos maiores produtores do mundo. Atualmente, a região Centro-Oeste firmou-se como a principal produtora de grãos e o Estado de Mato Grosso o maior produtor do país (COMPANHIA NACIONAL DE ABASTECIMENTO - CONAB, 2013). Para manter o crescimento e a competitividade no setor deve-se melhorar o gerenciamento do sistema de produção com investimentos em novas tecnologias pela sua relevância na economia do estado e do país.

O crescimento da produção agrícola tem-se beneficiado com o avanço dos mecanismos de automação de processos e de gerenciamento administrativo com o uso de tecnologias computacionais, comunicação, instrumentação e controle, proporcionando uma melhor coordenação de todas as etapas no processo de produção agrícola (MELLO; CAIMI, 2008). Dependendo do tipo de cultura de plantio o objetivo é diferente para um maior retorno financeiro. Por exemplo, a cultura de grãos visa principalmente a maior produção total ou minimização de custo, diferente da cultura de uva cujo objetivo principal está na qualidade do produto.

A Agricultura de Precisão (AP) visa melhorar o sistema de produção agrícola no uso de tecnologias recentes, com o intuito de aumentar a produtividade, reduzir os custos e minimizar a degradação do meio ambiente. Para um gerenciamento eficiente da colheita é necessário o monitoramento adequado dos recursos essenciais as plantas, sem deixar ocorrer a falta ou sobrecarga de nutrientes. Isso é feito por meio de técnicas que inferem as variações espaciais e temporais ocorridas, e determinam a inserção dos recursos necessários em cada talhão ou área de plantio, além de reduzir o uso de insumos agrícolas em todo o sistema de produção, como fertilizantes e pesticidas.

Diversas soluções tecnológicas podem ser empregadas dependendo da cultura a ser

explorada. Na produção de grãos, principalmente soja e milho, existe a necessidade de monitoramento e obtenção de dados periódicos da colheita e dos recursos naturais, para facilitar a tomada de decisão e melhorar a qualidade do sistema de produção. O uso de TI (Tecnologias da Informação) possibilita realizar diversas tarefas para a AP, como o sensoriamento remoto para a obtenção de dados em tempo real, arquiteturas computacionais no processamento analógico e digital, o mapeamento de rotas e áreas com Sistema de Posicionamento Global (GPS), o controle automático do maquinário, a inserção de implementos agrícolas de forma eficiente para determinada área, Sistemas de Informação Geográfico (SIG) na administração de recursos, entre outros.

A construção de sistemas automáticos com tecnologias embarcadas podem operar de modo combinado com outros equipamentos do manejo agrícola (Tratores, semeadoras, colheitadeiras), e a dificuldade aumenta com as diferenças de tecnologias de construção e incompatibilidade de interfaces de comunicação (MELLO; CAIMI, 2008). Existe grande necessidade do desenvolvimento nacional de tecnologias embarcadas com baixo custo, para se obter resultados mais expressivos em todo o sistema de produção, e contribuir com o desenvolvimento agropecuário e industrial do país. A utilização de comunicação padronizada é essencial no desenvolvimento de tecnologias embarcadas para a AP, por facilitar o reuso de tecnologia e a expansão do sistema, o uso de padrões IEEE (*Institute of Electrical and Electronics Engineers*) é um atrativo para disponibilizar tais vantagens.

Os sistemas embarcados estão presentes em vários dispositivos do nosso cotidiano, desde um simples portão eletrônico até equipamentos com as tecnologias mais recentes, como as TVs, máquinas fotográficas, geladeiras, máquinas de lavar, entre outros. Esses sistemas são definidos por terem todos recursos de hardware presentes no computador em apenas um chip (SoC – *System on Chip*) destinado a uma aplicação alvo, independente da utilização de software ou hardware embarcado. Os sistemas embarcados ou específicos contêm um hardware organizado conforme a sua aplicação, e o software é diretamente relacionado ao hardware. Diferente dos sistemas de propósito geral, em que o hardware cobre uma ampla abrangência de software e aplicações. Com o uso de dispositivo reconfigurável em FPGA (*Field Programmable Gate Arrays*), os sistemas embarcados tentam unir as vantagens no desenvolvimento de sistemas em software e hardware, como a flexibilidade, reuso, alto desempenho e menor tempo de projeto.

A utilização de dispositivos reconfiguráveis (FPGA) tem aumentado, principalmente por dispor de desempenho associado a um alto nível de paralelismo, flexibilidade na descrição por HDL (*Hardware Description Language*), reuso por IP Cores (*Intellectual Property*) e barramentos padronizados. Uma das principais características desses dispositivos é a facilidade na atualização através da reconfiguração do sistema tornando-se mais dinâmico, inclusive para o desenvolvimento de novos projetos no mesmo dispositivo, ao contrário da construção de hardware em ASIC (*Application-Specific Integrated Circuit*)

que necessita de máscaras e *layouts*.

O dispositivo reconfigurável (FPGA) é empregado principalmente em aceleradores por hardware, em diversas aplicações como processamento de sinais e imagens (RAKVIC et al., 2009), compressão de dados (RIGLER; BISHOP; KENNINGS, 2007), criptografia (CHRISTY; KARTHIGAİKUMAR, 2012), bioinformática (GU; VANCOURT; HERBORDT, 2008).

Para facilitar e agilizar o desenvolvimento de projetos, as atuais ferramentas CAD (*Computer-Aided Design*) incorporam em seus ambientes as linguagens para a descrição de hardware em alto nível de abstração, e automatizam etapas do processo de desenvolvimento do sistema. Tais características têm permitido o desenvolvimento de projetos para aplicação de RNA (Rede Neural Artificial). O projetista do sistema pode definir detalhes para melhorar alguns requisitos como desempenho, a área a ser utilizada, a portabilidade e a potência consumida, conforme a necessidade da aplicação alvo. Por exemplo, no caso de uma RNA, o tamanho da rede e a precisão necessária pode comprometer a área do chip trazendo uma maior complexidade ao sistema.

A utilização de RNA é popular em áreas relacionadas a engenharia e a computação, principalmente quando não existem modelos matemáticos para a resolução de um determinado problema, devido a sua complexidade ou não linearidade, e na necessidade de aplicações em tempo real (SILVA; SPATTI; FLAUZINO, 2010).

A organização de uma RNA é bem definida em camadas, neurônios, pesos e Função de Ativação (AF). A capacidade de generalização, adaptação, reconhecimento e a classificação de padrões são características que diferenciam as RNA das demais metodologias. A RNA pode ser facilmente associadas com outras técnicas como processamento de imagem e sinais, lógica nebulosa, algoritmos genéticos, entre outros. Trata-se de uma estrutura intensamente paralela e adequada para implementação em FPGA (FERREIRA; BARROS, 2010; ALI; MOHAMMED, 2010), cuja principal característica é o paralelismo de grão pequeno. Outra aplicação que se beneficia do paralelismo na FPGA é o Processamento Digital de Imagens (PDI), principalmente pelo desempenho e a portabilidade.

Com a facilidade na aquisição de sensores de imagem, as técnicas relacionadas ao PDI têm aumentado consideravelmente em diversas áreas de aplicação, por exemplo, no monitoramento e na detecção de objetos em imagens e vídeos. Na AP existe um grande crescimento no uso de processamento de imagens para o monitoramento da colheita, principalmente na identificação de doenças (CUI et al., 2010), pragas (BURGOS-ARTIZZU et al., 2011) e deficiência de nutrientes (LIU; PATTEY, 2010).

A utilização da foliar da planta para o monitoramento da colheita é feita por ser a parte da planta em que ocorrem os primeiros sintomas de doenças e de falta ou excesso de nutrientes, com monitoramento não intrusivo. Diversas características relevantes podem ser destacadas como: o dispositivo de aquisição de imagens, os métodos utilizados

na identificação e em várias etapas do PDI, o ambiente de aquisição (real ou casa de vegetação), a cultura de plantio, o tipo de colheita, entre outros.

A possibilidade de identificar o estado nutricional de uma planta por meio de imagens provê informações essenciais ao produtor para tomar uma decisão e recuperar uma colheita que poderia ser mal sucedida. Na AP existe uma relação de domínio não linear entre a leitura de determinados componentes de uma imagem da foliar e o seu teor de clorofila (VOLLMANN et al., 2011), sustentando o emprego de RNA. A união entre o processamento de imagem e RNA tem grande aceitação entre os pesquisadores em diversas áreas da engenharia. Na agricultura de precisão o uso de técnicas baseadas em processamento de imagens contém inúmeras vantagens, como:

- Baixo custo na aquisição de equipamentos para captura de imagens;
- Monitoramento mais ágil e com maior periodicidade;
- Supervisionamento da colheita de forma não destrutiva;
- Gerenciamento adequado de fertilizantes e pesticidas.

Neste trabalho foi desenvolvido um sistema cujo modelamento foi realizado em diferentes níveis de abstrações, com o intuito da aplicação do método de inteligência computacional com redes neurais artificiais, do tipo Perceptron Multicamadas (MLP - *MultiLayer Perceptron*), para a segmentação da imagem. O sistema em alto nível de abstração efetua as análises necessárias e garante resultados promissores pela análise de erro. Depois da validação do sistema em alto nível, realiza-se a transcrição para linguagem de descrição de hardware VHDL em dispositivo reconfigurável (FPGA). O objetivo do sistema é identificar a deficiência do macronutriente de potássio pela folha da soja utilizando RNA.

A implementação das funcionalidades do sistema em linguagem de alto nível de abstração é necessária para viabilizar a conversão para HDL. O sistema em baixo nível de abstração é composto por módulos de pré-processamento, pós-processamento e da RNA, conforme apresentado na Figura 1.

O pré-processamento realiza o controle de entradas, normalização e conversão dos dados em ponto flutuante conforme a inserção de cada píxel da imagem, com o intuito de disponibilizar os dados de maneira adequada à RNA. A segmentação da imagem é feita a partir da RNA definida em hardware com os seus respectivos pesos e topologia. O módulo RNA possui a funcionalidade de segmentar a imagem principalmente quanto ao nível de intensidade ou à taxa de variação do teor de clorofila, identificando a existência de deficiência de nutriente na planta (soja ou outra cultura). No pós-processamento da

Figura 1 – Modelo conceitual do sistema embarcado desenvolvido.

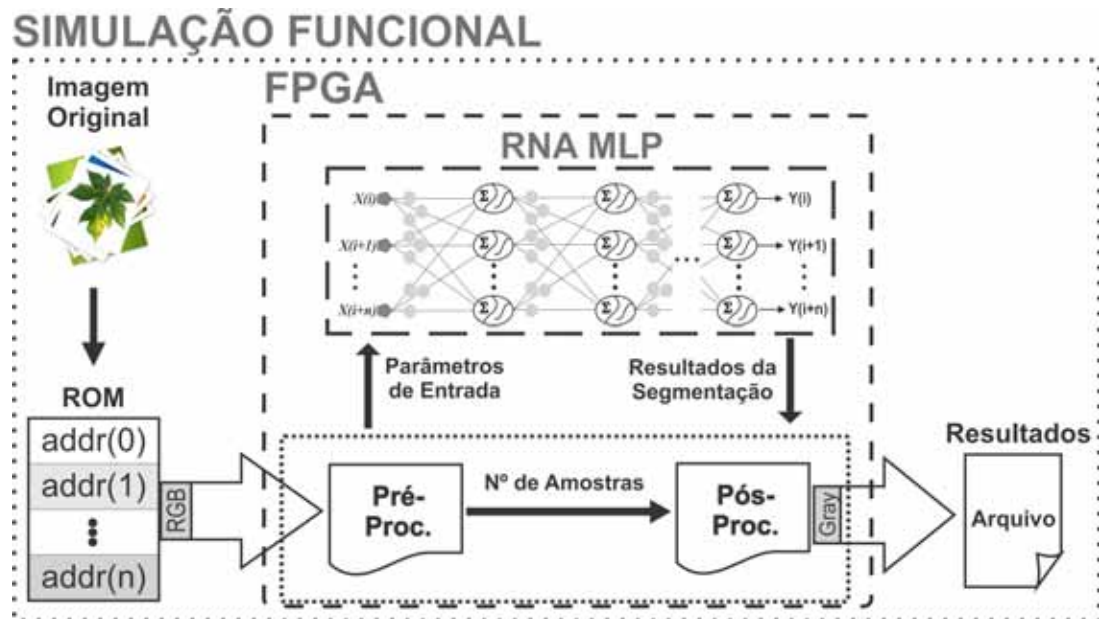


imagem é realizado o processo reverso do pré-processamento e armazena os resultados em arquivo e memória dedicada. Em todo o sistema utiliza-se a padronização IEEE Std 754 na computação dos dados da RNA, com precisão simples em ponto flutuante. Por parte do sistema estar em simulação funcional, o mesmo não está sendo executado em tempo real.

### 1.1 Justificativa

O Estado de Mato Grosso é o maior produtor de soja do país, e o Brasil é o segundo maior exportador de soja do mundo (COMPANHIA NACIONAL DE ABASTECIMENTO - CONAB, 2013; CRUVINEL, 2009). O aumento na eficiência do sistema de produção agrícola está diretamente relacionado à automatização dos processos com maquinário agrícola e com sistemas de gerenciamento de dados de todas as etapas da produção, desde o manejo do solo, passando pelo plantio, colheita, armazenamento e distribuição da produção.

Atualmente, nos sistemas de produção agrícola tem-se uma grande diversidade de processos realizados manualmente por técnicos ou pelos próprios produtores. O monitoramento e o diagnóstico de pragas, doenças e deficiência de nutrientes são exemplos dessas situações. Muitas vezes, essas tarefas são prejudicadas por estarem sensíveis a erros de manuseio ou, por exemplo, na identificação errada de doenças. A aplicação errada de herbicida, em determinada região da colheita, pode desencadear perdas ou prejuízos com novas aplicações.

Anualmente, uma grande soma de recursos financeiros é destinada para aplicação

de insumos agrícolas por parte dos produtores com a inserção da agricultura de precisão mais refinada. O custo com insumos agrícolas é um dos maiores fatores nos gastos fixos do sistema de produção. Com a produção de soja convencional, os fertilizantes contêm maior impacto financeiro, correspondendo a 25,1% do total da produção (RICHETTI, 2011).

A cultura na produção de grãos no Brasil é feita pelo plantio direto, no qual geralmente, realiza-se a aplicação de insumos agrícolas em excesso, com isso pode-se causar degradação no solo e na cultura. A adoção de técnicas de agricultura de precisão pode evitar perdas causadas pelo manejo agrícola incorreto. O uso adequado e eficiente de fertilizantes no manejo agrícola traz um retorno financeiro ao produtor e menor degradação do meio ambiente (DELAMEA, 2008).

O sistema de produção atual é beneficiado pelo uso de máquinas agrícolas, esse recurso contém uma grande diversidade de tecnologias embarcadas, com intuito de melhorar a produtividade da cultura e automatizar, principalmente, a colheita e o mapeamento de áreas de plantio. No entanto, o maquinário agrícola atual tem elevados custos (centenas de milhares de reais) para sua aquisição e não levam em consideração as especificidades nacionais. No maquinário agrícola era comum o uso de diversos computadores de bordo, ferramentas e interfaces proprietárias, dificultando a obtenção de dados e a comunicação com outras tecnologias ou subsistemas. Atualmente, no maquinário agrícola empregam-se equipamentos eletrônicos padronizados para facilitar a aquisição de dados, o padrão mais utilizado é o ISO 11783 (*International Organization for Standardization*) (SARAIVA; CUGNASCA, 2006).

O maior desafio da pesquisa em AP é o desenvolvimento de metodologias e tecnologias atuais com sensores em redes sem fio, sensoriamento remoto, sensoriamento *on-the-go* (Sensores embarcados em máquinas agrícolas) e ferramentas para rastreabilidade, pois exigem investimentos elevados para sua concepção (INAMASU et al., 2011).

A implementação e a aplicação prática da AP torna-se difícil com o alto custo dos maquinários agrícolas, e com a escassez de tecnologia e fabricantes nacionais para automatização agrícola (INAMASU et al., 2011). O emprego de AP atualmente concentra-se na automatização do mapeamento de solo em áreas, conforme a produtividade ou de recursos disponíveis. Porém, o uso do monitoramento da colheita por meio de imagens traz uma forma de reduzir os custos na aquisição de tecnologias para AP.

As dificuldades encontradas com a utilização dos sistemas de monitoramento e análise por imagens estão no baixo desempenho do processamento, requisitos que utilizem de grande quantidade de memória, bem como os elevados custos do sistema de hardware, por causa das limitações dos hardwares específicos (embarcados) utilizados e o grande volume de dados adquiridos (INAMASU et al., 2011). Porém, os dispositivos reconfiguráveis podem sanar essas dificuldades com as novas tecnologias de fabricação de FPGA com regras de

projeto em escala de 28 a 45nm, que possibilitam a execução de sistemas mais complexos com baixo consumo de energia. Além disso, contêm grande quantidade de espaço para lógica programável (elementos ou células lógicas), junto a diversidade de elementos embarcados como o DSP (*Digital Signal Processor*), processador, multiplicadores, e elementos das plataformas de desenvolvimento de sinais mistos, diversos tipos de memória (Cartão de Memória, SDRAM, SRAM, *Block RAM* e *Flash*), interfaces de E/S (*Ethernet*, USB, VGA, HDMI, Serial).

Essa ampla abrangência de recursos, em conjunto com o alto nível de paralelismo e a consolidação como aceleradores por hardware atestam a possibilidade de obter aplicação de tempo real ou com alto desempenho. A vasta diversidade de recursos disponíveis possibilita atualizações e versatilidade para auxiliar o desenvolvimento de sistemas de forma mais ampla nas aplicações.

A aplicação de tecnologias embarcadas na agricultura de precisão necessita de profissionais especializados em áreas como o processamento digital de sinais e sistemas computacionais embarcados. A quantidade de profissionais nessas áreas está muito a quem do ideal, o que dificulta o desenvolvimento de equipamentos para AP (INAMASU et al., 2011). O desenvolvimento desses dispositivos trazem o interesse de produtores em geral, empresas ou fabricantes, que necessitam agregar diferentes equipamentos de sensoriamento remoto com a variação temporal e espacial existente no sistema de produção.

## 1.2 Objetivo

O objetivo deste trabalho foi desenvolver um sistema que identifica a deficiência do macronutriente de potássio em plantações de soja por meio da análise de suas folhas. A principal contribuição do trabalho está no desenvolvimento de um sistema em FPGA de uma rede neural artificial multicamadas para a segmentação de imagens conforme a deficiência do macronutriente. O sistema faz uso de arquitetura parcialmente paralela para computação dos neurônios em ponto flutuante, com precisão de 32 bits padronizada. A aproximação da função de ativação foi investigada com diversos métodos. O método com maior equilíbrio entre área e precisão foi o HPC (Híbrido com divisão em partes lineares (PWL) e expressões booleanas simplificadas (Combinacional)). O sistema completo da RNA em FPGA é comparado à linguagem de alto nível (Matlab) com imagens de folhas, trifólios e ambiente externo na plantação de soja. Diversas outras contribuições são realizadas para tornar possível a implementação da RNA em baixo nível de abstração, como:

- Aquisição e armazenamento de imagem em FPGA;
- Escolha de técnicas para o pré-processamento da imagem em FPGA;

- Comparação de métodos de segmentação de imagem para a identificação de falta de nutrientes em folhas de soja;
- Escolha de técnicas e métodos para o processamento de RNA em FPGA;
- Definição dos melhores parâmetros na estrutura da RNA;
- Gerenciamento da complexidade de projeto envolvendo arquitetura reconfigurável;
- Computação, comunicação e controle dos dados na RNA em FPGA;
- Automação da transição das fases do projeto em dispositivo reconfigurável;
- Padronização e prototipação do sistema;
- Simulação do sistema de hardware com grande quantidade de dados para a segmentação de imagens.

Para tratar a complexidade e as dificuldades mencionadas, propõem-se uma abordagem com uma metodologia de projeto definida conforme os níveis de abstrações definidos para o sistema, e que permita uma melhor reutilização dos componentes de hardware. As bases teóricas necessárias à compreensão da arquitetura, métodos e técnicas aplicados neste trabalho são descritas no Capítulo 2 e no Capítulo 3 os principais trabalhos relacionados. Para o desenvolvimento em alto nível de abstração utilizou-se o Matlab e em baixo nível a linguagem de descrição de hardware VHDL (*Very High Speed Integrated Circuits Hardware Description Language*). O desenvolvimento inicial da RNA de camada simples é feita em dois níveis de abstrações com a função coincidência ou *Não OU exclusivo* (XNOR), apresentada no Capítulo 4. A consolidação do sistema da RNA com múltiplas camadas e função de ativação não linear é desenvolvida no Capítulo 5. Dentre as etapas de investigação, podem-se citar os principais itens averiguados como a unidade de ponto flutuante, o uso de memória dedicada, o controle do sistema por meio de máquinas de estados finitos, métodos para aproximação da função Tangente Hiperbólica e normalização de dados.

Os métodos verificados para o processamento digital de imagem são apresentados no Capítulo 6. Na segmentação foram analisados a limiarização simples com um único limiar global e com múltiplos limiares por meio do método de Otsu. A filtragem espacial foi avaliada com o intuito de supressão de ruídos possíveis na aquisição de imagens externas, utilizou-se filtros lineares de média simples e ponderada. Os métodos comuns de PDI foram comparados com RNA em alto nível, e a necessidade de alteração de diversos parâmetros de entrada e estrutura da RNA foram evidenciados pela análise de parâmetros de qualidade.

Após a comprovação dos resultados da RNA em processamento de imagem, o Capítulo 7 analisa o método na segmentação de imagens na cultura do algodão. A comparação da RNA e o método de Otsu são avaliados na identificação de dois macronutrientes, fósforo e potássio. A cultura da soja é verificada no Capítulo 8 para identificação do macronutriente de potássio. A comparação do sistema da RNA em dois níveis de abstrações são avaliados no Capítulo 8.

## 2 REFERENCIAL TEÓRICO

Este capítulo tem o objetivo de apresentar ao leitor uma breve base teórica para compreensão dos conceitos principais expostos no trabalho. Na seção 2.1 aborda-se a comparação entre a computação reconfigurável e outras tecnologias semelhantes, e a definição dos tipos de recursos pertencentes a um dispositivo reconfigurável (FPGA). Na seção 2.2 está definida a representação de dados em ponto flutuante, o funcionamento e a aplicação de redes neurais artificiais é apresentada na seção 2.3. Uma introdução ao processamento digital de imagens é apresentada na seção 2.4.

### 2.1 Computação Reconfigurável

Os sistemas computacionais têm como principal função a resolução de problemas nas mais variadas áreas do conhecimento. Atualmente, com o advento da tecnologia embarcada, os sistemas computacionais podem ser implementados em hardware, software ou trabalhando em conjunto, comumente empregando técnicas de *co-design* (COFER; HARDING, 2006).

Computadores pessoais são caracterizados por conter processador de propósito geral, utilizados em uma grande variedade de aplicações. O desenvolvimento de aplicações é feito por software, com as vantagens de alta flexibilidade, rápido desenvolvimento de um sistema, desempenho suficiente à um grande número de aplicações e preços razoavelmente baratos.

Existem aplicações que necessitam de alto desempenho e uso de recursos, e muitas vezes, não são solúveis em um tempo aceitável. Essas aplicações são impraticáveis em computadores de uso geral, assim recorre-se a diferentes abordagens de hardware. O uso de processamento paralelo, mesmo com processadores de propósito geral, melhoram o desempenho da aplicação com técnicas, por exemplo, de dividir e conquistar. Essa abordagem traz um aumento na complexidade com a necessidade de incorporar técnicas de comunicação em grupo. As interligações entre os nodos podem se tornar um gargalo ao sistema dependendo da quantidade e tamanho dos dados necessários para cada nodo. O paralelismo da aplicação (ou do seu núcleo) muitas vezes pode ser prejudicado pelas limitações impostas pela lei de Amdahl (HARTENSTEIN, 2001). Com o uso de

dispositivos reconfiguráveis obtém-se maior flexibilidade e um maior nível de paralelismo no hardware. Isso pode trazer melhorias no desempenho principalmente no núcleo da aplicação (HERBORDT et al., 2007).

Os dispositivos dedicados (hardware) projetados em CI (Circuito Integrado) são orientados a aplicação, e suas principais características estão marcadas pelo alto desempenho e a falta de flexibilidade nas alterações de suas funcionalidades. O ASIC, considerado um dispositivo dedicado, tem seu desenvolvimento mais dispendioso, e necessita de um grande número de unidades fabricadas para diminuir o custo. Além disso, pode-se citar algumas desvantagens antes e depois da fabricação do chip: o sistema necessita de um maior tempo de projeto e desenvolvimento, pela complexidade e o grande número de testes; após a fabricação é impossível alterações no chip, tornando-se necessária a sua refabricação para qualquer alteração ou atualização.

A meta da computação reconfigurável é unir as vantagens proporcionadas pelas características do hardware e as facilidades oferecidas pelo software em um só conceito, com isso satisfazer os problemas da melhor forma possível. A ideia é agregar as principais vantagens de cada abordagem, do hardware: maior desempenho; software: maior flexibilidade da solução e menor tempo de desenvolvimento.

A computação reconfigurável é uma área promissora para pesquisa acadêmica e comercial, por possibilitar uma maior agilidade no desenvolvimento de projetos de hardware com o reuso a partir de IP Cores e o uso de barramentos padronizados. Além disso, a facilidade na atualização do sistema com técnicas de configuração do hardware, geralmente é beneficiada pelo uso de dispositivos reconfiguráveis em projetos de *hardware*.

No desenvolvimento desses tipos de projetos é necessário o uso intensivo de ferramentas CAD para automatizar etapas do processo de validação do sistema e da síntese dos circuitos, geralmente descritos em RTL (*Register-Transfer Level*). Cabe ao projetista do sistema definir características como desempenho, área utilizada, portabilidade e potência consumida.

Com o uso de FPGA, na computação reconfigurável, obtém-se excelentes resultados de desempenho (SCROFANO et al., 2008; KHRAISHA; LEE, 2010; FERREIRA; BARROS, 2010). A granularidade é um fator importante para os dispositivos reconfiguráveis (FPGA) alcançarem um melhor desempenho (BEAUCHAMP; HAUCK, 2006; STRENSKI, 2007). A granularidade é relativa ao tamanho, escala e nível de detalhamento de componentes sendo que os dois principais tipos são: grão grande e grão pequeno. Em componentes de grão pequeno, pode-se aproveitar melhor o hardware para um determinado problema, utilizando técnicas e alto nível de paralelismo, independente do tamanho dos dados ou da complexidade. Os dispositivos reconfiguráveis contêm granularidade de grão pequeno, diferente do processamento paralelo convencional ou agregado de máquinas que são

consideradas de grão grande, por utilizarem processadores de propósito geral. Além disso, com a implementação direta em HDL, por exemplo, VHDL ou Verilog (*Verifying Logic*), é feita uma documentação e padronização do projeto de hardware que facilita futuras manutenções.

Existem diversos dispositivos que trazem o suporte a computação reconfigurável, o mais popular é o FPGA. A sua arquitetura é baseada em uma matriz lógica totalmente reconfigurável, os componentes de E/S podem ser configurados de acordo com especificações eletrônicas (padrões E/S, direção, diferença de potencial, etc), distintos tipos de memórias, entre outros, são exemplos da flexibilidade suportada pelo dispositivo. As inúmeras configurações possíveis para os dispositivos reconfiguráveis podem estar relacionadas a diversas áreas ou aplicações com a utilização do mesmo dispositivo, o que não acontece no desenvolvimento com um ASIC.

O uso de diferentes metodologias de projeto para sistemas reconfiguráveis empregando FPGAs tem recebido destaque quando se trata de computação reconfigurável. Existem diversas metodologias de projeto como a reconfiguração em tempo de execução ou compilação, a reconfiguração estática ou dinâmica e a reconfiguração parcial ou total.

Devido ao tempo significativo de desenvolvimento de um projeto de hardware com FPGA, em comparação ao desenvolvimento de software, é importante fazer o uso intensivo de reutilização de componentes. Para isso, conta-se com o auxílio da VHDL para padronizar e documentar o hardware proposto, associado ao objetivo de uma melhor utilização dos recursos da FPGA.

### **2.1.1 FPGA**

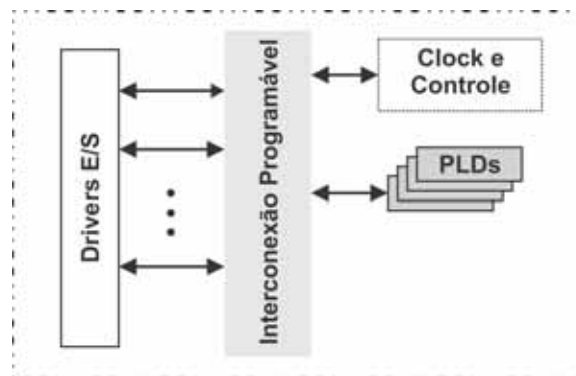
Os dispositivos reconfiguráveis (FPGA) foram desenvolvidos a partir de estruturas mais simples chamadas de PLD (*Programmable Logic Devices*). O PLD é um termo genérico que representa diversos dispositivos com características semelhantes, inicialmente foi definido para construir equipamentos capazes de executar circuitos combinacionais. Atualmente, existem não só combinacionais e sequenciais, mas uma ampla abrangência de aplicações possíveis. Dispositivos lógicos programáveis ou PLDs são circuitos integrados com portas lógicas internas e interconectado por meio de chaves programáveis entre recursos heterogêneos (Blocos Lógicos Programáveis, DSP, Memória dedicada, etc). O PLD não possui uma funcionalidade fixa, podendo ser reconfigurado pelo usuário quando houver necessidade, facilitando as alterações no projeto.

Geralmente, os autores referem-se a SPLDs (*Simple Programmable Logic Devices*) como sendo os primeiros dispositivos, e as suas subdivisões o PAL (*Programmable Array Logic*) e o PLA (*Programmable Logic Array*). Os PLDs com portas AND programáveis

e portas OR fixas é chamado de PAL e não tinham disponíveis a programação pelo usuário, apenas por máscara do próprio fabricante, semelhante a memória ROM. Já o PLA continham portas AND e OR programáveis, diferente do PAL, esse dispositivo possibilitou a programação pelo usuário (COFER; HARDING, 2006).

Os dispositivos SPLDs eram fabricados com tecnologia MSI (*Medium Scale Integration*) com até centenas de portas lógicas. A geração seguinte dos PLDs continham uma estrutura complexa com dois níveis de portas lógicas AND e OR, era chamado de CPLD (*Complex Programmable Logic Devices*). Além disso, esses dispositivos acomodavam uma coleção de vários PLDs em um mesmo chip, conforme apresentada na Figura 2 (PEDRONI, 2008).

**Figura 2 – Arquitetura básica de um CPLD.**



Fonte: Pedroni (2008).

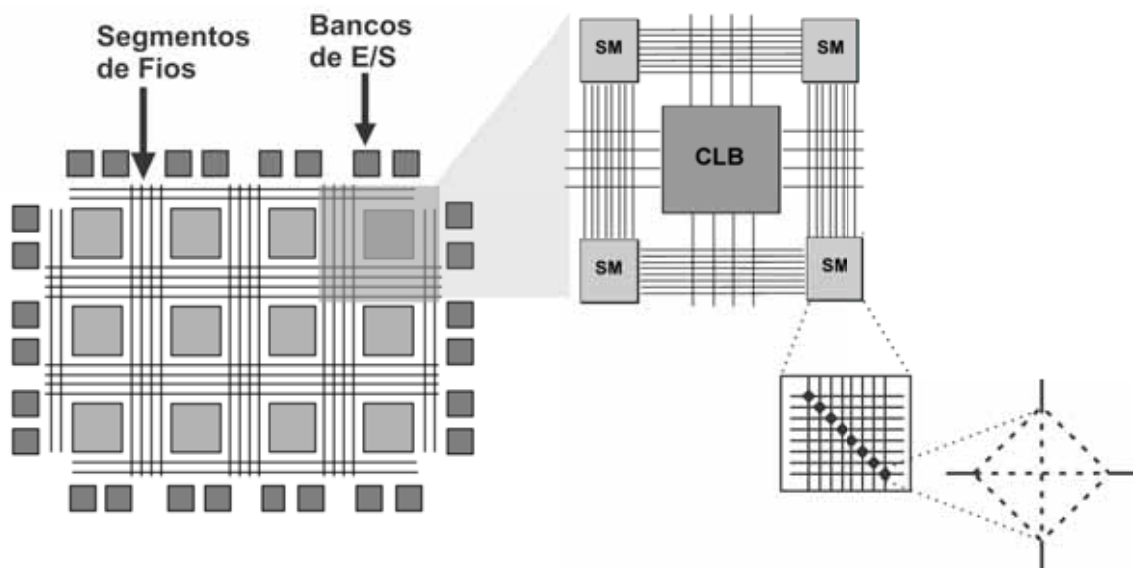
Os FPGAs são exemplos de PLDs com uma estrutura baseada em blocos lógicos, diferente dos CPLDs. Os FPGAs consistem em uma matriz de blocos lógicos, com pinos e fios de comunicação, todos configuráveis a partir do preenchimento de uma memória de controle interna (tipicamente do tipo RAM). Essa funcionalidade torna mais dinâmico o desenvolvimento do projeto. Uma vantagem é que sua implementação produz um hardware totalmente dedicado à aplicação, diferente de um microcontrolador que executa software em uma Unidade Central de Processamento (UCP). Atualmente, a comparação de dispositivos FPGAs por meio de portas lógicas não é determinística, por causa dos inúmeros recursos lógicos e dedicados disponibilizados por esses dispositivos.

#### 2.1.1.1 Arquitetura de FPGAs

Os FPGAs são circuitos lógicos que consistem em um grande arranjo de portas lógicas configuráveis contidos em um único circuito integrado, que podem ser reconfiguradas conforme a necessidade do usuário. Diversos tipos de organizações ao arranjo lógico são possíveis, dependendo do modelo e do fabricante. A matriz simétrica é a mais difundida, pela flexibilidade no roteamento e por possuir canais horizontais e verticais

em suas interconexões. Sua estrutura básica é formada de blocos lógicos, blocos de entrada e saída e segmentos de fios, conforme apresentado na Figura 3. Para interconectar os segmentos de fios ou canais horizontais e verticais existem blocos chamados de interruptores configuráveis ou *Switch Matrix* (SM). A interconexão entre os segmentos de fios, que chegam à interface da SM, pode ser configurada para a comutação dos dados de forma mais otimizada, e traçar o caminho ideal para os Blocos Lógicos Configuráveis (CLB).

**Figura 3 – Arquitetura FPGA baseada em matriz lógica.**



Fonte: Elaboração do próprio autor.

A tecnologia de programação adotada para configurar os CLBs, conforme a aplicação alvo, é uma importante característica da plataforma FPGA. A antifusível e a baseada em memória são as duas tecnologias empregadas em FPGA. A primeira tem sua configuração limitada em uma única vez e a outra com inúmeras possibilidades de reconfiguração, utilizando principalmente memórias do tipo SRAM e Flash.

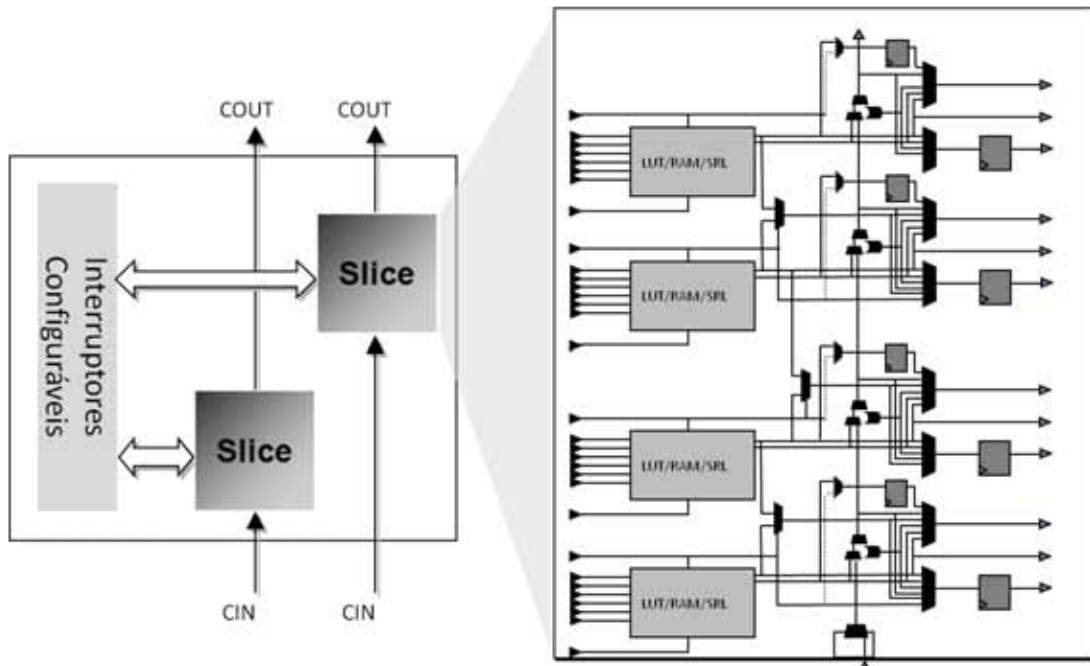
Os nomes e as estruturas adotadas aos “componentes lógicos” que formam a arquitetura da FPGA podem apresentar algumas diferenças conforme o tipo de fabricante e o modelo do chip.

As especificações teóricas neste trabalho sobre as plataformas de FPGA serão definidas a partir do fabricante Xilinx e suas novas famílias da série 7.

Os Blocos Lógicos Configuráveis (CLB) são as maiores unidades lógicas na FPGA. Cada CLB contém, tipicamente, capacidade computacional para implementar um conjunto de funções lógicas, realizar roteamento para comunicação entre blocos e uma pequena capacidade de armazenamento. Um FPGA contém centenas de milhares de CLBs os quais possibilitam o seu uso em aplicações complexas. Nas novas famílias de FPGA cada CLB contém dois *Slices*, um conector a matriz de blocos lógicos e um *carry*

ou “vai-um” In/Out para cada *Slice*, conforme apresentado na Figura 4. Além disso, cada CLB possui comunicação com a matriz lógica por meio de canais horizontais e verticais, os horizontais com transferência bi-direcional e os verticais com direção única (XILINX, 2012).

**Figura 4 – Estrutura de um CLB.**



Fonte: Xilinx (2012).

Os *Slices* são componentes lógicos formados geralmente por quatro LUTs (*Look-Up Tables*), vários Multiplexadores, propagadores de *carry*, oito elementos de armazenamento, registradores e *Latches*. Existem *Slices* com diversas funcionalidades, organizadas em três tipos:

- Slice-M - Completo com uso para lógica e armazenamento;
- Slice-L - Usado somente para lógica e operações aritméticas;
- Slice-X - Usado somente para lógica.

Os CLBs sempre contêm um *Slice* do tipo X e outro do tipo M ou L, para oferecer suporte com estruturas combinadas em dois modos. O conector para matriz lógica serve como comutador de dados destinados a outras regiões da FPGA, e os *carries* fazem a comunicação com CLBs vizinhos para facilitar a propagação dos dados (XILINX, 2012).

A LUT é o menor componente lógico na arquitetura da FPGA e pode conter características diferenciadas, por exemplo, existem dispositivos que usam LUTs de várias quantidades de entradas, dependendo do fabricante ou modelo da FPGA. A LUT é

uma memória SRAM simples que implementa a tabela verdade de uma função lógica dependendo da sua quantidade de entradas ( $n$ ), o tamanho da LUT é  $2^n$ . Uma LUT de 4 entradas tem capacidade de implementar uma função lógica com a mesma quantidade de entradas, já as LUTs de 6 entradas podem representar 2 tipos de funções, com a mesma quantidade de entradas ou duas funções independentes de 5 entradas (XILINX, 2012).

As principais características que diferenciam os fabricantes de FPGA estão relacionadas à arquitetura dos blocos lógicos, ao tipo de método de configuração da FPGA, ao tipo de memória utilizada e à capacidade de recursos lógicos. Na Tabela 1 apresentam-se os principais fabricantes de FPGA e as características de suas plataformas. Dentre essas características, vale destacar as diferenças entre as unidades que encapsulam as LUTs, células lógicas (equivalentes às *Slices*).

**Tabela 1 – Comparação de características em distintos fabricantes de FPGA.**

Características	Fabricantes			
	Altera	Lattice	Microsemi(Actel)	Xilinx
Células Lógicas	Elementos Lógicos	Slices	Módulos Lógicos	Slices
Blocos Lógicos	ALM <sup>2</sup>	PFU <sup>2</sup>	-	CLB
Memória Embarcada	M-RAM Block	sysMEM	LSRAM Blocks	Block RAM
Densidade Log. Comb. LUT(bits)	128	128	16	256
Fios Globais e Locais de Relógio	SIM	SIM	-	SIM
DSPs Embarcados	SIM	SIM	SIM	SIM
Reconfiguração Parcial Dinâmica	SIM	SIM	-	SIM
Tecnologia de Configuração	SRAM	SRAM/Flash	Antifuse/Flash	SRAM
Volume NASDAQ <sup>1</sup>	28,5%	19,0%	4,4%	43,4%
Novas Tecnologias Agregadas	Optical bandwidth	3G SERDES	Low-Power	ADC/3D IC

Os blocos lógicos contêm uma ou mais células lógicas e, nesse caso, a FPGA do fabricante Microsemi não contém essa característica, devido à sua granularidade pequena. Os métodos de configuração da FPGA mais comuns entre os fabricantes são por meio de memórias voláteis (SRAM) ou não voláteis (Flash) e antifuse. A antifuse é uma tecnologia em que o dispositivo não tem a possibilidade de reconfiguração.

### 2.1.1.2 Recursos Dedicados

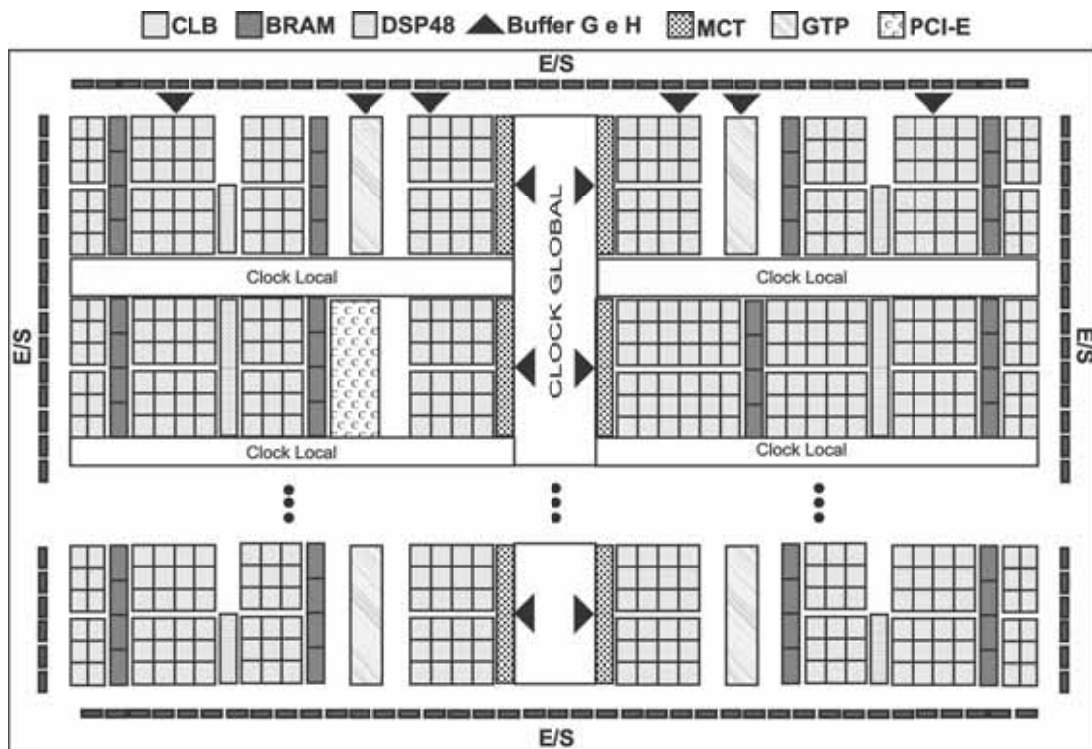
Atualmente, os recursos dedicados agregados ao chip fornecem uma das principais diferenças entre as famílias de FPGA, com um maior número desses componentes aumenta o desempenho e diminui a utilização de CLBs na implementação. Outras características também divergem conforme o fabricante e o modelo do chip, como a tecnologia de fabricação, os componentes lógicos (CLB, Slice e LUT), a posição dos componentes dedicados, e a frequência máxima atingida pelo chip (*Speed Grade*).

<sup>1</sup>O cálculo foi realizado entre os principais fabricantes de FPGA na área de semicondutores conforme o volume financeiro ativo da NASDAQ, os dados foram atualizados em Fev. 2013

<sup>2</sup>ALM - Adaptive Logic Modules; PFU - Programmable Function Unit

O advento da tecnologia de fabricação em 28 nm possibilitou a disponibilidade de muitos recursos em FPGA, tanto de blocos lógicos configuráveis, quanto de componentes embarcados dedicados. As famílias de FPGA da atualidade têm sua matriz lógica rodeada de vários componentes embarcados visando um melhor funcionamento global, flexibilidade e alto desempenho na aplicação. Os principais componentes de hardware embarcados são: os blocos de memória dedicados, dispositivos de E/S mais próximos das extremidades do chip e de alto desempenho (GTP-*Low-Power Gigabit Transceivers*), multiplicadores, DSPs, o MCT (*Manage Clock Tile*) que contém PLL (*Phase Locked Level*) e gerenciador de *clock* digital (DCM), processador dedicado, entre outros. Na Figura 5 apresentam-se os componentes básicos e embarcados de uma FPGA típica das novas famílias de chip da Xilinx.

**Figura 5 – Componentes embarcados típicos em uma FPGA.**



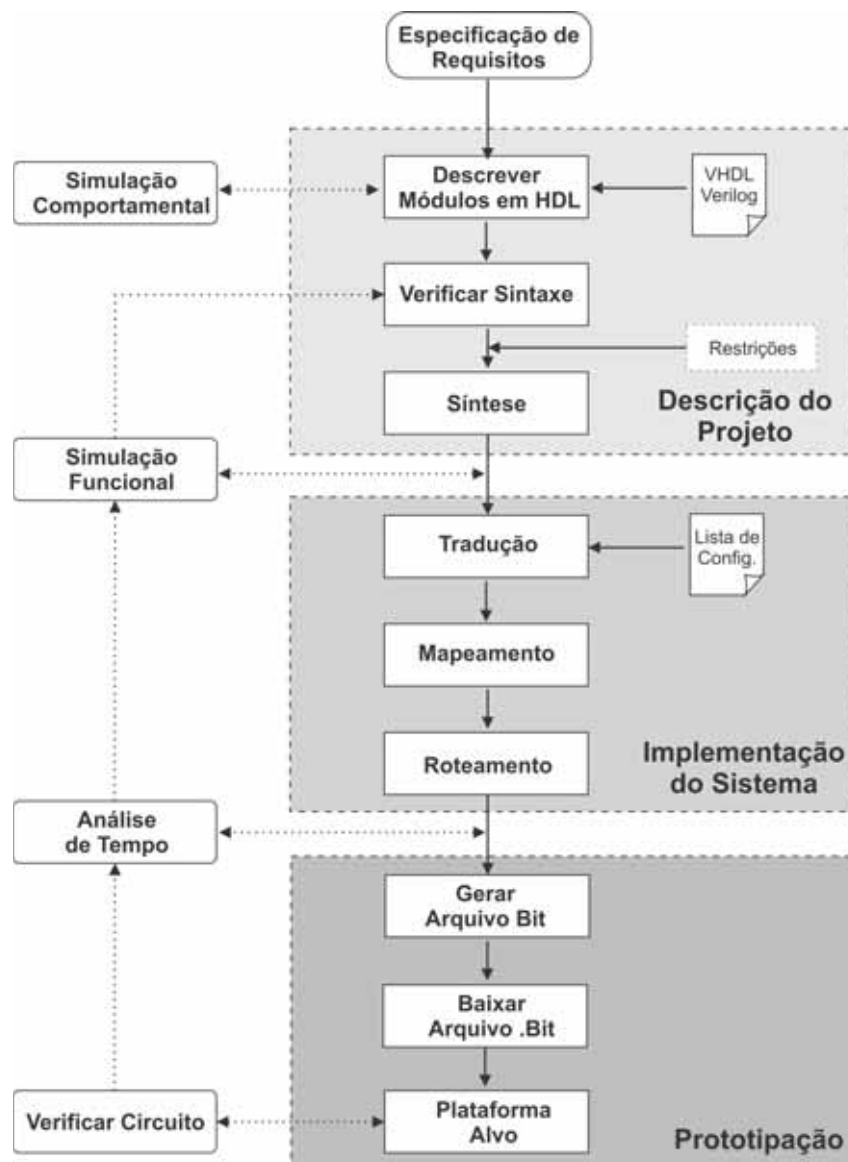
Fonte: Elaboração do próprio autor.

No projeto de aplicações em FPGA é comum o emprego de recursos dedicados para gerenciar o *clock*, armazenamento e processamento de sinais. Além disso, o projetista aproveita a possibilidade de reuso por meio de módulos IP visando agilizar o desenvolvimento do sistema. Dentre esses módulos destaca-se o uso de processamento digital de sinais, vídeos e imagens, ponto flutuante, processadores, interfaces de memória e E/S, operações aritméticas, entre outros. A escolha de diferentes padrões e níveis de corrente também é possível ser parametrizado para os pinos de E/S da FPGA. Essa gama de parametrização do hardware mostra a ampla flexibilidade de configurações do sistema para uma aplicação específica, bem como a diversidade de áreas de utilização.

### 2.1.2 Projeto de Hardware Reconfigurável

O início do projeto de um sistema baseado em hardware reconfigurável é semelhante ao de software. Nos dois casos deve-se fazer um levantamento ou estudo da aplicação a ser desenvolvida. Inicialmente, deve-se especificar todas as informações a respeito do domínio da aplicação. Antes de iniciar o projeto de hardware, se possível, é importante o desenvolvimento inicial da aplicação em linguagem de alto nível para adquirir um conhecimento prévio da aplicação e depois iniciar a especificação dos requisitos do sistema. Na Figura 6 apresenta-se o fluxo de projeto para o desenvolvimento em um dispositivo reconfigurável.

Figura 6 – Fluxo de desenvolvimento de um dispositivo reconfigurável (FPGA).



Fonte: Cofer e Harding (2006).

Existem diversas metodologias para desenvolvimento de hardware como a clássica, a

comportamental, a hierárquica, a co-design, , neste trabalho foi adotada a hierárquica. Essa metodologia possibilita a verificação individual dos blocos, e facilita a manutenção e a identificação de erros no sistema. Geralmente, o projeto é iniciado pela descrição dos blocos que irão compor a base do sistema ou o nível mais baixo da hierarquia, e os outros módulos no sentido para o nível mais alto ou topo do projeto. O processo de desenvolvimento de um sistema descrito em HDL para dispositivos reconfiguráveis (FPGA) contém 3 etapas principais: Descrição do Projeto, Implementação do Sistema e Prototipação.

O desenvolvimento do hardware inicia-se com a descrição de módulos em HDL, VHDL ou Verilog, e se necessário módulos IP Cores. Semelhante ao software, também há a necessidade de verificação da sintaxe. A VHDL é muito popular entre as linguagens de descrição de hardware. A VHDL em comparação a uma linguagem de alto nível, tem grande diferença quando não há um conhecimento prévio de sistemas digitais. Isso deve-se pelo fato de conter restrições e sintaxe diferenciada, principalmente nas definições básicas e tipos de dados de variáveis, sinais e portas. Para cada fabricante de FPGA existe uma ferramenta CAD com interface gráfica para a descrição do sistema em uma ou mais linguagens HDL. Isso auxilia em todo o processo de desenvolvimento do hardware, reduzindo o tempo de projeto do sistema. Antes da síntese, é possível efetuar a simulação comportamental do código HDL descrito, sem relação ao dispositivo reconfigurável.

A síntese é uma etapa em que a ferramenta CAD verifica vários requisitos do sistema como a sintaxe, problemas entre barramentos, instanciações efetuadas erroneamente, otimização do código, área e desempenho. Em descrição comportamental é feita uma transformação ao hardware correspondente, codificação da máquina de estados, entre outros. Após realizada essa etapa, o projetista cria um módulo chamado *testbench* para possibilitar a realização dos primeiros testes do sistema, a simulação funcional.

Com a ferramenta CAD adequada, a simulação funcional possibilita a definição de formas de onda correspondentes ao funcionamento do sistema e a aquisição de resultados gerados pelo hardware. Geralmente as ferramentas gráficas para visualização das formas de onda disponibilizam a verificação de cada sinal ou barramento em qualquer intervalo da simulação, tornando acessível ao projetista a investigação minuciosa do sistema ou módulo. A realização da simulação de blocos independentes é possível com a inclusão de um novo módulo *testbench* com os estímulos relacionados a esse bloco, assim a maioria dos erros podem ser encontrados até a simulação funcional.

Atualmente as ferramentas CAD não utilizam o tipo de simulação comportamental, apenas a funcional, por ser executada após a síntese e com código HDL relacionado à plataforma alvo.

A etapa de implementação do projeto tem início com a tradução que converte a lista

de configuração de E/S em uma listagem completa com a inclusão de todos os requisitos de hardware contidos no sistema, junto a ligações de bibliotecas. No mapeamento realiza-se a definição dos recursos específicos do sistema, já o roteamento faz a conexão física dos componentes com a disponibilização à plataforma FPGA.

Após a etapa de implementação é possível fazer uma análise do tempo, que consiste em um monitoramento físico do sistema na Plataforma FPGA. Geralmente são inseridos monitores no sistema que coletam informações de sinais especificados pelo projetista.

Na última fase, a prototipação é feita quase automaticamente pela ferramenta CAD. Quando o projetista gera o arquivo .bit e transfere para a FPGA, o hardware correspondente ao sistema desenvolvido é executado na plataforma FPGA. Nas plataformas de desenvolvimento ou kits didáticos, o usuário pode interagir com o sistema através de dispositivos de E/S (chaves, botões e leds) para verificar o funcionamento correto do sistema, conforme definido no projeto.

A tarefa de organizar o projeto do sistema com a plataforma FPGA geralmente é feita pela ferramenta CAD, isso é de extrema importância e pode comprometer todo o sistema. A ferramenta CAD deve identificar os componentes que realizam grande comunicação de dados entre si e deixá-los o mais próximo possível, com o intuito de diminuir o atraso físico entre esses componentes e tentar alcançar um tempo de resposta próximo ao ideal. Outra característica importante a ser avaliada é a inserção de componentes de armazenamento, *Flip-Flops*, entre as descrições de lógicas puramente combinacionais ou entre módulos.

O uso de componentes de armazenamento provê uma arquitetura com confiabilidade nos resultados gerados pelos módulos combinacionais. Dependendo do sistema, o atraso físico do hardware pode não ser considerado pelo projetista e isso pode trazer problemas nos resultados gerados pelo sistema, muitas vezes difíceis de serem encontrados.

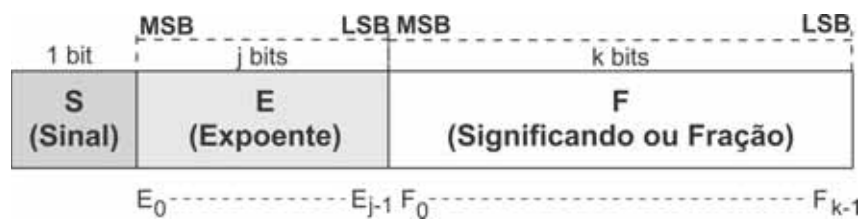
No desenvolvimento de projeto em dispositivo reconfigurável é necessário o uso intensivo de ferramentas CAD para automatizar etapas do processo de validação do sistema e da síntese dos circuitos, geralmente descritos em RTL. Cabe ao projetista do sistema definir as características essenciais do sistema como desempenho, a área utilizada, portabilidade e potência, para fazer as devidas alterações e proporcionar um sistema próximo ao ideal para a aplicação.

Os dispositivos reconfiguráveis (FPGA) possibilitam uma grande agilidade no desenvolvimento de projetos de hardware, principalmente pela evolução das ferramentas CAD. Além disso, a facilidade na atualização do sistema traz facilidade em eventuais testes físicos na busca de problemas ou adaptações. A disponibilidade de ferramentas e linguagens para a testabilidade do hardware provê formas eficientes de eliminar erros ou falhas de projetos.

## 2.2 O Padrão IEEE Std 754

A norma IEEE Std 754 resume-se a padronizar a representação de dados para números racionais em sistemas computacionais. Existe diversas tipos de representações conforme a precisão desejada do número, as mais populares são a precisão simples, dupla, estendida e quádrupla. Para compreender o funcionamento desse tipo de representação apresenta-se na Figura 7 uma característica básica, a divisão do número em três partes: Sinal, Expoente e a Fração. Em cada parte, com mais de um bit, contém a indicação do bit mais significativo (MSB) e o bit menos significativo (LSB).

**Figura 7 – Forma de representação de números racionais na base binária.**



Fonte: Elaboração do próprio autor.

No padrão IEEE Std 754, revisado de 2008, não se determina mais os formatos em precisões simples, dupla ou quádrupla, apenas identifica-se pela quantidade de bits em cada formato como binário de 16, 32, 64 ou 128 bits. A opção estendida para 32, 64 e 128 também é definida pela norma. Na Tabela 2 define-se o formato e a quantidade de bits em cada parte nas principais representações de números em ponto flutuante. Por opção adotou-se os nomes e a quantidade de bits para identificar a precisão para melhor compreensão. A precisão estendida para os três formatos apresentados é formalizado pela norma IEEE Std 754 (IEEE COMPUTER SOCIETY, 2008).

**Tabela 2 – Formatos de representações em ponto flutuante.**

Padrão IEEE-754	Simple (32 bits)	Dupla (64 bits)	Quádrupla (128 bits)
Sinal (S)	1	1	1
Expoente (E)	8	11	15
Significando (F)	23	52	112
Total (bits)	32	64	128
Expoente Max. ( $e_{max}$ )	127	1023	16.383

Outra definição apresentada na padronização IEEE Std 754 é a normatização das formas de arredondamento e tratamento de exceções em operações envolvendo ponto flutuante. Existe a possibilidade de reconhecimento específico para números do tipo infinito e NaN (*Not a Number*). A formalização na utilização dos atributos, para a direção do arredondamento e os tipos de exceções são descritas com a finalidade de reconhecer um problema em uma operação de ponto flutuante. Uma das principais informações contidas na norma é a descrição das exceções, como exemplo, tem-se o *overflow*, *underflow*, divisão

por zero, entre outros.

A conversão e as operações aritméticas básicas em ponto flutuante precisam seguir regras específicas para proporcionar o resultado desejado. Dependendo da operação a ser realizada pode aumentar a complexidade do algoritmo. Neste trabalho foi necessário fazer a conversão dos números reais, base decimal, para ponto flutuante na base binária, em vista a utilização desse tipo de dado no dispositivo reconfigurável (FPGA). O sentido inverso também foi necessário para análise dos resultados. A Equação (1) demonstra essa conversão, onde “ $S$ ” é o sinal, “ $E$ ” é o expoente e “ $F$ ” é a fração, todos definidos por números inteiros com a precisão definida pelo formato utilizado conforme apresentado na Tabela 2.

$$Valor_{Real} = (-1)^S \times \left(1 + \sum_{i=1}^{k_{max}} F_{-i} \times 2^{-i}\right) \times 2^{(E-E_{max})} \quad (1)$$

Existem diversos módulos IP para o processamento de operações em ponto flutuante, elaborados pelos próprios fabricantes da FPGA ou disponibilizados de forma livre. Geralmente a interface dos módulos são semelhantes e de fácil implementação. Porém, existem módulos com grandes quantidades de dados a serem instanciados ao módulo e formas de controle diferenciadas, que dificultam sua utilização. Isso acontece não só nas unidades de ponto flutuante, como em outras unidades ou interfaces entre componentes em sistemas digitais.

### 2.3 Redes Neurais Artificiais

Atualmente, métodos computacionais se inspiram em comportamentos de animais ou aspectos biológicos para a resolução de problemas de otimização linear e não-linear, reconhecimento de padrões, sistemas computacionais e de controle, entre outros. As redes neurais artificiais (RNA) são modelos bio-inspirados no funcionamento do cérebro humano ou redes neurais naturais (RNN).

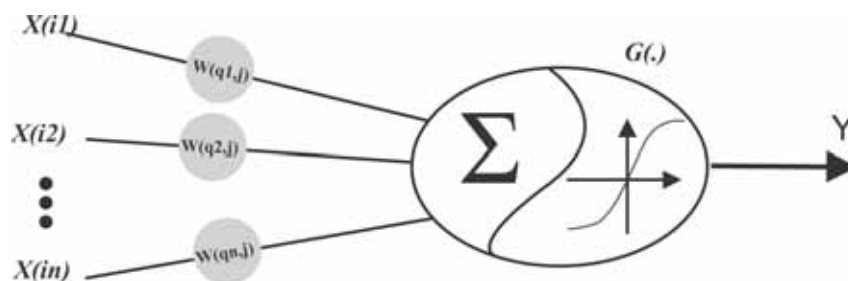
A RNA é geralmente utilizada quando, não há modelos matemáticos para a resolução de um determinado problema não linear, por prover o desenvolvimento com técnicas de processamento paralelo, por obter resultados mais rápidos (após o treinamento) que outros modelos matemáticos e por melhorar a utilização de área na FPGA em comparação à implementação direta. A robustez nos resultados e a vasta abrangência de aplicações como robótica, automação e controle, processamento de imagem, entre outros, tornam-se um atrativo aos pesquisadores na resolução de uma diversidade de problemas.

Frequentemente, as RNA são empregadas na resolução em aplicações de tempo real, e

quando existe um custo alto na implementação de um modelo matemático em hardware. Cada parte de uma RNA pode ser associada a um fenômeno biológico da RNN. O modelo de neurônio biológico muito se assemelha ao artificial, criado por McCulloch e Pitts (MCCULLOCH; PITTS, 1943).

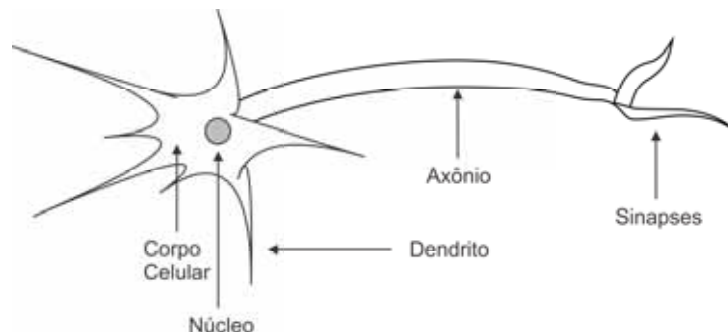
Na comparação entre o modelo de neurônio artificial, apresentado na Figura 8, com o biológico, apresentado na Figura 9, pode-se definir, respectivamente, que a entrada de dados são os dendritos. A soma ponderada das entradas e a função de ativação estão relacionados ao corpo celular do neurônio biológico, que definem a existência de um sinal elétrico ou nervoso transportado pelo axônio até as sinapses ou saídas da RNA (SILVA; SPATTI; FLAUZINO, 2010).

**Figura 8 – Modelo geral neurônio artificial.**



Fonte: Elaboração do próprio autor.

**Figura 9 – Neurônio biológico.**



Fonte: Elaboração do próprio autor.

### 2.3.1 Neurônio Artificial

O neurônio artificial é uma soma ponderada de suas entradas com seus respectivos pesos, conforme definido na Equação (2). Para cada entrada do neurônio existe um peso relacionado e distinto dos demais. Na Figura 8 são apresentados o neurônio artificial com suas entradas ( $X_n$ ) e seus pesos ( $W_q$ ). O índice  $j$  corresponde a quantidade de neurônios existentes na camada ( $k$ ) e o índice  $q$  a quantidade de neurônios existentes na camada anterior ( $k - 1$ ).

$$\eta_j^k = \sum_{q=0}^n W_{qj}^k * X_j^k \quad (2)$$

Um componente importante do neurônio artificial é a função de ativação ( $G(\cdot)$ ). A sua característica está em limitar a saída parcial da RNA no neurônio artificial, conforme o tipo de resposta proporcionada pela função escolhida. A função de ativação consiste em um filtro do resultado da Equação (2) no neurônio artificial.

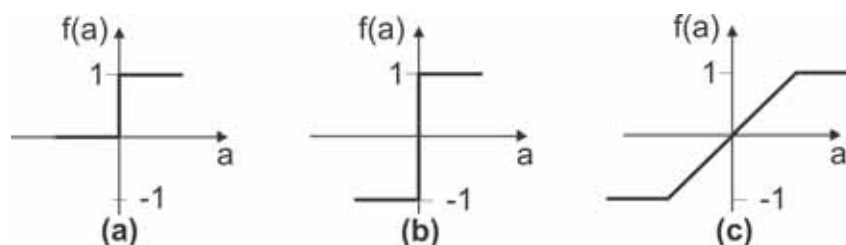
A saída parcial ( $Y_j^k$ ) da RNA no neurônio correspondente é definida pela Equação (3), onde  $G$  é a função de ativação e  $\eta$  é definido pela Equação (2).

$$Y_j^k = G(\eta_j^k) \quad (3)$$

Existem vários tipos de função de ativação e podem ser classificadas em dois grupos, parcial e totalmente diferenciável. Os tipos mais simples estão na parcialmente diferenciável cujas transições são mais abruptas e possuem pontos em que a derivada de primeira ordem é inexistente. Na Figura 10 são apresentadas as três principais funções: Degrau, Degrau Bipolar e Rampa.

Nas totalmente diferenciável são consideradas mais complexas e a derivada de primeira ordem existe em todo os pontos das funções. Na Figura 11 ilustra-se as principais funções dessa categoria, como exemplo tem-se a Logística, Gaussiana, Tangente Hiperbólica. A sigmoide é a mais utilizada nas redes MLP pela sua robustez nas mais variadas aplicações (SILVA; SPATTI; FLAUZINO, 2010).

**Figura 10 – Principais funções de ativação parcialmente diferenciável: (a) Degrau; (b) Degrau Bipolar; (c) Rampa.**

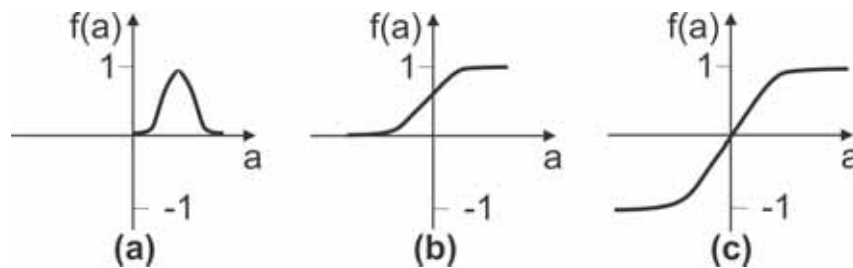


Fonte: Elaboração do próprio autor.

### 2.3.2 Arquitetura

As redes neurais artificiais podem ter suas estruturas de duas formas, com camadas simples ou múltiplas. Na camada simples, geralmente constitui-se de duas camadas, a

Figura 11 – Principais funções de ativação totalmente diferenciável: (a) Gaussiana; (b) Logística; (c) Tangente Hiperbólica.



Fonte: Elaboração do próprio autor.

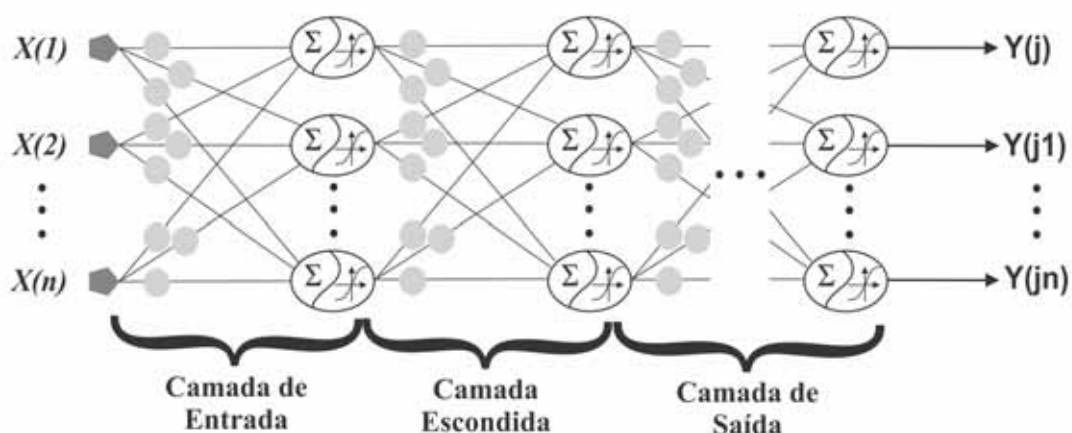
camada (1) de entrada e a camada (2) de saída. Em cada camada deve-se ter pelo menos um neurônio para fazer as interconexões entre as camadas.

A estrutura com múltiplas camadas é a mais frequentemente utilizada pela maior robustez em diversas aplicações. Esse tipo de RNA contém pelo menos uma camada escondida ou intermediária, além das camadas de entrada e de saída habituais.

A quantidade de camadas escondidas pode variar de acordo com a necessidade. Em cada RNA pode-se conter uma quantidade de neurônios e camadas diversificada, dependendo exclusivamente do tipo de aplicação onde será implementada.

O tipo de RNA mais popular é a MLP, que consiste em uma rede Perceptron multicamadas, com pelo menos uma camada intermediária ou escondida. Na Figura 12 ilustra-se a forma como pode ser organizada as redes neurais artificiais MLP.

Figura 12 – Rede neural perceptron de múltiplas camadas (MLP).



Fonte: Elaboração do próprio autor.

### 2.3.3 Treinamento

O treinamento da RNA é um processo que determina a capacidade de generalização e a funcionalidade da RNA de acordo com o ajuste de pesos em sua estrutura. Os dois

principais métodos de treinamentos são: o supervisionado e não supervisionado.

No método não supervisionado a RNA é ajustada automaticamente sem auxílio de informações desejadas, ou seja, não conhece o resultado ideal. Os ajustes são realizados automaticamente pela RNA apenas com base nas suas entradas.

O método supervisionado é o mais utilizado em RNA do tipo MLP. Esse treinamento baseia-se em um determinado número de amostras de entrada e saídas desejadas para ajustar os pesos da RNA. Geralmente, adota-se a análise de erro entre o resultado alcançado e o desejado, em conjunto com o gradiente descendente para o ajuste. Assim, no treinamento da RNA utiliza-se o conhecido algoritmo *backpropagation* junto ao *feedforward*.

Inicialmente, executa-se o *feedforward* para efetuar a soma ponderada das entradas com seus respectivos pesos atuais, para obter um limiar de ativação conforme a função de ativação definida. Isso é feito em todos os neurônios da rede, camada a camada até produzir as Saídas Parciais (SP) e final da RNA. No sentido contrário de execução, o algoritmo *backpropagation* faz a análise do erro quadrático médio, conforme a Equação (4), para cada neurônio até alcançar a tolerância imposta. O erro quadrático médio ( $E_{QM}$ ) é a diferença entre a saída do neurônio ( $Y_j^{k_n}$ ) da última camada ( $k_n$ ) e a saída desejada ( $\hat{Y}_j^{k_n}$ ), lembrando que  $j$  está relacionado ao índice do neurônio da última camada podendo variar de 1 a  $n$ .

$$E_{QM} = \frac{1}{n} \sum_{j=1}^n \left( \frac{1}{2} \sum_{j=1}^n (Y_j^{k_n} - \hat{Y}_j^{k_n})^2 \right) \quad (4)$$

A execução conjunta dos dois modos *feedforward* seguido pelo *backpropagation* consiste em uma época da RNA. Os critérios de parada podem estar relacionados ao erro ( $E_{QM}$ ) ou com a quantidade de épocas determinadas, ou seja, a quantidade de repetições em que completou-se um ciclo com a execução de um *feedforward* e *backpropagation*.

Para produzir o resultado esperado precisa-se comparar a saída da RNA com a desejada, em seguida ajustar os pesos de acordo com a resposta ideal à rede (convergência) ou o mais próximo do ideal, conforme o critério de parada do treinamento. Essa metodologia é conhecida como regra delta.

A regra delta é amplamente utilizada em RNA MLP, e o ajuste dos pesos no modo *backpropagation* é realizado para a execução da nova época ( $m$ ) ou ciclo a ser avaliado. Os pesos da próxima época ( $m+1$ ) é a soma dos pesos da época atual ( $m$ ) com o produto do  $\sigma$  da última camada e saída da camada anterior, conforme a Equação (5).

$$W_{qj}^{(m+1)k_n} = W_{qj}^{mk_n} + (\sigma_j^{k_n} * (Y_j^{k_{n-1}})) \quad (5)$$

Para determinar um novo conjunto de pesos, o principal fator é o valor de  $\sigma$  definido pela análise do erro em conjunto com gradiente descendente. O valor de  $\sigma$  é encontrado pela Equação (6), onde  $\alpha$  é a taxa de aprendizagem (TA) e  $\delta$  é definido na Equação (7). O  $\delta$  é encontrado a partir do erro  $E_{QM}$ , e  $\theta$  está relacionado a camada vigente. O valor de  $\theta$  corresponde a Equação (8) (SILVA; SPATTI; FLAUZINO, 2010).

$$\sigma_j^{k_n} = 2\alpha * \delta_j^{k_n} \quad (6)$$

$$\delta_j^{k_n} = E_j^{k_n} * \theta_j^{k_n} \quad (7)$$

A Equação (9) é utilizada para adquirir o erro em todas as camadas com exceção da última. A atualização dos pesos pelo algoritmo de *backpropagation* pode ser definida conforme as Equações (4), (5), (6), (7), (8) e (9) (SILVA; SPATTI; FLAUZINO, 2010).

$$\theta_j^{k_n} = \frac{1}{2} * ((1 + (Y_j^{k_n})) * (1 - Y_j^{k_n})) \quad (8)$$

$$E_j^{k_{n-1}} = \sum_{q=1}^n W_{qj}^{k_{n-1}} * \delta_q^{k_n} \quad (9)$$

Com os pesos adquiridos após o treinamento da RNA é necessária a execução final do modo *feedforward*. A execução com os pesos atualizados obtém os resultados com as novas amostras na validação da rede com a funcionalidade da aplicação. Geralmente o treinamento é realizado com um determinado conjunto de amostras. A execução do *feedforward*, após o treinamento, é feito com outro conjunto de amostras para a validação do sistema.

O algoritmo de treinamento supervisionado em alto nível de abstração utilizado neste trabalho é apresentado no Algoritmo 1. A entrada de dados ( $X_j^k$ ) corresponde aos componentes RGB da imagem atual. Na linha 4 determina a normalização das entradas (componentes da imagem) entre o intervalo [0:1] com números reais. As saídas desejadas também passam por esse processo de forma inversa.

Os resultados retornam aos seus valores inteiros, correspondentes aos componentes da imagem, na parte de avaliação dos resultados e para gerar as novas imagens segmentadas, este aspecto está descrito na seção 6.3. As imagens de entrada definidas no domínio do espaço por matrizes, são convertidas para vetores com tamanhos menores apenas

no treinamento. Os vetores não podem ultrapassar 70% das amostras e o restante das amostras são utilizadas para a validação do sistema.

A convergência do algoritmo ocorre quando uma das duas condições de guarda da linha 5 for verdadeira. A quantidade de repetições dos laços nas linhas 7 e 8 correspondem respectivamente, a quantidade de imagens de amostras de entrada e a quantidade de píxeis analisados para o treinamento em cada imagem. O modo *feedforward* e *backpropagation* são executados nas linhas 12 e 13, respectivamente.

---

**Algoritmo 1:** Treinamento da RNA com critério de parada baseado em erro ou épocas.

---

**Entrada:**  $X_j^k$  - Entradas,  $\hat{Y}_j^{k_n}$  - Saídas Desejadas  
**Saída:**  $Y_j^k$  - Equação (3)

```

1 início
2   Inicialização de variáveis;
3   Abrir Arquivos;
4   Normalização e Particionamento dos dados de treinamento e validação;
5   enquanto  $Epocas < NMaxEpocas \vee E_{QM} < ErroMax$  faça
6     Inicializar variáveis de saída desejada, erros e contador;
7     para  $ki \geq Vlr_{Inicial} \wedge ki \leq QtdeIMGs$  faça
8       para  $kp \geq Vlr_{Inicial} \wedge kp \leq QtdePíxeis$  faça
9         Zerar saídas parciais (SP) e erro da RNA;
10        Carregar as entradas;
11        Carregar as saídas desejadas;
12         $SP \leftarrow FeedForward(SP, W_{qj}^{(m+1)k_n});$ 
13         $[E_j^{k_n-1}, W_{qj}^{(m+1)k_n}] \leftarrow$ 
14         $BackPropagation(SP, W_{qj}^{(m+1)k_n}, E_j^{k_n-1}, \theta_j^{k_n}, \sigma_j^{k_n}, \delta_j^{k_n}, \alpha);$ 
15        Calcular a função de custo baseado no erro ;
16      fim
17    fim
18    Calcular erro médio quadrático - Equação (4);
19     $Epocas \leftarrow Epocas + 1;$ 
20    Salvar erros;
21    Salvar pesos;
22  fim
23 fim
```

---

A execução final da RNA é feita com a alteração das amostras de entrada e execução do modo *feedforward* apenas uma vez, conforme a funcionalidade da aplicação e respeitando o seguinte modelo:  $\forall f(x', y') \in A = \{f1(x, y), f2(x, y), \dots, fn(x, y)\}$ , onde  $x$  e  $y$  constituem a resolução da imagem  $f(x, y)$ .

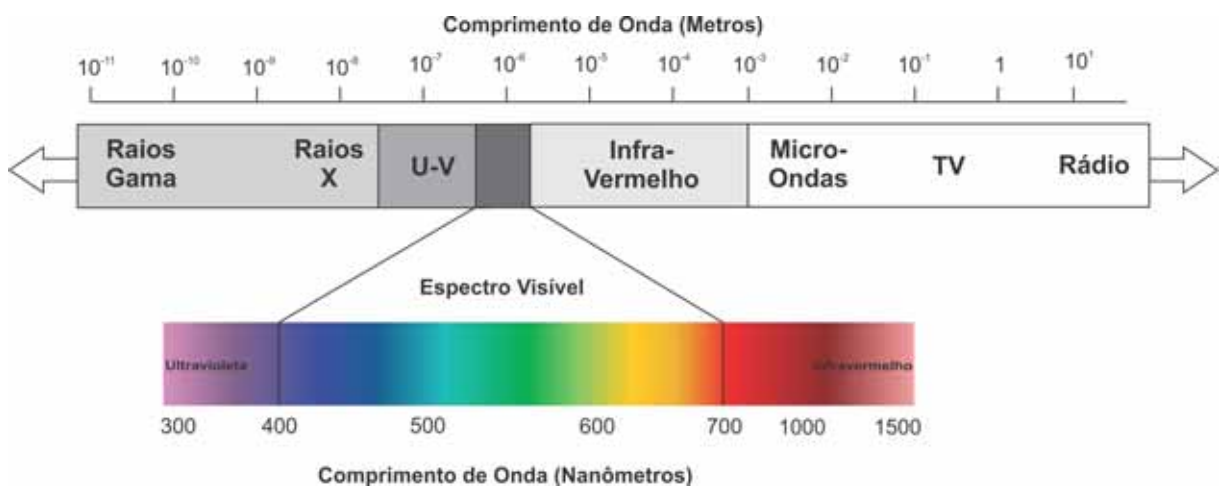
A função  $f(x, y)$  é definida pela Equação (10), apresentada na seção 2.4.2, e  $A$  é o conjunto de imagens de amostras utilizados. Os parâmetros  $x'$  e  $y'$  são os conjuntos

de píxeis das imagens  $f(x, y)$  que não participaram do treinamento. As definições apresentadas a partir do Algoritmo 1 estão relacionadas a execução da RNA com imagens, que representa o principal objetivo do trabalho. Porém, nos Capítulos 5 e 6 apresentam-se estruturas da RNA com diferentes tipos de entradas para execução dos sistemas.

## 2.4 Processamento Digital de Imagens

O processamento digital de imagens é um conjunto de fórmulas matemáticas e probabilísticas aplicadas em uma imagem. Muitas vezes, a visão humana é responsável por analisar e definir entre uma ou outra técnica para um determinado problema, geralmente se baseia em características visuais subjetivas. Nos seres humanos, a percepção visual é realizada pelo olho. O olho humano capta as ondas eletromagnéticas correspondentes ao espectro de luz visível. O espectro contínuo de cores, luz visível, foi comprovado de maneira prática por Isaac Newton (1666) com uma luz branca através de um prisma. A visão humana identifica objetos conforme a luz refletida pelo mesmo, isso devido a todo o resto de espectro ser absorvido pelo objeto. A faixa de espectro visível é estreita entre o violeta (Início do azul), menos intenso, até o vermelho, conforme apresentada na Figura 13 (MARQUES FILHO; VIEIRA NETO, 1999; GONZALEZ; WOODS, 2010).

**Figura 13 – Faixa de comprimentos de onda e o espectro visível**



Fonte: Gonzalez e Woods (2010).

A teoria principal do modelo adotado em imagens com cores aditivas RGB (*Red Green Blue*) é formulado pela hipótese de Young (1801), em que os Bastonetes são em maior número para dar uma imagem geral do campo de visão e estão relacionados à visão de luz escura. Já os Cones (células fotossensíveis da retina) são responsáveis pela definição de detalhes na imagem relacionados a visão de luz clara e se dividem em três classes R(*Red*), G(*Green*) e B(*Blue*). A partir das cores primárias pode-se formar qualquer outra cor, a princípio as cores secundárias são formadas por duas primárias em uma mistura aditiva.

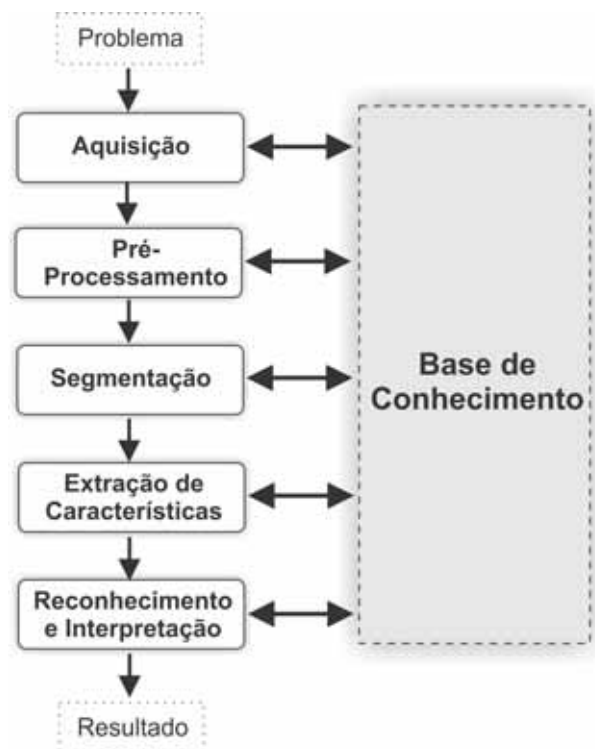
Outros modelos de cores podem ser formados pela mistura subtrativa. Três características importantes devem ser levadas em consideração no processamento de imagens (MARQUES FILHO; VIEIRA NETO, 1999; GONZALEZ; WOODS, 2010):

1. Base de dados de Qualidade;
2. Poder de processamento;
3. Capacidade de trabalhar em condições ou ambientes adversos.

#### 2.4.1 Etapas do PDI

O processamento digital de imagens (PDI) é dividido em 5 etapas fundamentais: Aquisição, Pré-Processamento, Segmentação, Extração de Características e Reconhecimento de Padrões ou Inferência. Dependendo da aplicação pode-se utilizar algumas ou todas as etapas definidas diante de uma base de conhecimento para auxiliar as etapas. A Figura 14 apresenta a relação entre as etapas fundamentais do PDI (MARQUES FILHO; VIEIRA NETO, 1999; PEDRINI; SCHWARTZ, 2007).

**Figura 14 – Etapas do processamento digital de imagens.**



Fonte: Pedrini e Schwartz (2007).

A aquisição da imagem é feita por meio de sensor óptico que transforma a luz em elétrons. Geralmente, esses sensores são fabricados por meio das tecnologias CMOS

(*Complementary Metal–Oxide–Semiconductor*) ou CCD (*Charge-Coupled Device*). A tecnologia CCD traz melhor qualidade e tem uma maior sensibilidade na captura da imagem, em contraste ao CMOS que tem baixa qualidade na imagem, porém os atrativos desse sensor é a baixa potência utilizada e dispositivos mais baratos (LITWILLER, 2001).

O pré-processamento da imagem consiste em melhorias na qualidade da imagem, retirando ruídos indesejáveis e modificações no contraste ou no brilho da imagem.

A segmentação é o processo para selecionar um ou mais elementos em uma imagem, e evidenciá-lo(s) resultando em uma nova imagem ou características do elemento a ser selecionado. Em determinadas aplicações, a seleção de partes ou objeto específico pode determinar o sucesso no seu reconhecimento. Depois do objeto selecionado, a etapa de extração de características define outras informações a respeito do objeto, para determinar limitações e fronteiras de uma determinada região ou conjunto de píxeis (GONZALEZ; WOODS, 2010). Por exemplo, a definição do contorno de uma região específica da folha que contém a deficiência da folha, delimitando a separação entre a parte saudável ou não da folha por meio de segmentos e descritores de fronteiras.

No reconhecimento de padrões ou inferência, relaciona a informação ao objeto, ou seja, atribui um significado ao caractere reconhecido. Um exemplo é verificar se uma determinada folha corresponde à uma cultura de planta específica.

#### **2.4.2 Características, Vizinhança e Conectividade**

A imagem pode ser definida como uma função bidimensional  $f(x, y)$ , a intensidade ou nível de cinza da imagem é  $f$ ,  $x$  e  $y$  são as coordenadas do plano. Uma imagem digital é formada por um conjunto finito de elementos relacionados em matriz  $N \times M$ . Para cada elemento da matriz, com suas coordenadas e valores de intensidade, pode-se definir um dos principais componentes de uma imagem digital, o píxel. Na aquisição e digitalização das imagens existem duas características essenciais, a iluminância ( $i(x, y)$ ) e refletância ( $r(x, y)$ ). Com uma imagem ruidosa ou com altas variações de luz ambiente ou cena, a segmentação e a extração das características de uma imagem pode-se tornar difícil o processamento da imagem por causa das variações nas características de iluminância e a refletância. Para não ocorrer essas variações, deve-se tomar os devidos cuidados na aquisição das imagens quanto à dinâmica na variação de luz ambiente na cena, a iluminância, e a luz refletida pelo objeto ou elemento contido em uma cena, a refletância (GONZALEZ; WOODS, 2010).

Uma imagem é gerada com suas devidas intensidades de forma proporcional à uma fonte real de energia irradiada, e os valores de  $f(x, y)$  devem ser diferentes de zero e infinito. O produto da iluminância e refletância formam a imagem  $f(x, y)$  conforme Equação (10)

e todos esses fatores influenciam na digitalização da imagem (GONZALEZ; WOODS, 2010).

$$f(x, y) = i(x, y) * r(x, y) \quad (10)$$

onde

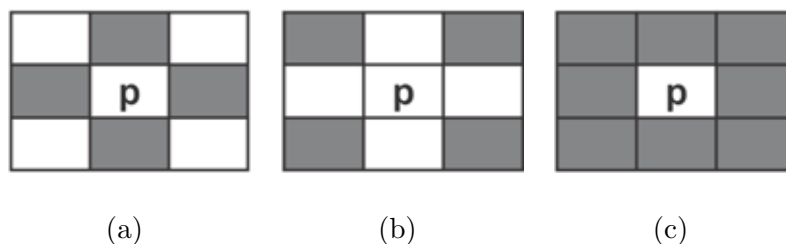
$$0 < i(x, y) < \infty \quad (11)$$

$$0 < r(x, y) < 1 \quad (12)$$

A digitalização da imagem é a amostragem ou quantização do sinal analógico, que consiste no processo de discretização espacial da imagem transformando em uma matriz de píxeis. Se a imagem foi digitalizada em níveis de cinza, cada píxel corresponde a um elemento da matriz com um número inteiro que define a intensidade de cinza daquele píxel. A qualidade da imagem pode ser definida pela resolução e a quantidade possível de níveis de cinza da imagem.

A partir das coordenadas e dos valores de intensidade de uma imagem, pode-se determinar duas propriedades importantes: a vizinhança e a conectividade. Na Matriz 2D correspondente a imagem, um determinado píxel com coordenadas (x,y) possui no máximo 8 vizinhos adjacentes ou “vizinhança-8”. A “vizinhança-4” são os píxeis adjacentes de forma horizontal e vertical a um determinado píxel. A vizinhança diagonal está relacionado com os píxeis adjacentes nas diagonais primária e secundária. A Figura 15 apresenta os três tipos de vizinhança conforme a disposição dos píxeis na imagem. A conectividade é o agrupamento de píxeis que tenham relação por meio da vizinhança e de um determinado intervalo de valores de intensidade de cinza.

**Figura 15 – Tipos de Vizinhança: (a) vizinhança-4; (b) vizinhança diagonal; (c) vizinhança-8.**



Fonte: Pedrini e Schwartz (2007).

O uso do termo níveis ou escala de cinza é equivalente a uma imagem que constitui

de cor cinza em diversas intensidades de branco à preto. Geralmente a faixa de níveis de cinza são definidas, por causa do hardware, em múltiplos de  $2^n$ . As imagens coloridas podem ser definidas conforme o modelo de cores adotado para permitir uma padronização de formatos em um sistema de coordenadas para facilitar o emprego em hardware.

Os principais modelos de cores são: RGB, o sistema subtrativo CMYK (*Cyan Magenta Yellow black*), YIQ (padrão NTSC<sup>3</sup> de TV em cores) e HSI (*Hue Saturation Intensity*). A aplicação de destino é um fator importante na definição do tipo de modelo de cor. Geralmente estão relacionados ao hardware em que será utilizado. O modelo RGB para televisores coloridos e máquinas fotográficas, o CMY e CMYK para impressoras e o HSI é utilizado no processamento de imagens pela separação de componentes importantes da imagem para a visão humana (MARQUES FILHO; VIEIRA NETO, 1999).

Na visualização de imagens coloridas do tipo HSI, três características importantes devem ser observadas: o Brilho, a Matiz e a Saturação. O brilho subjetivo é a intensidade de luz incidente no olho humano com características logarítmicas. A matiz é o comprimento de onda predominante em uma faixa limitada do espectro. A saturação é o grau de mistura entre a matiz original e a luz branca. Nas imagens coloridas do tipo RGB, a cromaticidade é a união entre a matiz e a saturação, e os percentuais RGB de uma cor são chamados de coeficientes tricromáticos. A intensidade de luz em cada cor (RGB) geralmente é normalizada entre 0 e 1 e a soma dos três coeficientes (rgb) é 1 (MARQUES FILHO; VIEIRA NETO, 1999; GONZALEZ; WOODS, 2010).

### 2.4.3 Histogramas

No processamento de imagem existem diversas técnicas utilizadas na seleção de objetos em uma cena. A segmentação de imagens por histogramas é frequentemente empregada na identificação do objeto. Diversas outras técnicas podem auxiliar na segmentação de imagens, como filtros, vizinhança, conectividade, operações de convolução, aritméticas e lógicas, entre outros.

O histograma é um método estatístico que define a quantidade de píxeis em todos os níveis de cinza possíveis em uma imagem. Diversas características podem ser obtidas com esse método. As mais importantes estão relacionadas a qualidade da imagem em relação ao seu brilho e contraste, e a segmentação das mesmas. O Histograma ( $H_k$ ) da imagem é definido por  $H_k = PI_k$ , onde  $k = 0, \dots, L - 1$  níveis de intensidade de cinza e  $PI_k$  é o número de elementos (Píxel) com determinada intensidade  $k$ . Os histogramas são utilizados principalmente para equalização e segmentação de imagens.

A equalização de histogramas é uma técnica muito utilizada na correção de contraste

---

<sup>3</sup>NTSC - *National Television Standards Committee*

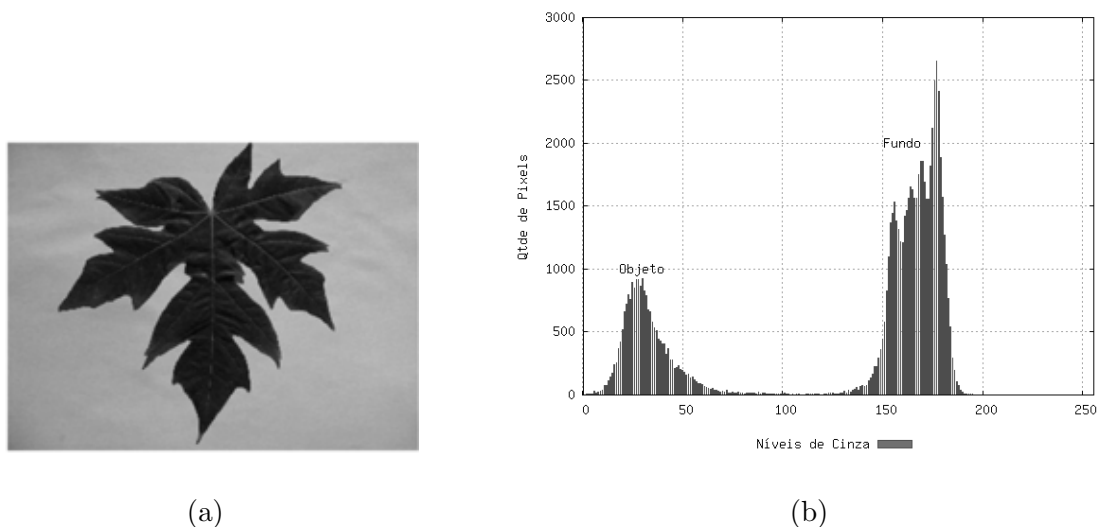
de uma imagem. Essa técnica tenta deixar o histograma de forma contínua, ou seja, definir valores para tentar cobrir o máximo de níveis de intensidade da imagem (MARQUES FILHO; VIEIRA NETO, 1999).

O histograma possibilita também a segmentação de regiões de imagem ou a redução dos níveis de cinza. Por exemplo, na seleção, ou transformação da imagem em determinada região.

A limiarização é uma técnica muito utilizada na separação do fundo e objeto da imagem, por meio de dois picos do histograma, conforme apresentado na Figura 16-(b)<sup>1</sup>. Tal histograma corresponde de forma visível na identificação do fundo e o objeto na imagem na Figura 16-(a).

A limiarização de uma imagem consiste na identificação de dois agrupamentos em intervalos de intensidade distintos localizados entre os picos do histograma. Geralmente com um vale entre os agrupamentos, correspondente ao limiar (T) global da imagem, que separa os dois agrupamentos. Nesse caso, com fundo claro e objeto escuro, ao contrário também é possível fazer a identificação. Na Figura 16-(a) pode-se identificar o limiar em aproximadamente o valor 100 no eixo  $x$ , os níveis de cinza (L) da imagem correspondem a  $2^8 = 256$  ou 8 bits.

**Figura 16 – Limiarização por histogramas: (a) Imagem original; (b) Histograma.**



Fonte: Elaboração do próprio autor.

A limiarização com apenas um limiar global possibilita a transformação da imagem, por exemplo, uma imagem em escala de cinza para binária. Assim, a imagem original ( $f(x, y)$  em escala de cinza) sofre uma transformação ( $g(x, y)$ ) em uma determinada

<sup>1</sup>A ferramenta GNUPLOT na versão 4.6 foi utilizada na confecção de todos os gráficos e histogramas apresentados na seção 2.4 e Capítulos 4, 5 e 6.

região. Respeitando o limiar global da imagem pode-se alterar o objeto da segmentação para preto e o resto da imagem em branco. A Equação (13) apresenta a forma de segmentação por um limiar global. Quando o valor de  $T$  é alterado no processamento, a imagem é dita como variável e o limiar local também é variável, mas está relacionado com a sua vizinhança. Por exemplo, o limiar é definido como parâmetro local e um determinado intervalo de intensidades é o limite para encontrar uma determinada região da imagem (GONZALEZ; WOODS, 2010).

$$g(x, y) = \begin{cases} 0 & \text{se } f(x, y) > T \\ 255 & \text{se } f(x, y) \leq T \end{cases} \quad (13)$$

A definição de limiares múltiplos é utilizada quando a imagem contém várias classes predominantes, se  $K$  é o número de classes,  $K - 1$  é o número de limiares a serem encontrados. No histograma é definido por mais de dois picos de intensidade e a quantidade de limiares corresponde ao número de picos menos um.

A obtenção de um limiar global ótimo para um histograma com os picos e o vale bem definidos, torna a segmentação mais simples. Porém, com a inserção de ruídos, iluminação diferenciada, e textura complexa pode trazer grandes dificuldades para encontrar um limiar ótimo. Por exemplo, em uma situação em que o histograma tem apenas um pico e a imagem contém plano de fundo e objeto. Nesse caso, torna-se mais complexa a determinação de um limiar ótimo, maiores detalhes são apresentados no Capítulo 6. Na seção 2.4.4 apresenta-se um método eficiente para obtenção de limiares ótimos na imagem, mesmo com histogramas definidos com apenas um pico.

#### 2.4.4 Método de Otsu

Existem diversos métodos para determinar o limiar ótimo de um histograma, o mais conhecido é o método de Otsu (OTSU, 1979). Este método verifica duas características baseadas no nível de intensidade da imagem, a probabilidade de ocorrência de um nível de intensidade em um determinado píxel e a variância de intensidades da imagem, que possibilitam diferenciar as classes pertencentes a uma imagem, por exemplo, o objeto e o fundo da imagem. Além disso, o método pode ser aplicado com limiares múltiplos e em partes determinadas de uma imagem de forma independente na identificação de objetos.

O método de Otsu possibilita a determinação da variância de níveis de intensidade entre distintas classes em uma imagem. Este método pode ser utilizado na segmentação ou extração de características de objetos em uma determinada cena.

Para obter um limiar ótimo a partir do método de Otsu deve-se implementar duas

partes do método, uma para obter valores estatísticos iniciais e outra para as variâncias entre as classes. A primeira parte inicialmente adquire o histograma da imagem para servir de base para encontrar seis valores estatísticos, Probabilidade de Ocorrência ( $PO_k$ ), Soma Acumulada da Imagem ( $S_a$ ), Intensidade Média Global da imagem ( $M_g$ ), Intensidade Acumulada ( $I_k$ ) e Soma Cumulativa das Probabilidades da Classe 1 ( $P1_k$ ) e Classe 2 ( $P2_k$ ), respectivamente conforme as Equações (14), (15), (16), (17), (18) e (19) (GONZALEZ; WOODS, 2010).

$$PO_k = \frac{H_k}{NPI} \quad (14)$$

$$S_a = \sum_{k=0}^{L-1} S_a + PO_k \quad (15)$$

$$M_g = \sum_{k=0}^{L-1} M_g + ((k - 1) * PO_k) \quad (16)$$

$$I_k = \sum_{k=0}^{L-1} I_{k-1} + ((k - 1) * PO_k) \quad (17)$$

$$P1_k = \sum_{k=0}^{L-1} P1_{k-1} + PO_k \quad (18)$$

$$P2_k = \sum_{k=0}^{L-1} 1 - P1_k \quad (19)$$

Os valores de k correspondem aos níveis de intensidade de cinza da imagem e NPI é o Número de Píxeis da Imagem. A primeira parte do método está relacionada a probabilidade de ocorrência de determinado nível de intensidade definida por distintas características, como os somatórios e médias do tipo acumulada, local e global.

A segunda parte do método define as variâncias de intensidade encontradas na imagem, a Variância Global ( $\sigma_g$ ) e Variância entre as Classes ( $\sigma_c$ ). Outra importante característica é a Medida de Separabilidade entre as Classes ( $MS_c$ ). Esses três valores

podem ser encontrados respectivamente pelas Equações (20), (21) e (22).

O valor máximo de  $\sigma_c(k)$  para  $k$  variando entre  $(0:L-1)$  é a intensidade que caracteriza o limiar ótimo (T) da imagem e a separação entre duas classes analisadas. Por exemplo, se for necessário encontrar uma terceira classe, o método é repetido apenas com a parte da imagem que não contém essa terceira classe.

$$\sigma_g = \sum_{k=0}^{L-1} (k - M_g)^2 * PO_k \quad (20)$$

$$\sigma_c(k) = \frac{(M_g * P1_k - I_k)^2}{P1_k * (1 - P1_k)} \quad (21)$$

$$MS_c(k) = \frac{\sigma_c(k)}{\sigma_g} \quad (22)$$

Para algumas aplicações, o método de Otsu pode resolver o problema de inexistência de vales em um histograma. O método faz a análise com base em valores estatísticos da imagem, e pode reduzir o consumo de memória por trabalhar com vetores de intensidades e não com a imagem inteira durante toda a execução. Com isso, a utilização da limiarização torna-se um atrativo para o desenvolvimento de um sistema embarcado para a computação dessa aplicação em hardware. Porém, a investigação da eficiência do método deve ser analisada em cada aplicação.

Uma das principais preocupações na segmentação e no processamento digital de imagens é a utilização de imagens ruidosas e com iluminação não uniforme e variável. Dependendo da anomalia na imagem, existem métodos para contornar esses problemas por meio de suavização por filtros, particionamento da imagem, detecção de linhas e bordas, entre outros. Para determinar a necessidade da utilização de uma técnica e corrigir tais problemas, o ideal é primeiramente fazer as aquisições das imagens em um local adequado ou próximo da realidade da aplicação e verificar a incidência de algum tipo de anomalia, e depois determinar a melhor técnica de correção ou a utilização de múltiplas técnicas.

No Capítulo 6 serão investigados os métodos para segmentação de imagens e filtros lineares no auxílio da segmentação. No Capítulo 3 serão investigados os trabalhos relacionados as três partes principais do trabalho, a arquitetura da RNA, a aproximação da função de ativação não linear em hardware e os métodos de PDI empregados em agricultura de precisão.

### 3 MÉTODOS E FERRAMENTAS DE APOIO

Este capítulo contém uma revisão de estudos realizados nas principais vertentes que definem este trabalho. O objetivo é verificar as metodologias e técnicas apresentadas na literatura com o intuito de analisar os resultados, as principais características e dificuldades no desenvolvimento da pesquisa deste trabalho. Três principais abordagens foram elencadas: duas relacionadas à implementação de RNA em hardware, seção 3.1, definidas para a computação dos neurônios e da AF, respectivamente nas Seções 3.1.1 e 3.1.2; e a terceira abordagem traz os principais métodos computacionais para a AP, contido na seção 3.2.

#### 3.1 Redes Neurais Artificiais em Hardware

Na implementação de redes neurais artificiais, em hardware, deve-se determinar algumas características importantes no seu desenvolvimento como: (i) a necessidade de treinamento online; (ii) a topologia da RNA; (iii) o tipo de arquitetura; (iv) o método e o tipo de função de ativação; (v) a forma de entrada e saída dos dados da RNA; (vi) o modelo de representação dos dados no sistema.

Com o conhecimento do tipo de aplicação na qual deve ser empregada a RNA e a sua prévia implementação em software, muitas características tornam-se mais fáceis de serem definidas no sistema. Por exemplo, a necessidade da RNA ser dinâmica ou estática, respectivamente, com a inclusão ou não de treinamento *online* (i); uma topologia adequada com a quantidade de camadas, neurônios e conexões (ii); a escolha do tipo da AF (iv) conforme os resultados prévios alcançados e a disponibilização dos dados de entrada e saída (v).

Em contato com o desenvolvimento do hardware, o projetista depara-se com os principais desafios dos pesquisadores da área: a representação de dados, a computação dos dados no neurônio e a aproximação da AF. A primeira característica é sobre o modelo de representação dos dados no sistema (vi), essa escolha tem influência sobre o modo adotado para a computação dos neurônios na RNA. A forma de computação dos dados no neurônio, segunda característica, pode ser realizada de diversas formas, dependendo da quantidade de neurônios e entradas utilizadas na RNA. A terceira característica está no método a ser

empregado na aproximação da AF (iv), geralmente uma função não linear (FNL). Essas duas características devem ser cuidadosamente escolhidas, sempre verificando o *tradeoff* entre desempenho e área utilizada na FPGA. Neste trabalho adotou-se a representação de dados conforme o padrão, IEEE Std 754 (32 bits), e a função de ativação Tangente Hiperbólica é aproximada por métodos híbridos, apresentada no Capítulo 5.

A entrada e a saída do sistema são definidas conforme a necessidade de alimentação e aquisição dos dados, seja para comparação dos resultados ou para servir de entrada a outro sistema, por exemplo, um software em um computador. Na próxima seção são descritos alguns trabalhos relacionados à pesquisa do sistema da RNA em hardware, e trazem características importantes a serem levantadas para futuros pesquisadores da área. As duas principais vertentes estão relacionadas às arquitetura e computação dos dados no neurônio, e à forma de implementação da função de ativação com não linearidade.

### 3.1.1 *Arquitetura e Computação dos Dados no Neurônio*

Para representar o neurônio artificial é comum o desenvolvimento de diversas arquiteturas que empregam tecnologias baseadas em SoC, com o intuito de obter um hardware eficiente que encontre um limiar entre desempenho e área utilizada. Em Savich, Moussa e Areibi (2007) foram apresentados três tipos de modelos de arquiteturas em hardware para o neurônio artificial, a serial, a parcialmente paralela e a completamente paralela.

Em Al-Kazzaz e Khalil (2008) todas as arquiteturas desenvolvidas foram baseados em três abstrações: (i) software: aplicação sobre *softcore*, (ii) Codesign: comunicação entre processamento em software e hardware dedicado para AF, (iii) hardware: hardware totalmente dedicado.

A arquitetura serial ou MAC (*Multiply and Accumulate*) é utilizado frequentemente para reduzir área de *hardware* para o neurônio (FERREIRA; BARROS, 2010; ZHANG; CHEN, 2011; NEDJAH; SILVA; MOURELLE, 2012; TISAN et al., 2009). Em contrapartida, neste trabalho empregou-se a arquitetura parcialmente paralela para obter um desempenho satisfatório com FPU (*Floating Point Unit*) e o uso de área não foi priorizado. No Capítulo 5 apresentam-se mais detalhes.

Em aplicações que envolvem RNAs em projetos com FPGA, frequentemente as FPU são desenvolvidas de forma dedicada, em ponto fixo (BENJEMMA et al., 2012; BAHOURA; PARK, 2011; NEDJAH; SILVA; MOURELLE, 2012) com utilização de arquitetura MAC. Nessa abordagem o uso da RNA fica restrito ao projeto, dificultando o seu reuso e prejudicando a precisão com o truncamento realizado na otimização da FPU. Geralmente, a particularização é feita com diferentes representações de números e caminhos de cálculos

dificultando a compreensão (DAMAK; KRID; MASMOUDI, 2008; LEE; KO, 2006).

A obtenção de um neurônio artificial otimizado conduz a uma RNA mais eficiente, a busca pela computação dos dados no neurônio e da AF de maneira que satisfaça os requisitos de desempenho, área e precisão torna a implementação de RNA mais complexa. Na computação dos neurônios utilizam-se diversas técnicas, geralmente com ponto fixo.

Em Ferreira e Barros (2010) os autores fazem a computação dos neurônios por meio do produto de matrizes. O processamento é feito em uma única entrada com todos os seus pesos (coluna) e depois por meio de deslocadores e somadores adquire a saída parcial do neurônio. Isso é repetido para todas as entradas. O número de multiplicadores e somadores é dependente da quantidade de entradas. Nesse trabalho os autores implementaram 3 RNAs, uma para aproximação de uma função senoide e duas para classificação (iris e semeion) adquiridas por repositório público, respectivamente com 6, 14 e 20 neurônios, alcançando desempenhos de 34, 26 e 36 vezes mais rápidos do que as correspondentes em software para cada aplicação. A implementação da função sigmoide é feita por LUT utilizando um modo particular para acesso aos valores armazenados, em todos os casos com 128 elementos.

De forma semelhante na computação dos dados, Ali e Mohammed (2010) apresentam um sistema digital com arquitetura definida por meio de multiplicadores e somadores no neurônio para efetuar a soma ponderada. Assim, a quantidade de multiplicadores é igual ao seu número de conexões, e número de somadores são encontrados pelo número de conexões da camada anterior menos um. A arquitetura desenvolvida é completamente paralela, rápida e inflexível, com 4 neurônios (3-1) que utiliza apenas 3 multiplicadores e 2 somadores. A função sigmoide foi implementada em quatro partes, duas regiões de saturação e duas regiões entre as transições por meio de função não-linear simples de segunda ordem.

Em Nedjah, Silva e Mourelle (2012) foram exploradas as otimizações do neurônio e da camada para obter um módulo simples e reutilizável. O reuso do módulo possibilita construir RNAs mais complexas utilizando o conceito de camadas virtuais. A representação dos dados é feita por meio de fração com números inteiros para facilitar os cálculos em sistemas digitais. A função de ativação logística sigmoide é aproximada por polinômio quadrático com seleção de polinômios dependendo do intervalo de entrada.

No trabalho de Benjemmaa et al. (2012) foram utilizados ponto fixo e complemento de 2 na implementação da RNA em FPGA. Nesse trabalho, um robô inteligente que detecta e desvia de obstáculos foi controlado por meio de vários sensores instalados no veículo. Dois métodos foram analisados para a implementação da sigmoide, com LUT para valores codificados em 14 bits e polinomial, e com um polinômio de ordem 7 avaliado em um diagrama para possibilitar a implementação em paralelo.

Outro fator importante no desenvolvimento do sistema é o uso de hardware dedicado para aumentar o desempenho da aplicação e diminuir o uso de área lógica da FPGA. Zhang e Chen (2011) evidenciam o uso de DSPs, em arquitetura MAC, para fazer o processamento do neurônio. Os pesos, LUT, entradas e saídas da RNA na FPGA são armazenados por meio de memória dedicada (BRAM - Block RAM). O número de recursos lógicos na arquitetura é reduzido, com o uso de BRAM, além de trazer desempenho adequado para a aplicação. O uso de multiplicadores dedicados é um fato comum entre os pesquisadores (AL-KAZZAZ; KHALIL, 2008), e obrigatório principalmente quando se adota uma arquitetura paralela.

Em contrapartida, no desenvolvimento de RNA com ponto flutuante (IEEE Std 754) padroniza o sistema tornando-se mais fácil e ágil a manutenção, reutilização e expansão da arquitetura. Os projetos de hardware usando HDL tem maior tempo de desenvolvimento quando comparado a linguagem de alto nível, mesmo com uso ferramentas CAD avançadas. Por isso, o desenvolvimento de um sistema portátil e de fácil reutilização é essencial na sua construção.

Lee e Ko (2006) desenvolveram um sistema de detecção de face utilizando RNA, cuja parte de reconhecimento pode ser acessada por diferentes dispositivos. No trabalho, usaram uma RNA MLP contendo duas camadas escondidas com 25 e 6 nodos. Os dados da face no nó de entrada é calculado em ponto flutuante de diversas precisões e arquitetura MAC. A função sigmoide foi implementada com 3 diferentes métodos, Taylor, Polinomial, CORDIC (*COordinate Rotation DIgital Computer*), e escolheram o método de Taylor para utilizar no trabalho.

Existem trabalhos que fazem a implementação particular de ponto flutuante em conformidade com o IEEE Std 754, mas com precisão não determinada por este padrão ou seja reduzida. Em Damak, Krid e Masmoudi (2008) é realizada a detecção de bordas em imagens por meio de uma RNA MLP em FPGA com formato de ponto flutuante e 9 bits de precisão. A RNA tem 4 neurônios na camada escondida e função de ativação utilizada foi a Rampa dedicada. Os resultados da RNA são comparados ao método de processamento digital de imagens (*canny*).

Diversos cenários com RNAs complexas foram desenvolvidos por Leboeuf, Muscedere e Ahmadi (2011), utilizando ponto flutuante e treinamento supervisionado. Algumas das variações nas implementações são o tipo de treinamento, a topologia da RNA, o conjunto de dados reais para validação, o erro máximo de aproximação, além dos métodos de aproximação da função de ativação (LUT, RALUT (*Range Addressable Look-Up Table*) e Híbrido), maiores detalhes na próxima seção. Resultados satisfatórios de erro e desempenho foram adquiridos na simulação pelo Synopsys *Design Compiler* (SDC) e sintetizado para um hardware CMOS 0,18 $\mu$ m, com arquitetura baseada na AF RALUT,

e os treinamentos LM (Levenberg-Marquardt) e RPROP (*Resilient Backpropagation*).

Uma abordagem distinta, proposta por Santos et al. (2011), foi a utilização de *softcore* (NIOS II) para a execução de uma RNA *feedforward* em ponto flutuante e método híbrido para a Função de Ativação, foram obtidos resultados satisfatórios em comparação ao método sem a utilização da instrução customizada (Híbrido).

Uma análise mais específica na comparação entre ponto fixo e ponto flutuante com diversas precisões é feita por Savich, Moussa e Areibi (2007). As análises realizadas entre as implementações mostraram que o uso de ponto flutuante em hardware tem um custo em área e desempenho. Os autores efetuaram diversas comparações de desempenho e área entre as implementações, porém a comparação da precisão (dados de erros) entre os sistemas não foram feitas entre as mesmas representações em ponto fixo e flutuante. A análise da precisão, que seria o fator mais importante para as implementações em ponto flutuante não foram feitas, prejudicando a análise desse tipo de implementação.

Com o advento de FPGAs com grande capacidade de recursos lógicos e dedicados, o desenvolvimento de tecnologias que priorizam sistemas padronizados, com o intuito de facilitar o reuso e a expansão da automação de processos em *hardwares* dedicados, torna-se prioritários para ampliar o número de aplicações em computação reconfigurável. Assim, neste trabalho adota-se o uso de ponto flutuante de precisão simples na computação dos dados no neurônio e diversas formas de aproximação da AF Tangente Hiperbólica. No Capítulo 5, mostra-se que mesmo com uma plataforma de baixo custo em FPGA é possível a implementação padronizada em ponto flutuante da RNA. A comparação dos trabalhos investigados na implementação de RNA em hardware e algumas de suas características podem ser verificados na Tabela 3.

### **3.1.2 Função de Ativação em Hardware**

O desenvolvimento de arquiteturas de hardware para representação de diversas funções não lineares é um desafio aos pesquisadores. A implementação direta da AF (Tangente Hiperbólica - TH) é inviável, por ser uma série exponencial infinita. Diversos métodos são utilizados no emprego eficiente de não linearidade em hardware, geralmente visam encontrar um equilíbrio entre área, desempenho e precisão.

Os métodos frequentemente utilizados na implementação da AF TH em dispositivos reconfiguráveis (FPGA) são: Polinômios com diferentes ordens, mapeamento por LUT, divisão em partes lineares (PWL) e Híbrido.

Tabela 3 – Comparação entre os trabalhos na implementação da RNA em hardware

Ref.'s	Características							
	Dados <sup>17</sup>	Método AF <sup>17</sup>	Qtde Neurônios	Aplicação	Área	Plataforma	Erro	Desempenho
Ref. <sup>1</sup>	FXP	LUT	2-8-2	-	39 CLBs	Spartan 3	-	-
Ref. <sup>2</sup>	FXP	Linear	35-10-26	Reconhecimento de Caracter	-	Virtex-2	-	-
Ref. <sup>3</sup>	FLP	Taylor	25-6-2	Detector de Face	4438 Slices	Spartan 3 - 1500	0	1,7 ms
Ref. <sup>4</sup>	FXP e FLP	Linear	2-2-1/outros	XOR	4149 Slices	Virtex-2	-	-
Ref. <sup>5</sup>	FXP	Vários	12-Na-Na	-	170 Slices	Spartan 3E	-	7 Cycles Neuron
Ref. <sup>7</sup>	FXP	POTB	10-20-3	Aprox. Func. NL	30 CLBs	Xilinx	-	1,31 ms
Ref. <sup>8</sup>	FLP*	Linear	Na-4-Na	Reconhecimento de Imagens	1779 Slices	Virtex-2	3,91%	60,99 mhz
Ref. <sup>9</sup>	-	Taylor	3-2-2	Processos Industriais	63%	Simulink (Spartan 2/3)	-	-
Ref. <sup>10</sup>	FXP	LUT	6:14:20	Várias	4.591 ALUTs	Stratix III	0,0021	10, 137 $\mu$ s
Ref. <sup>11</sup>	FXP	Poli	3-1	-	-	FPGA	-	-
Ref. <sup>12</sup>	FXP	LUT	220-24-10	Reconhecimento de Caracter	89 Slices	Virtex-4	-	21,5 ms
Ref. <sup>13</sup>	FLP	Vários	39;17;69	Várias	253 $\mu$ m <sup>2</sup>	CMOS0, 18 $\mu$ m	1%	0,8 ns
Ref. <sup>14</sup>	FLP	Híbrido	4-4-3	Iris	10.984 LEs	DE2-Cyclone II	MSE-0,015	7,1 ms
Ref. <sup>15</sup>	FXP	Poli	3-2-1/outros	-	6 Slices	Virtex-5	-	3,33ns
Ref. <sup>16</sup>	FXP	LUT/Poli	10-7-1	Controle de rodas robô	11,234 LEs	DEII/Stratix III	-	-

<sup>1</sup> (BONNICI et al., 2006)<sup>2</sup> (HOFFMAN et al., 2006)<sup>3</sup> (LEE, KO, 2006)<sup>4</sup> (SAVICH; MOUSSA; AREIBI, 2007)<sup>6</sup> (AL-KAZAZ; KHALIL, 2008)<sup>7</sup> (HAVEI; VLCEK, 2008)<sup>8</sup> (DAMAK; KRID; MASMOUDI, 2008)<sup>9</sup> (KHODJA; KHELDOUN; REFOUFI, 2010)<sup>10</sup> (FERREIRA; BARROS, 2010)<sup>11</sup> (ALI; MOHAMMED, 2010)<sup>12</sup> (ZHANG; CHEN, 2011)<sup>13</sup> (LEBOEUF; MUSCEDERE; AHMADI, 2011)<sup>14</sup> (SANTOS et al., 2011)<sup>15</sup> (NEDJAH; SILVA; MOURELLE, 2012)<sup>16</sup> (BENJEMMA et al., 2012)<sup>17</sup> FLP - Floating Point; FXP - Fixed Point; POTB - Power-Of-Two-Based

A implementação direta de modelos matemáticos por meio de **polinômios** com ordens elevadas torna-se impraticável devido ao alto custo na utilização de recursos computacionais e baixo desempenho. Entretanto, o uso da série de Taylor é popular com termos de 4<sup>a</sup> (LEE; KO, 2006) e 5<sup>a</sup> (KHODJA; KHELDOUN; REFOUFI, 2010) ordem. Porém, mesmo com termos de baixa ordem existe as mesmas desvantagens da implementação direta. A principal vantagem desses métodos está na alta precisão alcançada na aproximação da função não linear (LIN; WANG, 2008).

Um método muito utilizado na implementação de funções não lineares em hardware é por meio de LUT, principalmente pela simplicidade no desenvolvimento e desempenho satisfatório. O aumento da precisão no método baseado em LUT clássica exige maior utilização de área no hardware, assim o uso de memória dedicada na FPGA é necessário para utilização de poucos recursos lógicos. A LUT clássica é definida por uma entrada ser relacionada apenas a uma saída ou resposta da função a ser aproximada. Já o método RALUT, contém um conjunto de entradas endereçadas à mesma saída, tornando o intervalo de entrada da função mais abrangente para possibilitar o desenvolvimento de tabelas menores, sem perdas de precisão.

Uma comparação entre as aproximações LUT e RALUT é apresentada em Leboeuf et al. (2008), que utilizaram-se de um limiar baseado no erro máximo até 0,04 e 0,02 com respectivos 9 e 10 bits de endereçamento para as entradas e saídas dos dois métodos respectivamente. Com esses valores de erros máximos, na LUT são necessários 512 e 1024 pontos, já com a RALUT necessitam-se de 61 e 127 pontos, respectivamente, para 0,02 e 0,04. A implementação foi sintetizada usando o SDC para o TSMC (Taiwan *Semiconductor Manufacturing Company*) CMOS 0,18 $\mu$ m.

Um outro método para aproximação da função de ativação Tangente Hiperbólica é baseado em PWL e LUT, visando economia em área do hardware e alto desempenho é desenvolvido por Namin et al. (2009). O método utiliza dois segmentos lineares como comparador do sistema e um módulo RALUT para tornar mais eficiente o endereçamento das entradas com as suas saídas. Um módulo subtrator, subtrai a saída do módulo comparador com a saída do módulo RALUT, para obter a saída da AF Tangente Hiperbólica.

O método PWL busca reduzir a complexidade da não linearidade da função utilizando distintos agrupamentos em um conjunto de elementos com domínio linear. O método PWL divide a resposta da função em uma determinada quantidade de partes lineares, com a intenção de simplificar a complexidade da função e reduzir área. Dependendo da técnica utilizada pode comprometer a precisão do sistema.

Os métodos clássicos que empregam PWL em hardware na aproximação não linear são: A-law (MYERS; HUTCHINSON, 1989), Alippi (ALIPPI; STORTI-GAJANI, 1991) e PLAN

(AMIN; CURTIS; HAYES-GILL, 1997). Em Tisan et al. (2009) são desenvolvidos os métodos clássicos para aproximação da AF com a ferramenta de síntese proprietária (Xilinx) acoplada ao Simulink do Matlab. Os principais métodos são a aproximação A-law, cuja curva modificada, da respostas da AF, é obtida com o gradiente de cada segmento expresso por meio de potência de 2 (POTB - *Power-Of-Two-Based*). A aproximação de Alippi, seleciona um conjunto de pontos de interrupção da primeira derivada e define a função pela soma de números na potência de 2. O PLAN utiliza portas digitais para compor a função sigmoide e a definição de equações de reta. A comparação entre os métodos clássicos analisados concluiu que a melhor aproximação é dada pelo método PLAN com quantidade maior de neurônios, com menor quantidade de neurônios escolhe-se a LUT.

Uma comparação dos métodos clássicos em PWL é realizada por Tommiska (2003), que propõe um método totalmente combinacional. A implementação proposta utilizou simplificação de expressões booleanas, com *McBoole Logic Minimiser* (DAGENAIS; AGARWAL; RUMIN, 1986), para obter um circuito combinacional otimizado com diversas quantidades de entradas, saídas e precisão. No trabalho de Havel e Vlcek (2008) é realizada a comparação dos métodos clássicos de aproximação da AF com o método proposto, baseado em potência de 2 (POTB), o proposto obteve melhores resultados em várias características.

O desenvolvimento de métodos híbridos busca a união de dois ou mais métodos geralmente envolvendo os métodos clássicos a LUT, polinômios e PWL.

O método utilizado por Meher (2010) no endereçamento e na organização da LUT obteve resultados satisfatórios com erro máximo em 0,02 e apenas 15 palavras armazenadas utilizando a customização entre LUT e expressões lógicas, que correspondem a 1024 palavras em uma LUT clássica. A customização é feita por um decodificador de domínio para selecionar uma das palavras armazenadas na LUT, composto de um seletor responsável pela conversão do sinal e um decodificador dos intervalos, tudo isso junto a um multiplexador que possibilita a saída correspondente da Tangente Hiperbólica.

Uma combinação distinta de métodos híbridos é apresentada com os métodos, PWL com LUTs (LEBOEUF; MUSCEDERE; AHMADI, 2011; NAMIN et al., 2009) e PWL com Polinômios de 2º ordem (LIN; WANG, 2008). Em Saichand et al. (2008) apresenta-se dois experimentos na implementação da AF sigmoide logística utilizando LUT e PWL. Na primeira a representação foi de 10 bits de entrada e 8 bits de saída, totalizando 8192 bits necessários para armazenar a LUT. O PWL faz a aproximação linear com 5 segmentos ou partes e precisão de 8 bits com 710 amostras, em um endereçamento de 10 bits, com 1K de RAM e tamanho de passo (Delta X) de 0,015625. Diversos blocos individuais foram construídos para codificar e decodificar o módulo computacional do experimento, conseguiu-se reduzir a área utilizada com precisão aceitável no método PWL.

Outro tipo de método híbrido é apresentado por Bajger e Omondi (2008) na aproximação da função sigmoide logística combinando interpolação de baixa ordem de polinômios e LUTs de tamanho reduzido. Bajger e Omondi (2008) realizaram uma investigação do erro relativo e adicionaram uma restrição relacionada ao ponto estacionário que corresponde ao extremo local da função. A estrutura do hardware para realizar o cálculo da função de avaliação, para a aproximação com diferentes coeficientes, é exatamente a organização de um DSP48 da Xilinx.

Um relato distinto com método híbrido é utilizado por Santos et al. (2011), semelhante ao realizado neste trabalho, seção 5.1.3.2. Porém, existem três diferenças primordiais: (i) na quantidade e forma de divisão dos intervalos; (ii) na definição dos valores da LUT pelo tamanho do passo entre as amostras de forma reduzida; (iii) na implementação em *softcore* com linguagem de alto nível de abstração.

A busca por um limiar entre o alto desempenho, menor uso de área na FPGA e precisão são encontrados na utilização de métodos híbridos (LIN; WANG, 2008; NAMIN et al., 2009; LEBOEUF et al., 2008).

Este trabalho busca o desenvolvimento de métodos híbridos para resolver a aproximação AF Tangente Hiperbólica com a utilização dos conceitos de Híbrido de PWL e RALUT (HPR) e Híbrido de PWL com Combinacional (HPC). O método HPR utiliza a divisão em partes (PWL), como outros métodos, o principal é o RALUT, que visa a definição de cada segmento com uma quantidade de elementos diferenciada e adequada a variação do segmento. O método HPC baseia-se na união entre PWL e expressões lógicas simplificadas conforme o seu endereçamento. Para efeito de comparação de algumas características na literatura revisada a Tabela 4 sintetiza os trabalhos neste capítulo.

Tabela 4 – Comparação entre os trabalhos na implementação da AF em hardware

Ref's	AF	Método <sup>19</sup>	Área	Características				
				Erro_Max	Erro_Med	Plataforma	Desempenho	Entrada
Ref. <sup>1</sup>	log	POTB	-	-	-	-	-	-
Ref. <sup>2</sup>	log	PLAN	-	-	-	-	11 gates	[-5,5]
Ref. <sup>3</sup>	log	Comb	45 LEs	$3,90 \times 10^{-3}$	$1,70 \times 10^{-3}$	Simulation VHDL	76,4MHz	[-8,8]
Ref. <sup>4</sup>	log	PWL	-	-	-	Altera's APEX II	-	[-8,8]
Ref. <sup>5</sup>	TH	Híbrido	$32069,8\mu m^2$	$2,20 \times 10^{-2}$	$4,10 \times 10^{-3}$	Actel Pro ASIC (FPGA)	1,1 MHz	[-8,8]
Ref. <sup>7</sup>	TH	RALUT	$11871,5\mu m^2$	$1,78 \times 10^{-2}$	$2,00 \times 10^{-3}$	UMCO, $18\mu m$	2,12 ns	[-4,4]
Ref. <sup>8</sup>	log	Vários	170 Slices	-	-	Xilinx Spartan 3E	7 Ciclos (Neur.)	[-4,4]
Ref. <sup>9</sup>	log	PWL	58 Slices	-	-	Xilinx Virtex-4	-	[-8,8]
Ref. <sup>10</sup>	log	Híbrido	-	$1,01 \times 10^{-4}$	$1,01 \times 10^{-4}$	Xilinx Virtex-4	-	[-8,8]
Ref. <sup>11</sup>	TH	POTB	30 CLBs	$4,30 \times 10^{-2}$	-	-	1,31 ms (RNA)	[-8,8]
Ref. <sup>12</sup>	log	Vários	12 Slices	$1,89 \times 10^{-2}$	$6,30 \times 10^{-3}$	Simulink (Virtex-4)	39 MHz	[-5,5]
Ref. <sup>13</sup>	TH	RALUT	$5130,78\mu m^2$	$1,89 \times 10^{-2}$	$1,21 \times 10^{-2}$	CMOS0, $18\mu m$	2,80 ns	[-8,8]
Ref. <sup>14</sup>	log	Taylor	63% Spartan3	$5,10 \times 10^{-3}$	-	Simulink (Spartan 2 e 3)	-	-
Ref. <sup>15</sup>	TH	LUT	181 Gates	$2,00 \times 10^{-2}$	$1,72 \times 10^{-2}$	TSMC 90 nm	2,46 ns	[-3,3]
Ref. <sup>16</sup>	log	PLAN	8 Slices	-	-	Simulink (Virtex 5)	17 MHz	[-8,8]
Ref. <sup>17</sup>	TH	Vários	-	-	1%	CMOS0, $18\mu m$	-	[-8,8]
Ref. <sup>18</sup>	TH	Híbrido	541 LEs	$1,00 \times 10^{-2}$	-	Altera DE2(Cyclone II)	2 ciclos	[-4,4]

<sup>1</sup> (ALIPPI; STORTI-GAJANI, 1991)<sup>2</sup> (AMIN; CURTIS; HAYES-GILL, 1997)<sup>3</sup> (TOMMISKA, 2003)<sup>4</sup> (MISHRA; ZAHEERUDDIN; RAJ, 2007)<sup>6</sup> (LIN; WANG, 2008)<sup>7</sup> (LEBOEUF et al., 2008)<sup>8</sup> (AL-KAZAZ; KHALIL, 2008)<sup>9</sup> (SAICHAND et al., 2008)<sup>10</sup> (BAJGER; OMONDI, 2008)<sup>11</sup> (HAVEK; VLCEK, 2008)<sup>12</sup> (TISAN et al., 2009)<sup>13</sup> (NAMIN et al., 2009)<sup>14</sup> (KHODJA; KHELDOUN; REFOUFI, 2010)<sup>15</sup> (MEHER, 2010)<sup>16</sup> (PRATAP; SUBADRA, 2011)<sup>17</sup> (LEBOEUF; MUSCEDERE; AHMADI, 2011)<sup>18</sup> (SANTOS et al., 2011)<sup>19</sup> PLAN - Piecewise Linear Approximation of a Nonlinear Function; Comb - Combinacional

## 3.2 Métodos Computacionais para Agricultura de Precisão

A soja é atualmente um dos produtos com maior produção no país, ficando atrás apenas da cana de açúcar. A produção da soja em grãos é de cerca de 71 milhões de toneladas e 34 milhões são destinados a exportação, o restante é destinado ao consumo interno (DOSSA et al., 2012). Todavia, anualmente o produtor chega a perder de 15% a 20% de sua safra devido a ocorrência de doenças, que no Brasil está em torno de 40 tipos diferentes causadas por fungos, bactérias, nematoides e vírus (HENNING, 2009).

A agricultura de precisão pode proporcionar um retorno financeiro de três formas, segundo Cruvinel (2009): o aumento da produção, o aumento da qualidade do produto, e o melhor uso de recursos e insumos. Um dos grandes desafios da AP é a utilização de recursos de forma exata. Para isso, deve-se identificar o problema a ser solucionado. Geralmente, buscam-se técnicas para identificação de deficiência de nutrientes, no aparecimento de pragas e doenças nas plantas e no solo.

No Brasil o monitoramento e o diagnóstico de doenças em geral são realizados de forma manual pelos produtores. A identificação de deficiência de nutrientes e doenças como a ferrugem asiática, o Míldio, bacteriose, Oídio, etc, são feitas pelas foliares da cultura da soja por meio de lupa binocular e registrado em planilha eletrônica. Existem empresas que cobram por essa identificação ou projetos que fazem a detecção gratuitamente, cujo exemplo é o projeto olho vivo (parceria entre a Coopermibra de Campo Mourão-PR (Cooperativa Mista Agropecuária do Brasil), Faculdade Integrado de Campo Mourão, e a Iharabrás S/A Ind. Químicas de Sorocaba-SP) (BONALDO; RIEDO; LIMA, 2009). A dificuldade, junto ao atraso na identificação de nutrientes pode causar doenças e o desenvolvimento retardado nas plantas. Esses problemas podem ser sanados com a inserção de novas tecnologias embarcadas para automatizar esse processo (VOLLMANN et al., 2011).

### 3.2.1 *Monitoramento*

O monitoramento do plantio na AP pode ser feito de diferentes formas, relacionado a metodologia empregada na aquisição e no processamento dos dados, e no tamanho da área a ser analisada. A aquisição dos dados pode ser feita por meio de sensores de vários tipos, os mais comuns para o ambiente são de temperatura, umidade, vento e precipitação. Os sensores relacionados ao solo e a planta, são sensores de umidade e fertilidade do solo, já na identificação de objetos, plantas ou vegetação o monitoramento por imagens é comum.

Atualmente, o uso eficiente de imagens está em ascensão no monitoramento do plantio para AP. Os baixos custos na aquisição de câmeras digitais são um incentivo no uso dessa tecnologia. No monitoramento é comum a utilização de diferentes sensores e definições

para a aquisição de imagens, geralmente relacionado ao modelo de cores, resolução e distância. A principal diversidade nas aquisições de imagens são definidas de acordo com os modelos de cores: o RGB (*Red, Green e Blue*), HSI (*Hue, Saturation e Intensity*), imagens em escala de níveis de cinza e histograma; no tipo de resolução: espacial com a matriz MxN de píxeis, espectral e multiespectral com infravermelho; e as distâncias estão associadas ao tamanho da região a ser monitorada, por exemplo, orbital, aérea e terrestre.

A análise de imagens orbitais, por satélite, são as mais populares dentre os pesquisadores pela grande abrangência de regiões, em aplicações como o reconhecimento de culturas, a determinação de áreas degradadas e a análise de solo. Porém, a aquisição das imagens é feita em um intervalo de tempo grande, geralmente em semanas ou meses, a periodicidade é limitada pelo seu posicionamento na área específica (plantio) e pode comprometer o monitoramento. Imagens terrestres e aéreas supervisionam uma região menor, mas em tempo real, ou seja, com limites de tempo mais rígidos para a aquisição dos dados (MORAES et al., 2008). O monitoramento de uma região menor pode ser realizado por meio de VANT (Veículos Aéreos Não-Tripulados), por veículos terrestres (robôs), ou nas unidades de controle eletrônicas (ECU) de implemento para máquinas agrícolas.

O monitoramento é realizado de forma mais específica em uma região menor, geralmente está voltada à análise da variação de cores e formas das folhas, com o intuito de identificar deficiências de nutrientes (LIU; PATTEY, 2010), ervas daninhas (BURGOS-ARTIZZU et al., 2011) e doenças (CUI et al., 2010). O reconhecimento de deficiência de nutrientes pelas folhas, traz uma forma não invasiva de identificar e aplicar fertilizantes de maneira correta, além de proporcionar agricultura de precisão no plantio.

### ***3.2.2 Identificação, Reconhecimento e Classificação de Padrões***

A maioria dos métodos relacionados a uma região menor para o monitoramento, empregam a análise pela folha da cultura para identificação, reconhecimento e classificação de padrões. Os principais alvos, direcionados pelos pesquisadores da área, estão relacionados à deficiência de macronutrientes, ervas daninhas, pragas e doenças. A razão principal pela análise da folha é por ser o órgão que melhor reflete o estado nutricional das plantas, ou seja, responde mais prontamente às variações de suprimentos, como a falta de um determinado nutriente, ou o excesso de elementos tóxicos existentes no solo ou fornecidos com adubos (SFREDO, 2008).

Atualmente, existe a tecnologia para a aplicação de insumos líquidos, mas o maior desafio está na detecção e no reconhecimento de pragas, doenças e nutrientes (CRUVINEL, 2009). A análise da foliar é comum entre essas aplicações. Diversos métodos são utilizados na análise de nutrientes da planta como por refletância (PACUMBABA JUNIOR; BEYL, 2011), pela intensidade de clorofila nas folhas (VOLLMANN et al., 2011; FANG; MA; CHEN,

2010) e pelo espectro da imagem da folha (CUI et al., 2010; KOSAKA et al., 2006).

Na observação das folhagens por espectrometria com índices de refletância identifica-se de forma satisfatória a alteração dos valores de macronutrientes como o N (Nitrogênio), P (Fósforo), K (Potássio), Ca (Cálcio) e Mg (Magnésio) (PACUMBABA JUNIOR; BEYL, 2011).

Na análise de imagens por meio do espectro e refletância da imagem, geralmente utilizam-se de espectrômetros e imagens por satélite, respectivamente. O custo elevado na aquisição dos equipamentos ou da coleta dos dados é a principal desvantagem na utilização de tais métodos. De forma distinta, a utilização de câmeras de baixo custo podem trazer resultados satisfatórios na identificação de índices vegetativos que se relacionam com o estado nutricional da planta.

A identificação de doenças é semelhante à deficiência de nutrientes. Já na identificação de pragas, geralmente, utilizam-se métodos distintos para classificação das folhas. Em Ma, Guan e Tan (2010) é apresentado o reconhecimento de doença pela análise de folhas de soja com o uso do processamento digital de imagens tradicional, redes neurais e algoritmo genético com 100% de precisão. A identificação e segmentação de doença da ferrugem em folhas de soja é realizada por Cui et al. (2010). Os métodos de processamento digital de imagens são baseados no modelo de cores HSI, o qual possibilita verificar a distribuição das cores das folhas em coordenadas polares. Neste caso definem-se parâmetros para análise da folha tais como a área infectada e saudável, a taxa de área infectada e o índice de cor da ferrugem. Nos dois casos, o processamento é realizado em PC e a aquisição da imagem é realizada por meio de uma câmera de baixo custo.

A detecção precoce da deficiência de macronutrientes pode melhorar o crescimento da colheita e diminuir a proporção de doenças. Essa detecção na soja é frequentemente feita com processamento de imagem nas foliares (MA; GUAN; TAN, 2010; LI-SHU, 2010). A deficiência de macronutrientes na soja e em diversos outros plantios promove o decréscimo da estrutura da planta, a omissão de diversos macronutrientes primários ou secundários e torna-se perceptível nas foliares da soja (PRADO; FRANCO; PUGA, 2010). A falta de micronutrientes como o Cu (Cobre), Fe (Ferro) e Zn (Zinco) também apresenta diferenças foliares de clorofila (BERNAL et al., 2007).

A detecção do macronutriente de nitrogênio na soja pode viabilizar o reconhecimento de um padrão que permite verificar a qualidade da soja, pois o excesso é prejudicial ao homem e a falta desse nutriente causa a necrose da planta e a falta de proteínas (SFREDO, 2008). O uso em excesso de herbicidas (por exemplo, glifosato) altera os índices do macronutriente de nitrogênio, no cultivo da soja e pode ocasionar danificação (necrose) e diferença na pigmentação da folhas (YAO et al., 2012).

O reconhecimento de macronutrientes (nitrogênio) por meio de imagens é feito em

Lishu (2010), Li-shu (2010), Fang, Ma e Chen (2010). A análise dos modelos de cores, RGB e HSI, são investigados e relacionados a quantidade de nitrogênio aplicado na cultura da soja em Fang, Ma e Chen (2010). Em Lishu (2010), Li-shu (2010) verificam a análise de histogramas com a quantidade de nitrogênio e imagem em escala de cinza, os trabalhos discutem a coloração das imagens das folhas e métodos para o processamento de imagens. A aquisição de imagens são realizadas de forma semi-automática em PC.

Em Pagola et al. (2009) apresenta-se uma comparação de vários métodos desenvolvidos, entre o índice de verde e o medidor SPAD (mensura o teor de clorofila na folha da planta). Os experimentos foram realizados em casa de vegetação com diferenças em níveis de nitrogênio e culturas da cevada. A análise da imagem é feita pela diferença entre o índice de RGB. Quatro métodos foram desenvolvidos e os cálculos são baseados no índice global da folha, com valores médios, índice em cada pixel e determinar o médio da imagem inteira, e dois métodos dividindo a imagem em segmentos. A correlação entre os métodos e o medidor SPAD foram averiguados nos três experimentos, na literatura os valores foram mais elevados em aproximadamente 10% do melhor valor alcançado.

Em Vollmann et al. (2011) faz-se a medição pelo teor da clorofila da folha para prover informação sobre a nodulação e fixação de nitrogênio na colheita de soja. O equipamento é frequentemente utilizado para medir o teor de clorofila da folha, de forma não destrutiva, mas a área de leitura do equipamento é de  $8mm^2$ , além de ser muito caro. Dentre os resultados foi possível a correlação dos resultados com análise da imagem e o SPAD-502. Os algoritmos utilizando os sistemas de cores, RGB e HSI, estão respectivamente relacionados aos componentes, verde e de saturação, para comparar o teor de clorofila da folha com o SPAD-502.

A mensuração do teor de clorofila por meio de imagens da folha, com o uso do índice LAI (*Leaf Area Index*), torna-se possível o monitoramento do estado nutricional da planta, e separar o verde da vegetação e o fundo do solo ou de resíduos materiais. A separabilidade é feita em Liu e Pattey (2010) utilizando o método de limiarização por histogramas em colheitas de milho, soja e trigo. Geralmente a coleta de imagens para o monitoramento do estado nutricional da soja é realizado no fim do estágio de desenvolvimento vegetativo e início do reprodutivo, ou seja, no desenvolvimento da vagem (LISHU, 2010; FANG; MA; CHEN, 2010).

A maioria dos métodos para a análise dos nutrientes estão relacionados com a medida de conteúdo de clorofila (MOGHADDAM; DERAFSHI; SHIRZAD, 2011) ou nível de verde (PAGOLA et al., 2009), usualmente medido por refletância (PACUMBABA JUNIOR; BEYL, 2011). O uso da RNA como método de análise de folhas é realizado nas mais variadas situações. Em Moghaddam, Derafshi e Shirzad (2011) apresenta-se a estimativa da clorofila pela folha da beterraba com imagens adquiridas por uma câmera digital convencional. Um

pré-processamento foi realizado na imagem da folha de beterraba para retirar o plano de fundo uniforme, feito por meio de segmentação por limiarização global. A saída da RNA corresponde aos valores encontrados no leitor de clorofila SPAD e as entradas os valores RGB das imagens com e sem deficiência de nitrogênio. Os resultados da RNA e regressão linear são comparados aos valores medidos pelo índice SPAD. A RNA obteve melhores resultados do que a regressão linear e os índices encontrados em Pagola et al. (2009).

A classificação de pragas na colheita de soja apresenta-se em Pan, Huang e He (2007). O uso de operações morfológicas, em processamento de imagem, e a classificação por meio de rede neural base radial obteve 80% de classificação correta nos resultados. O processamento é realizado no computador com câmera de baixo custo para a aquisição da imagem.

A classificação de diferenças foliares para o reconhecimento de doenças na árvore da borracha é feita em Abdullah et al. (2007). As imagens da folha da borracha são adquiridas em laboratório e aplicado um pré-processamento com filtro de mediana. As amostras de entrada foram reduzidas com a utilização de Análise de Componentes Principais (PCA), a RNA conseguiu alcançar bons resultados com a melhor precisão em 96,5%.

Em Kulkarni et al. (2013) faz uso da rede neural base radial (RBPNN) para classificação de plantas. Diversas características das imagens foram extraídas para possibilitar o treinamento da RBPNN. A proposta foi comparada com outras literaturas e a melhor precisão alcançada foi de 93,44%. Já o trabalho de Fu e Chi (2006), os autores propõem a classificação de plantas baseadas na segmentação fina das veias da folha utilizando um classificador por RNA. Um pré-processamento é realizado com limiarização e combinando os detectores de borda de sobel, laplaciano e derivada de segunda ordem, para encontrar as sementes das veias da folha e parâmetros específicos para servir de entrada à RNA. Com RNA obtiveram-se melhores resultados em comparação com métodos tradicionais de PDI.

Husin et al. (2012), implementam a classificação de 20 tipos de ervas por RNA. As entradas da RNA são baseadas em decomposição em valores singulares (Singular Value Decomposition - SVD) e esqueletização, baseado em detecção de bordas pelo método de Canny. A aplicação foi embarcada em um microcontrolador da Dallas (DS89C450) com um tempo de resposta de 30 segundos por cada reconhecimento. A Taxa de reconhecimento das espécies de ervas obteve-se resultados satisfatórios com precisão de 98,9% no reconhecimento.

Neste trabalho é feita análise de várias folhas de forma individual e alguns casos em campo de diferentes cultivares. A análise é realizada para poder observar a presença de deficiência de nutrientes na folha da soja, algodão e mamão, conforme a mudança de sua coloração ou índice vegetativo. O objetivo é realizar a aquisição e o processamento da

imagem na identificação da falta de nutriente nas folhas da soja e do algodão, nesses dois casos contêm a comprovação de deficiências de macronutrientes, respectivamente nos Capítulos 7 e 8. Na literatura encontrou-se trabalhos relacionados com RNA e imagens para identificação de doenças, classificação de plantas e estimação de teor de clorofila, nenhum relacionado à segmentação de imagens quanto a deficiência de macronutriente de potássio e fósforo, especificado neste trabalho. Além disso, todos os trabalhos implementaram em computador convencional, nenhum está relacionado ao emprego de FPGA, com recursos específicos para implementação da RNA em hardware para a segmentação, com operações em ponto flutuante, método híbrido para a aproximação de função de ativação, IP Cores, memórias, entre outros.

Em Moghaddam, Derafshi e Shirzad (2011) foi realizado o processamento de imagem com RNA e entradas RGB semelhante a este trabalho, mas a saída é a estimação do índice SPAD. Neste trabalho foram analisados diversos tipos de entradas e escolheu-se a RGB, porém, a função da RNA é a segmentação da imagem e não valores do índice SPAD. Em Abdullah et al. (2007) contém a RNA para classificação de doenças de imagens e não deficiência de nutrientes, a proposta tem parâmetros de entrada e saída distintos, ou seja, entradas e funcionalidade diferente deste trabalho. Fu e Chi (2006), Kulkarni et al. (2013), Husin et al. (2012) utilizam RNAs distintas para classificação de plantas, diferentes da segmentação, correspondente a este trabalho.

Diferente dos outros trabalhos relatados, neste texto desenvolveu-se um sistema em alto nível de abstração e outro totalmente dedicado a dispositivos reconfiguráveis (FPGA). Outra importante vertente é a utilização da RNA MLP para a segmentação de imagens ou visão de máquina. O uso de planos de fundo uniforme e não uniforme (complexos) também foram analisados com regiões de tamanhos diferentes, e ampla abrangência de culturas e aquisições de imagens das folhas. A etapa inicial do desenvolvimento do trabalho está descrito no Capítulo 4, para explorar a plataforma de hardware, o uso de IP Cores e uma organização inicial da RNA em hardware.

## 4 RNA EM HARDWARE PARA FUNÇÃO XNOR

Os estudos apresentados neste capítulo e no Capítulo 5 foram elaborados para obter uma estrutura mais consistente, com o padrão IEEE Std 754 visando uma fácil manutenção e expansão do sistema. Os dois primeiros estudos foram desenvolvidos em linguagem de descrição de *hardware* VHDL. No primeiro, foram utilizadas as ferramentas gratuitas da Xilinx (ISE 13.1) e Altera (Quartus II versão 11.1) para o desenvolvimento e a prototipação nas plataformas FPGAs *Spartan 3E-500*(Nexys 2) e *Cyclone II*, respectivamente. As plataformas destinadas à realização e à validação do estudo, *Spartan* e *Cyclone II*, contêm diversas interfaces com o usuário e são kits de desenvolvimento de custo relativamente baixo.

Nas arquiteturas de *hardware* para implementar a RNA em FPGA, utilizou-se diversos conceitos de IP *Cores*, como a FPU para o cálculo dos neurônios, além de divisores de frequências, multiplexadores e registradores para os dispositivos de E/S.

A RNA desenvolvida é baseada no modelo de McCulloch-Pitts (MCCULLOCH; PITTS, 1943) com duas entradas, duas camadas e três neurônios. Esta rede é não-recorrente e com função de ativação degrau bipolar. O treinamento da rede não foi prototipado, porém os cálculos dos pesos foram feitos e adicionados ao projeto. Atualizando-se os pesos pode-se obter novas funcionalidades para a RNA, diferente da XNOR.

Neste teste a arquitetura de hardware para uma RNA foi a função XNOR e o usuário interage com o sistema por meio de dispositivos de E/S. Uma análise de área e desempenho da FPGA foi realizada com vários cenários distintos para verificar a melhor escolha no uso de FPU.

As principais contribuições deste estudo estão na análise específica de vários requisitos de um sistema com RNA e FPU em dispositivo reconfigurável, na utilização de distintas plataformas e IP *Cores*, e no uso de dispositivos de E/S dispostos nas plataformas FPGA. Essa análise poderá auxiliar de forma adequada o projetista na escolha dos componentes a serem utilizados em seu sistema.

O projeto foi construído e prototipado nas plataformas FPGAs da Altera e Xilinx e suas respectivas ferramentas CAD. O *Coregen* (XILINX, 2011) e o *plugin Megafunctions* (ALTERA, 2011) foram usados para a concepção dos IP *Cores* nas duas unidades de ponto

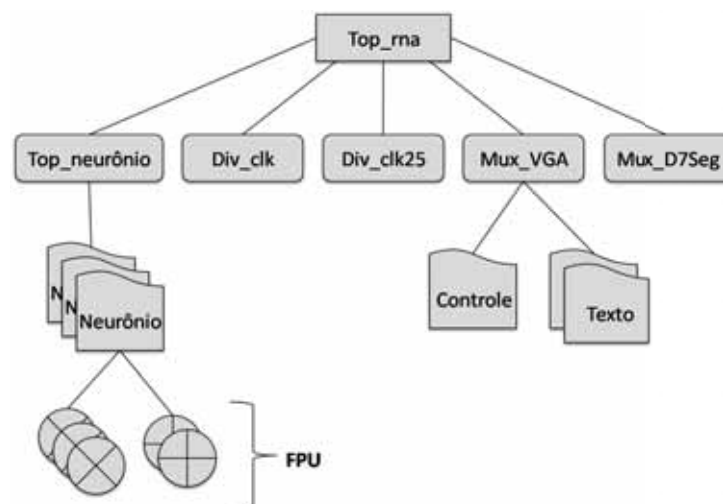
flutuante, adição e multiplicação, para realizar os cálculos das somas ponderadas nos neurônios. Outro IP *Core* analisado foi o Havoc (GUILLERMO, 2010; MARCUS et al., 2004), usado para as duas plataformas, ou seja, com portabilidade para diferentes fabricantes.

Além do desenvolvimento do sistema da RNA, a substituição dos IP *Cores* foi realizada para uma análise de área e desempenho nas diferentes plataformas, e vários outros componentes integram a estrutura do sistema.

A alimentação do sistema é feita por meio de seis chaves, quatro para os sinais de entrada  $x_1$  e  $x_2$  da RNA, que podem assumir valores  $+1$  e  $-1$ , sendo duas para cada entrada poder representar o sinal, e mais duas para os sinais de *load* e *reset* que são responsáveis pelas transições da máquina de estados. O sistema da RNA dispõe de seis leds como indicadores das entradas, e as saídas da RNA podem ser acompanhadas pelo *display* de 7 segmentos e pela interface VGA.

A metodologia *bottom-up* em conjunto com a descrição estrutural foram utilizadas para melhor legibilidade do projeto, para aumentar o paralelismo e facilitar o reuso dos componentes. Na Figura 17 apresenta-se a organização do sistema implementado em FPGA, conforme a sua divisão por níveis. Nessa organização é possível verificar todos os blocos existentes no sistema e a forma como são instanciados.

**Figura 17 – Organização do projeto.**



Fonte: Elaboração do próprio autor.

Nas unidades de ponto flutuante foram realizados os cálculos de cada neurônio, a ponderação dos pesos com as entradas e o somatório para gerar a saída parcial de cada neurônio. Para cada saída parcial é executada a função de ativação degrau bipolar e determinado um novo resultado em ponto flutuante. A função de ativação segue a regra apresentada na Equação (23). Esta função foi utilizada apenas para este primeiro estudo.

$$G(\cdot) = \begin{cases} +1 & \text{se } x \geq 0 \\ -1 & \text{se } x < 0 \end{cases} \quad (23)$$

Considerando-se o último neurônio, a saída da função degrau bipolar é o resultado da RNA na representação binária com 2 bits. Cada função de ativação é determinada por um processo e habilitado com um CE (*Chip Enable*), tornando mais fácil a substituição desta função.

Salienta-se que outras funções de ativação descritas na literatura podem ser utilizadas como: a sigmoide (logística), Tangente Hiperbólica e rampa. No próximo capítulo será utilizada a função de ativação Tangente Hiperbólica.

#### 4.1 Estrutura Geral

As plataformas de hardware utilizadas são consideradas de custo relativamente baixo. A Spartan disponibiliza pouca área de chip (*Slices*) da FPGA, cerca de 4 vezes menos quando comparado aos Elementos Lógicos (LE) disponíveis na *Cyclone II*.

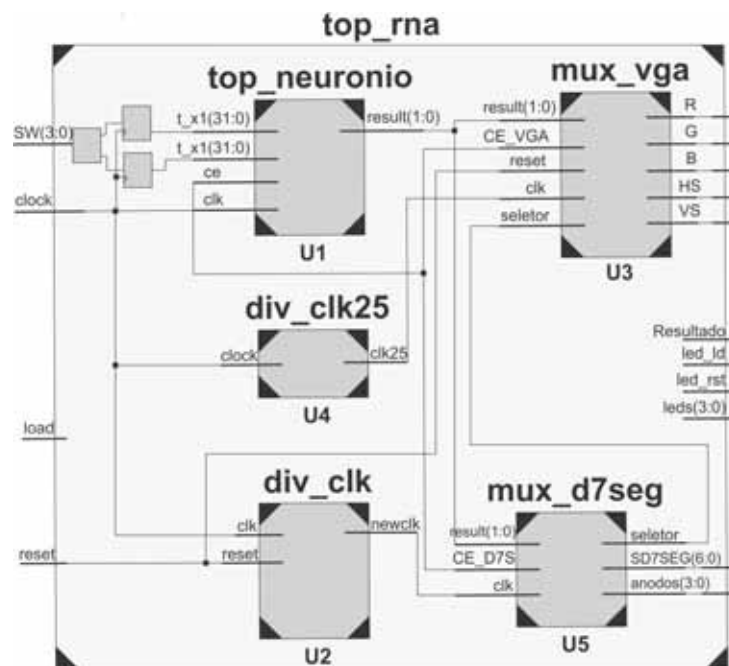
Os *Slices* e os LEs, por serem de plataformas e fabricantes distintos, contêm diferenças principalmente nos tipos de LUT, e nas quantidades de entradas e de *Flip-Flops* empregados nessas estruturas lógicas (*Slices* e LE). Um exemplo é que na *Cyclone* possui LE de dois modos, o normal e o aritmético. Já na *Spartan* existem *Slices* que podem representar funções de memória e lógica, e outro tipo para efetuar apenas função lógica. Independente da plataforma ou fabricante, a utilização de VHDL com descrição comportamental internamente nos módulos, torna o sistema portátil e flexível.

A vista de topo da estrutura do sistema, com as características de controle e das interfaces de E/S dos componentes está ilustrada na Figura 18. Essa estrutura está em nível de lógica de transferência de registradores (RTL).

As entradas *clock* e *reset* são propagadas por todo o circuito, as chaves são armazenadas nos registradores que são convertidos para seus respectivos valores em ponto flutuante. O *load* inicializa a máquina de estados da entidade denominada topo para controlar o armazenamento correto dos registradores. Em seguida, a execução da RNA e os multiplexadores são habilitados para exibir a saída da RNA no monitor e nos *displays* de 7 segmentos.

A interface VGA e o *display* de 7 segmentos são utilizados para apresentar os resultados para o usuário. Definiram-se divisores de frequência e multiplexadores para alternar entre a apresentação pelos *displays* de 7 segmentos e por meio de texto na

Figura 18 – Estrutura RTL do nível topo do projeto.



Fonte: Elaboração do próprio autor.

interface VGA. O multiplexador para o *display* de 7 segmentos foi utilizado apenas na plataforma Nexys2. A DE II contém um barramento de dados para cada *display* tornando desnecessário o multiplexador, porém manteve-se uma estrutura semelhante para não alterar a análise.

O bloco *Top\_Neurônio* corresponde à implementação das camadas da RNA, a execução dos neurônios e da função de ativação em paralelo. Os pesos sinápticos estão definidos nesse bloco e processa os neurônios (idênticos) modificando-se apenas em seus respectivos pesos na instanciação. Internamente, cada neurônio contém a soma ponderada das entradas com seus respectivos pesos, cada tipo de operação possui a sua própria unidade de ponto flutuante.

## 4.2 Cenários

Na implementação da RNA em FPGA geralmente ajusta-se a unidade de ponto flutuante para menor consumo de área e maior desempenho. A reutilização de um projeto otimizado obriga ao estudo do componente para poder utilizá-lo, isso demanda maior tempo de projeto e muitas vezes deve ser adaptado à nova realidade ou sistema.

Nesse estudo foram empregadas unidades de ponto flutuante padronizadas pela IEEE Std 754 de precisão simples (32 bits). Essas unidades estão disponíveis e podem ser utilizadas pelas ferramentas dos próprios fabricantes de FPGA ou por IP *Cores* livres.

As FPU apresentadas são todas gratuitas e possíveis de serem empregadas em qualquer sistema que disponha das plataformas definidas neste estudo. Na utilização das ferramentas CAD dos fabricantes Altera e Xilinx é possível definir diversos tipos de parametrização para as unidades de ponto flutuante.

Os cenários avaliados utilizam 4 tipos de IP *Cores* para ponto flutuante, um da Altera, um da Xilinx, e dois IP *Cores* Livres: o Havoc e o FPU100 (JIDAN, 2011).

A FPU100 é uma unidade de ponto flutuante com os tipos de operações aritméticas básicas (soma, subtração, multiplicação e divisão) e a raiz quadrada. O pacote em VHDL contém sinais de controle e exceção, e diversas características que podem ser parametrizadas como as operações aritméticas, o tipo de implementação (com ciclo simples ou pipeline) e o tipo de arredondamento pode ser definido por sinais no próprio código fonte. Porém, não foi possível fazer o mapeamento na plataforma da Xilinx, pois ocorreu um *Overmapped*<sup>1</sup>, mesmo retirando os componentes de divisão e raiz quadrada. Assim, apenas adicionaram-se os valores de área para comparação com os outros cenários, já que os valores de temporização são coletados após a etapa de alocação e roteamento.

O IP *Core* Havoc opera com o padrão IEEE-754 de precisão simples (32bits) e possui dois tipos de operação, a adição e a multiplicação. Na adição a implementação tem suporte a ciclo simples e pipeline de 6 estágios, já a multiplicação opera em ciclo simples e pipeline de 4 estágios. A Havoc foi o único IP *Core* a ser prototipado nas duas plataformas e cada tipo de operação contém módulos separados como na Altera e Xilinx.

A ferramenta Coregen da Xilinx gera o seu IP *Core* de ponto flutuante e pode ser parametrizado conforme o padrão IEEE-754 com precisão simples, dupla ou customizada, onde define-se a quantidade de bits para cada elemento da FPU (Sinal, Expoente e Mantissa). O uso de multiplicadores, a exigência de latência ao sistema, os sinais de exceção, controle e protocolo de comunicação são características possíveis de serem incluídas na construção de uma unidade de ponto flutuante.

Cada FPU contém um tipo de operação, com exceção da adição e subtração que podem ser agrupadas em uma mesma unidade. A prototipação da FPU com operação de multiplicação e a opção de uso completo de multiplicadores 18x18, não foi possível na plataforma Spartan<sup>2</sup>, em uma plataforma mais robusta o uso de multiplicadores dedicados pode reduzir a área de uso pela FPU e aumentar o desempenho do sistema.

O ambiente da Altera tem uma grande biblioteca chamada *Megafunctions* e um *plugin* associado para facilitar a utilização de funcionalidades dessa biblioteca. A construção da

---

<sup>1</sup>Overmapped é quando uma determinada plataforma não contém todos os recursos necessários para a prototipação do sistema.

<sup>2</sup>A quantidade de multiplicadores dedicados na plataforma Spartan 3E é limitada e insuficiente para a prototipação da RNA.

FPU segue o padrão IEEE-754 e a parametrização tem a opção de precisão simples, dupla ou simples estendida, além disso, o desenvolvedor pode selecionar um tipo de otimização de área ou velocidade, escolheu-se o padrão, velocidade. A adição de portos de controle e exceções é viável em qualquer tipo de operação escolhida. Os tipos de operações de ponto flutuante possíveis para os dois fabricantes são semelhantes, as principais são as básicas: adição, subtração, multiplicação e divisão; e as específicas: raiz quadrada, comparação, conversão, entre outras.

Além da variação de IP *Cores*, foram analisadas duas funcionalidades para gerenciar com maior qualidade o caminho do sinal de *clock*. Nesse controle são inseridas memórias temporárias (*buffers*) globais no sinal de entrada de *clock* externo e várias outras características para melhorar a propagação do sinal de *clock* em toda a FPGA. Para cada fabricante analisado contém um tipo de unidade de controle de *clock*, o DCM na Xilinx e o ALTCLKCTRL na Altera. Esses controladores também podem ser inseridos na saída de um PLL para ter uma melhor qualidade no sinal em altas frequências. Nesse estudo foram implementados apenas os controles digitais, não se utilizou PLLs.

As unidades de controle de *clock* visam melhorar principalmente os problemas de *jitter*, *skew* e *metastability* na FPGA. Os problemas de *jitter* e *skew* são imperfeições ou ruídos, que deformam ou adicionam atrasos ao sinal de *clock*, e podem causar problemas em todos os componentes ligados a este (XILINX, 2006). A *metastability* ocorre em sistemas síncronos, provoca instabilidade no sistema por não haver tempo suficiente para o *setup time* e *hold time* do registrador, isto causa inconsistência nos dados de saída dos registradores prejudicando o sistema (STEPHENSON et al., 2009).

O ALTCLKCTRL e o DCM são tipos de unidades de controle de *clock* dos dois principais fabricantes de FPGA, Altera e Xilinx, respectivamente. O ALTCLKCTRL (ALTERA, 2012) pode ser parametrizado conforme o modo de operação, determinando a habilitação de *clock* global, regional e externo. Para a Cyclone II permite-se apenas a opção global e externo, sendo o último para o uso em conjunto com o PLL. Já o DCM (XILINX, 2006), contém 3 unidades importantes: o DLL (*Delay-Locked Loop*), DFS (*Digital Frequency Synthesizer*) e o PS (*Phase Shift*). Dessa forma, o DCM possibilita a habilitação do *clock*, adiciona buffer global e de entrada, além de trazer a opção de saída para múltiplos *clocks* com divisão ou multiplicação do *clock* externo ou referência.

Diante disso, na Tabela 5 apresentam-se 17 cenários analisados neste estudo, variando as unidades de ponto flutuante, controle de *clock* e o tipo de implementação (Ciclo Simples (CS) ou pipeline) no IP *Core* Havoc. Todos os cenários obedeceram o padrão IEEE-754 de precisão simples, com exceção dos cenários 9 e 10 nos quais a precisão em ponto flutuante foi ajustada para 24 bits, dependendo da aplicação essa precisão tem baixa taxa de erro, conforme Lee e Ko (2006).

Os dez primeiros cenários foram prototipados na plataforma Spartan e o restante na Cyclone. Entretanto, nas duas plataformas utilizou-se *clock* externo a FPGA com frequência de 50MHz em todas as comparações analisadas.

**Tabela 5 – Cenários analisados.**

Nº	Cenários
1	Xilinx+Coregen(S/ DCM)
2	Xilinx+Coregen(C/ DCM)
3	Xilinx+Coregen(S/ DCM+Multip.)
4	Xilinx+Havoc+CS(S/ DCM)
5	Xilinx+Havoc+CS(C/ DCM)
6	Xilinx+FPU100(S/ DCM)
7	Xilinx+Havoc+Pipeline(S/ DCM)
8	Xilinx+Havoc+Pipeline(C/ DCM)
9	Xilinx+Coregen(24 bits)(S/ DCM)
10	Xilinx+Coregen(24 bits)(C/ DCM)
11	Altera+Megafuncions(S/ CTRLCLK)
12	Altera+Megafuncions(C/ CTRLCLK)
13	Altera+Megafuncions(S/ CTRLCLK+Multip.)
14	Altera+Havoc+CS(S/ CTRLCLK)
15	Altera+Havoc+CS(C/ CTRLCLK)
16	Altera+Havoc+Pipeline(S/ CTRLCLK)
17	Altera+Havoc+Pipeline(C/ CTRLCLK)

### 4.3 Resultados

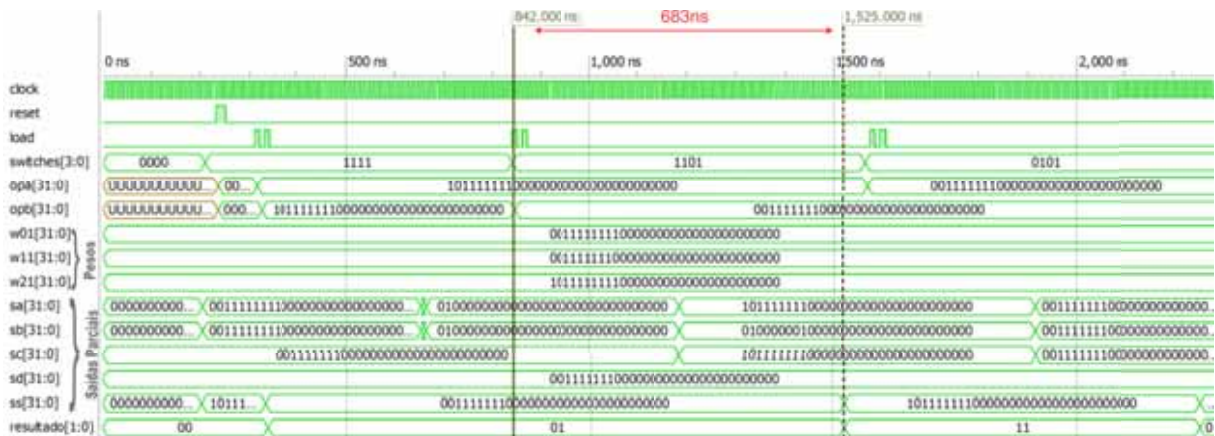
O diagrama de tempo do sistema é apresentado na Figura 19, destaca a funcionalidade da RNA e os principais sinais do sistema. A entrada de dados pelas chaves são convertidas para ponto flutuante e armazenadas nos registradores OPA e OPB, os pesos do primeiro neurônio e as saídas parciais também podem ser observadas na marcação. O sinal de “resultado” na saída do neurônio também contém um registrador, para manter os dados por mais tempo sem perdas, antes de serem transferidos aos outros níveis e dispositivos externos.

As saídas parciais SA e SB são os resultados dos neurônios da primeira camada e após a realização da função degrau bipolar obtém-se as saídas SC e SD. A saída parcial SS é o resultado do neurônio da última camada, o sinal “resultado” está após a função degrau bipolar e finaliza o sistema.

No diagrama de tempo são apresentadas três transições nas entradas da RNA conforme a simulação do sistema completo implementado e ilustrado pela Figura 19. Na segunda transição apresentada na Figura 19, o tempo de execução da RNA é marcado em aproximadamente 683 nanosegundos, esse tempo foi alcançado pelo primeiro cenário da Tabela 5.

O cenário 3 obteve o melhor desempenho com a plataforma e o IP *Core* da

Figura 19 – Diagrama de tempo.



Fonte: Elaboração do próprio autor.

Xilinx alcançando 645 nanosegundos, pela utilização de multiplicadores dedicados a área utilizada na FPGA também é menor, conforme apresentado na Tabela 6. No cenário 7 o tempo de execução do sistema ficou em cerca de 330 nanosegundos, mostrando a eficiência do pipeline no desempenho global do sistema. Já o cenário 9, que opera com precisão reduzida, teve uma resposta de 665 nanosegundos, um pouco menor que a versão com 32 bits. Com a plataforma Cyclone II, o cenário 11 obteve o tempo de execução em 1330 nanosegundos e o cenário 15 com o Havoc chegou a 660 nanosegundos, menos da metade do valor da Altera.

Nas Tabelas 6, 7 e 8 apresentam-se várias informações dos cenários prototipados nas plataformas Spartan e Cyclone. Os dados estão divididos em dois grupos, em área consumida na FPGA (5 primeiras linhas) e parâmetros usados na análise de otimização (4 últimas linhas). Salienta-se que não houve modificações no tipo de otimização por esforço nas ferramentas de síntese dos dois fabricantes, mantendo-se o padrão normal.

As FPU's para as operações de cálculos em ponto flutuante nos neurônios, ocupam uma parte considerável da área da FPGA da Spartan dependendo do IP *Core* utilizado, comprometendo o seu uso nesta plataforma. Porém, percebe-se que o IP *Core* Havoc foge um pouco dessa suposição mostrando-se econômico em relação à área utilizada em FPGA.

Nos cenários de 1 a 6, fica claro que a área consumida (*Slices*) do IP *Core* Havoc é menor em relação aos outros cenários, chegando a 112% menor que a FPU100 e 30% menor que o da Xilinx, conforme representado na Tabela 6. A redução da precisão da FPU em 9 e 10, apresentada na Tabela 7, diminui a área consumida em 28% nas LUTs e FlipFlops, e 11% para os *Slices* em relação a versão completa.

A comparação de todos os cenários de forma geral, em relação à área utilizada na FPGA, é apresentado na Figura 20. Mesmo com as diferenças entre os componentes lógicos

Tabela 6 – Cenários analisados - Spartan 3-E.

Spartan 3-E	1	2	3	4	5
Slices	4.654(99%)	4.654(99%)	3.990(85%)	2.565(55%)	2.356(50%)
Slices FF	8.430(90%)	8.429(90%)	5.262(56%)	1.564(16%)	1.483(15%)
4-input LUT	8.334(89%)	8.334(89%)	4.619(49%)	4.120(44%)	3.777(40%)
MULTI18X18s	0(0%)	0(0%)	36(180%)	10(50%)	10(50%)
IOBs	31(13%)	31(13%)	31(13%)	31(13%)	63(27%)
Limite de Período(ns)	7	7	-	18	19
Clock Skew(ns)	0,089	0,089	-	0,084	0,083
Slack-Worst Case(ns)	0,081	0,117	-	0,028	0,050
FanOut(Worst Case)	4.527	4.519	-	1.153	1.091
Freq. Máx.(MHz)	144,530	112,575	-	55,642	52,770

Tabela 7 – Cenários analisados - Spartan 3-E.

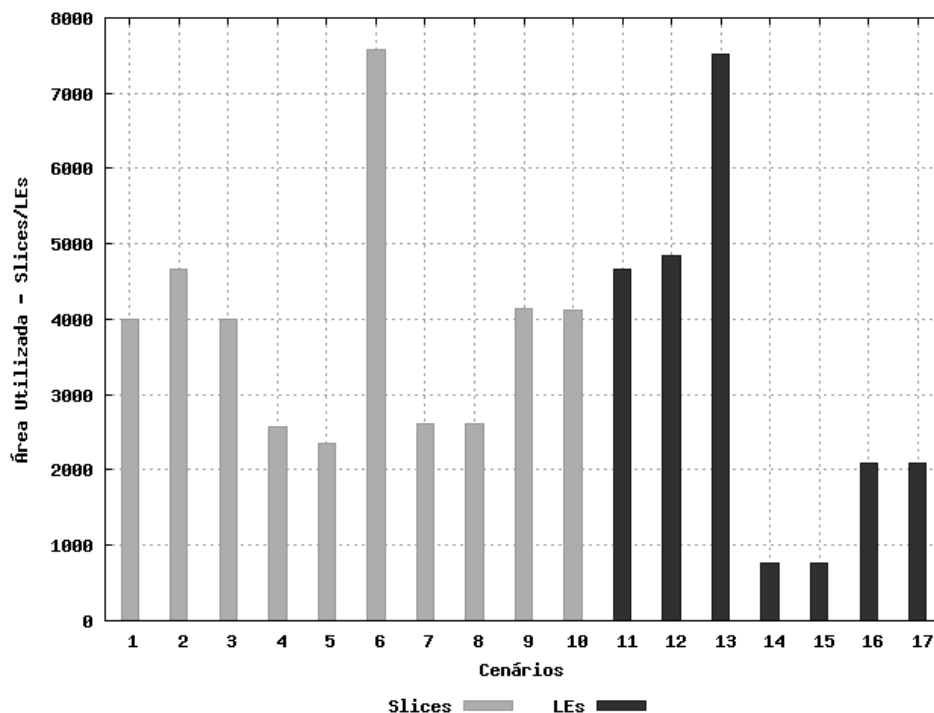
Spartan 3-E	6	7	8	9	10
Slices	7.575(162%)	2.614(56%)	2.611(56%)	4.133(88%)	4.128(88%)
Slices FF	1.680(18%)	1.588(17%)	1.587(17%)	5.809(62%)	5.804(62%)
4-input LUT	14.554(156%)	4.515(48%)	4.513(48%)	5.703(61%)	5.697(61%)
MULTI18X18s	4(20%)	10(50%)	10(50%)	0(0%)	0(0%)
IOBs	63(27%)	31(13%)	31(13%)	31(13%)	31(13%)
Limite de Período(ns)	-	19	19	7	7
Clock Skew(ns)	-	0,082	0,086	0,089	0,089
Slack-Worst Case(ns)	-	0,132	0,012	0,083	0,083
FanOut(Worst Case)	-	4.527	4.519	1.153	1.091
Freq. Máx.(MHz)	-	50,332	52,665	144,571	87,589

Tabela 8 – Cenários analisados - Cyclone II.

Cyclone II	11	12	13	14	15	16	17
LE	4.671(25%)	4.838(25%)	7.518(40%)	763(4%)	763(4%)	2.088(11%)	2.082(11%)
Registers	3.107(17%)	3.107(17%)	5.171(28%)	64(1%)	64(1%)	720(4%)	720(4%)
4-input LUT	982(5%)	982(5%)	1206(6%)	359(2%)	359(2%)	864(5%)	864(5%)
MULTI 9-bits	0(0%)	0(0%)	63(121%)	0(0%)	0(0%)	0(0%)	0(0%)
IOBs	34(11%)	34(11%)	34(11%)	34(11%)	34(11%)	34(11%)	34(11%)
Limite de Período(ns)	7	7	-	4	4	15	16
Clock Skew(ns)	-0,047	-0,004	-	0,000	0,002	-0,002	-0,001
Slack-Worst Case(ns)	0,627	0,562	-	0,136	0,152	0,004	0,519
Fan-out	3.198	3.198	-	59	59	684	684
Freq. Máx.(MHz)	156,91	155,33	-	258,80	259,88	66,68	64,6

das distintas plataformas, pode-se observar uma correlação de área entre os cenários.

**Figura 20 – Comparação da área utilizada (*Slices*) na FPGA**



Fonte: Elaboração do próprio autor.

O IP *Core* da Xilinx, cenário 1, obteve a maior frequência no caminho do sinal de *clock* dentre os cenários de 1 a 7, chegando a ser quase 3 vezes mais rápido em relação à frequência do Havoc. Vale ressaltar, que a característica de frequência máxima está relacionada ao desempenho em que o sistema pode atuar em determinados caminhos e não do sistema completo.

Na inserção do DCM, nos cenários 2 e 5 da Tabela 6, a área consumida ficou praticamente a mesma dos cenários 1 e 4, o DCM foi parametrizado com dois *clocks* de saída com buffer, ou seja dois DCM, um na mesma frequência da plataforma (50 MHz) e outro dividido por 2, para eliminar o módulo divisor de *clock* de 25 MHz. Esse *clock* está na saída do DCM interconectado ao divisor de *clock* lento (60Hz), para diminuir o seu contador, e também utilizado como referência ao *clock* do VGA com resolução de 640x480.

Na análise dos dados atribuíram-se diferentes valores para a limitação de período de *clock* em cada cenário, esses números foram encontrados aumentando essa limitação até não ter mais problemas de *metastability* no sistema. Nos dados de *slack* (pior caso) em todos os cenários esse valor fica próximo ao zero mostrando que a implementação do cenário está no limite de período de tempo sustentável.

A inclusão do DCM não prejudica o custo em área da FPGA e viabiliza melhorias

em qualidade no sistema. Nesse estudo de caso pode-se retirar o divisor de frequência e reduzir o consumo de *Slices* em 5%, cenários 4 para 5, além disso, nota-se uma suave redução ao *clock skew* e no *FanOut* máximo, isso pode ser visto de forma semelhante em todos os cenários com e sem DCM para a plataforma da Xilinx.

Na prototipação com a plataforma Cyclone II, Tabela 8, o comportamento do IP *Core Havoc* alterou-se de maneira satisfatória em área consumida e desempenho. O número de Elementos Lógicos (LE) ocupados nos cenários 11 e 14 são distintos, o IP *Core Havoc* é 6 vezes menor que o da Altera e a frequência é aproximadamente 65% maior.

A utilização do ALTCTRLCLK não apresentou mudanças relevantes nos cenários analisados, com exceção da melhoria no *clock skew* dos cenários 11 e 12. No cenário 13 não foi possível adquirir os dados, por ter ocorrido um *overmapped* em dois requisitos, multiplicadores e DSPs, mesmo com alteração da otimização do IP *Core* para “area”.

A Cyclone II, com os cenários que utilizaram o IP *Core Havoc*, ocuparam um menor número de LUTs de 4 entradas do que na Xilinx, devido a maior utilização de LUTs de 2 e 3 entradas nesses cenários, isso pode ser observado em uma análise mais detalhada do relatório de recursos. Vale ressaltar, que a plataforma da Xilinx só contém LUTs de 4 entradas (XILINX, 2009), já em relação aos Slices e LE a área utilizada nos dois fabricantes e a frequência alcançada são semelhantes.

Diante dos resultados encontrados nos diversos cenários apresentados para esse sistema, a melhor escolha é dos cenários utilizando o IP *Core Havoc*. Nas duas plataformas obtiveram-se excelentes resultados principalmente de área usada na FPGA. O cenário 14, com a plataforma Cyclone, resultou na menor área consumida e na maior frequência em um determinado caminho dentre todos os outros cenários.

Para a plataforma Spartan, o cenário 7 com *pipeline* foi duas vezes mais rápido em tempo de execução que o IP *Core* da Xilinx. A frequência máxima encontrada em cada cenário é correspondente a um caminho específico e o tempo de execução é a validação do processamento de todo o sistema.

O mesmo estudo de caso foi desenvolvido em software, com processador Celeron 2,2 GHz (Linux Fedora com GCC), e obteve o resultado em 2 milissegundos, mesmo com uma RNA simples, a diferença de desempenho entre a implementação em hardware e software é extremamente alta, evidencia o paralelismo encontrado na RNA e FPGA.

O emprego do IP *Core Havoc* economizou o uso de recursos de área da FPGA em 49% na Spartan e 21% na Cyclone, comparado aos IP *Cores* dos seus respectivos fabricantes. Em Nichols (2003), utilizou-se 16 bits de ponto fixo para a XOR RNA e sem o padrão IEEE-754, a área utilizada foi de 2478 *Slices* na plataforma xcv2000e, semelhante ao Havoc com precisão de 32 bits em ponto flutuante. Em Ferreira e Barros (2010), a implementação

de uma RNA MLP com 6 neurônios, para aplicação de uma senóide teve um consumo de 4.591 ALUTs. O bloco lógico denominado ALUT é equivalente a duas LUTs comuns de 4 entradas.

#### 4.4 Conclusão deste Capítulo

Neste estudo foi prototipado e validado uma arquitetura de hardware para uma RNA que se utiliza de vários tipos de interfaces como chaves, led, *display* de 7 segmentos e VGA. Foi comprovada a portabilidade com a utilização de duas plataformas dos principais fabricantes de FPGA.

O sistema utilizou unidades de ponto flutuante para os cálculos dos neurônios e vários componentes de controle que foram implementados em hardware empregando FPGA. Na organização do sistema utilizou-se de metodologia hierárquica de projeto, com intuito de facilitar a manutenção e o reuso de componentes. A prototipação do sistema empregou diferentes tipos de FPU e unidades de controle de *clock*, organizados em diferentes cenários. Em cada cenário apresentou-se várias informações de área consumida e otimização para a FPGA.

No diagrama de tempo do estudo de caso, a RNA processa e mostra os resultados em cerca de 683 nanosegundos com o IP *Core* da Xilinx e 330 nanosegundos com o Havoc. Já a Altera obteve resultado de 1330 nanosegundos e o Havoc com 660 nanosegundos. Os resultados apresentados em todos os cenários foram divididos em duas plataformas de hardware, comprovando que nas duas plataformas de hardware o IP *Core* Havoc é a melhor escolha dentre os analisados. O cenário 3 obteve um desempenho melhor (645 nanosegundos), e uma redução de aproximadamente 15% de *Slices* e 40% de LUTs, em comparação com os outros cenários que não utilizam multiplicadores dedicados. No Capítulo 5 apresenta-se o aprimoramento do sistema em hardware para uma RNA MLP e com função de ativação não linear.

## 5 RNA MLP NA APROXIMAÇÃO DE FUNÇÃO NÃO LINEAR

Neste capítulo é apresentada o desenvolvimento em hardware de um sistema com RNA MLP para a aproximação de uma função não linear. A arquitetura do sistema desenvolvida neste capítulo, serve como base para o sistema do Capítulo 8, que apresenta uma aplicação com segmentação de imagem. Diversos cenários foram analisados, alterando-se o tipo de método para a aproximação da função de ativação Tangente Hiperbólica, e os resultados foram comparados à literatura.

A implementação em hardware de RNA MLP é um dos principais desafios aos pesquisadores da área. Diversos trabalhos (ALI; MOHAMMED, 2010; ZHANG; CHEN, 2011; NEDJAH; SILVA; MOURELLE, 2012) investigam melhorias principalmente na implementação de neurônios artificiais e funções de ativação com maior precisão e menor área utilizada.

Neste estudo apresenta-se uma arquitetura de hardware padronizada para uma RNA MLP, em ponto flutuante IEEE Std 754, que realiza a aproximação de uma função não linear. A RNA é dividida em três partes principais a unidade de controle (UC), unidade de armazenamento (UA) e unidade de processamento (UP), respectivamente descritos nas Seções 5.1.1, 5.1.2 e 5.1.3.

A RNA foi desenvolvida em linguagem de programação de alto nível, com o Matlab sem *toolbox*, para comparação dos resultados obtidos em hardware. O Matlab é um interpretador e foi utilizado em todos os testes com precisão dupla (format long). O desenvolvimento auxilia o projetista na escolha dos componentes a serem utilizados em seu sistema de baixo nível. A computação dos dados foram feitas em ponto flutuante IEEE Std 754 (32 bits), a padronização é uma forma de facilitar o desenvolvimento, manutenção, expansão e compreensão do sistema. Todos os neurônios contém uma AF Tangente Hiperbólica desenvolvida em vários métodos distintos, pode-se destacar os dois principais baseados em métodos híbridos, RALUT com PWL e combinacional com PWL.

Para o treinamento supervisionado da RNA adotou-se a regra delta em linguagem de alto nível e modo *offline*. O critério de parada adotado foi pelo erro quadrático com limite em  $10^{-12}$  e a convergência ocorreu com 56.987 épocas. No treinamento utilizou-se 256 amostras de entrada, com o devido valor desejado na saída da rede. Na implementação em hardware os pesos da RNA foram transportados ao dispositivo FPGA com os mesmos

valores encontrados em alto nível.

No estudo de caso empregou-se o Matlab, para o desenvolvimento em software e comparação dos resultados. A ferramenta CAD da Xilinx utilizada foi o ISE na versão 14.2 e a plataforma FPGA *Spartan 3E-500* (Nexys 2). Os dados utilizados na entrada da RNA, para o seu treinamento com o Matlab são diferentes dos dados utilizados na obtenção dos resultados comparados em software e hardware na seção 5.3. No treinamento utilizou-se 256 amostras de entrada, com o devido valor desejado na saída da rede. Após o treinamento, para a execução da RNA utilizaram-se 128 amostras, na comparação entre as implementações em hardware e software.

O sistema implementa uma RNA MLP com três camadas, uma escondida, quatro neurônios organizados na estrutura 1-2-1 e AF Tangente Hiperbólica em todos os neurônios. A funcionalidade da rede está na aproximação de uma senoide representada pela Equação (24),  $x_i$  é a entrada de dados e  $sen$  é uma função seno.

$$S(x_i) = \frac{x_i}{e^{0,03}} \times sen(0,03 \times x_i) \quad (24)$$

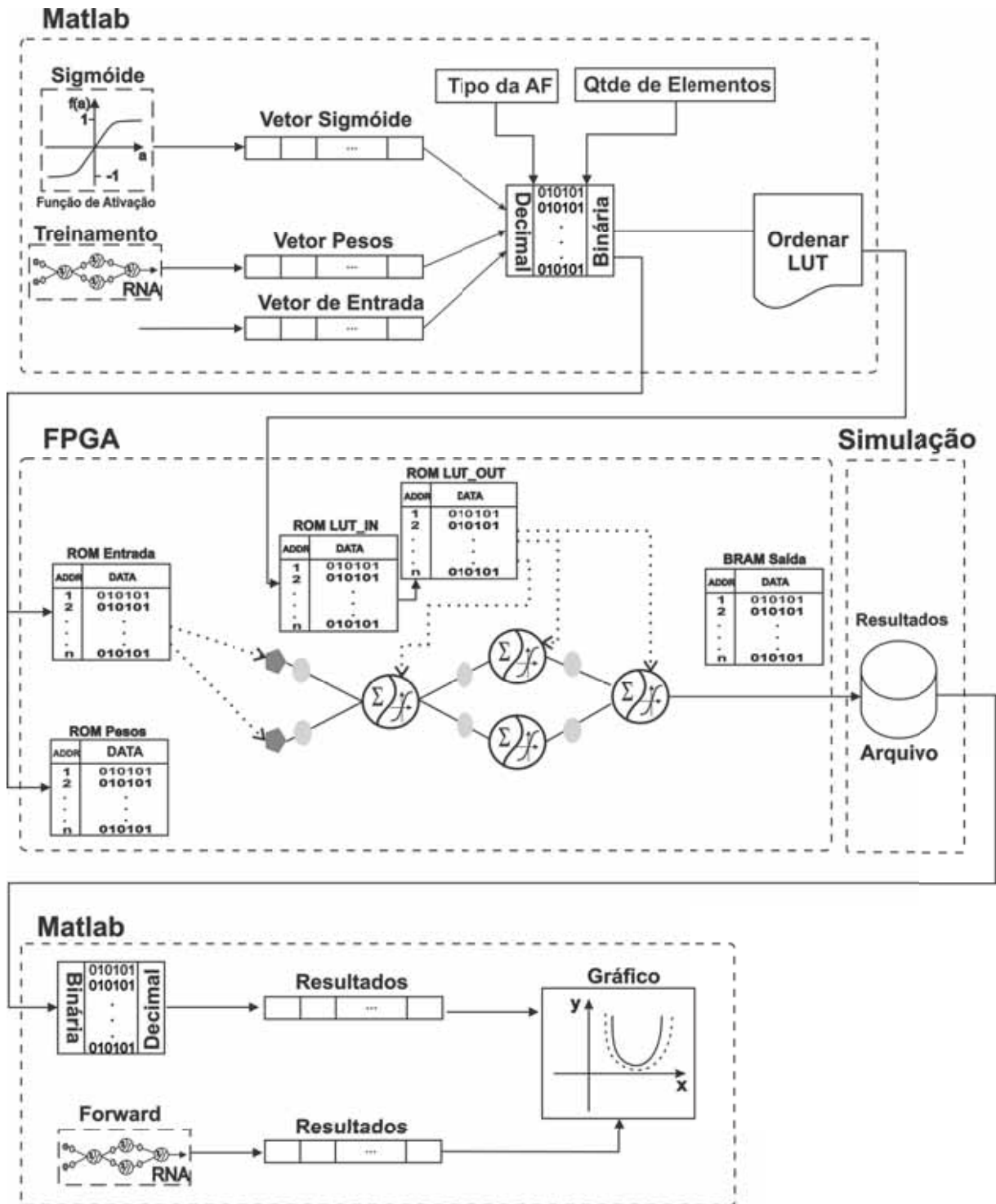
Para validação do sistema aplicou-se uma metodologia dividida em três etapas: preparação dos dados, execução do sistema e análise dos resultados. Na Figura 21 ilustra-se a metodologia utilizada para implementar e comparar os resultados adquiridos do sistema.

Na preparação dos dados definiu-se os parâmetros iniciais do sistema respeitando o padrão IEEE 754 de precisão simples. Assim, foi necessário manipular os dados de entrada na RNA e AF, com o intuito de garantir a qualidade, organização e padronização dos dados para embarcar no hardware. A execução do sistema realizou-se em linguagens de níveis de abstrações alto e baixo.

Os resultados obtidos no hardware (FPGA) pela simulação funcional foram decodificados em alto nível para gerar os gráficos e tabelas. A análise dos resultados é feita pela apresentação dos erros e área utilizada. Os dados foram obtidos pelas distintas implementações da RNA, principalmente na alteração da AF do sistema e da plataforma de hardware. O sistema em baixo nível foi comparado ao de alto nível, e para validação do sistema no Matlab, diversas tarefas adicionais foram desenvolvidas para tornar possível a execução de todas as etapas, como:

- Processo de codificação e decodificação dos dados no padrão IEEE Std 754;
- Definição do mapeamento para algumas LUTs
- A organização das memórias ROM baseadas em LUT;

Figura 21 – Fluxo de execução no funcionamento e comparação dos resultados do sistema.



Fonte: Elaboração do próprio autor.

- Obtenção dos dados da equação não linear ideal;
- Implementação do treinamento da RNA MLP;
- A conversão dos resultados obtidos da FPGA;
- A concepção dos dados na análise de erros;
- Gerar os arquivos de resultados e disponibilizar de forma adequada à ferramenta GNU PLOT.

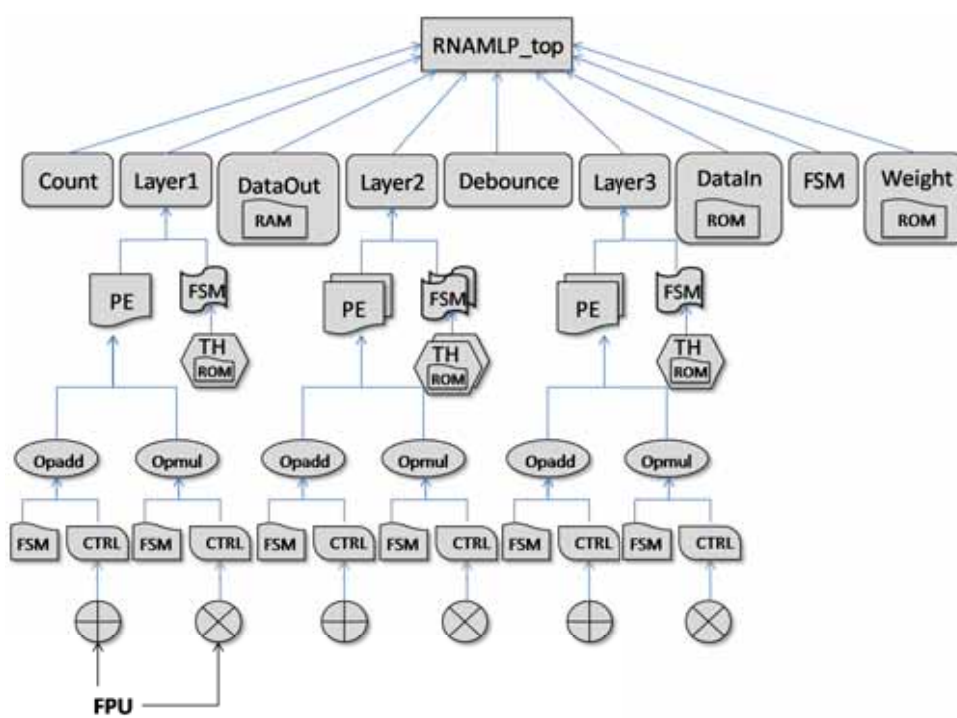
### 5.1 Arquitetura do Hardware

A organização do sistema está definida em 7 níveis, contando com o testbench desenvolvido para realizar a simulação funcional do hardware. A estrutura composta de todos os seus níveis e módulos são apresentadas na Figura 22. Os níveis mais baixos correspondem as unidades de ponto flutuante com operações de soma e multiplicação, junto ao controle de cada operação por meio de máquinas de estados finitos. Os elementos de processamento contêm os módulos para operações aritméticas e estão no mesmo nível dos módulos de aproximação da AF. Para cada camada existe um ou mais elementos de processamento e AF conforme a quantidade de neurônios e entradas na camada. A relação entre todos os módulos, a quantidade de instâncias e alguns submódulos, também são identificados. Outros módulos adjacentes são utilizados para o controle, armazenamento e E/S de dados.

Na Figura 23 apresenta-se a arquitetura do sistema de uma RNA no modelo RTL com os principais módulos e componentes, além das memórias, máquinas de estados finitos, circuitos combinacionais e sequenciais dispostos no sistema.

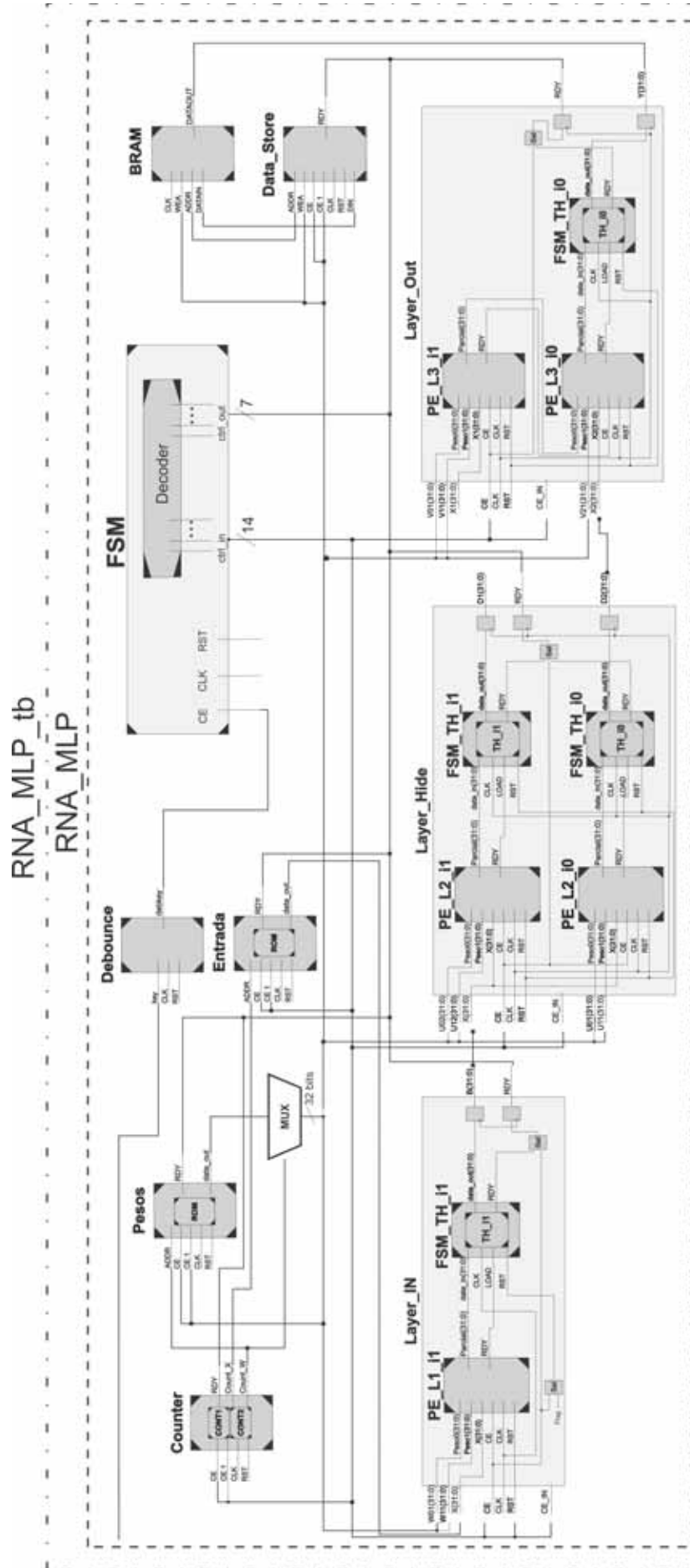
As interligações de alguns sinais foram suprimidas para melhorar a visualização completa. Nesse caso, foi disponibilizado ao nível superior, o *testbench*, responsável pelos estímulos da simulação do sistema e pela coleta dos dados de saída para armazenamento em arquivo. Posteriormente, os resultados são recuperados no Matlab para comparação gráfica entre Matlab e FPGA.

Figura 22 – Árvore com a hierarquia de níveis do sistema.



Fonte: Elaboração do próprio autor.

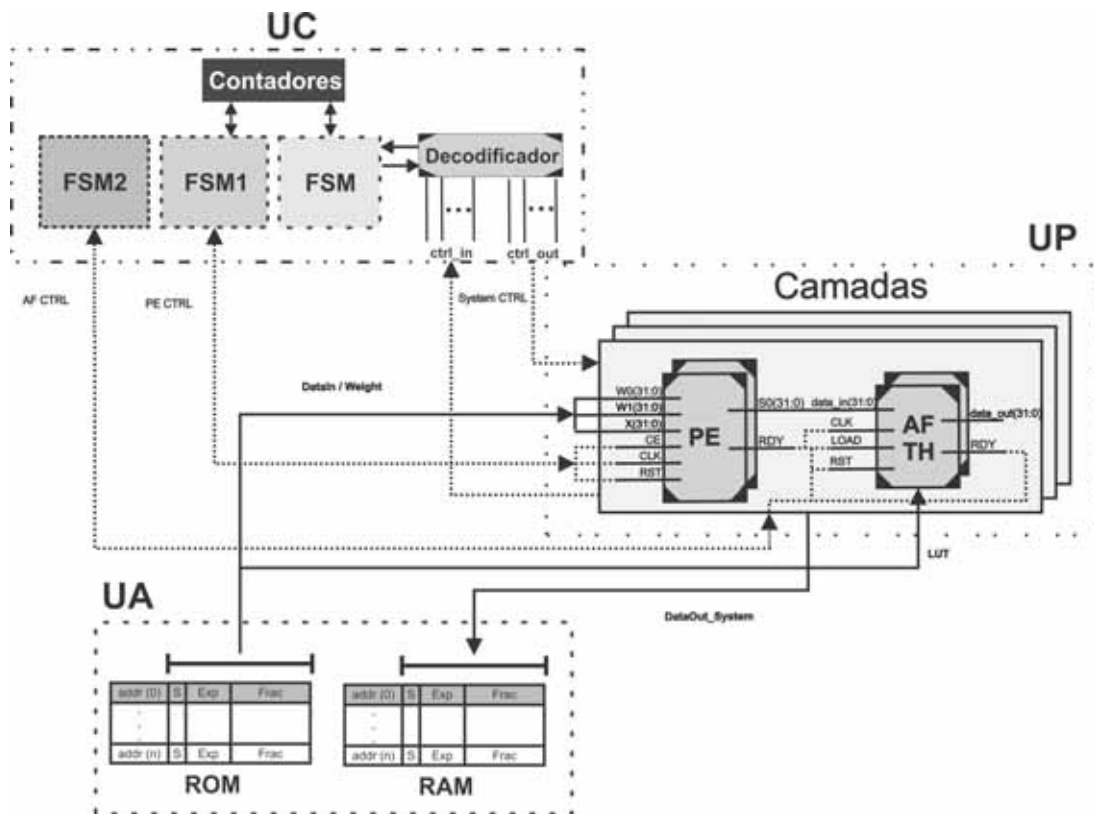
Figura 23 – Arquitetura de hardware para a rede neural artificial.



Fonte: Elaboração do próprio autor.

O sistema está organizado em três partes principais: a unidade de controle (UC), unidade de armazenamento (UA) e unidade de processamento (UP). A relação entre os módulos e os componentes utilizados no hardware do sistema está ilustrado na Figura 24, com destaque para as principais unidades UC, UA e UP.

**Figura 24 – Hardware do sistema com as três principais unidades: UC, UA e UP.**



Fonte: Elaboração do próprio autor.

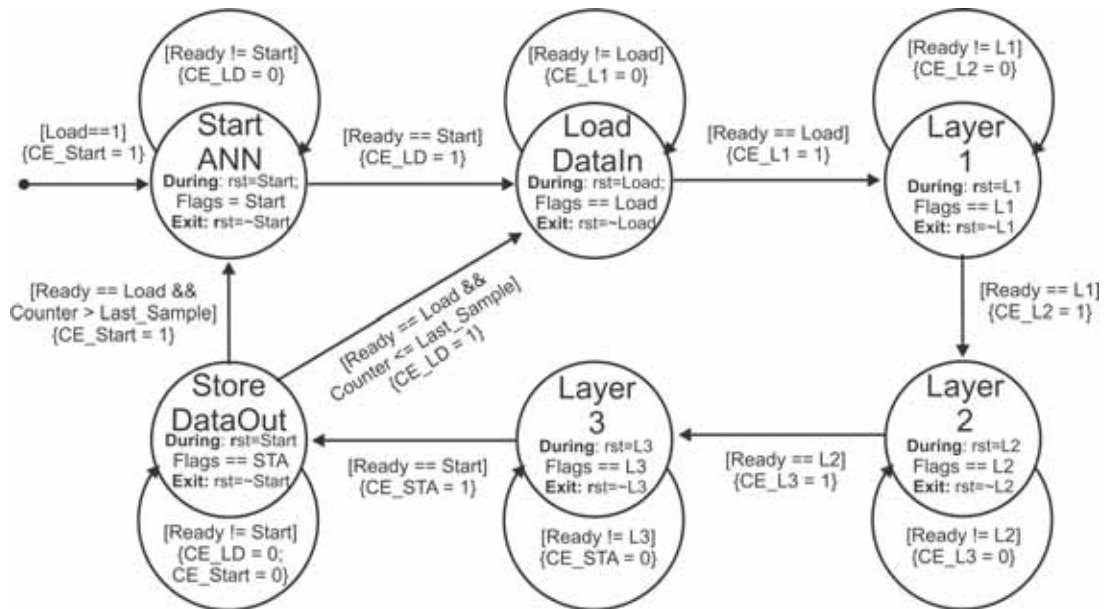
Para a implementação em hardware (FPGA) usou-se VHDL, com módulos IP *Core* do próprio fabricante para a unidade de ponto flutuante e blocos de memória dedicados. Desta forma, para prototipar o sistema em outra plataforma basta integrar os módulos e respeitar suas características específicas, trazendo portabilidade e expansão ao sistema.

### 5.1.1 Unidade de Controle

A principal característica na unidade de controle está na implementação de três máquinas de estados finitos e contadores que proporcionam o fluxo correto dos dados em todo o sistema. A unidade de controle é responsável pelo fluxo de execução direcionado pela máquina de estados e pelos contadores. Esses componentes permitem o percurso adequado dos dados na RNA. O fluxo correto dos dados na RNA não admite problemas ocasionados na execução concorrente das operações, por exemplo, a informação de saída de uma camada não pode ser sobrescrita antes que a próxima camada tenha feito a sua

aquisição. Para isso, definiu-se para todo o sistema uma máquina de estados formada por seis estados (Start, Load, Layer1, Layer2, Layer3, Store), representada na Figura 25.

**Figura 25** – Máquina de estados finitos do controle geral da rede neural artificial.



Fonte: Elaboração do próprio autor.

Cada estado possui sua saída atualizada somente durante a execução do estado, por meio de habilitação do sinal *Ready*, a partir da borda de subida do *clock* do sinal pode-se recuperar os dados de saída da camada de maneira correta.

Por causa da necessidade de um grande controle de todos os módulos do sistema com o intuito de evitar a propagação de valores indesejados em momentos distintos da execução do estado atual da FSM, cada estado contém um sinal de *reset* para reiniciar todos os sinais internos aos módulos e para facilitar posteriores implementações utilizando *pipeline*.

Sinais de controle interno a FSM também são utilizados para codificação e decodificação das entradas e saídas do próprio módulo da FSM. Os sinais de *chip enable* iniciam a execução das funcionalidades contidas no estado. Esses sinais são definidos de forma independente, por não respeitarem o princípio de exclusão mútua, devido principalmente ao *reset* de cada estado e a utilização desses sinais por outros componentes, contadores e interfaces de memória.

Existe uma funcionalidade única em cada estado no fluxo da RNA, e o sinal de *reset* do sistema inicia a FSM. O estado *Start* é executado apenas uma vez após o início do sistema e carrega os pesos transferidos da memória ROM. O estado *Load* carrega as entradas da memória ROM de forma sequencial, uma a uma. Dois contadores, um dos pesos e um das entradas, são definidos para conduzir e alimentar o sistema de forma adequada.

Após a entrada da RNA carregada, o estado Layer1 inicia a execução da RNA para

efetuar as operações no neurônio e habilitar a função de ativação com as quantidades correspondentes em cada camada. Essa funcionalidade é realizada sucessivamente nos três estados correspondentes a cada camada.

O estado *Store* armazena o resultado da RNA em memória dedicada BRAM e verifica se o endereço atual corresponde a última entrada do sistema, caso verdadeiro retorna-se ao estado inicial (*Start*), caso contrário faz a transição para o estado *Load* e carrega a próxima entrada ou amostra do sistema.

### 5.1.2 *Unidade de Armazenamento*

A Unidade de Armazenamento (UA) faz o armazenamento dos dados e resultados do sistema utilizando diferentes tecnologias de memória dedicada. A unidade de armazenamento consiste de memórias do tipo BRAM embarcada e registradores baseados em área lógica da FPGA. A parametrização dos componentes embarcados segue a plataforma utilizada. Geralmente é possível escolher dentre vários tipos e tecnologias distintas dependendo do fabricante e da família da FPGA. O uso de memória dedicada possibilita a redução de área em recursos lógicos da FPGA e o acesso com desempenho superior quando comparado a outros tipos de memória, por exemplo, a SDRAM.

O sistema utilizou dois tipos distintos de memória BRAM, uma apenas de leitura (ROM) e outra para escrita dos dados (RAM). As duas tecnologias utilizam memória de porta simples para acessar os dados. A memória do tipo ROM está relacionada aos dados de entrada e a função de ativação. Na plataforma da Xilinx é carregado um arquivo inicial com extensão do tipo “.coe”, já na Altera o arquivo é do tipo “.mif”. Na memória RAM são armazenados os resultados da RNA para possibilitar posterior recuperação por outro módulo na expansão do sistema. Em todas as memórias utiliza-se precisão simples (32 bits) nas palavras. Por meio de simulação funcional pode-se recuperar os dados de saída do sistema, que também são armazenados em arquivo para a leitura no Matlab.

### 5.1.3 *Unidade de Processamento*

A unidade de processamento é dividida em duas partes, o Elemento de Processamento (PE) e a AF para a execução das três camadas da RNA, com um total de 4 neurônios e topologia 1-2-1. O desenvolvimento da AF em vários cenários é discutido nas Seções 5.1.3.2 e 5.2.

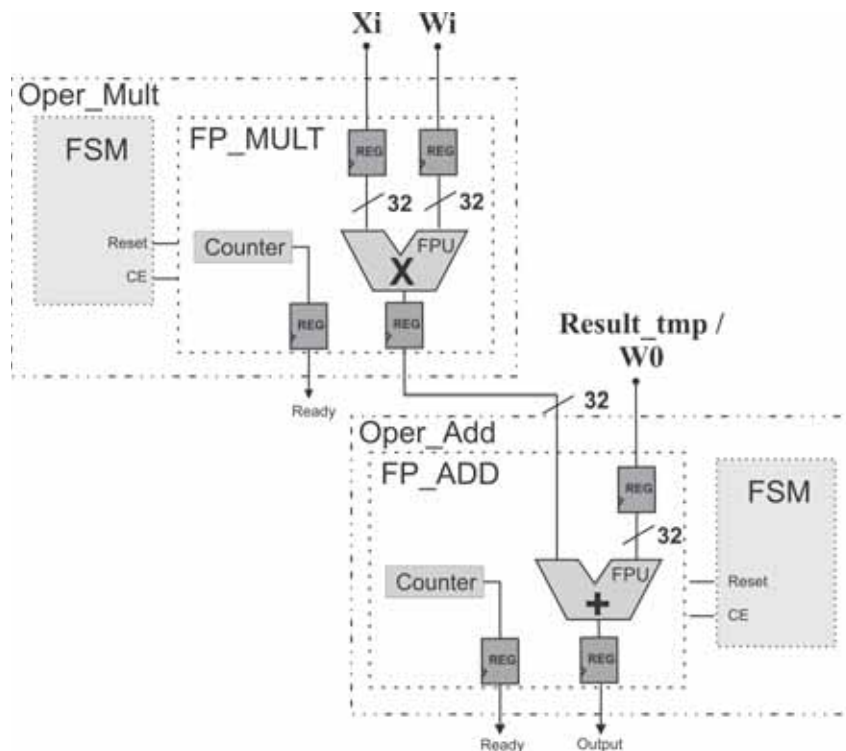
A unidade de processamento contém a soma ponderada entre as entradas e os pesos relacionados ao neurônio, conforme o padrão IEEE 754 de 32 bits. Para fazer esse processamento utiliza-se uma unidade de ponto flutuante baseado em um IP *Core*

correspondente ao fabricante e com otimização de síntese para área. O processamento realizado no sistema é formado pela função de ativação e pelos elementos de processamento nos neurônios. O número desses componentes no sistema está diretamente relacionada à quantidade de entradas em cada neurônio, à quantidade de neurônios em determinada camada e à quantidade de camadas da RNA. Cada entrada em um neurônio contém um elemento de processamento, assim se  $n$  é a quantidade de entradas a  $Qtde_{PE} = 2^{n-1}$ , e cada neurônio contém apenas uma AF baseada na Tangente Hiperbólica.

### 5.1.3.1 Elemento de Processamento

O Elemento de Processamento (PE) restringe-se a um multiplicador e um somador em ponto flutuante, conforme apresentado na Figura 26. Dependendo da quantidade de entradas associadas ao neurônio, necessita-se de mais elementos de processamento para executar os cálculos. A inserção de novos elementos de processamento depende da quantidade de entradas associadas ao neurônio. A unidade de processamento contém a soma ponderada nos neurônios baseada no padrão IEEE 754 de precisão simples com 32 bits. Para fazer esse processamento utiliza-se uma unidade de ponto flutuante baseado em um IP *Core*, neste caso do próprio fabricante.

**Figura 26 – Elemento de processamento do neurônio.**

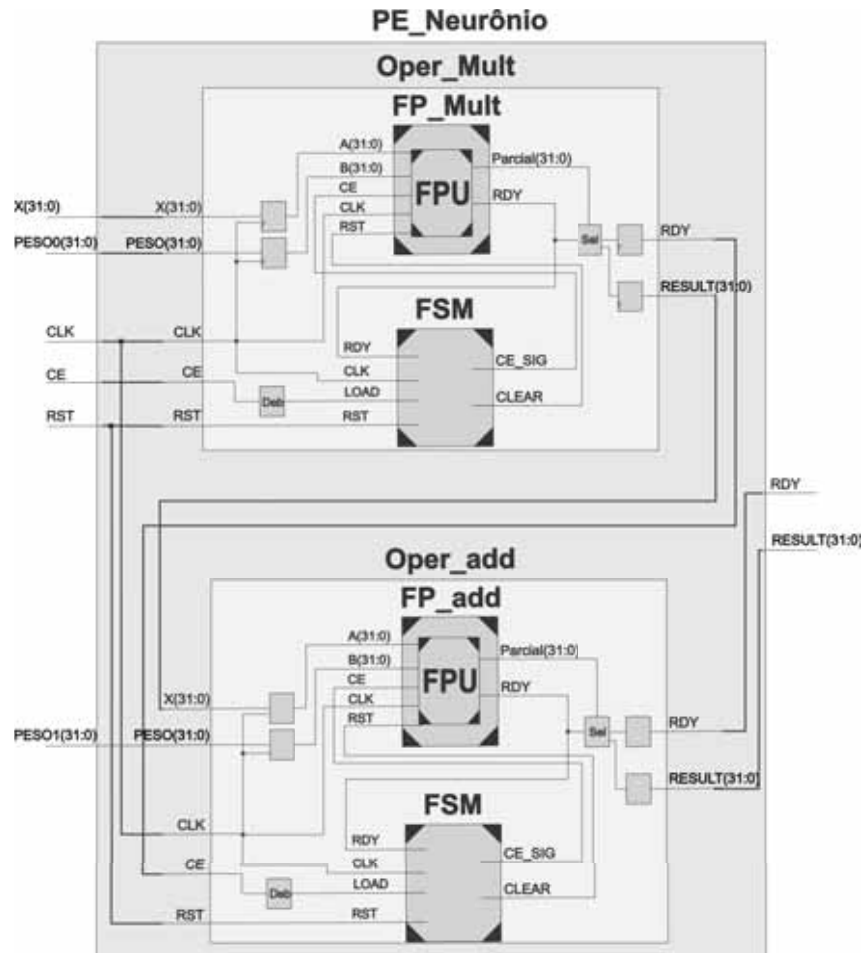


Fonte: Elaboração do próprio autor.

Na Figura 27 apresenta-se o PE organizado em nível RTL extraído do sistema completo da Figura 23. Na Figura 27 pode-se notar os detalhes do módulo e submódulos presentes.

Este módulo contém várias instâncias no sistema e tem a função de ordenar as operações de multiplicação e soma em ponto flutuante.

**Figura 27 – Arquitetura do elemento de processamento.**

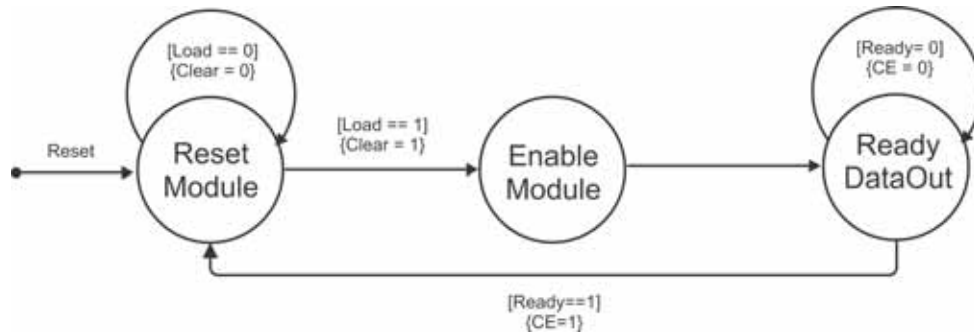


Fonte: Elaboração do próprio autor.

Para ter um maior controle no elemento de processamento foi necessário criar uma estrutura que organiza a inserção das entradas e saídas nos módulos de operações em ponto flutuante. Para isso, foi necessária a inserção de uma máquina de estados, com 3 estados: *Reset*, *Enable* e *Ready*.

O estado *Reset* tem a função de preparar o módulo de ponto flutuante, reiniciando todos os seus sinais com um sinal de *clear*. O segundo estado *Enable*, envia um sinal alto de *chip Enable* para a operação correspondente ser iniciada naquele ciclo. O último estado, *Ready*, aguarda o sinal de pronto, decorrente da verificação dos ciclos de *clock* de cada unidade de ponto flutuante para indicar o término da operação. Esse controle disponibiliza os dados de maneira correta aos módulos relacionados e aos níveis superiores, na Figura 28 apresenta-se o diagrama de estado de controle das operações em ponto flutuante no neurônio.

Figura 28 – Máquina de estados para o controle das operações em ponto flutuante no neurônio.



Fonte: Elaboração do próprio autor.

### 5.1.3.2 Função de Ativação

As funções com características não lineares são mais complexas para implementação em FPGA. A vantagem está na boa precisão e na robustez dos resultados da RNA, tornando o seu emprego necessário para aplicações que necessitam de uma precisão mais refinada.

A resposta da função Tangente Hiperbólica corresponde à Equação (25). A variável  $x$  está relacionada à saída parcial do neurônio na RNA cuja a AF irá gerar um valor correspondente conforme a resposta da função Tangente Hiperbólica, e  $\lambda$  é a inclinação da função. A ativação da AF TH é determinada conforme a característica da função, além de inserir a não linearidade no sistema da RNA. Quando o numerador, na Equação (25), for igual a 1, corresponde a função não linear logística, que possui o comportamento semelhante ao da Tangente Hiperbólica.

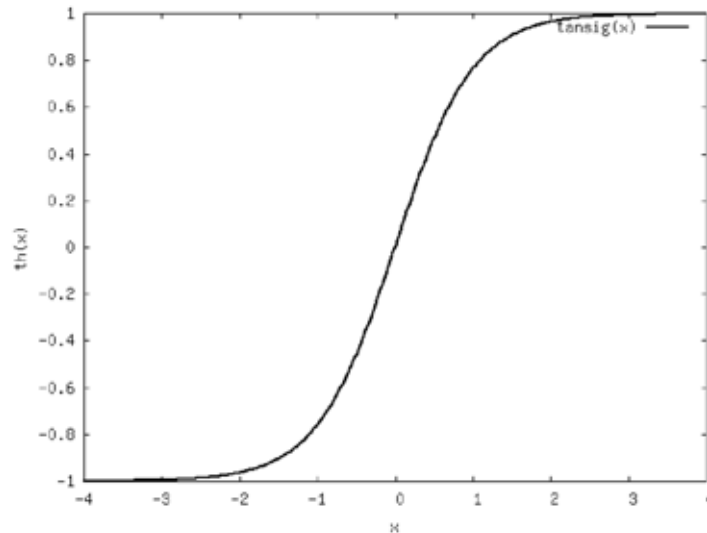
$$Th(x) = \frac{1 - e^{-\lambda x}}{1 + e^{-\lambda x}} \quad (25)$$

A curva “S” da função é uma característica da sigmoide, e a função Tangente Hiperbólica contém valores positivos e negativos, diferente da sigmoide (Logística), composta apenas de valores positivos.

O comportamento da AF é não linear e possui uma curva mais acentuada antes da saturação das duas extremidades, conforme ilustrado na Figura 29. Quatro características a serem observadas na função Tangente Hiperbólica são: (i) A função é ímpar e anti-simétrica, ou seja,  $f(-x) = -f(x)$  (BENJEMMA et al., 2012; MEHER, 2010); (ii) Para valores muito grandes ou muito pequenos de  $x$ , independente do sinal,  $y \rightarrow \pm 1$ ; (iii) A parte central da função é próxima a uma equação de reta (linear); (iv) A saída da função ( $Th(x)$ ) é igual a sua entrada para valores de  $x$  próximos de zero (BENJEMMA et al.,

2012).

**Figura 29 – Resposta da função de ativação Tangente Hiperbólica com 512 pontos.**



Fonte: Elaboração do próprio autor.

## 5.2 Métodos Propostos na Aproximação da Tangente Hiperbólica

A RNA foi desenvolvida utilizando diferentes métodos para a aproximação da função de ativação. Assim, o desenvolvimento do hardware foi dividido em duas etapas para a análise da aproximação e verificação de erros, uma apenas com a função de ativação e outra com o hardware da rede neural artificial completo, inclusive com o módulo da AF. Nessas duas etapas são analisados dois diferentes grupos de cenários, correspondentes às Tabelas 9 e 10.

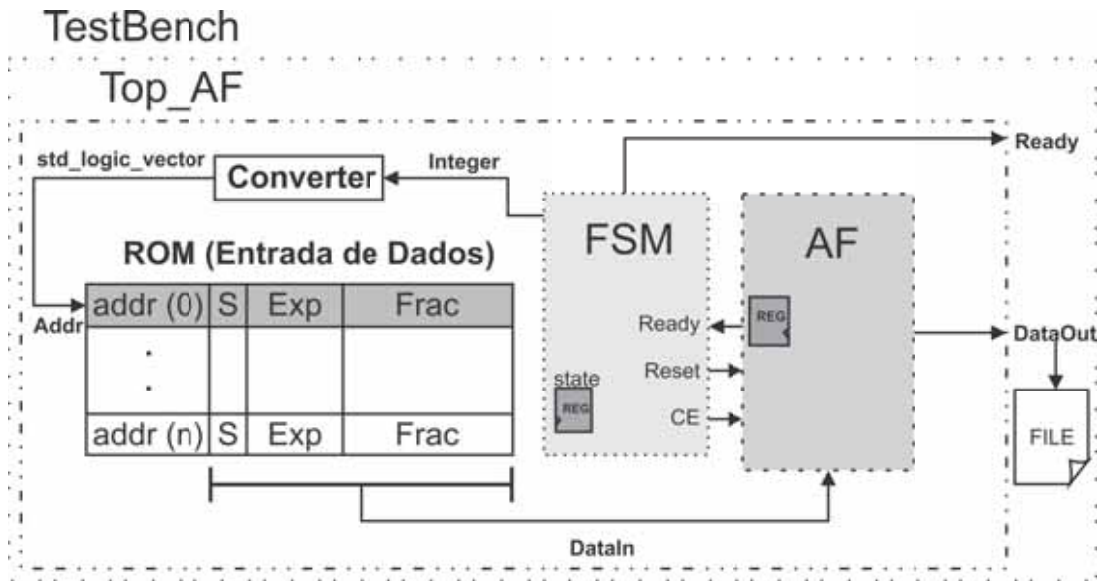
Pelas dificuldades e grande variedade de técnicas na implementação da AF com comportamento não linear, faz-se necessária a implementação de diversas arquiteturas para obter um equilíbrio entre precisão, desempenho e área utilizada na FPGA.

### 5.2.1 Arquitetura de Hardware para Função de Ativação

Os testes com a AF foram feitos em separado, antes do seu acoplamento à RNA. Nesse subsistema foram utilizadas 500 amostras armazenadas em memória dedicada BRAM junto a uma máquina de estados para disponibilização dos dados, esse subsistema é detalhado na Figura 30.

Um novo módulo em nível superior foi criado para incluir os módulos da FSM, a AF e a memória dedicada para as entradas. Os resultados são apresentados na seção 5.3. O

Figura 30 – Arquitetura destinada aos experimentos realizados com AF independente da RNA.



Fonte: Elaboração do próprio autor.

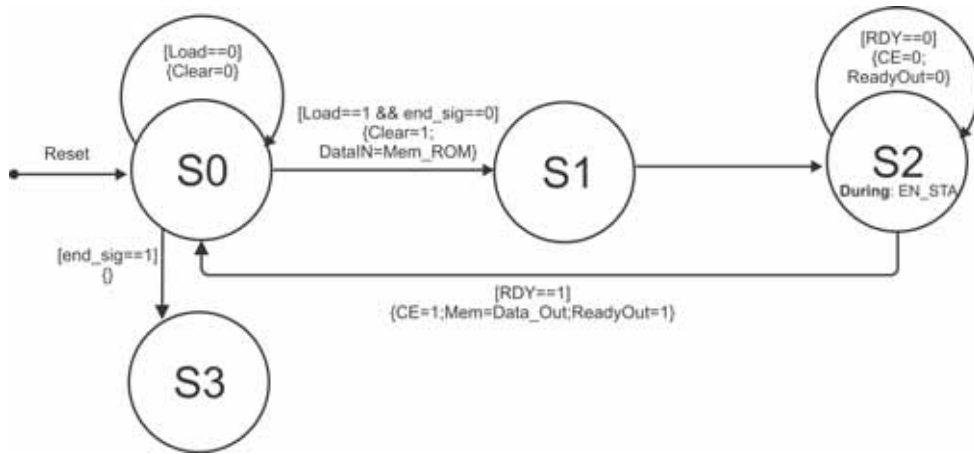
objetivo é ter um maior controle sobre as ações dos módulos principais da RNA, facilitar a manutenção e solucionar possíveis problemas com maior agilidade.

Os valores de entrada em todos os cenários da AF, cujo intervalo  $[-4:+4]$  não é válido, são codificados da mesma forma. Isso devido à característica discutida em (ii) na seção 5.1.3.2 para função Tangente Hiperbólica. Outra importante característica em todas as implementações da tabela (LUT) é a sua organização em ordem decrescente, isso reduz a quantidade de comparadores, sendo necessário apenas um operador relacional para fazer o mapeamento dos endereços da memória dedicada.

A FSM para o controle do subsistema da função de ativação é ilustrado pela Figura 31. O funcionamento da FSM inicia-se após o sinal de *reset*, no estado “S0” espera-se o sinal de *load* para carregar os dados de entrada da memória dedicada em um sinal interno do módulo, e posterior disponibilização ao módulo da AF. Um sinal de *clear* é habilitado para zerar os dados do módulo da AF. O estado “S1” faz o processamento da AF com um atraso de 1 ciclo de *clock*. O estado “S2” aguarda o sinal de controle “*ready*” que o módulo de processamento da AF irá gerar, para disponibilizar os dados de saída e a habilitação da próxima entrada. Quando retornar ao estado “S0” e a entrada anterior não seja a última, é realizado todo o processo novamente, caso contrário é habilitado o sinal de controle *end\_sig* para finalizar a FSM no estado “S3”.

O módulo desenvolvido para disponibilizar os dados de entrada e mapear a saída conforme a resposta da AF incorporado na RNA é semelhante ao da arquitetura do subsistema. Porém, a FSM é distinta e a entrada de dados não está relacionada a uma memória dedicada, pois essa entrada corresponde à saída parcial do neurônio pertencente

Figura 31 – Máquina de estados para o controle da função de ativação e testes iniciais do subsistema.

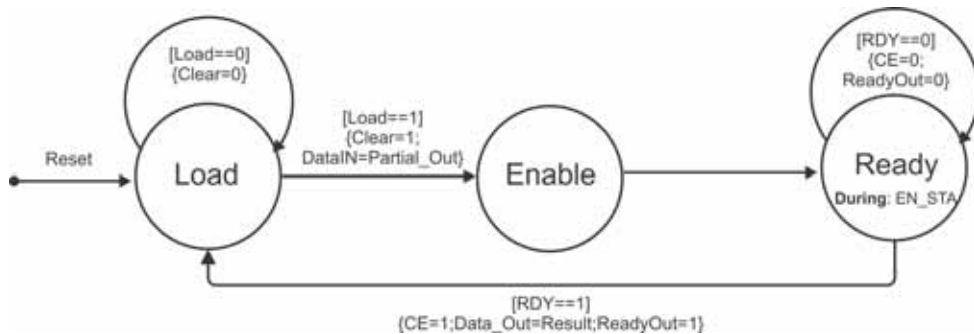


Fonte: Elaboração do próprio autor.

à RNA.

O módulo adicional da FSM apresentado na Figura 32 foi elaborado com o intuito de incorporar uma das implementações (cenários), e controlar a entrada e a saída do subsistema correspondente à função de ativação. A FSM contém 3 estados: *Load*, *Enable* e *Ready*. O estado *Load* inicia a máquina de estados, carrega a entrada adquirida da memória ROM e faz um reset no módulo da AF, quando é habilitado pelo sinal de CE faz a transição para o estado *Enable*. O *Enable* não produz nada, e serve apenas para estabilizar os dados de entrada da AF. O *Ready* tem a função de habilitar a execução do módulo que retorna a resposta (saída) da Tangente Hiperbólica, e avisa o momento em que a saída está disponível de forma correta no barramento. Vale ressaltar, que o módulo superior da Figura 30 e a FSM da Figura 32 foram elaborados apenas para os testes da AF, não estão inclusos nos resultados da RNA completa.

Figura 32 – Máquina de estados para o controle adequado da execução da função de ativação na E/S.



Fonte: Elaboração do próprio autor.

### 5.2.2 Cenários e Intervalos

O desenvolvimento da AF fundamentou-se em quatro diferentes métodos para sua implementação: comportamental, PWL com comportamental, PWL com RALUT personalizada (HPR) e PWL combinacional (HPC). Todos os métodos empregados são considerados híbridos, com exceção do primeiro (comportamental). Para validar as implementações dos cenários é necessária a seleção dos dados pertinentes à entrada da AF e depois a disponibilização da saída conforme o mesmo padrão, neste caso, precisão simples (32 bits).

O primeiro grupo de cenários trouxe uma importante informação a respeito dos vários métodos adotados, em que os métodos híbridos (HPR e HPC) contêm melhor qualidade em relação às características de área, desempenho e precisão. O segundo grupo de cenários foi utilizado para a análise com maior cuidado da implicação da alteração de quantidades de elementos em cada intervalo dos métodos híbridos (HPR e HPC). Com o segundo grupo de cenários foi possível determinar os melhores cenários de cada método híbrido, relacionado principalmente, com a área e a precisão dos resultados.

Todos os cenários, com exceção do primeiro, apresentado na Tabela 9, contêm um multiplexador para fazer a seleção do ponto fixo da entrada, conforme o expoente em representação com ponto flutuante. A entrada e a saída da Tangente Hiperbólica estão relacionadas aos valores de expoentes de acordo com a Equação (26) e parte da fração (1 bit), por exemplo, para diferenciar os expoentes com os mesmos valores (128).

$$\text{Exp} = \begin{cases} 129\dots255 & \text{for } x \geq 4 \\ 128 & \text{for } 4 > x \geq 3 \\ 128 & \text{for } 3 > x \geq 2 \\ 127 & \text{for } 2 > x \geq 1 \\ 126 & \text{for } 1 > x \geq 0,5 \\ 125 & \text{for } 0,5 > x \geq 0,25 \\ 0\dots124 & \text{for } 0,25 > x \geq 0 \end{cases} \quad (26)$$

Em todas as implementações utilizou-se a seleção dos expoentes, para reduzir a quantidade de bits manipulados no módulo da AF. Além disso, a codificação dos dados dos expoentes garantiu a saída desejada, pois os bits relacionados ao expoente servem como parte da entrada no mapeamento dos endereços e na resolução das expressões Booleas, respectivamente, nas implementações com (PWL e RALUT) e (PWL e Combinacional).

No primeiro grupo de cenários realizaram-se 6 implementações de distintas arquiteturas para AF Tangente Hiperbólica, listadas na Tabela 9. Os cenários híbridos

na Tabela 9 estão relacionados com a união dos métodos PWL e Comportamental nos cenários 2 e 3, os métodos PWL e RALUT nos cenários 4 e 5, e o cenário 6 é desenvolvido baseado nos métodos PWL e Combinacional.

**Tabela 9 – Grupo 1 - cenários analisados para AF Tangente Hiperbólica.**

Nº	Cenários
1	Comportamental
2	Comportamental com Híbrido de 128 Elementos
3	Comportamental com Híbrido de 256 Elementos
4	Híbrido de 212 Elementos
5	Híbrido de 442 Elementos
6	Combinacional (Funções Lógicas baseadas no Híbrido de 212 Elementos)

Após a obtenção dos resultados do primeiro grupo, observou-se a necessidade de uma investigação com diferentes quantidades de elementos nos intervalos, para os dois principais métodos híbridos escolhidos o HPR (Híbrido - PWL com RALUT) e HPC (Híbrido - PWL com Combinacional). No segundo grupo de cenários foram analisados esses dois métodos híbridos com a variação na quantidade de elementos em cada intervalo.

A RNA foi desenvolvida utilizando os dois métodos híbridos com divisão em partes lineares (PWL), sendo cinco HPR dedicados (cenários - 1 a 5) e quatro HPC (cenários - 6 a 9), Tabela 10. A principal diferença é a combinação com métodos RALUT (HPR) e Combinacional (HPC). No entanto, existem cinco diferentes quantidades de elementos especificados para as divisões dos segmentos. Nos cenários HPC são necessárias uma quantidade de elementos em potência de 2, assim o intervalo com 314 elementos não foi possível ser implementado.

**Tabela 10 – Grupo 2 - cenários analisados para AF Tangente Hiperbólica.**

N.	Cenários
1	HPR de 180 Elementos
2	HPR de 212 Elementos
3	HPR de 314 Elementos
4	HPR de 340 Elementos
5	HPR de 404 Elementos
6	HPC de 180 Elementos
7	HPC de 212 Elementos
8	HPC de 340 Elementos
9	HPC de 404 Elementos

A diversidade de cenários analisados são necessárias pela grande diferença encontrada nos resultados de área e precisão, dependendo do tipo de cenário escolhido, e isso poderia ocasionar grandes problemas em todo sistema da RNA relacionados a área, precisão e desempenho.

Todas as implementações contêm características do método RALUT, com exceção do comportamental, isso devido ao número de bits usado na fração da RNA (Entrada de

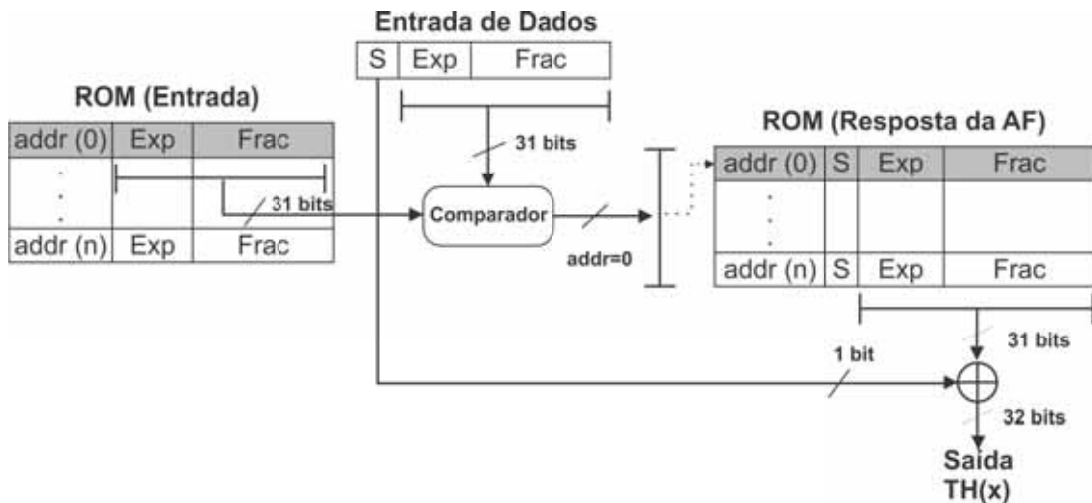
Dados) ser maior que o do módulo da AF, sempre em conformidade com o IEEE Std 754. Os tipos de mapeamentos utilizados em todos os cenários estão na seção 5.2.3.

### 5.2.3 Mapeamento dos Métodos

No primeiro grupo, o cenário 1 apresentado na Tabela 9 faz o mapeamento por meio de código comportamental descrito em VHDL. O algoritmo segue de forma semelhante a implementação em alto nível, assim criam-se duas tabelas ou conjuntos em memória, uma relacionada à entrada e outra à resposta da AF, conforme apresentado na Figura 33.

O valor de entrada  $x$  é comparado aos bits do expoente (operador de igualdade) no conjunto de entrada e os bits da fração seleciona o primeiro endereço que contenha uma fração maior que a entrada, já que o conjunto foi ordenado de forma decrescente na memória. Com o endereço selecionado a resposta da função está contida no mesmo endereço do segundo conjunto.

**Figura 33 – Implementação correspondente ao 1º cenário (comportamental).**

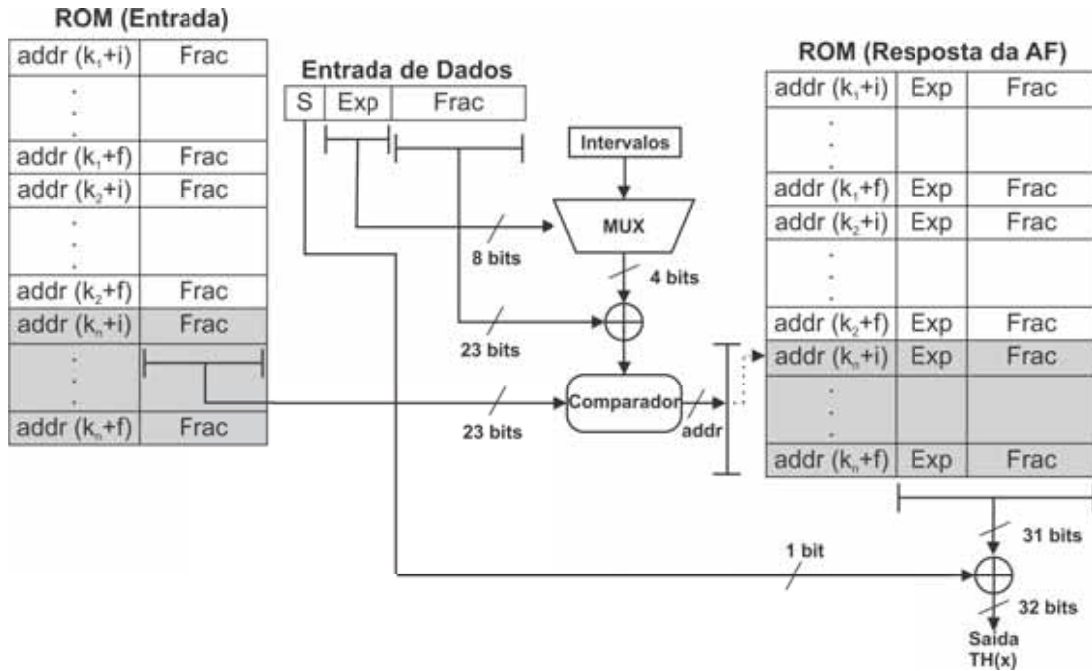


Fonte: Elaboração do próprio autor.

Os cenários (2 e 3) do primeiro grupo, apresentado na Tabela 9, diferem-se do cenário 1 por terem um multiplexador para encontrar o expoente de entrada e mapear um valor de início (i) e fim (f) no conjunto de entrada na busca do maior valor de fração. Assim, o multiplexador faz a divisão em intervalos de endereçamentos, de acordo com o expoente de entrada, e o comparador é utilizado para verificar a fração mais próxima ao valor de entrada. A busca da fração é feita por diversos laços, um para cada intervalo, com quantidades de repetições menores que no primeiro cenário. Na Figura 34 apresenta-se o fluxo dos dados nas implementações dos cenários 2 e 3 do primeiro grupo.

Para os cenários (2 e 3) do primeiro grupo, a característica (iv) considerou-se duas abordagens distintas para o mesmo conjunto de dados de entrada,  $0 \leq x < 0,5$  e  $0 \leq$

Figura 34 – Implementação correspondente aos cenários 2 e 3 (comportamental).



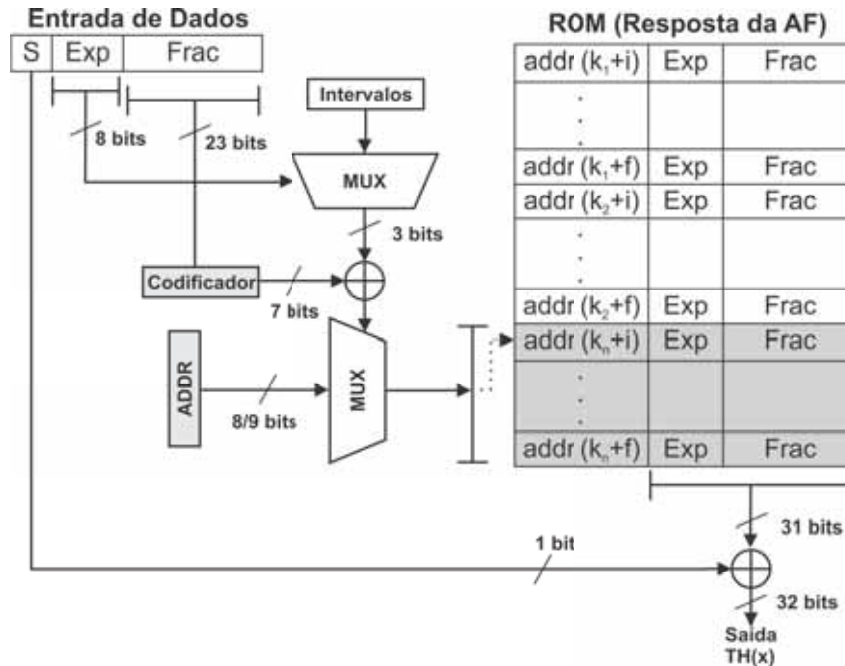
Fonte: Elaboração do próprio autor.

$x < 0,375$ , respectivamente. Esses intervalos foram feitos com base no erro absoluto apresentado nas diferentes abordagens e encontrou-se valores que apresentavam a mesma relação.

Os cenários (4 e 5) do primeiro grupo da Tabela 9, e os cenários (1 a 5) do segundo grupo da Tabela 10, fazem o mapeamento da RALUT com endereçamento de 8 bits (método HPR). Na memória armazenam-se apenas valores positivos da resposta da AF, e antes da disponibilização dos dados na saída é feita a seleção do sinal pelo valor de entrada. A divisão em pequenos intervalos permite um mapeamento para a memória dedicada com relação ao número de elementos apropriados em cada intervalo, baseado em erros absolutos. Assim, no pior caso todos os dados podem ser armazenados em aproximadamente 9KB, ou seja, metade de uma BRAM de 18KB. Na manipulação da AF utilizou-se distintas representações para essas implementações, os valores adotados são de S=1, E=3, F=7. Desta forma, “S” representa o sinal, “E” o expoente e “F” a fração do número baseado na abordagem em ponto flutuante, para fazer o mapeamento da LUT. Com essa organização pode-se ter no máximo 8 intervalos e até 256 elementos em cada intervalo.

As implementações do método HPR, nos dois grupos de cenários, contêm as mesmas arquiteturas para o mapeamento, a diferença está na quantidade de elementos determinados, conforme a necessidade de cada segmento. Essa abordagem diminui a quantidade de elementos em segmentos com variação baixa e determina valores adequados

Figura 35 – Implementação correspondente aos cenários 4 e 5 (PWL e RALUT).



Fonte: Elaboração do próprio autor.

nos segmentos com variação alta. A divisão dos segmentos e as quantidades de elementos em cada implementação é apresentada na Tabela 11.

Tabela 11 – Número de elementos nos intervalos da função de ativação Tangente Hiperbólica.

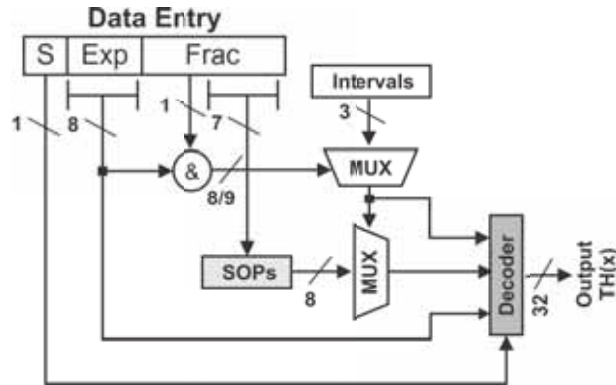
Intervalos	180	212	314	340	404	442
$x \geq 4$	1	1	1	1	1	1
$4 > x \geq 3$	2	2	2	2	2	2
$3 > x \geq 2$	16	16	16	16	16	16
$2 > x \geq 1$	64	64	102	128	128	102
$1 > x \geq 0,5$	64	64	128	128	128	256
$0,5 > x \geq 0,25$	32	64	64	64	128	64
$x < 0,25$	x	x	x	x	x	x

As entradas do sistema e as LUTs são armazenadas em memória dedicada (BRAM) em todos os cenários HPR, já nos cenários HPC somente as entradas são armazenadas em memória dedicada. O cenário 6 do primeiro grupo da Tabela 9, e os cenários (6 a 9) do segundo grupo da Tabela 10, implementam o método HPC e utilizam as mesmas quantidades de elementos do método HPR, especificados na Tabela 11. O mapeamento é direcionado para a saída da AF por meio de lógica puramente combinacional, como apresentado na Figura 36.

Como a AF Tangente Hiperbólica é uma função ímpar, a aproximação nos métodos HPR e HPC são realizadas apenas para valores positivos. Nos mapeamentos dos métodos apresentados nas Figuras 35 e 36 pode-se verificar que o sinal dos dados de entrada são

propagados para a resposta da AF. Assim, a prototipação dos métodos utilizam menor quantidade de área na FPGA, e no método HPR menor quantidade de memória BRAM.

**Figura 36 – Implementação correspondente aos cenários 6 a 9 (PWL e combinacional).**



Fonte: Elaboração do próprio autor.

Os cenários HPR fazem o mapeamento para uma memória de endereçamento que contém as saídas da AF. Já os cenários HPC, usam expressões booleanas para gerar a saída da AF de acordo com o intervalo. Para efetuar o mapeamento nos cenários HPC, cada intervalo contém uma soma de produtos (SOP) para cada bit de saída. SOPs foram feitas baseadas nas quantidades de elementos nos intervalos, e simplificadas com o algoritmo de Quine-McCluskey utilizando a ferramenta Karma (KLOCK et al., 2010). A quantidade de bits de saída da expressão booleana é o múltiplo da quantidade de elementos no intervalo, o resto é constante. A escolha do expoente de saída correspondente é feito com os dados de entrada e o resultado da fração, sempre com múltiplo de 2. A saída contém 8 bits, pelo fato de 1 bit da fração de saída ser utilizado para diferenciar os expoentes de cada intervalo, quando necessário. Na Figura 36 apresenta-se a arquitetura descrita para o mapeamento do circuito combinacional em hardware.

A comparação com a literatura atual é feita na seção 5.3, porém geralmente as implementações são realizadas com padrões distintos ou tamanhos reduzidos para os bits de expoente e fração. Além disso, nem sempre os parâmetros de resultados analisados são iguais, isso trouxe grandes dificuldades na comparação dos resultados com a literatura.

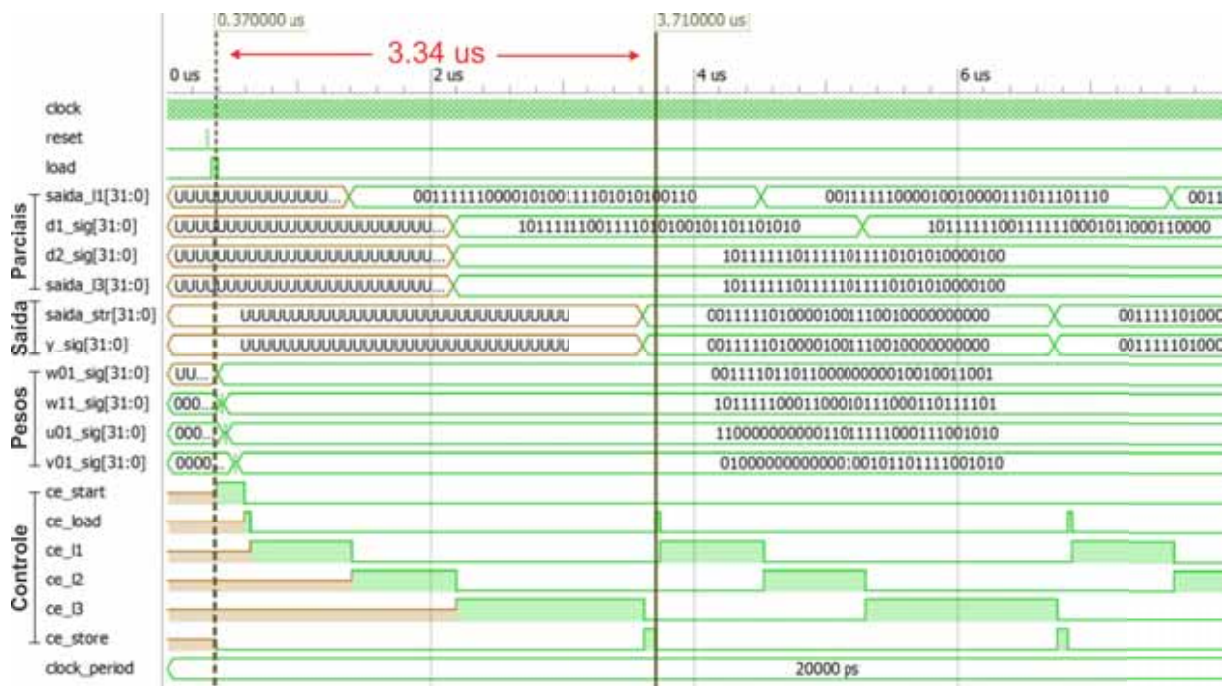
### 5.3 Resultados

A análise dos resultados foram definidas com base em três principais partes do sistema a UC, UA e UP. Para cada parte são apresentados os resultados específicos das implementações. O sistema da RNA foi simulado com 128 entradas para verificar a sua robustez. Na comparação dos resultados foram feitas entre distintos cenários, com a implementação no Matlab e a literatura.

A UC consiste principalmente na máquina de estados. A simulação funcional do componente com a primeira amostra ou ciclo de execução pode ser observada na Figura 37. Duas importantes observações são o tempo de execução de cada estado da FSM e o tempo total de execução da RNA. O tempo de execução em cada estado correspondente às camadas da RNA e está diretamente relacionados a UP.

A primeira amostra que determina o tempo de execução em 3,34 microssegundos está em destaque na Figura 37. Essa amostra é a que gasta maior tempo, pelo fato de conter um estado a mais na FSM, o *Start*, responsável por carregar os pesos da RNA. A partir da segunda amostra, esse estado não é mais executado e o tempo diminui para 3,12 microssegundos. Com 128 entradas o sistema executou em 392 microssegundos para os cenários 1, 2 e 3, os cenários 3, 4 e 5 obtiveram um desempenho de 399 microssegundos. O desempenho do sistema pode ser melhorado com o uso de *pipeline*, assim como a execução do sistema está dividida em várias partes (estados), é possível a implementação de um pipeline de até 5 estágios, um para cada estado da FSM, com exceção do *Start*. Em linguagem de alto nível o tempo de execução alcançado é de aproximadamente 109 milissegundos, o hardware é 275 vezes mais rápido que o software.

**Figura 37 – Simulação da máquina de estados da RNA.**

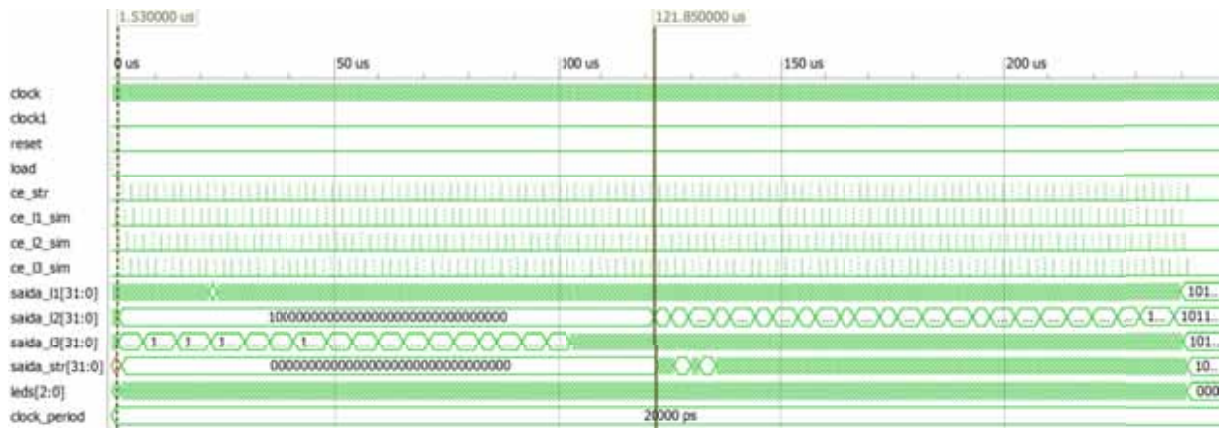


Fonte: Elaboração do próprio autor.

O emprego do IP *Core* de ponto flutuante Havoc escolhido no estudo de caso da seção 4 apresentou erros (*Bugs*) com arquitetura desenvolvida, impossibilitando a sua utilização. Na Figura 38 apresenta-se o erro registrado em destaque na simulação da RNA com o IP *Core*, pode-se observar o funcionamento correto da FSM, mas as saídas das camadas 2

e 3 foram prejudicadas com a repetição de dados anteriores. Os ciclos de tempo foram verificados de forma equivalente aos demais, porém não houve sucesso na utilização desse IP *Core*. Assim, a utilização de IP *Cores* dos próprios fabricantes com multiplicadores dedicados, foram as melhores opções pelos resultados de desempenho e área alcançados no estudo apresentado no Capítulo 4.

**Figura 38 – Erro na simulação com IP core Havoc.**



Fonte: Elaboração do próprio autor.

O componente analisado na unidade de processamento foi a função de ativação, Tangente Hiperbólica. Com a intenção de avaliar a validação desse componente essencial ao sistema, criou-se um projeto independente para fazer a aquisição dos erros, absoluto e relativo, além do custo de área entre as distintas implementações, seção 5.3. A verificação dos resultados de aproximação da AF Tangente Hiperbólica e da Equação (24) pela RNA com todas as implementações pode ser verificada nas Figuras 39 e 41. A função de ativação Tangente Hiperbólica foi baseada na “tansig” do Matlab e utilizada em todos neurônios da RNA, nos cenários abordados.

As Seções 5.3.1 e 5.3.2 apresentam os resultados, respectivamente, dos grupos 1 e 2 de cenários conforme apresentado nas Tabelas 9 e 10.

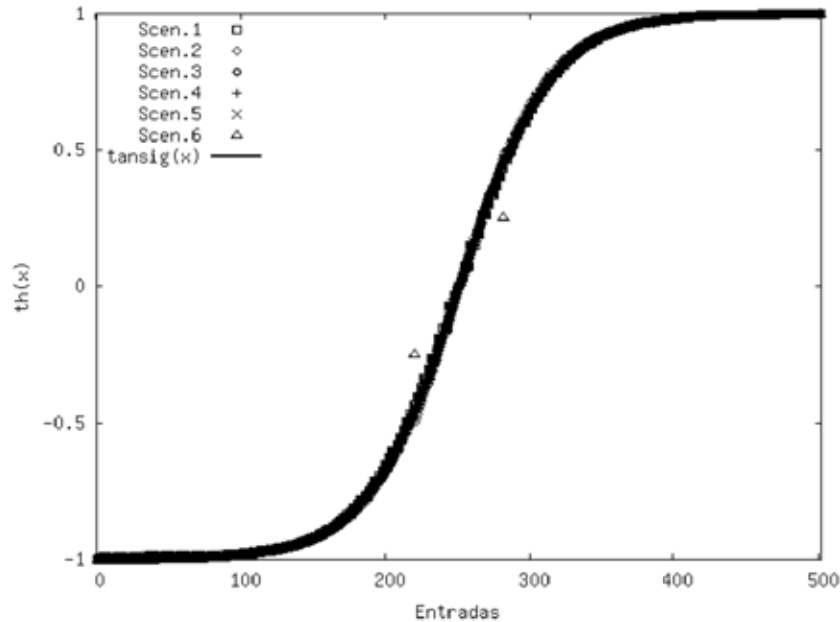
### 5.3.1 Cenários do Grupo 1

#### 5.3.1.1 Aproximação

A Figura 39 ilustra a aproximação de todas as implementações da AF Tangente Hiperbólica com 500 amostras. Para uma melhor visualização das aproximações, o gráfico foi particionado em três partes com 80 pontos, correspondentes à parte negativa da função Tangente Hiperbólica, conforme apresentado nas Figuras 40-(a;c;e). As Figuras 40-(b;d;f), correspondem às partes específicas das Figuras 40-(a;c;e) com 80 pontos, respectivamente.

A quantidade de pontos foi reduzida para possibilitar uma visualização com escala menor, tornando evidente a diferença entre as implementações.

**Figura 39 – Aproximação da AF Tangente Hiperbólica com 500 pontos em todas as implementações.**



Fonte: Elaboração do próprio autor.

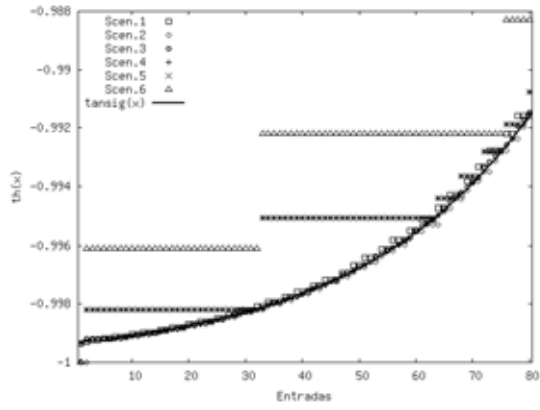
A investigação dos resultados na implementação do sistema da RNA completo para o primeiro grupo de cenários é apresentada na Figura 41. O sistema completo visa aproximar a função não linear da Equação (24). De forma equivalente ao apresentado nos resultados da AF, na Figura 42 apresenta-se a aproximação dividida em partes com escala reduzida de valores e especificados em suas legendas.

Resultados satisfatórios foram obtidos na aproximação da função senoide utilizando a RNA e na aproximação da AF Tangente Hiperbólica. As diferenças encontradas entre as aproximações e os resultados ideais das equações (24) e (25) foram mínimos, com isso a boa precisão dos resultados mostra a relevância dos métodos adotados.

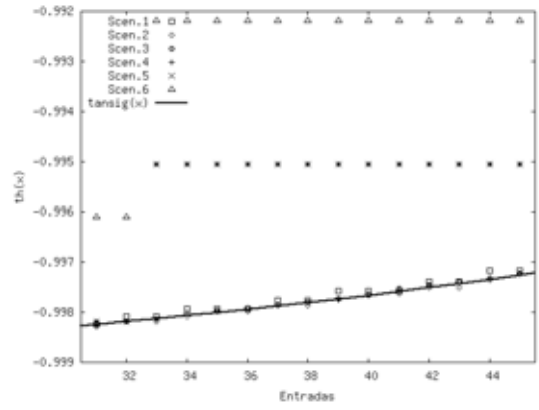
A comparação entre a implementação na FPGA com o valor ideal contém uma validação adequada com baixa taxa de erro, como pode ser visto na seção 5.3.1.2, proporcionando um sistema adequado para implementações com RNA. Com os resultados apresentados nas Figuras 40 e 42 pode-se observar que as diferenças na aproximação do sistema completo da RNA têm grande influência no método utilizado na função de ativação. Isso evidencia a importância da função de ativação para a aplicação da RNA em hardware, tornando-se um fator essencial nesse tipo de implementação.

Os dados coletados na RNA para a análise de erro do sistema foram adquiridos pela simulação funcional. As entradas e os resultados finais da RNA são armazenados em

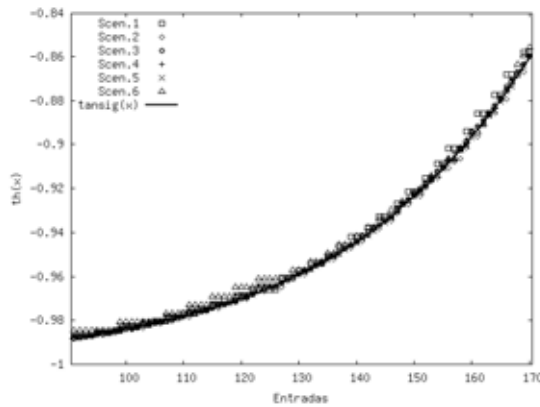
Figura 40 – Aproximação da AF Tangente Hiperbólica: (a) 80 pontos; (b) 15 pontos; (c) 80 pontos; (d) 15 pontos; (e) 80 pontos; (f) 15 pontos.



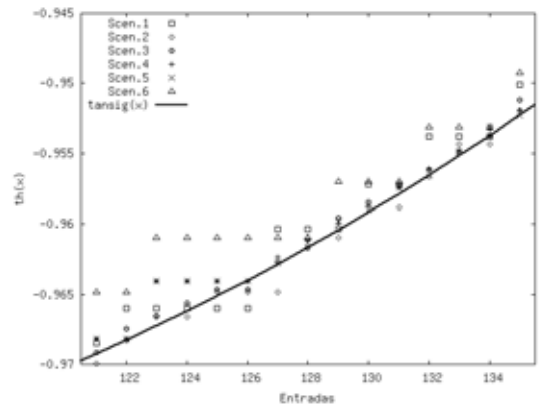
(a)



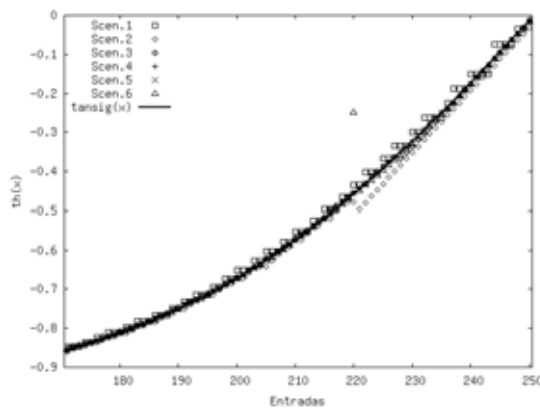
(b)



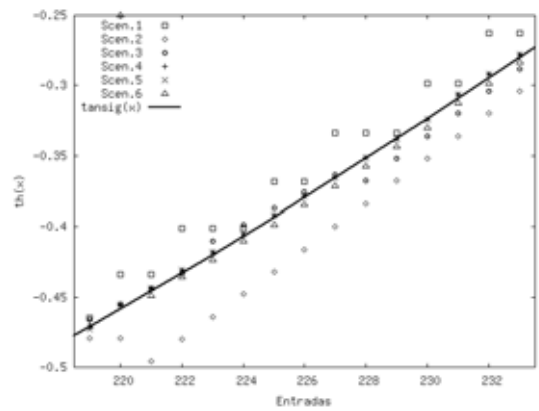
(c)



(d)



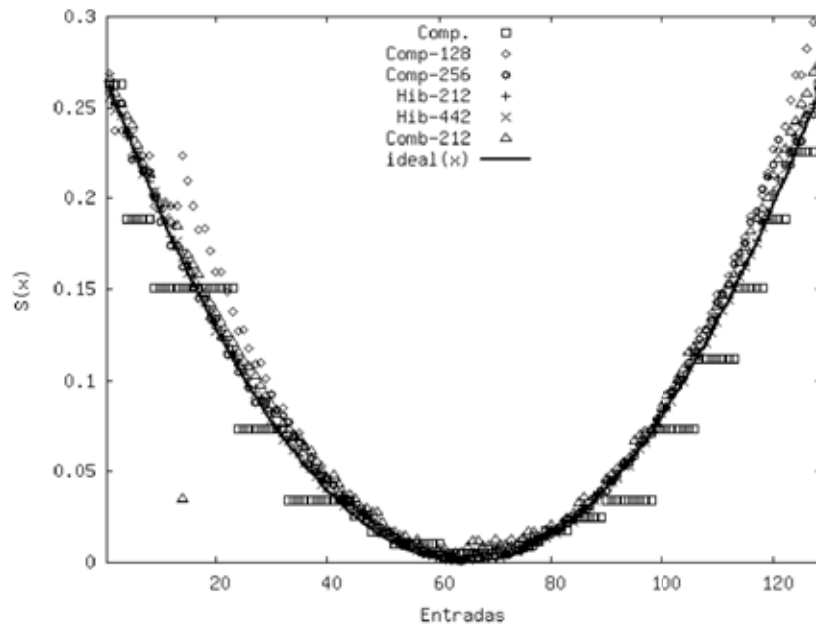
(e)



(f)

Fonte: Elaboração do próprio autor.

Figura 41 – Aproximação da função não linear pela RNA em todas as implementações.



Fonte: Elaboração do próprio autor.

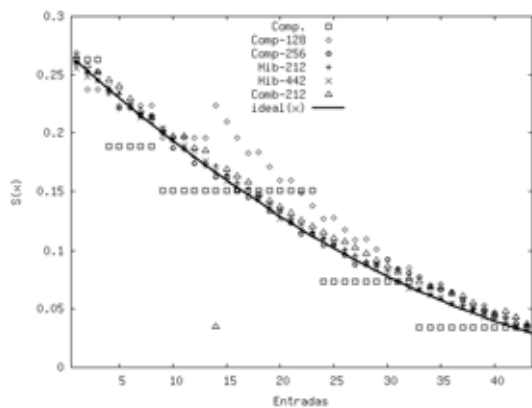
memória dedicada, isso facilita posteriores implementações em tempo real na transferência de dados por um barramento de comunicação interligado ao sistema.

A aproximação da função não linear no sistema da RNA obtiveram resultados satisfatórios nos cenários apresentados, com exceção do cenário 6 que apresentou um erro próximo ao valor 0,25 no eixo y. Essa diferença, apresentada pelo cenário 6, foi verificada e solucionada. Os resultados são apresentados na seção 5.3.2 com a solução do problema para esse mapeamento. O problema ocorria devido a valores muito próximos de 0,25 e não tinha o mapeamento para um determinado expoente.

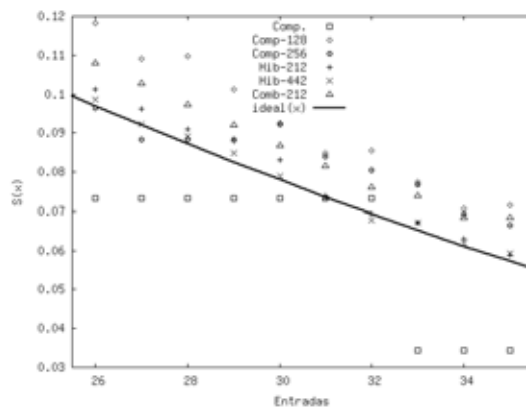
### 5.3.1.2 Análise de Erros e Área

A análise de erros contém a relação de diversos tipos de erros para uma melhor verificação do sistema da RNA e seus componentes. A função  $f(x_k)$  representa o resultado da RNA conforme Equação (24) e também a resposta da AF Tangente Hiperbólica da Equação (25), apenas para poder definir os tipos de erros analisados. A função  $\hat{f}(x_k)$  é o valor aproximado obtido conforme as implementações realizadas. O erro absoluto foi adquirido conforme Equação (27), o relativo é representado pela Equação (28) e o médio quadrático pela Equação (29). A média dos erros absoluto, relativo e médio quadrático é computada pela média aritmética simples e a quantidade de amostras analisadas na

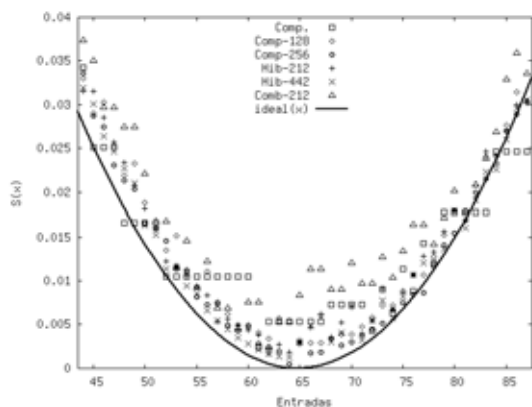
Figura 42 – Aproximação da função não linear pela RNA: (a) 45 pontos; (b) 10 pontos; (c) 45 pontos; (d) 10 pontos; (e) 45 pontos; (f) 10 pontos.



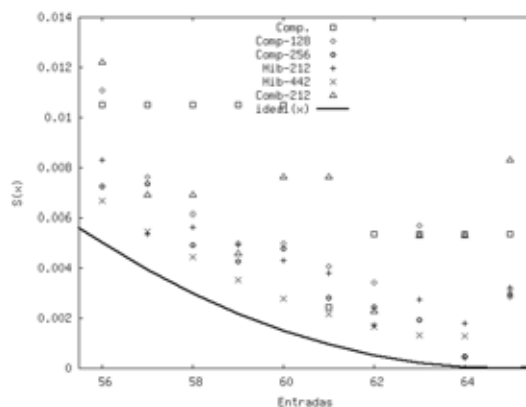
(a)



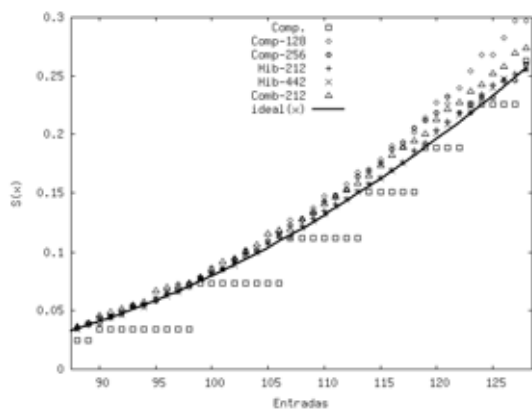
(b)



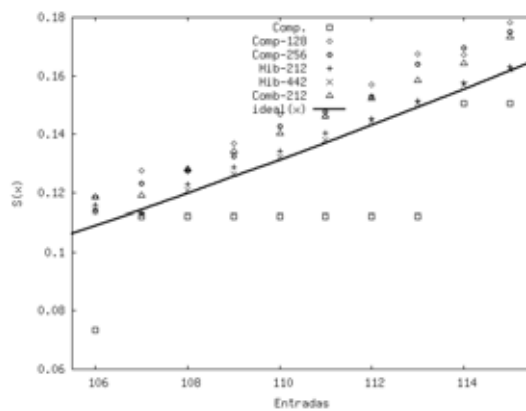
(c)



(d)



(e)



(f)

Fonte: Elaboração do próprio autor.

entrada da RNA (128) ou da AF (501).

$$E_{abs} = |f(x_k) - \hat{f}(x_k)| \quad (27)$$

$$E_{rel} = \frac{f(x_k) - \hat{f}(x_k)}{f(x_k)} \quad (28)$$

$$E_{quad} = \frac{(f(x_k) - \hat{f}(x_k))^2}{2} \quad (29)$$

O cálculo dos erros absoluto, relativo e médio quadrático foram truncados conforme a Equação (30), que corresponde ao erro máximo possível para determinada representação de dados (TOMMISKA, 2003). Assim, para diferentes valores da fração ( $f$ ), ou na quantidade de elementos do conjunto de dados disponíveis na LUT, das implementações respeitam a Equação (30) de truncamento. As exceções *Overflow* e *Underflow* também foram definidas na análise dos erros apresentados.

$$E_{truncMax} = 2^{-(f+1)} \quad (30)$$

Nas Tabelas 12 e 13 apresentam-se os erros máximos e médios, nos dois sistemas AF e RNA. Na Tabela 12 são definidos os resultados dos erros encontrados em cada cenário para o subsistema da AF. Os erros no sistema completo com todos os cenários são apresentados na Tabela 13. As abordagens contêm distintas precisões conforme descrito na seção 5.2. A comparação dos diversos tipos de erro, absoluto e relativo são feitas com literatura atual. Desta forma, pode-se observar a boa precisão na AF e na RNA do sistema com padronização IEEE 754.

**Tabela 12 – Erros na implementação do hardware para função de ativação.**

Cenários	Max_Abs	Med_Abs	Max_Rel	Med_Rel
Comportamental	$4,06 \times 10^{-3}$	$1,03 \times 10^{-3}$	$8,14 \times 10^{-2}$	$1,05 \times 10^{-3}$
Comp. PWL e RALUT - 128	$2,88 \times 10^{-2}$	$8,02 \times 10^{-4}$	$5,00 \times 10^{-1}$	$8,81 \times 10^{-4}$
Comp. PWL e RALUT - 256	$1,96 \times 10^{-3}$	$2,96 \times 10^{-4}$	$1,03 \times 10^{-2}$	$3,59 \times 10^{-4}$
PWL com RALUT - 212	$4,18 \times 10^{-3}$	$1,03 \times 10^{-3}$	$1,66 \times 10^{-2}$	$1,20 \times 10^{-3}$
PWL com RALUT - 442	$3,07 \times 10^{-3}$	$7,03 \times 10^{-4}$	$1,03 \times 10^{-2}$	$8,69 \times 10^{-4}$
PWL e Combinacional	$5,94 \times 10^{-3}$	$2,91 \times 10^{-3}$	$1,66 \times 10^{-2}$	$3,17 \times 10^{-3}$
Ref. (MEHER, 2010)	$2,00 \times 10^{-2}$	$1,72 \times 10^{-2}$	-	-
Ref. (BENJEMMAA et al., 2012)	-	$1,00 \times 10^{-2}$	-	-
Ref. (KHODJA; KHELDOUN; REFOUFI, 2010)	$5,10 \times 10^{-3}$	-	-	-
Ref. (NAMIN et al., 2009)	$1,89 \times 10^{-2}$	$1,21 \times 10^{-2}$	-	-
Alippi - Ref. (TISAN et al., 2009)	$1,89 \times 10^{-2}$	$1,11 \times 10^{-2}$	-	-
PLAN - Ref. (TISAN et al., 2009)	$1,89 \times 10^{-2}$	$6,30 \times 10^{-3}$	-	-

**Tabela 13 – Erros na implementação do hardware para o sistema da RNA.**

Cenários	Max_Abs	Med_Abs	Max_Rel	Med_Rel
Comportamental	$4,98 \times 10^{-2}$	$8,39 \times 10^{-3}$	$4,96 \times 10^{-1}$	$1,77 \times 10^{-1}$
Comp. PWL e RALUT - 128	$5,72 \times 10^{-2}$	$6,55 \times 10^{-3}$	$4,97 \times 10^{-1}$	$1,63 \times 10^{-1}$
Comp. PWL e RALUT - 256	$2,93 \times 10^{-2}$	$2,56 \times 10^{-3}$	$4,64 \times 10^{-1}$	$5,76 \times 10^{-2}$
PWL com RALUT - 212	$9,83 \times 10^{-3}$	$2,99 \times 10^{-3}$	$4,97 \times 10^{-1}$	$4,21 \times 10^{-2}$
PWL com RALUT - 442	$7,12 \times 10^{-3}$	$1,54 \times 10^{-3}$	$4,79 \times 10^{-1}$	$2,80 \times 10^{-2}$
PWL e Combinacional	$6,25 \times 10^{-2}$	$7,79 \times 10^{-3}$	$4,81 \times 10^{-2}$	$1,02 \times 10^{-1}$

A avaliação gráfica individual dos erros (absoluto, relativo e quadrático) em cada implementação da AF são demonstradas para todos os cenários do primeiro grupo. O Apêndice A contém as aproximações da AF e do sistema completo da RNA de forma individual com todos os erros, respectivamente em cada sistema.

### 5.3.1.3 Área Utilizada na FPGA

Nas Tabelas 14 e 15, apresentam-se os resultados finais da AF e RNA em relação a área utilizada. A Tabela 14 determina a área utilizada por cada arquitetura e plataforma implementada para a AF. Em nenhuma das implementações da AF utilizou-se multiplicadores ou somadores. Os resultados da AF foram comparados com a literatura, diferentemente da RNA, para os quais não foram encontrados resultados com a mesma aplicação. Os resultados de área do sistema completo da RNA para o primeiro grupo de cenários é apresentado pela Tabela 15.

**Tabela 14 – Utilização de área na FPGA para a função de ativação.**

Aproximações	Slices FF	Slices	LUT - 4 input	BRAM(Mem)
Comportamental	16(1%)	14(1%)	16(1%)	0(0%)
Comp. PWL e RALUT - 128	99(1%)	1029(22%)	1922(20%)	0(0%)
Comp. PWL e RALUT - 256	68(1%)	2095(44%)	3158(33%)	0(0%)
PWL com RALUT - 212	26(1%)	123(2%)	223(2%)	2(10%)
PWL com RALUT - 442	27(1%)	160(3%)	295(3%)	2(10%)
PWL e Combinacional	27(1%)	132(2%)	242(2%)	1(5%)
Alippi - Ref. (TISAN et al., 2009)	-	127	218	-
PLAN - Ref. (TISAN et al., 2009)	-	109	138	-
Ref. LUT (SAICHAND et al., 2008)	-	102	-	-
Ref. PWL (SAICHAND et al., 2008)	-	58	-	-
Ref. (HAVEL; VLCEK, 2008)	-	60	-	-

Os cenários 4 e 5 são as melhores opções relacionadas com a área e a precisão para AF Tangente Hiperbólica. Além disso, com valores abaixo da potência de 2 podem utilizar a BRAM para o armazenamento de outros dados do sistema, como os pesos, entradas e resultados parciais. Uma BRAM da plataforma Spartan 3E pode conter 512 elementos de 32 bits, ou seja, aproximadamente 16Kbits de memória. O cenário 5 não chega a ocupar uma BRAM e o cenário 4 não ocupa nem metade da BRAM, assim nos dois

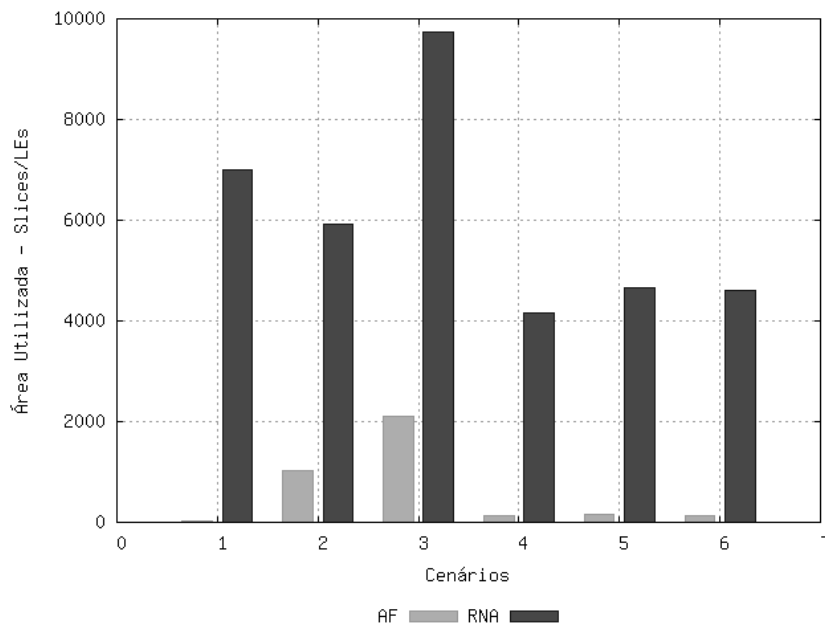
Tabela 15 – Utilização de área na FPGA para o sistema da RNA.

Cenários	Slices FF	Slices	LUT - 4 input	BRAM(Mem)	Multip.(18x18)
Comportamental	6.037(64%)	6.993(150%)	13.808(148%)	1(5%)	20(100%)
Comp. PWL e LUT - 128	5.921(63%)	5.933(127%)	11.672(125%)	1(5%)	20(100%)
Comp. PWL e LUT - 256	5.925(63%)	9.729(208%)	19.137(205%)	1(5%)	20(100%)
PWL com RALUT - 212	5.865(62%)	4.164(89%)	4.536(48%)	1(5%)	20(100%)
PWL com RALUT - 442	5.885(63%)	4.648(99%)	5.496(59%)	5(25%)	20(100%)
PWL e Combinacional	5.885(63%)	4.605(98%)	5.416(58%)	1(5%)	20(100%)

cenários sobram espaços para o armazenamento de outros dados. O cenário 6 é um método promissor para a implementação com uma quantidade maior de neurônios, por não necessitar de memória dedicada.

A comparação dos *Slices* contidos nas Tabelas 14 e 15, estão ilustrados no gráfico da Figura 43 para uma melhor análise da diferença entre a implementação independente da aproximação da AF e o sistema completo da RNA.

Figura 43 – Comparação da área utilizada (Slices) na FPGA com a implementação da AF e RNA.



Fonte: Elaboração do próprio autor.

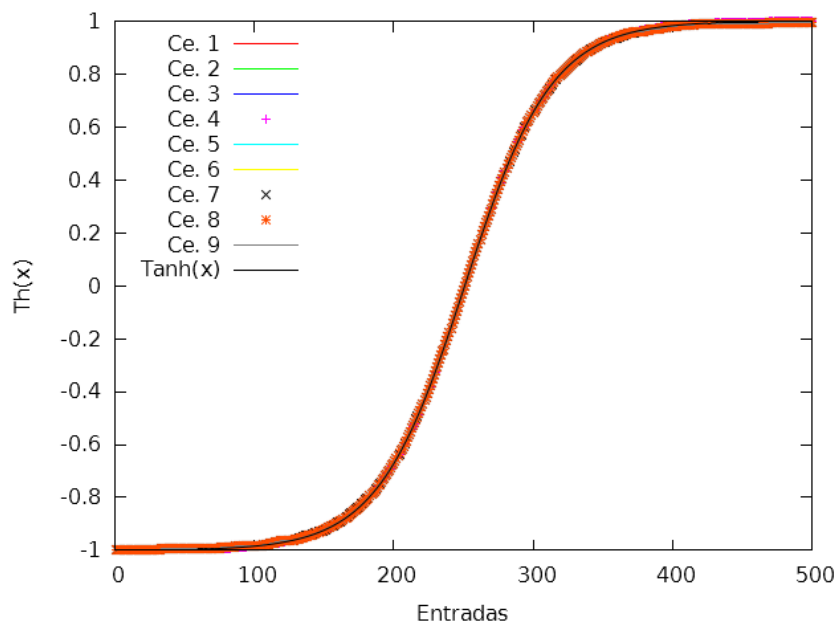
### 5.3.2 Cenários do Grupo 2

No segundo grupo foram desenvolvidos nove cenários em abstração de baixo nível. A arquitetura para computação dos neurônios foi a mesma com exceção para a AF. Assim, em todas as arquiteturas implementadas o tempo de execução é o mesmo, porque todas as AF gastam dois ciclos de *clock*.

### 5.3.2.1 Aproximação

A aproximação da AF Tangente Hiperbólica em todos os cenários seguiram o comportamento ideal como apresentado na Figura 44, definida na Equação (24). As aproximações feitas pela AF TH alcançam resultados satisfatórios, com os cenários 3, 4 e 5 atingindo os melhores resultados, os demais obtiveram resultados piores e semelhantes, mas ainda com boa precisão. Isto pode ser verificado na Figura 45, que contém escala reduzida para visualizar de forma adequada a precisão em todos os cenários.

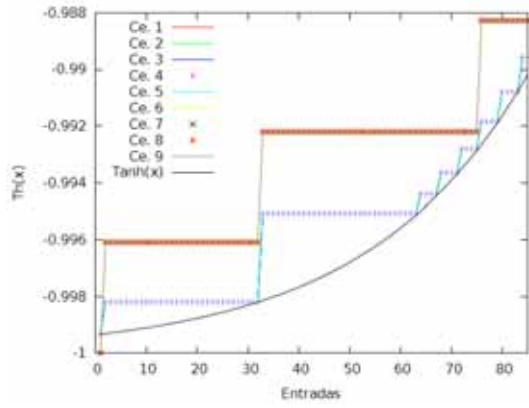
**Figura 44 – Aproximação da AF Tangente Hiperbólica com 500 pontos em todas as implementações.**



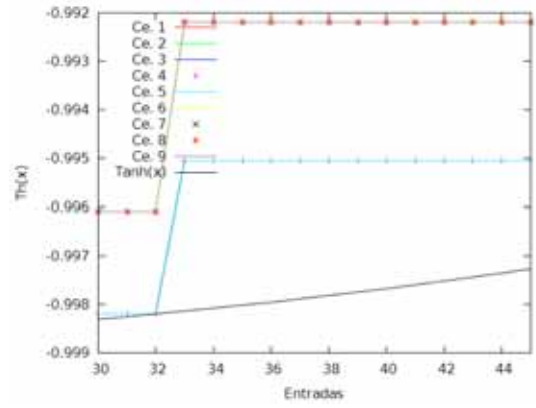
Fonte: Elaboração do próprio autor

A execução da RNA para a função não linear definida na Equação (24) também foi analisada para todos os cenários do segundo grupo. Os resultados das implementações no sistema da RNA apresentam um hardware adequado com ponto flutuante. A aproximação do sistema da RNA com 128 amostras e precisão em aproximadamente  $10^{-3}$  é apresentado na Figura 46. Os mesmos resultados são apresentados na Figura 47, mas com escala reduzida em somente 30 pontos e diferenciando as aproximações com linhas e pontos para facilitar a visualização. O cenário 1 foi o único com comportamento instável, todos os outros cenários seguiram a curva ideal com precisão adequada. A comparação entre a implementação na FPGA e com os valores ótimos ou ideais gerados pelo Matlab contém uma validação adequada, com baixa taxa de erro.

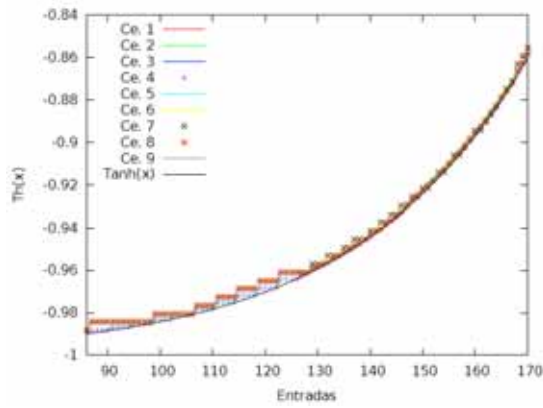
Figura 45 – Aproximação da AF Tangente Hiperbólica: (a) 85 pontos; (b) 15 pontos; (c) 85 pontos; (d) 15 pontos; (e) 80 pontos; (f) 15 pontos.



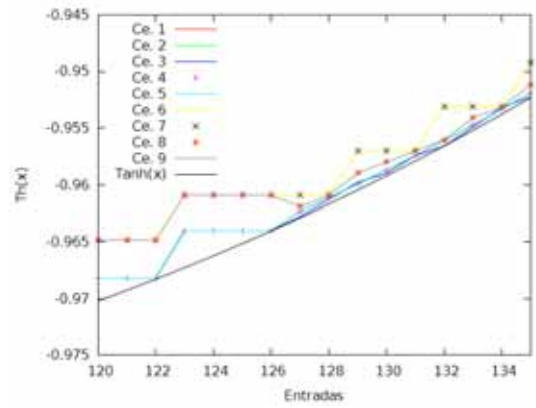
(a)



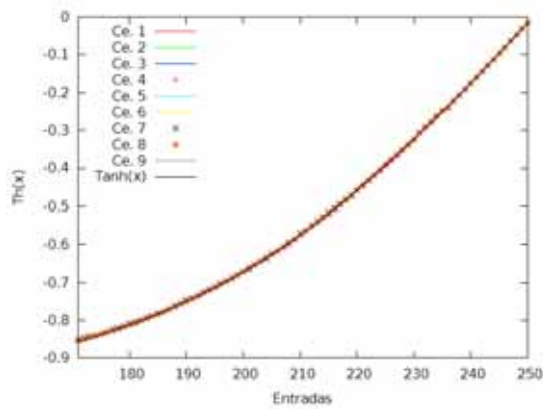
(b)



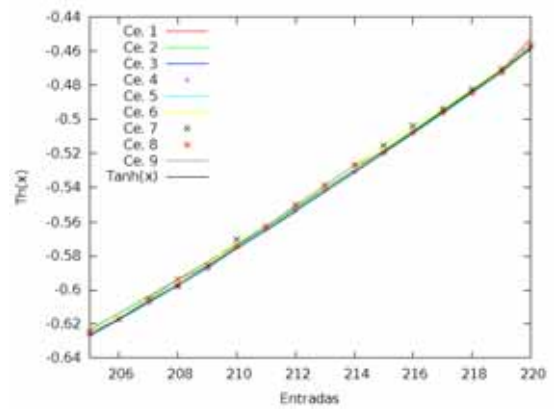
(c)



(d)



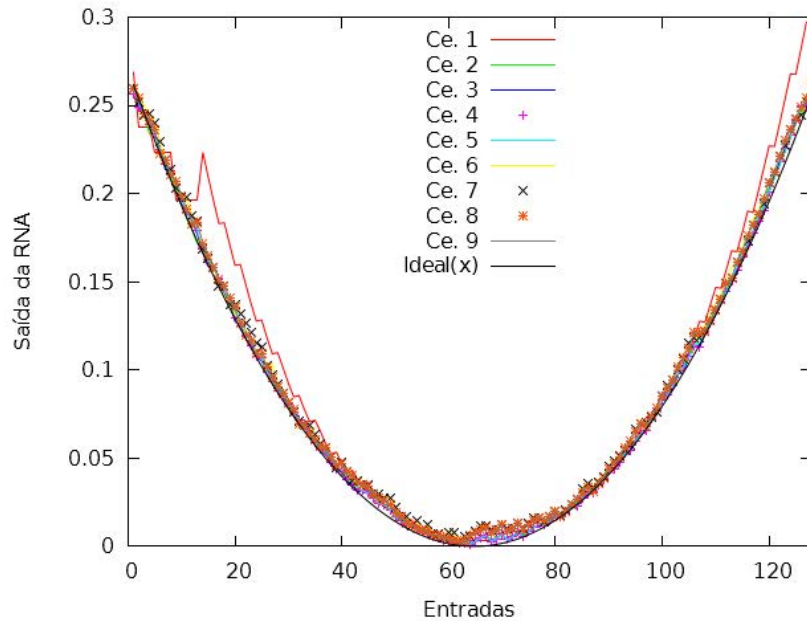
(e)



(f)

Fonte: Elaboração do próprio autor.

**Figura 46 – Aproximação da função não linear pela RNA em todas as implementações com 128 pontos.**



Fonte: Elaboração do próprio autor.

### 5.3.2.2 Análise de Erros

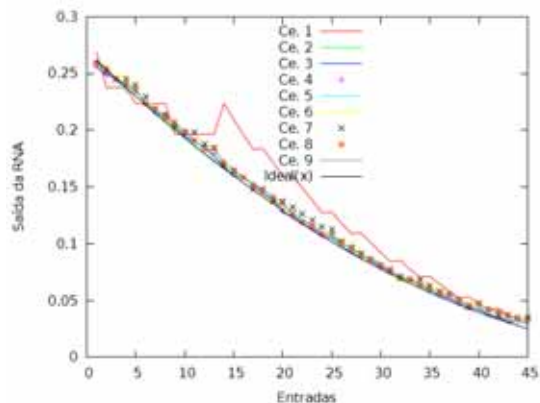
A análise de erros absoluto, relativo e quadrático foi realizada para o segundo grupo de cenários da mesma forma como apresentado na seção 5.3.1. O máximo erro relativo não foi relacionado por não ter resultados expressivos de comparação entre os cenários. No entanto, o erro médio relativo apresenta o comportamento das aproximações de forma adequada.

Nas Tabelas 16 e 17 apresentam-se análises dos resultados obtidos com as arquiteturas propostas para AF Tangente Hiperbólica e o sistema da RNA, respectivamente. Nessas tabelas são definidos os erros absolutos e relativos encontrados, juntamente com os seus valores médios e máximos associados em cada arquitetura do grupo 2 de cenários.

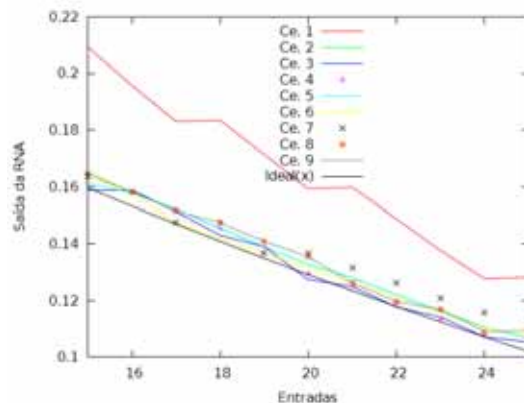
Os resultados dos erros e as melhores precisões são alcançadas pelos cenários HPR (1 a 5) (RALUT), quando comparados com os cenários HPC (6 a 9) (Combinational). Porém, os cenários HPR necessitam de memória dedicada (BRAM), diferentemente dos cenários HPC que têm apenas expressões booleanas para o mapeamento das saídas do sistema da AF.

Alguns cenários da literatura não contêm todas as características especificadas na Tabela 16, e estão representadas pelo caractere “-”. A maioria dos cenários tiveram precisão similar ou melhor que a encontrada na literatura pertinente. Na última linha da Tabela 16 apresenta-se a melhor precisão alcançada em Bajger e Omondi (2008), mas

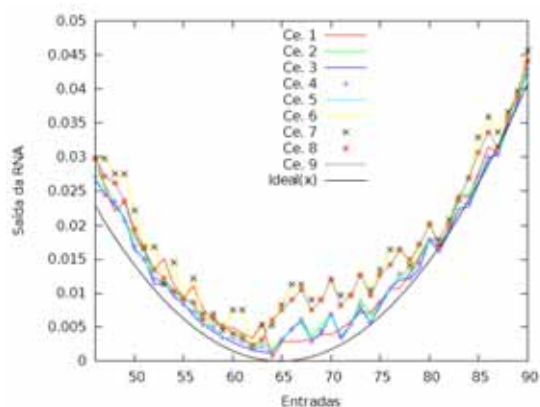
Figura 47 – Aproximação da função não linear pela RNA: (a) 45 pontos; (b) 10 pontos; (c) 45 pontos; (d) 10 pontos; (e) 40 pontos; (f) 10 pontos.



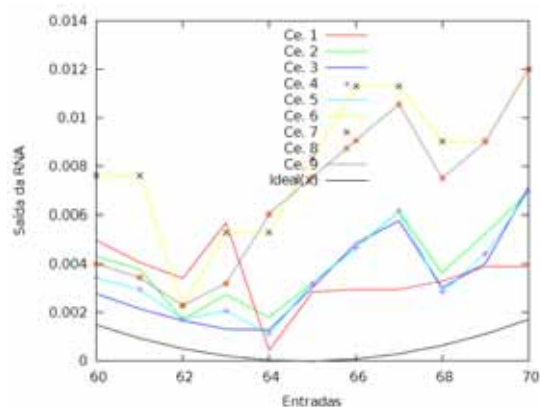
(a)



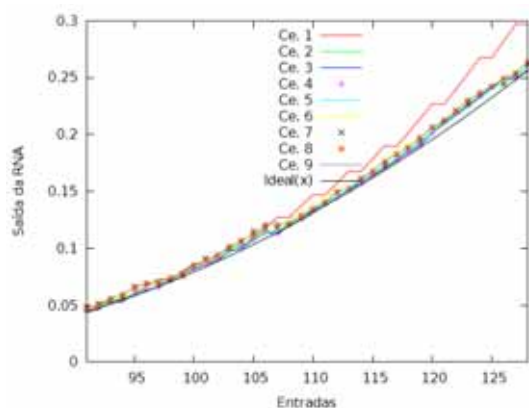
(b)



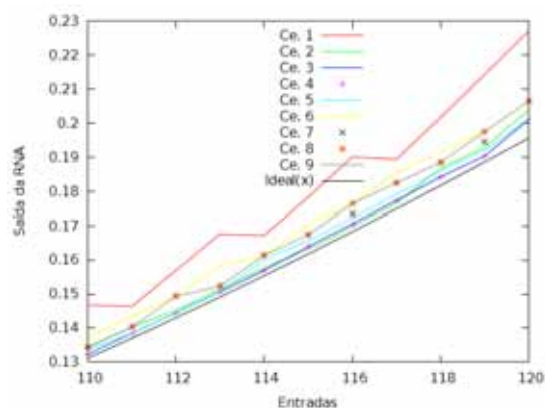
(c)



(d)



(e)



(f)

Fonte: Elaboração do próprio autor.

esta precisão é adquirida com 18 bits entre fração e expoente, a penúltima linha contém valores similares ao deste trabalho.

**Tabela 16 – Erros absolutos e relativos encontrados no hardware para a aproximação da AF.**

Aproximações	Erro_Máx	Erro_Méd	RelMéd
Ce. 1 (180)	$4,18 \times 10^{-3}$	$1,05 \times 10^{-3}$	$1,24 \times 10^{-3}$
Ce. 2 (212)	$4,18 \times 10^{-3}$	$1,03 \times 10^{-3}$	$1,20 \times 10^{-3}$
Ce. 3 (314)	$3,30 \times 10^{-3}$	$6,80 \times 10^{-4}$	$8,43 \times 10^{-4}$
Ce. 4 (340)	$3,30 \times 10^{-3}$	$7,18 \times 10^{-4}$	$8,99 \times 10^{-4}$
Ce. 5 (404)	$3,23 \times 10^{-3}$	$6,80 \times 10^{-4}$	$8,69 \times 10^{-4}$
Ce. 6 (180)	$5,94 \times 10^{-3}$	$2,75 \times 10^{-3}$	$3,04 \times 10^{-3}$
Ce. 7 (212)	$5,94 \times 10^{-3}$	$2,75 \times 10^{-3}$	$3,04 \times 10^{-3}$
Ce. 8 (340)	$5,94 \times 10^{-3}$	$2,36 \times 10^{-3}$	$2,72 \times 10^{-3}$
Ce. 9 (404)	$5,94 \times 10^{-3}$	$2,28 \times 10^{-3}$	$2,67 \times 10^{-3}$
POTB (HAVEŁ; VLCEK, 2008)	$4,30 \times 10^{-2}$	-	-
PLAN (TISAN et al., 2009)	$1,89 \times 10^{-2}$	$6,30 \times 10^{-3}$	-
Alippi (TISAN et al., 2009)	$1,89 \times 10^{-2}$	$1,11 \times 10^{-2}$	-
Combinational (TOMMISKA, 2003)	$3,90 \times 10^{-3}$	$1,70 \times 10^{-3}$	-
Combinational (TOMMISKA, 2003)	$7,70 \times 10^{-3}$	$4,00 \times 10^{-3}$	-
Hybrid (MEHER, 2010)	$2,00 \times 10^{-2}$	$1,72 \times 10^{-2}$	-
Hybrid (BAJGER; OMONDI, 2008)	$4,61 \times 10^{-4}$	$1,35 \times 10^{-4}$	$1,38 \times 10^{-3}$
Hybrid (BAJGER; OMONDI, 2008)	$2,67 \times 10^{-5}$	$9,11 \times 10^{-6}$	$5,66 \times 10^{-5}$

**Tabela 17 – Erros absolutos e relativos encontrados na implementação do hardware para o sistema da RNA.**

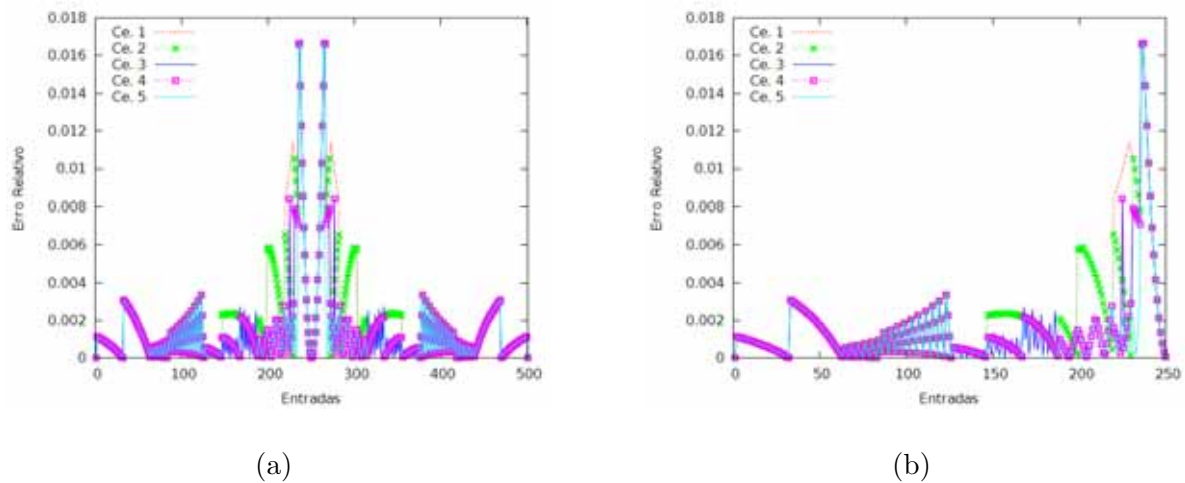
Cenários	Erro_Máx	Erro_Méd	RelMéd
1 (HPR - 180)	$6,25 \times 10^{-2}$	$4,75 \times 10^{-3}$	$2,86 \times 10^{-1}$
2 (HPR - 212)	$9,83 \times 10^{-3}$	$2,99 \times 10^{-3}$	$4,21 \times 10^{-2}$
3 (HPR - 314)	$9,17 \times 10^{-3}$	$1,75 \times 10^{-3}$	$3,00 \times 10^{-2}$
4 (HPR - 340)	$9,17 \times 10^{-3}$	$2,10 \times 10^{-3}$	$3,47 \times 10^{-2}$
5 (HPR - 404)	$9,17 \times 10^{-3}$	$2,69 \times 10^{-3}$	$3,39 \times 10^{-2}$
6 (HPC - 180)	$1,20 \times 10^{-2}$	$5,98 \times 10^{-3}$	$7,04 \times 10^{-2}$
7 (HPC - 212)	$1,20 \times 10^{-2}$	$5,84 \times 10^{-3}$	$7,34 \times 10^{-2}$
8 (HPC - 340)	$1,31 \times 10^{-2}$	$5,01 \times 10^{-3}$	$6,26 \times 10^{-2}$
9 (HPC - 404)	$1,31 \times 10^{-2}$	$5,01 \times 10^{-3}$	$6,26 \times 10^{-2}$

A diferença no comportamento das aproximações pode ser analisada pelo erro médio relativo de forma clara. Nas Figuras 48 e 49, apresentam-se uma análise completa de todos os cenários baseada no erro relativo, por esse tipo de parâmetro melhor demonstrar os dados analisados. Por meio destas figuras é possível verificar a implicação das alterações nas quantidades de elementos nos intervalos em cada aproximação ou cenário.

As análises detalhadas de cada cenário do segundo grupo, de forma independente, para cada tipo de erro (absoluto, relativo e quadrático) são apresentadas no Apêndice A. Com as informações desse apêndice pode-se apreciar de forma mais detalhada as diferenças entre cada cenário com a verificação das mudanças nos seus intervalos, conforme os intervalos da Tabela 11.

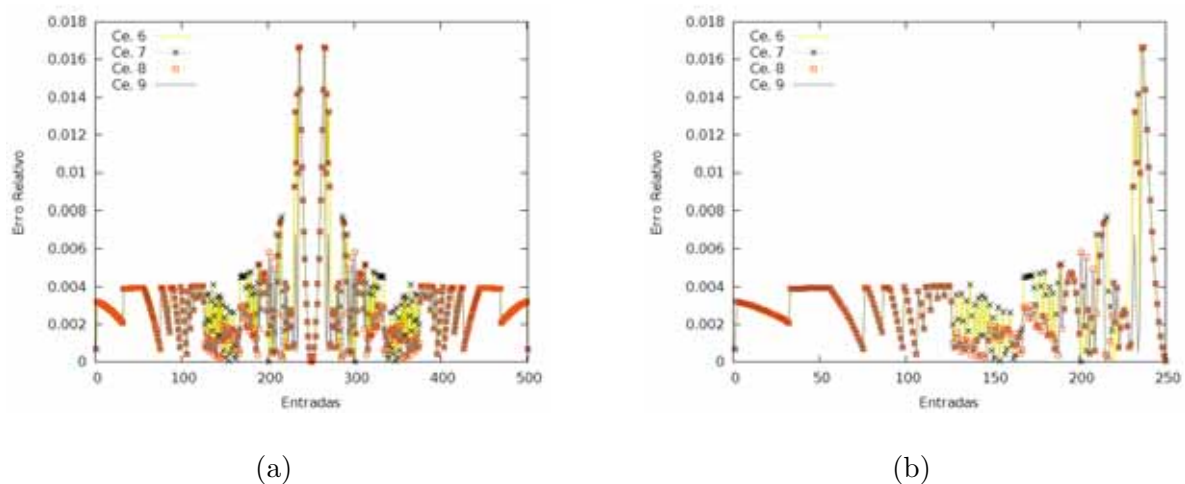
A análise do erro de forma geral foi dividida em dois grupos HPR e HPC com duas figuras para cada, uma com todas as amostras e outra com escala reduzida pela metade para diferenciar cada cenário. O volume de amostras entre 0,002 e 0,004 na Figura 49 é maior que na Figura 48, isso demonstra a precisão superior nos cenários HPR apresentados na Tabela 16.

**Figura 48 – Erro relativo dos cenários HPR para a AF: (a) 500 pontos; (b) 250 pontos.**



Fonte: Elaboração do próprio autor.

**Figura 49 – Erro relativo dos cenários HPC para a AF: (a) 500 pontos; (b) 250 pontos.**



Fonte: Elaboração do próprio autor.

Analisando as Figuras 48 e 49 e a Tabela 11 pode ser definida a igualdade entre os cenários de cada grupo respeitando as características discutidas na seção 5.1.3.2. As amostras são analisadas apenas na primeira metade, pela simetria da função de ativação.

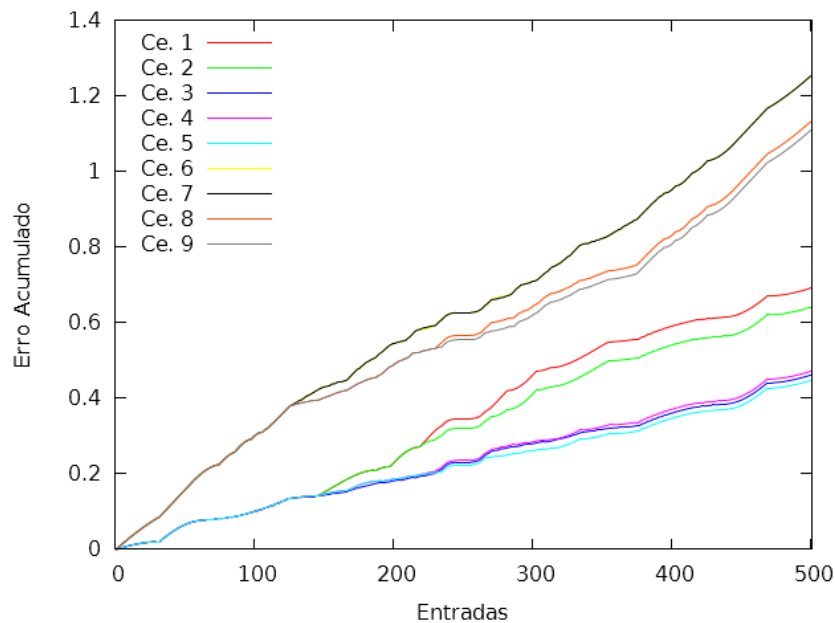
Os intervalos entre [0:125] e [235:250] têm a mesma resposta para todos os cenários de cada grupo, devido a igualdade entre a última e as três primeiras linhas da Tabela 11.

A relação entre o aumento na precisão e a quantidade de elementos no intervalo pode ser observada em vários momentos nas Figuras 48 e 49. No grupo HPR os cenários 1 e 2 têm a mesma resposta, exceto no intervalo entre as amostras 220 e 235 para o qual o cenário 2 obteve uma melhora pela maior quantidade de elementos. Esta situação ocorre de forma semelhante nos cenários 4 e 5, alterando as amostras no mesmo intervalo. Já, a diferença entre os cenários (1 e 2) e (4 e 5) é visível em todos os intervalos alterados. Nos grupo HPC, os cenários (6 e 7) e (8 e 9) podem ser comparados de forma similar à discussão feita para o grupo HPR nos cenários (1 e 2) e (4 e 5), respectivamente.

Na Figura 50 apresenta-se a diferença entre todos os cenários de forma evidente pelo erro acumulado. O cenário 5 alcança uma aproximação suavemente melhor em comparação aos cenários 3 e 4. As alterações de elementos do cenário 2 para 3 não tiveram efeito e atingiram os mesmos erros.

O cenário 3 e as amostras inseridas no sistema não são múltiplos de 2, isto provavelmente resultou na grande diferença com os demais. Com amostras intercaladas entre múltiplo de 2 e não múltiplo ou amostras apenas com múltiplo de 2 poderá servir de teste para futuras análises na interferência do tipo de amostras no resultado do sistema.

**Figura 50 – Erro acumulado para todos os cenários com 500 pontos.**

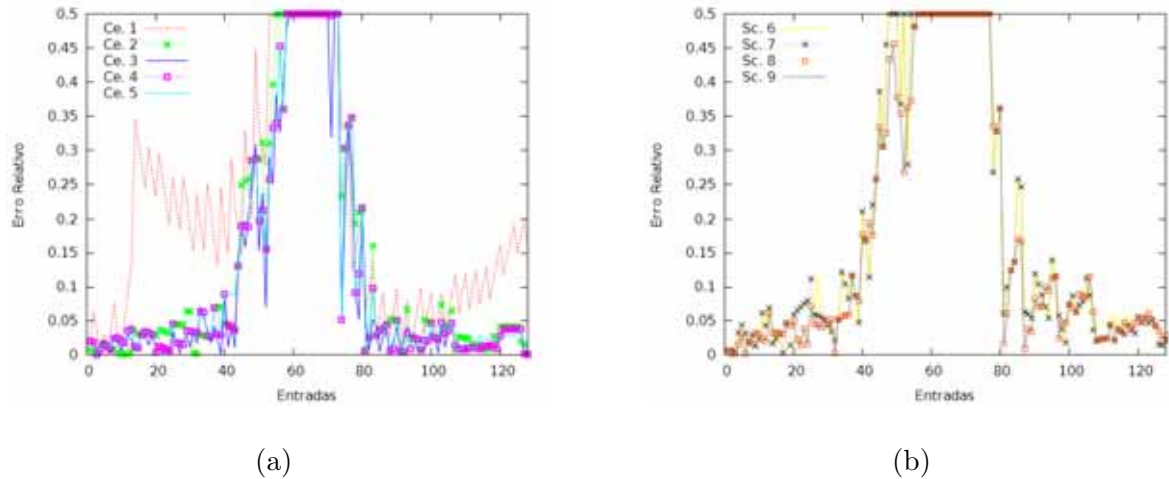


Fonte: Elaboração do próprio autor.

A análise do erro relativo para o sistema da RNA é apresentado na Figura 51. O primeiro cenário é o pior dentre todos na implementação do sistema da RNA, principalmente nas extremidades da função aproximada. Em todos os outros cenários

foram obtidos comportamentos similares, com os maiores erros com precisão de  $10^{-2}$ . Nas Figuras 51 e 52 ilustram-se a precisão superior dos cenários HPR conforme apresentado na Tabela 17.

**Figura 51 – Erro relativo com 128 pontos para o sistema da RNA: (a) Cenários HPR; (b) Cenários HPC.**



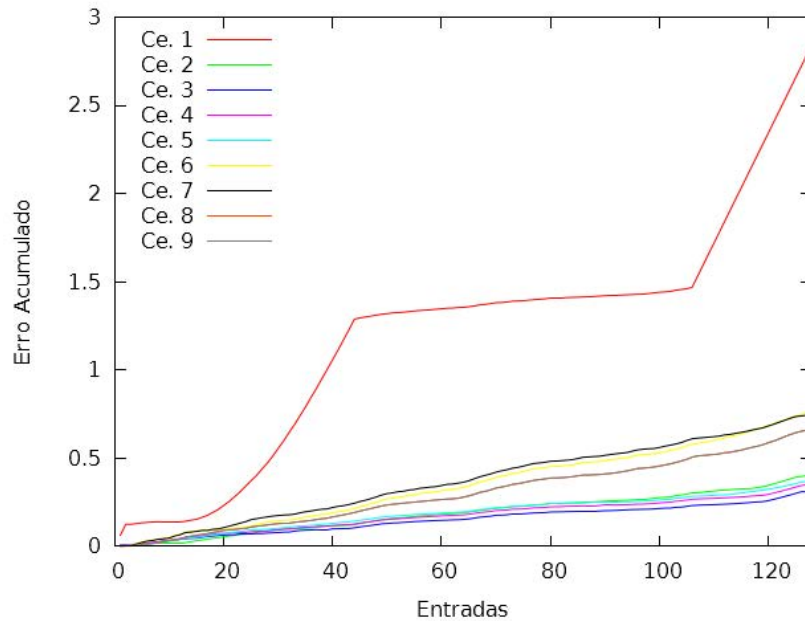
Fonte: Elaboração do próprio autor.

### 5.3.2.3 Área Utilizada na FPGA

Os resultados de área utilizada no dispositivo reconfigurável para o segundo grupo de cenários são apresentados nas Tabelas 18 e 19. Nos resultados dos cenários apresenta-se uma similaridade entre as áreas utilizadas. Os resultados no sistema da AF contêm valores abaixo de 15 *Slices*, apenas os cenários 6 e 7 com aproximadamente 135 *Slices*, semelhante a literatura recente (SANTOS et al., 2011; TISAN et al., 2009). No entanto, os mesmos cenários discrepantes obtiveram valores proporcionais aos demais no sistema completo da RNA. Na AF ocorreu uma alocação de recursos de maneira distinta pela ferramenta de síntese. O cenário 5 utilizou mais recursos da FPGA no sistema da RNA, Tabela 19, devido à quantidade de elementos para o mapeamento. Os resultados de síntese foram realizados pelo menos três vezes e são apresentados nas Tabelas 18 e 19.

A principal diferença do sistema da RNA, observada na Tabela 19, é a quantidade de BRAMs utilizadas pelos cenários. Em relação à memória utilizada, os cenários HPC e HPR, usam uma e quatro memórias dedicadas (BRAM), respectivamente, no sistema da RNA. Nos cenários HPR a BRAM serve apenas para as entradas do sistema. Em todos os cenários foram utilizados 20 (100%) dos multiplicadores dedicados disponíveis na plataforma Spartan-3e. Na Figura 53 compara-se a área utilizada para cada arquitetura do segundo grupo de cenários, na função de ativação e no sistema da rede neural artificial.

**Figura 52 – Erro acumulado para todos os cenários do sistema da RNA com 128 pontos.**



Fonte: Elaboração do próprio autor.

Na execução do Sistema da RNA os cenários 8 e 9 tiveram os mesmos resultados, analisando-se a relação entre área e precisão. O cenário 8 é o melhor escolha entre todos, por não utilizar BRAM, e apresenta reduzida utilização de área e precisão satisfatória. O cenário 2 é uma ótima opção por ter uma tabela pequena e boa precisão na execução dos dois sistemas AF e RNA.

#### 5.4 Conclusão deste Capítulo

Neste estudo apresentou-se a implementação de um sistema para aproximação de uma função utilizando RNA em linguagem de alto nível e dispositivo reconfigurável. A rede composta de 4 neurônios e função sigmoide em todos os neurônios na FPGA, para prover um melhor nível de paralelismo para o sistema. Nessa proposta pode-se observar um alto desempenho na RNA e com alto nível de precisão, com a utilização do padrão IEEE 754 de precisão simples (32 bits). Essa arquitetura mostra a possibilidade de implementar sistemas mais complexos com maior quantidade de neurônios, principalmente nas atuais famílias de FPGA, com chips de tecnologia de fabricação em 28 nm e uma vasta disponibilização de elementos lógicos.

Em todos os resultados nas três arquiteturas observa-se, sobretudo a possibilidade de aplicação de RNA em hardware com o uso do padrão IEEE 754. Com isso o sistema torna-

<sup>1</sup>Logic Elements (LE)

<sup>2</sup>Portas

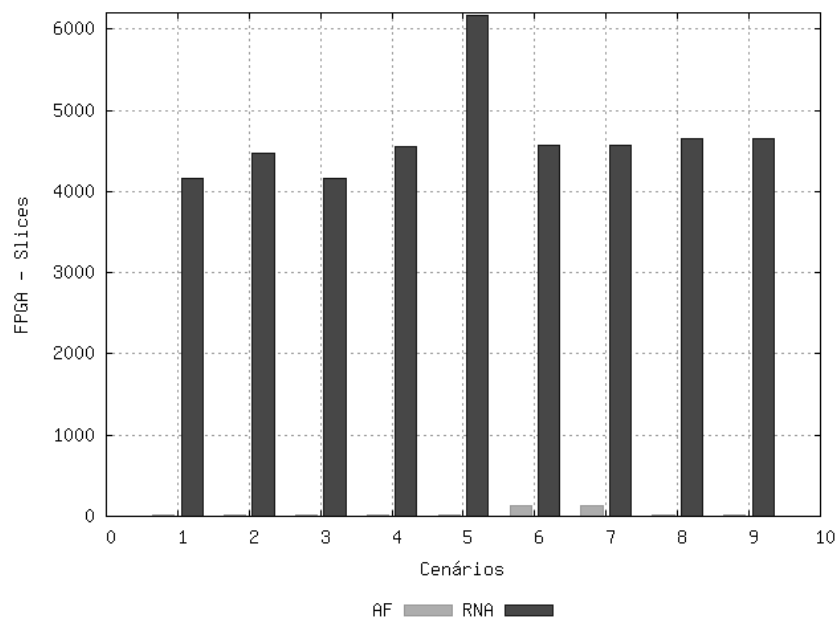
Tabela 18 – Área utilizada na FPGA para AF TH.

Aproximações	Slices	LUT-4in	BRAM	Intervalos
1 (HPR - 180)	12	16	1	[-4:4]
2 (HPR - 212)	12	16	1	[-4:4]
3 (HPR - 314)	12	17	1	[-4:4]
4 (HPR - 340)	12	17	1	[-4:4]
5 (HPR - 404)	12	17	1	[-4:4]
6 (HPC - 180)	134	244	0	[-4:4]
7 (HPC - 212)	133	243	0	[-4:4]
8 (HPC - 340)	12	17	0	[-4:4]
9 (HPC - 404)	12	17	0	[-4:4]
POTB (HAVEL; VLCEK, 2008)	60	-	0	[-8:8]
PLAN (TISAN et al., 2009)	109	138	0	[-5:5]
Alippi (TISAN et al., 2009)	67	110	0	[-5:5]
Combinacional (TOMMISKA, 2003)	45 <sup>1</sup>	-	0	[-8:8]
Combinacional (TOMMISKA, 2003)	25 <sup>1</sup>	-	0	[-4:4]
Híbrido (MEHER, 2010)	181 <sup>2</sup>	-	0	[-3:3]
PWL (SAICHAND et al., 2008)	58	108	0	[-8:8]

Tabela 19 – Utilização de área na FPGA para o sistema da RNA

Arquitetura	Slices FF	Slices	LUT - 4 input	BRAM(Mem)	Multip.(18x18)
Sc. 1 (180)	6.044(64%)	3.955(84%)	4.956(53%)	5(25%)	20(100%)
Sc. 2 (212)	5.892(63%)	4.476(96%)	5.143(55%)	5(25%)	20(100%)
Sc. 3 (314)	5.863(62%)	4.156(89%)	4.520(48%)	5(25%)	20(100%)
Sc. 4 (340)	5.883(63%)	4.556(97%)	5.312(57%)	4(20%)	20(100%)
Sc. 5 (404)	6.005(64%)	6.166(132%)	9.532(102%)	4(20%)	20(100%)
Sc. 6 (180)	5.892(63%)	4.572(98%)	5.347(57%)	1(5%)	20(100%)
Sc. 7 (212)	5.863(62%)	4.567(98%)	5.340(57%)	1(5%)	20(100%)
Sc. 8 (340)	5.872(63%)	4.654(99%)	5.640(60%)	1(5%)	20(100%)
Sc. 9 (404)	5.863(62%)	4.654(99%)	5.624(60%)	1(5%)	20(100%)

Figura 53 – Utilização de área na FPGA na AF e sistema da RNA.



Fonte: Elaboração do próprio autor.

se mais robusto, facilita sua manutenção e o reuso de módulos ou da própria aplicação. Os resultados apresentam a alta precisão alcançada na utilização de ponto flutuante (precisão simples) para a computação dos dados do neurônio. A comparação entre a implementação na FPGA com a linguagem de alto nível contém grande semelhança nos resultados e na resposta da função aproximada.

O emprego de FPGA para aplicações com RNA é adequado pelo nível de paralelismo previsto nas abordagens, o tempo de execução alcançado pela aplicação foi de aproximadamente 399 microssegundos (us) em 128 entradas. Porém, cálculos mais complexos junto ao uso de ponto flutuante é um desafio ao projetista no desenvolvimento de um sistema digital ou SoC, pelas inúmeras possibilidades na construção de hardware. O projetista deve analisar as possibilidades de inserção ou a melhor escolha de componentes para determinada situação, além da melhor forma de desenvolvimento para unir duas características essenciais ao projeto do sistema, poupar área consumida da FPGA e não perder desempenho.

O sistema foi desenvolvido para proporcionar a portabilidade a outras plataformas de hardware, e facilitar a adaptação de novas variações relacionadas com a quantidade de neurônios e de camadas na RNA. No Capítulo 6 apresentam-se as comparações entre os métodos de segmentação, para identificar a deficiência de nutrientes pelas folhas da cultura do Mamão e Caferana.

## 6 MÉTODOS PARA SEGMENTAÇÃO DE IMAGENS

Neste capítulo analisam-se três métodos distintos para a segmentação da imagem, o uso de diferentes tipos de limiarização, por histograma com limiar global único e com limiares múltiplos, por meio do método de Otsu, e as redes neurais artificiais. Estes métodos serão utilizados para fazer a segmentação de imagens para identificar a deficiência de nutrientes com base na coloração das folhas, ou seja, o teor de clorofila contido na folha do Mamão e Caferana.

Na segmentação de imagens, diversas características ou partes de uma imagem podem ser selecionados. Essas características podem estar relacionadas a descontinuidade ou similaridade na imagem. A descontinuidade é a diferença abrupta de níveis de cinza em determinadas regiões da imagem, definidas por meio de detecção de pontos, segmentos de retas ou bordas. A similaridade está relacionada a descoberta e o agrupamento de uma região que mantém uma correlação de características em um conjunto de píxeis. Neste trabalho será analisada a segmentação com relação a similaridade na folha, conforme a presença de indicadores que determinam a deficiência de nutriente em uma cultura.

Para verificar a possibilidade de implementação do processamento de imagem em FPGAs, realizou-se esse estudo, no qual os métodos foram desenvolvidos em linguagem de alto nível (Matlab) e as imagens adquiridas por meio de um celular com câmera de 3 MegaPíxel. As câmeras digitais de baixo custo atuais contêm baixa qualidade na captura de imagem, semelhante à câmera escolhida, a resolução foi reduzida para o tamanho de 320x240. As imagens foram adquiridas de diferentes tipos de culturas e iluminação ambiente, para averiguar possíveis dificuldades ao longo da pesquisa.

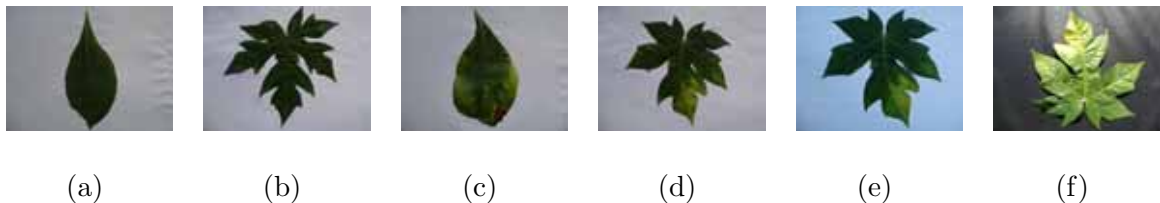
### 6.1 Limiarização Global por Histograma

A limiarização global por histograma foi realizada com a função de segmentação da imagem baseada no gráfico do histograma e na escolha de um único limiar ( $T$ ).

Inicialmente, as imagens adquiridas são analisadas com base em seus histogramas e transformadas em escala de cinza. As imagens originais são apresentadas na Figura 54, em sequência, sendo duas imagens sem deficiência e quatro com possível deficiência em

diferentes culturas. Na Figura 54, a sequência das imagens são das culturas da Caferana, Mamão, Caferana, e três do Mamão, sendo as duas últimas com diferenças de iluminação e plano de fundo. Nos testes iniciais, pode-se observar que há a possibilidade de fazer a segmentação da imagem pela falta de nutrientes na folha. Principalmente, nos casos em que a imagem tem fundo e objeto bem definidos. A imagem escolhida de teste, Figura 54-(e), contém uma possível deficiência de nutriente de nitrogênio na planta, pela diferença de coloração amarelada encontrada (OLIVEIRA et al., 2004).

**Figura 54 – Imagens originais do Mamão e Caferana: (a) Imagem (Caferana); (b) Imagem (Mamão); (c) Imagem (Caferana c/ deficiência); (d) Imagem (Mamão c/ deficiência); (e) Imagem (Mamão c/ distinta iluminação); (f) Imagem (Mamão c/ alta iluminação e fundo escuro).**



Fonte: Elaboração do próprio autor

Como apresentado na seção 2.4, a limiarização por histograma possibilita a seleção de objetos utilizando um limiar global e obtém a separação entre objeto e fundo. Neste caso, não é apenas separar o objeto e o fundo, mas selecionar uma parte do objeto que contenha coloração ou contraste diferenciado, que identifica a falta de nutriente pela folha da planta. Essa região de contraste diferenciado deverá conter o mesmo nível de intensidade e verificasse que a sua localização ficará próximo ao vale do histograma. Neste experimento os dados da região foram identificados de forma empírica. Porém, existem alguns métodos que podem ser utilizados para uma melhor definição do limiar e conseqüentemente da região a ser segmentada. Um método que encontra um limiar ótimo e tem boa aceitação entre os pesquisadores, é o método de Otsu (OTSU, 1979), que está apresentado na seção 6.2.

As transformações  $(g(x, y))$  são apresentadas nas Figuras 55-(c;f), 56-(c;f) e 57-(c;f). A transformação consiste na definição de dois limiares de forma empírica para dividir a imagem em fundo, objeto1 (folha sem deficiência) e objeto2 (folha com deficiência). De forma semelhante a transformação realizada na seção 2.4.3, porém com um limiar a mais. A Equação (31) determina a separabilidade entre a cena e os objetos da imagem. A transformação converte a imagem em escala de cinza em uma imagem binária, branca (255) e preta (0).

$$g(x, y) = \begin{cases} 0 & \text{se } T1 \leq f(x, y) \leq T2 \\ 255 & \text{se } T1 > f(x, y) \\ 255 & \text{se } T2 < f(x, y) \end{cases} \quad (31)$$

Nos dois tipos de culturas, Mamão e a Caferana, contêm imagens com aparência normal, ou seja, sem deficiência de nutrientes na planta, e imagens com deficiência na folha da cultura. A Figura 55-(a) corresponde a uma imagem em escala de cinza do Mamão, a Figura 55-(c) ao histograma da imagem e a Figura 55-(e) à transformação da imagem em binário. A Figura 55-(b;d;f) segue a mesma sequência, para a cultura da Caferana. As Figuras 55-(e;f) correspondem as imagens após a aplicação da técnica de limiarização por histograma, com limiar global. A segmentação ocorreu com limiares (T1 e T2) de intensidade entre 90 e 120, conforme Equação (31) para a Figura 56. No entanto, a Figura 55 não contém nenhuma intensidade de clorofila diferenciada na folha. Nas Figuras 55-(e;f) não ocorre a segmentação com os mesmos limiares, apenas pontilhados na borda da imagem.

Os histogramas representados pelas Figuras 55-(c;d) mostram a distinção entre o fundo e o objeto (folha), e o vale entre os lóbulos do histograma correspondem a separação do fundo e objeto. Um dos lóbulos do histograma contém uma largura maior, ou leve diferença de tons de cinza em determinadas regiões da imagem, ou as cores mais claras da folha. Essa análise pode ser feita com uma busca em todos os pontos da imagem selecionando apenas os pontos com níveis diferenciados, que correspondem à deficiência nutricional da foliar do Mamão. Após encontrados os pontos da imagem, a transformação desses pontos em preto (deficiência) e branco (resto da imagem), ou seja, a transformação da imagem em escala de cinza em imagem binária, pode-se determinar a segmentação da deficiência nutricional da folha.

As Figuras 56-(a;b) contêm intensidade de clorofila diferenciado próximos a borda da folha ou vestígios de falta de nutrientes, isso geralmente acontece com determinadas deficiências de nutrientes, por exemplo, o Potássio (K) ou nitrogênio (N), maiores detalhes são abordados nos Capítulos 7 e 8. Os histogramas estão ilustrados nas Figuras 56-(c;d), respectivamente para o Mamão e Caferana, pode-se observar que os histogramas são semelhantes aos das imagens anteriores. Nas Figuras 56-(e;f), respectivamente Mamão e Caferana, contém a imagem segmentada conforme o nível de intensidade de cor, apenas a deficiência de nutriente é transformada em cor preta, conforme Equação (31).

Algumas observações importantes em relação à Figura 56-(b) podem ser evidenciadas em sua segmentação na Figura 56-(f). Nesse caso, existem dois lóbulos com perfeitas distinções no histograma da Figura 56-(d), mas existe a necessidade de fazer a separação

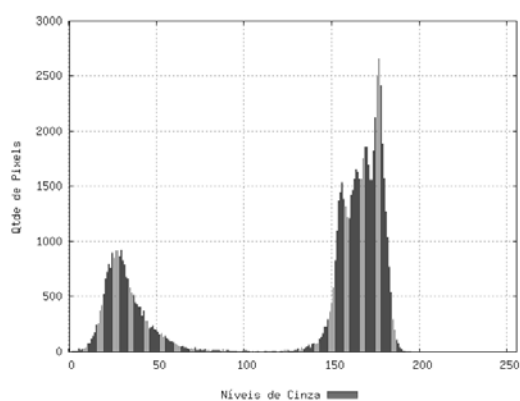
Figura 55 – Limiarização de imagens das folhas com aparência normal: (a) Imagem original (Mamão); (b) Imagem original (Caferana); (c) Histograma Mamão; (d) Histograma Caferana; (e) Segmentação Mamão; (f) Segmentação Caferana.



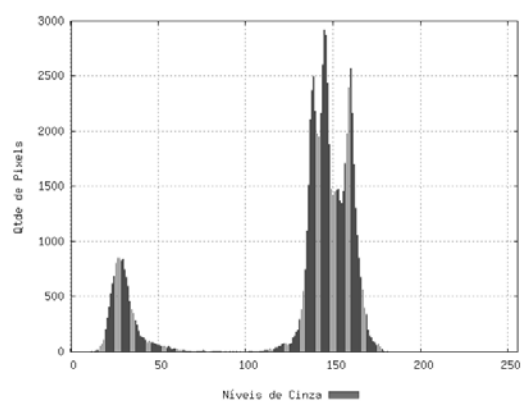
(a)



(b)



(c)



(d)



(e)



(f)

Fonte: Elaboração do próprio autor

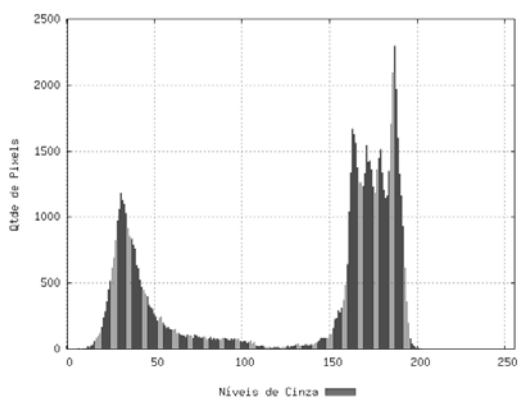
Figura 56 – Limiarização de imagens das folhas com deficiência de nutrientes: (a) Imagem original (Mamão); (b) Imagem original (Caferana); (c) Histograma Mamão; (d) Histograma Caferana; (e) Segmentação Mamão; (f) Segmentação Caferana.



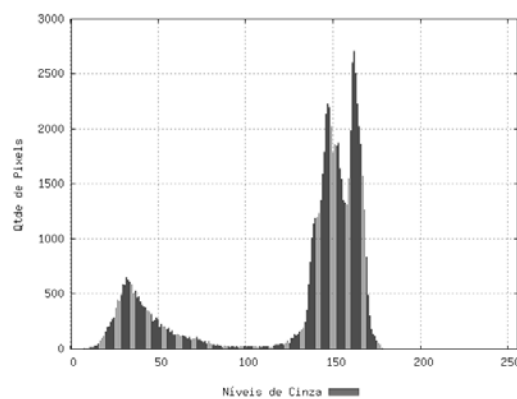
(a)



(b)



(c)



(d)



(e)



(f)

Fonte: Elaboração do próprio autor

de três partes (fundo e dois objetos). Assim, um dos lóbulos contém dois objetos a serem identificados. Essa é a dificuldade de obtenção de limiares empíricos, constatando a sua ineficiência para determinadas condições. Na Figura 56-(f) o método falha nas seguintes condições:

- Diferentes níveis de intensidade (amarelo) do lado esquerdo da imagem não foram segmentados;
- Na parte inferior da imagem existe uma necrose (parte escura) na folha que não foi segmentada;
- O canto esquerdo da imagem foi segmentado e não era para ter ocorrido.

A definição de um limiar ou um intervalo de intensidade de cinza são características essenciais na identificação e segmentação da deficiência de nutrientes nas folhas. Porém, a utilização deste tipo de método para encontrar um limiar ótimo é ineficiente, necessita-se de uma método mais robusto para tal finalidade. Nas imagens apresentadas nas Figuras 56 e 57 também ocorreram problemas na segmentação.

A interferência de iluminação pode ocasionar alguns problemas no processamento de imagens com histogramas. Nas Figuras 57-(a;b) apresentam-se duas imagens de folhas de Mamão com luminância e fundos de cores diferenciadas nas cenas. Na Figura 57-(a) a diferença de iluminância é baixa e mesmo com o vale bem definido entre objeto e fundo no histograma da Figura 57-(c), a segmentação ocorre com uma região menor na Figura 57-(e), quando comparado a Figura 56-(d). Porém, a Figura 57-(b) contém alta iluminância incidente no objeto, ou seja, parte da imagem contém iluminação solar, e isso traz um distúrbio no histograma da imagem, Figura 57-(d). Neste histograma a região do vale, e a separação entre o objeto e o fundo ficam unidas e impossíveis de serem distinguidas. Assim, com o uso de limiares empíricos nesse tipo de histograma é totalmente impossível de identificar a deficiência de nutriente, a partir do histograma, e prejudica a segmentação, conforme apresentado na Figura 57-(f).

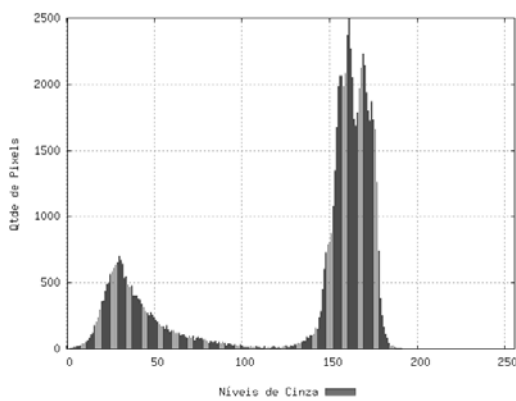
Figura 57 – Limiarização de imagens das folhas de Mamão com deficiência de nutrientes e iluminação distinta: (a) Baixa diferença na iluminação; (b) Alta diferença na iluminação; (c) Histograma (baixa); (d) Histograma (alta); (e) Segmentação (baixa); (f) Segmentação (alta).



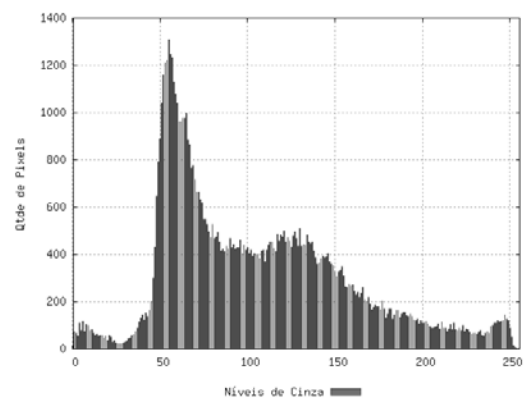
(a)



(b)



(c)



(d)



(e)



(f)

Fonte: Elaboração do próprio autor

## 6.2 Limiarização Múltipla pelo Método de Otsu

Para análise do método de Otsu utilizou-se as mesmas imagens das folhas do Mamão e da Caferana aplicados na seção 6.1, para efeito de comparação entre os métodos. Além do método de Otsu, aplicou-se operador morfológico de erosão para encontrar as sementes da imagem e um ponto de partida na identificação de nutrientes. Esse operador possibilita retirar partes ruidosas da imagem extraindo apenas características essenciais para posterior utilização com RNA.

O algoritmo que define o método de Otsu com operador morfológico de erosão é definido em sua essência pelo Algoritmo 2. Os dados de entrada do algoritmo é a imagem original  $(f(x, y))$ , feito na linha 2, e o método de Otsu é aplicado nas linhas 2 até 26. O operador de erosão é executado no final do Algoritmo 2, na linha 28, para definir melhor a semente da imagem retirando, por exemplo, pontos isolados. A descrição do método de Otsu e suas equações foram descritas na seção 2.4.4.

Nesta seção, todas as figuras estão dispostas em três etapas: original, segmentação pelo método de Otsu (SMO) e segmentação pelo método de Otsu em conjunto com operador de erosão (SMOOE). As imagens originais são apresentadas em níveis de cinza e a segmentação SMO é alcançada apenas por meio do Algoritmo 2 sem a linha 28.

A segmentação SMOOE consiste em incluir o operador de erosão linha 28 do Algoritmo 2. Este operador corresponde a  $A \ominus B$ , assim A e B são conjuntos de  $Z^2$ , a erosão é definida pela Equação (32), conforme (GONZALEZ; WOODS, 2010).

$$A \ominus B = \{z \mid (B)_z \subseteq A\} \quad (32)$$

O primeiro experimento é feito com folhas do Mamão e da Caferana com aparência normal, sem deficiência ou coloração evidente, conforme apresentado na Figura 58-(a;d). Nesse experimento nota-se que a principal diferença, em relação à Figura 55, está nas bordas das duas folhas que a limiarização de Otsu conseguiu capturar mesmo com baixa variação de tom de amarelo.

No caso das Figuras 55-(e;f), na limiarização global, apareceram apenas as bordas das folhas com baixa intensidade. Na Figura 55-(f) apareceu um ruído indesejado no canto inferior esquerdo. Já, no método de Otsu, nas Figuras 58-(b;c;e;f) apareceram detalhes de pigmentação referente à folha, imperceptíveis ao olho humano, permitindo uma segmentação com maior precisão e sem o ruído no canto inferior esquerdo.

Nas Figuras 59-(b;e) e 60-(b;e) apresentaram-se os resultados alcançados apenas com o método de Otsu. Já nas Figuras 59-(c;f) e 60-(c;f) os resultados apresentados com a

---

**Algoritmo 2:** Transformação ( $G'$ ) da imagem após duas segmentações ( $S$  e  $S'$ ) e uma reconstrução ( $R$ )

---

**Entrada:**  $f(x, y)$  - Imagem Original

**Saída:**  $G'(x, y) = R(S')$  com  $S'(xi, yi) \subseteq f(x, y)$  onde  $S'(xi, yi)$  é a última classe.

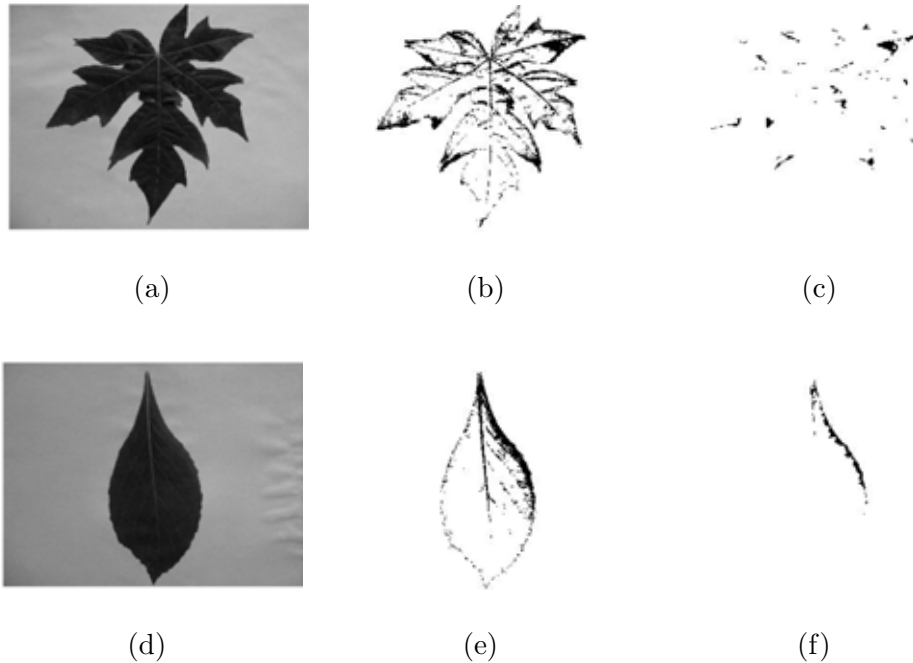
```

1 início
2   Inicialização e Preparação de f(x,y);
3   para  $i \leftarrow Vlr_{Inicial} \wedge i < Qtde_{Limiares}$  faça
4     para  $j \leftarrow Vlr_{Inicial} \wedge j < (L - 1)$  faça
5       Construir Histograma ( $H(0 - (L - 1))$ ) de  $f(x, y)$ ;
6     fim
7     para  $j \leftarrow Vlr_{Inicial} \wedge j < (L - 1)$  faça
8       Calcular Probabilidade de Ocorrência ( $PO_k$ ) - Eq. (14);
9       Calcular Soma Acumulada da Imagem ( $S_a$ ) - Eq. (15);
10      Calcular Intensidade Média Global da imagem ( $M_g$ ) - Eq. (16);
11      Calcular Intensidade Acumulada ( $I_k$ ) - Eq. (17);
12      Calcular Soma Cumulativa das Probabilidades da Classe 1 ( $P1_k$ ) e
13      Classe 2 ( $P2_k$ ) - Eq. (18) e (19);
14    fim
15    para  $j \leftarrow Vlr_{Inicial} \wedge j < (L - 1)$  faça
16      Calcular Intensidade Média Acumulada (m1k e m2k) Calcular
17      Variância Global ( $\sigma_g$ ) - Eq. (20);
18      Calcular Variância entre as Classes ( $\sigma_c$ ) - Eq. (21);
19      Encontrar a Variância Máxima de Intensidade ( $Max(\sigma_c)$ );
20    fim
21    para  $x \geq Vlr_{Inicial} \wedge x < N$  faça
22      para  $y \leftarrow Vlr_{Inicial} \wedge y < M$  faça
23        Segmentar Imagem com  $Max(\sigma_c)$  Encontrado ( $S(f(x, y))$ );
24        Definir Novos Valores para N e M (x1,y1);
25        Eliminar uma Classe da Imagem ( $S'(x1, y1)$ );
26      fim
27    fim
28  Reconstruir Imagem com a Última Classe e Ordem inicial (NxM)
29  ( $R(S'(x1, y1))$ );
30  Aplicar Método de Erosão para Encontrar Semente da Imagem ( $G'(x, y)$ ) - Eq.
31  (32);
32  Salvar e Apresentar Resultados;
33 fim

```

---

**Figura 58 – Limiarização múltipla de imagens das folhas com aparência normal: (a) Original (Mamão); (b) SMO (Mamão); (c) SMOOE (Mamão); (d) Original (Caferana); (e) SMO (Caferana); (f) SMOOE (Caferana).**

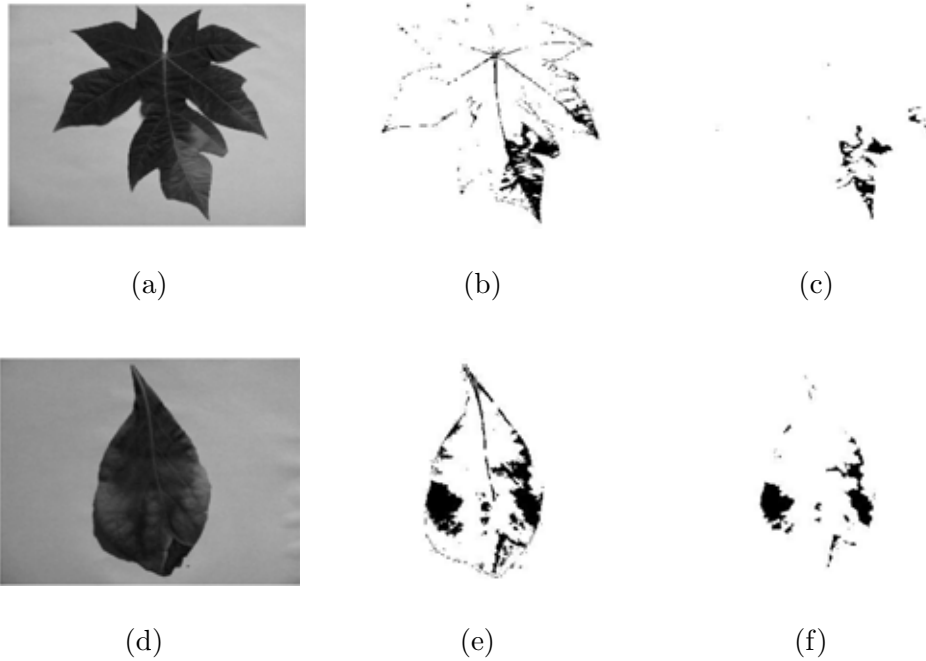


Fonte: Elaboração do próprio autor

utilização do método de Otsu em conjunto com o operador morfológico de erosão nas imagens com deficiência de nutrientes. A Figura 60-(d;e;f) ilustra que um fundo escuro junto a uma iluminação não uniforme na imagem prejudica seriamente a segmentação e a utilização de limiarização por níveis de intensidade.

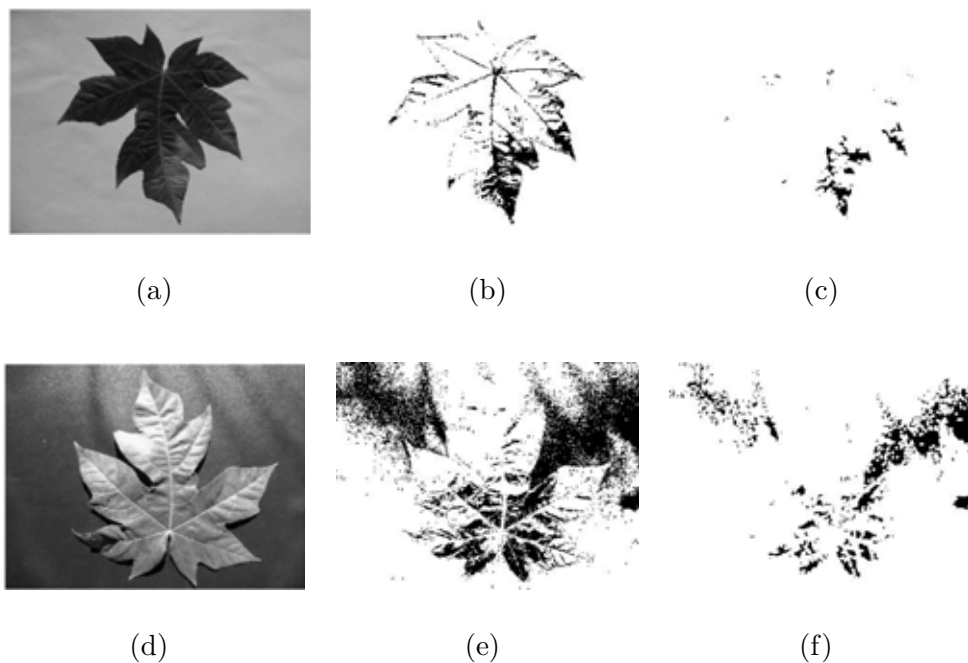
Uma importante observação estão nas Figuras 59-(a) e 60-(a), ou nas imagens coloridas da Figura 54. Mesmo com diferenças de luminosidade, mas com o mesmo fundo claro na imagem a primeira segmentação com o método de Otsu, Figuras 59-(b) e 60-(b), apresentaram segmentação adequada com a região específica com a diferença de coloração. Nessas figuras observaram-se que as bordas da folha com maior número de detalhes foi na Figura 60-(b). Já, nas segmentações de deficiência de nutrientes pelas Figuras 59-(c) e 60-(c), com o segundo tipo de segmentação, pelo método de Otsu em conjunto com a erosão, apresentaram os detalhes das sementes da segmentação de forma adequada.

Figura 59 – Limiarização de imagens das folhas de Mamão com deficiência de nutrientes e iluminação distinta: (a) Original (Mamão); (b) SMO (Mamão); (c) SMOOE (Mamão); (d) Original (Caferana); (e) SMO (Caferana); (f) SMOOE (Caferana).



Fonte: Elaboração do próprio autor

Figura 60 – Limiarização de imagens das folhas de Mamão com deficiência de nutrientes e iluminação distinta: (a) Original com baixa diferença na iluminação; (b) SMO (baixa); (c) SMOOE (baixa); (d) Original com alta diferença na iluminação; (e) SMO (alta); (f) SMOOE (alta).



Fonte: Elaboração do próprio autor

### 6.3 Segmentação com Redes Neurais Artificiais

As redes neurais artificiais são conhecidas pela sua capacidade de generalização e de classificação dos dados de forma não linear. Diversos trabalhos no sensoriamento remoto utilizam redes neurais principalmente para classificação de imagens de satélite (BOCCO et al., 2012), espectral (EDDY et al., 2008) e foliar (MOGHADDAM; DERAFSHI; SHIRZAD, 2011).

Neste trabalho tem como objetivo analisar a segmentação de deficiência de nutrientes pela RNA, em uma região específica pela folha da cultura. A RNA pode ser utilizada em diversas aplicações pela folha da planta, conforme definido na seção 3.2.

A aplicação de RNA com algoritmo de retro-propagação em determinado contexto, pode ser parametrizável em sete aspectos: (i) Quantidade de Camadas; (ii) Quantidade de Neurônios na Camada Escondida (QNCE); (iii) Quantidade e Tipos de Dados de Entrada; (iv) Taxa de Aprendizagem (TA); (v) Constante de Momento (CM); (vi) Critério de Parada; (vii) Função de Ativação. A habilidade de adaptação e robustez no reconhecimento de padrões são características que diferenciam as RNA das demais metodologias. Além disso, a RNA pode ser facilmente associada com outras técnicas de processamento de imagem e sinais, lógica nebulosa, algoritmos genéticos, entre outros. A RNA é aplicada nesta seção em duas funcionalidades, segmentação e filtragem da imagem.

Na segmentação por RNA utilizou-se o método de Otsu para aquisição das imagens desejadas (segmentadas), necessárias ao treinamento. Dentre as diversas imagens geradas foram selecionadas as mais adequadas para a análise da segmentação. Essas imagens definiram melhor a semente da região da folha com deficiência de nutriente, no qual, deve ser segmentada. Um grande número de variações entre imagens e parâmetros aplicados na RNA foram relacionados para a obtenção de melhores resultados, esses são os principais parâmetros a serem analisados na RNA. A relação de variações efetuadas nos aspectos parametrizáveis definidos para RNA estão listados a seguir:

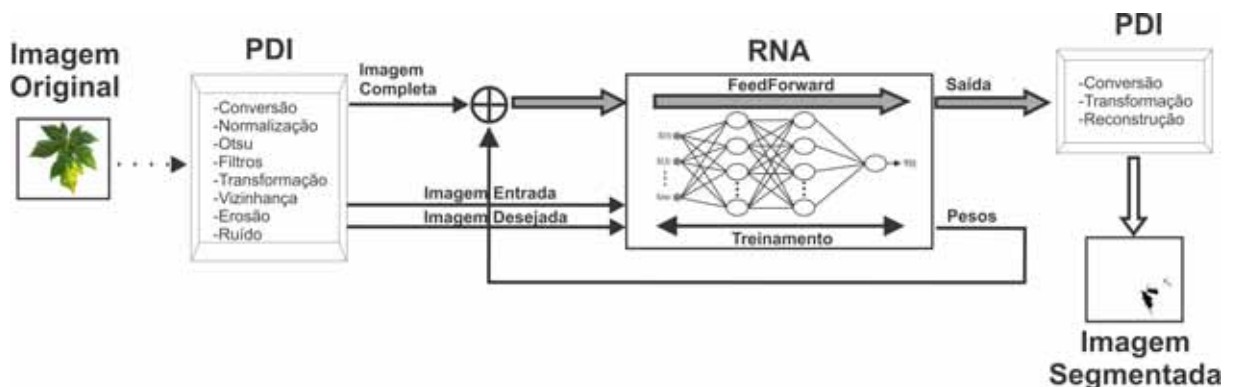
1. A quantidade de camadas não foram alteradas, utilizou-se apenas uma camada escondida;
2. A quantidade de neurônios na camada escondida alternaram-se entre 3, 4, 5 e 8;
3. A quantidade e tipos de dados de entrada foram diferenciadas com e sem aplicação de filtros na imagem de entrada, imagem do tipo RGB, imagem em escala de cinza, entradas por parâmetros probabilísticos (EPP), EPP e RGB, EPP e escala de cinza;
4. A taxa de aprendizagem alternaram-se entre os valores 0,01, 0,1, 0,3 e 0,5;
5. A constante de momento teve suas alterações entre os valores 0,3, 0,5, 0,7 e 0,9;

6. A quantidade de épocas foram alteradas em 100, 200, 500 e 5000, as primeiras implementações foram com 100 épocas, após a seleção dos melhores foram realizados testes com 500 e 5.000 épocas, e alguns casos com 200 épocas. Em cada etapa será especificada a quantidade de épocas utilizadas.

O desenvolvimento e a implementação de quatro etapas têm o objetivo de definir os melhores parâmetros para a organização do sistema e posteriormente o hardware na FPGA. Assim, uma grande investigação foi realizada na alteração desses parâmetros. A partir dos resultados adquiridos determinou-se uma estrutura adequada à RNA e o tipo de filtro mais adequado ao hardware. O sistema em *hardware* visa uma organização da arquitetura da RNA mais otimizada, com esse objetivo e por testes iniciais, não foi levado em consideração neste capítulo o aumento na quantidade de camadas escondidas na RNA. Além disso, dependendo da aplicação ou do método de aprendizagem utilizado, não existe a necessidade de se adicionar mais uma camada escondida (GONZALEZ; WOODS, 2010).

A organização e execução do sistema para obtenção dos resultados que irão determinar os melhores parâmetros discutidos está ilustrada na Figura 61, com seleção apenas da segmentação. A partir de uma imagem original são aplicados os métodos de processamento digital de imagens para gerar três tipos de imagens: **completa**, **entrada** e **desejada**.

**Figura 61 – Organização das principais etapas de execução do sistema.**

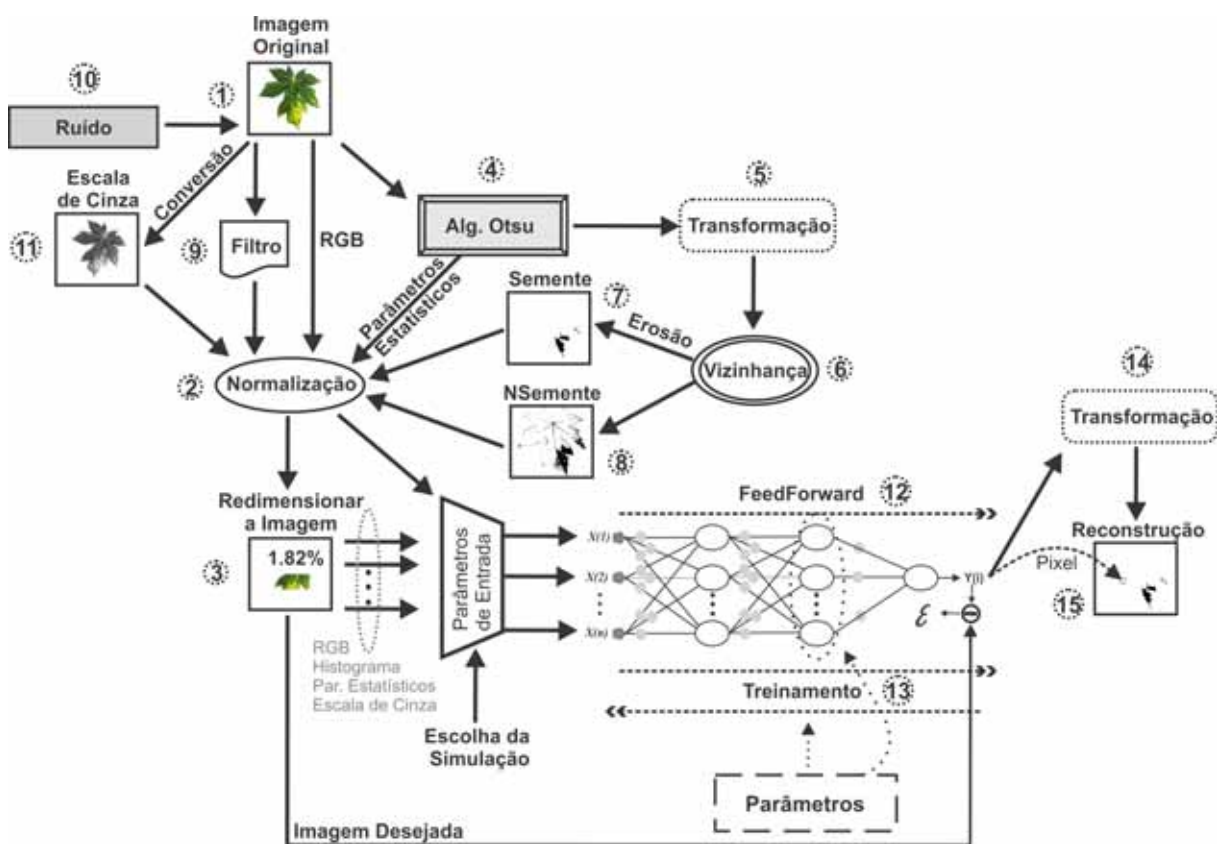


Fonte: Elaboração do próprio autor

A Figura 61 corresponde à execução do sistema de forma geral. Na Figura 62 apresentam-se todos os métodos utilizados no processamento digital de imagens e as distintas imagens definidas em cada método. A definição e as formas de execução dos métodos para gerar os dados e as imagens serão definidos em cada etapa das Seções 6.3.1, 6.3.2, 6.3.3 e 6.3.4, conforme as suas tarefas.

O treinamento é supervisionado e com critério de parada baseado em épocas. A fase de treinamento (13), apresentado na Figura 62, é executada para obtenção dos pesos por meio de entradas com dois tipos de imagens, entrada e desejada. Após a execução do

Figura 62 – Sequência de execução dos métodos utilizados nas etapas e tarefas.



Fonte: Elaboração do próprio autor

treinamento, é iniciado o modo *Feedforward* (12), apresentado na Figura 62, com a imagem completa sem redimensionar, ou seja, sequência 1-2 apenas, apresentada na Figura 62.

No treinamento, a imagem de entrada ( $f(x, y)$ ) é inserida na RNA após a normalização ( $f''(x, y)$ ) dos valores no intervalo [0:1], conforme Equação (33). A imagem completa também faz a normalização dos píxeis para valores com representação em ponto flutuante.

$$f''(x, y) = \frac{f(x, y)}{\text{Max}(f(x, y))} \quad (33)$$

A imagem completa pode ser utilizada antes ou após o treinamento, dependendo das tarefas em cada etapa ou para execução de outro método, por exemplo, um filtro de média ou a inserção de ruído. A imagem de entrada é reduzida para aproximadamente 1,82% da imagem completa, o restante é utilizado na validação. A quantidade de píxeis da imagem completa é de 76.800 e a imagem reduzida utilizada no treinamento possui apenas 1.400.

A imagem desejada tem a função de servir de referência para o treinamento da RNA. Esta imagem é gerada pela inserção de vários métodos, o principal é o algoritmo de Otsu com múltiplos limiares. A função do método é encontrar um limiar ótimo para efetuar a segmentação da imagem original e deixar apenas a parte da imagem correspondente a deficiência de nutrientes, ou seja, gerar a imagem desejada, conforme sequência 1-4-2-3 apresentada pela Figura 62.

O término do treinamento irá gerar os pesos necessários para executar o modo *Feedforward* (12) e segmentar a imagem completa, sequência 1-2 apresentada pela Figura 62, de acordo com a deficiência de nutriente. O treinamento no algoritmo de retropropagação é finalizado conforme o critério de parada (épocas), definido em cada seção.

A saída da execução do *Feedforward* é o píxel da imagem segmentada, ou seja, a execução em todas as simulações nas seções 6.3.1, 6.3.2, 6.3.3 e 6.3.4 são realizadas de forma sequencial píxel a píxel. A imagem de saída executa dois processos, em todos os casos, a transformação em escala de cinza (14) e reconstrução da imagem (15) como uma matriz NxM. Após os dois processos, têm-se a imagem segmentada de acordo com os parâmetros especificados no treinamento e na estrutura da RNA em cada simulação. Esse processo é realizado em todas as quatro etapas das seções 6.3.1, 6.3.2, 6.3.3 e 6.3.4, e as características específicas de cada seção são discutidas.

As quatro etapas realizadas nesse estudo estão relacionadas, respectivamente, às seções 6.3.1, 6.3.2, 6.3.3 e 6.3.4. A primeira etapa avalia os itens 2, 4, 5 e 6 relacionados nesta seção. A segunda etapa determina as melhores combinações de filtros lineares, itens 3 e 6. A terceira etapa altera os tipos e quantidades de entradas da RNA, item 2, 3, 4, 5 e 6.

A quarta etapa trata a influência dos filtros na segmentação e testes para averiguação da robustez da RNA, itens 3 e 6.

### 6.3.1 Primeira Etapa - Avaliação de Parâmetros

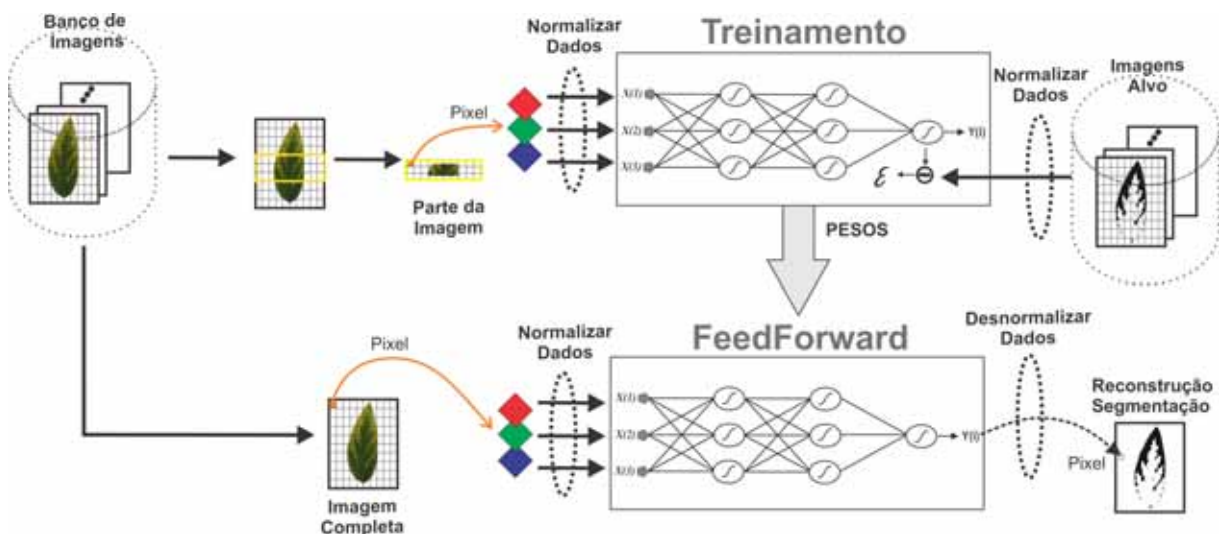
A primeira etapa consiste na identificação dos melhores parâmetros entre a imagem desejada, a taxa de aprendizagem, a constante de momento e a quantidade de neurônios na camada escondida. Nessa primeira etapa foram realizados o treinamento e a execução da RNA para duas tarefas:

1. Variações dos parâmetros: TA e CM;
2. Alteração da quantidade de neurônios na camada escondida.

A primeira tarefa contém uma RNA com 3 neurônios na camada de entrada correspondentes aos componentes RGB da imagem e a execução é sequencial píxel a píxel, a camada escondida contém 4 neurônios e a camada de saída com apenas 1 neurônio.

A camada de saída traz os resultados em uma imagem em escala de cinza para diferenciar entre duas classes, com e sem deficiência de nutriente na folha. Na Figura 63 apresenta-se a execução geral do sistema, com o treinamento feito com parte da imagem e o modo *feedforward* com a imagem completa. A função de ativação Tangente Hiperbólica foi utilizada em todos os neurônios e etapas.

Figura 63 – Organização geral do sistema em alto nível de abstração.



Fonte: Elaboração do próprio autor

As imagens de entrada e de referência (desejada) são apresentadas a partir das seguintes sequências, 1-2-3 e 1-4-5-6-7-2-3. Essas sequências de passos (métodos aplicados)

estão apresentadas na Figura 62, respectivamente. A segunda sequência corresponde à imagem “Semente” da Tabela 20, sendo que a imagem “NSemente ou Não-Semente” é obtida pela sequência 1-4-5-6-8-2-3, apresentada na Figura 62. Na primeira tarefa foram realizadas 32 simulações, sendo 16 com uma imagem desejada baseada no método de Otsu e o operador de erosão (Semente) e outras 16 com uma imagem desejada baseada apenas no método de Otsu (NSemente). O número de píxeis da área segmentada na imagem “Não-Semente” é de 11.189 e na “Semente” é de 2.621.

Nas primeiras 32 simulações investigaram-se as taxas de aprendizagem e a constante de momento, conforme os itens (iv) e (v) da seção 6.3. O primeiro conjunto de simulações tem o objetivo de analisar os melhores parâmetros e a forma da imagem desejada para a RNA.

A avaliação da porcentagem de acertos (PA) em cada simulação é gerado conforme Equações (34) e (35). O cálculo é baseado no incremento dos acertos e erros conforme um limite adequado de nível de cinza para o resultado.

Com apenas um neurônio na camada de saída da RNA e a determinação de uma imagem segmentada em binário no resultado, a variável “Acertos (A)” é incrementada apenas quando a saída da RNA está em um intervalo fechado entre 1/4 dos possíveis níveis de cinza para cima ou para baixo do valor desejado, caso contrário a variável “Erros (E)” é incrementada. Desta forma, analisa-se individualmente se cada píxel corresponde a uma margem aceitável de erro, outra forma simples para essa análise é definir um limiar, por exemplo a metade dos níveis de cinza, a parte superior do limiar corresponde a uma classe e a inferior a outra. Assim, investiga-se se o valor pertence a uma classe e os resultados adquiridos com essa metodologia foram melhores do que os apresentados na Tabela 20. Porém, escolheu-se o método da Equação (35) para se obter uma análise adequada de cada valor adquirido.

$$PA = 100 * (A / (A + E)) \quad (34)$$

$$PA = \begin{cases} A++ & \text{se } D + ((1/4) * LMax) > Saida > D - ((1/4) * LMax) \\ E++ & \text{se Caso Contrário} \end{cases} \quad (35)$$

Os resultados das 32 simulações estão apresentados na Tabela 20. As 6 melhores simulações com o percentual de acertos acima de 97% estão em destaque na Tabela 20. As melhores simulações foram repetidas com 500 e 5.000 épocas. A maior diferença nos resultados entre esses critérios de parada nos treinamentos é menor que 1% na

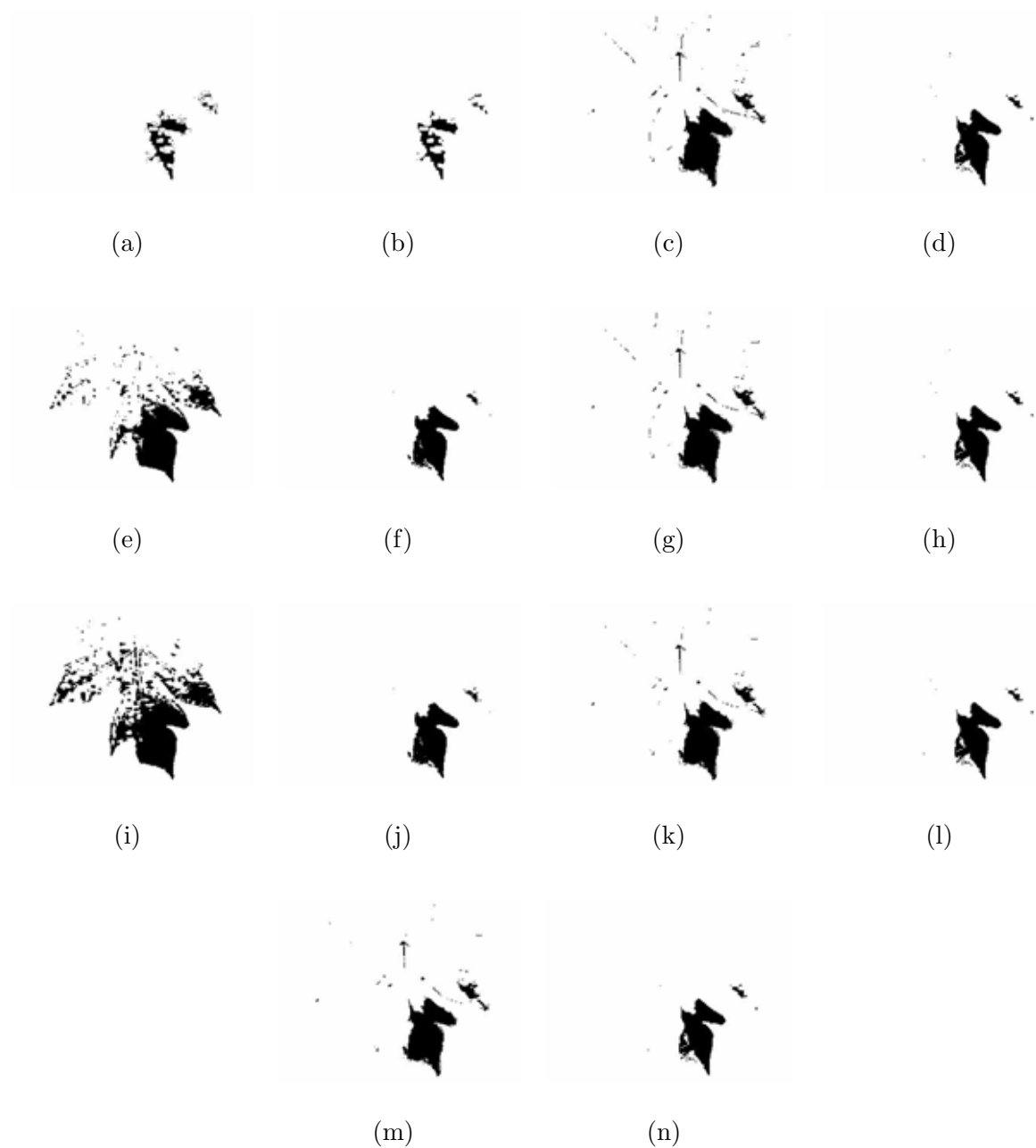
porcentagem de acertos, em relação aos treinamentos com 100 épocas. Assim, escolheu-se o treinamento com 100 épocas para uma validação adequada dos dados, e para obter os resultados dos conjuntos de simulações seguintes de forma mais rápida.

As simulações selecionadas correspondem aos números 1, 2, 5, 6, 9 e 13 (1ª coluna), todas pertencem ao grupo de imagens do tipo semente. Na Figura 64 apresenta-se as imagens segmentadas das melhores simulações. Na Figura 64-(i) observa-se a pior segmentação, de número 6, parte da área da folha que não deveria ser segmentada está em cor preta.

**Tabela 20 – Resultados da primeira etapa e primeira tarefa com 32 simulações. As colunas estão organizadas com o Tipo de Imagem (TI), a Quantidade de Neurônios na Camada Escondida (QNCE), a Taxa de Aprendizagem (TA), a Constante Momento (CM), a Porcentagem de Acertos (PA), o tempo de execução em segundos e o Erro Médio Quadrático (EMQ).**

Simulação	TI	QNCE	TA	CM	PA	Tempo (S)	EMQ
1	<i>Semente</i>	4	0,01	0,3	97,19%	208,25	0,01725
2	<i>Semente</i>	4	0,1	0,3	97,21%	207,61	0,01350
3	Semente	4	0,3	0,3	70,04%	205,64	0,02757
4	Semente	4	0,5	0,3	0,00%	206,21	0,19503
5	<i>Semente</i>	4	0,01	0,5	97,23%	204,11	0,01665
6	<i>Semente</i>	4	0,1	0,5	97,09%	204,3	0,01392
7	Semente	4	0,3	0,5	0,40%	203,03	0,06171
8	Semente	4	0,5	0,5	96,59%	203	0,22302
9	<i>Semente</i>	4	0,01	0,7	97,26%	204,35	0,01614
10	Semente	4	0,1	0,7	96,92%	209,47	0,01446
11	Semente	4	0,3	0,7	0,00%	206,51	0,10608
12	Semente	4	0,5	0,7	96,59%	206,29	0,34944
13	<i>Semente</i>	4	0,01	0,9	97,29%	218,65	0,01569
14	Semente	4	0,1	0,9	96,64%	241,21	0,01521
15	Semente	4	0,3	0,9	0,00%	225,35	0,13746
16	Semente	4	0,5	0,9	1,08%	230,46	0,30453
17	Não Semente	4	0,01	0,3	91,38%	219,11	0,02310
18	Não Semente	4	0,1	0,3	90,15%	221,13	0,00657
19	Não Semente	4	0,3	0,3	72,36%	218,66	0,04317
20	Não Semente	4	0,5	0,3	0,00%	225,2	0,18819
21	Não Semente	4	0,01	0,5	91,30%	222,45	0,02220
22	Não Semente	4	0,1	0,5	90,12%	209,84	0,00678
23	Não Semente	4	0,3	0,5	6,66%	209,76	0,07137
24	Não Semente	4	0,5	0,5	2,56%	213,57	0,23319
25	Não Semente	4	0,01	0,7	91,19%	212,41	0,02127
26	Não Semente	4	0,1	0,7	90,19%	217,89	0,00717
27	Não Semente	4	0,3	0,7	69,73%	214,85	0,09486
28	Não Semente	4	0,5	0,7	5,99%	205,03	0,26832
29	Não Semente	4	0,01	0,9	91,09%	207,92	0,02004
30	Não Semente	4	0,1	0,9	90,24%	205,89	0,00768
31	Não Semente	4	0,3	0,9	0,00%	208,52	0,13641
32	Não Semente	4	0,5	0,9	5,99%	210,33	0,33081

Figura 64 – As imagens geradas com as seis melhores segmentações com seleção em baixa (BI) e alta intensidade (AI): (a) Desejada - BI; (b) Desejada - AI; (c) N° 1 - BI; (d) N° 1 - AI; (e) N° 2 - BI; (f) N° 2 - AI; (g) N° 5 - BI; (h) N° 5 - AI; (i) N° 6 - BI; (j) N° 6 - AI; (k) N° 9 - BI; (l) N° 9 - AI; (m) N° 13 - BI; (n) N° 13 - AI;

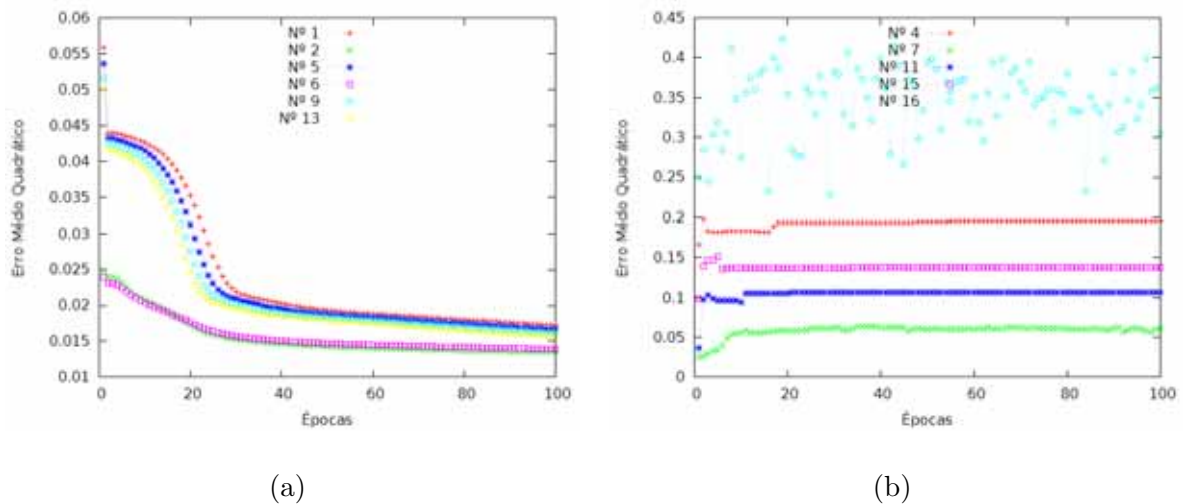


Fonte: Elaboração do próprio autor

As curvas de aprendizagem por meio do erro médio quadrático das seis melhores e das cinco piores simulações são apresentadas na Figura 65. As curvas de aprendizagem das melhores simulações apresentadas na Figura 65-(a) mostram o comportamento adequado do erro quadrático. As quatro primeiras simulações possuem grande semelhança entre si por causa do mesmo valor de taxa de aprendizagem, a diferença fica a cargo da constante de momento. Já as duas últimas simulações tem uma diferença mais expressiva das demais por causa da taxa de aprendizagem maior dessas simulações.

As curvas de aprendizagem apresentadas na Figura 65-(b), com as piores simulações, evidenciam valores altos de erro médio quadrático e em alguns casos o crescimento desse erro. A última simulação mostra um problema de oscilação do erro causado por um valor alto da constante de momento (HAYKIN, 2001). Esses comportamentos inadequados influenciam nos resultados da rede neural, por isso não foram utilizadas.

**Figura 65 – Curva de aprendizagem das melhores e piores simulações com 100 épocas: (a) 6 melhores. (b) 5 piores.**



Fonte: Elaboração do próprio autor

Na segunda tarefa foi feita a variação de neurônios na camada escondida com as 6 melhores simulações da Tabela 20 e os outros parâmetros não comentados são iguais a primeira tarefa. Os neurônios da camada escondida para execução foram alterados entre 5 e 8 neurônios, gerando mais 12 simulações. Respectivamente, as 6 primeiras com 5 neurônios e as 6 últimas com 8 neurônios conforme a Tabela 21.

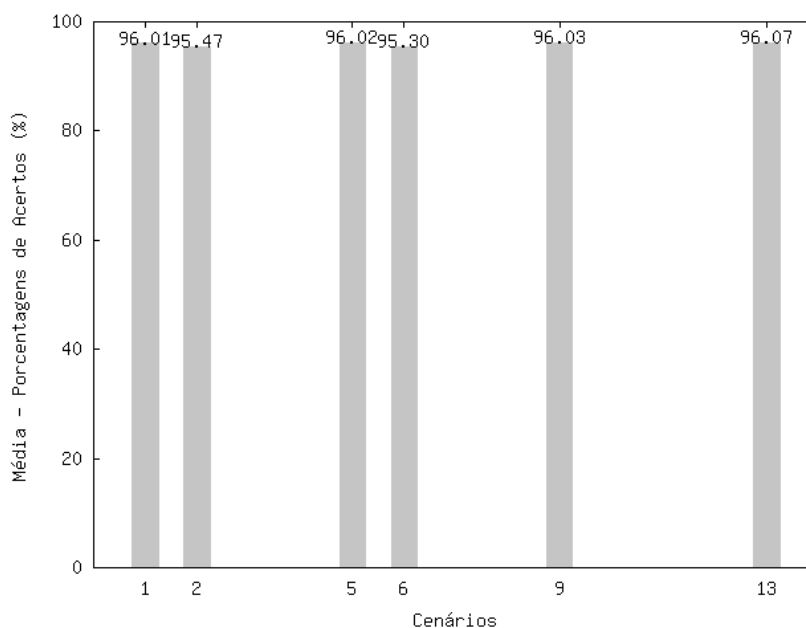
O melhor resultado escolheu-se por meio de média aritmética simples entre as melhores simulações com diferentes quantidades de épocas (100, 500 e 5000) e diferentes quantidades de neurônios na camada escondida (4, 5, 8). A melhor média foi a simulação número 13, com taxa de aprendizagem em 0,01, constante de momento em 0,9 e 4 neurônios na camada escondida, conforme Tabela 20. A porcentagem média obtida foi de 96,07% e os seus parâmetros de treinamento são utilizados em tarefas seguintes. As médias de todos

**Tabela 21** – Resultados da segunda tarefa da primeira etapa com as seis melhores simulações e alteração da quantidade de neurônios na camada escondida.

Simulação	QNCE	TA	CM	PA	Tempo (S)	EMQ
1	5	0,01	0,3	97,05%	236,13	0,01793
2	5	0,1	0,3	96,90%	238,71	0,01410
3	5	0,01	0,5	97,10%	237,79	0,01726
4	5	0,1	0,5	96,62%	235,6	0,01484
5	5	0,01	0,7	97,13%	227,6	0,01667
6	5	0,01	0,9	97,17%	228,51	0,01621
7	8	0,01	0,3	96,92%	291,9	0,01819
8	8	0,1	0,3	95,32%	294,26	0,01642
9	8	0,01	0,5	96,97%	297,7	0,01768
10	8	0,1	0,5	94,92%	296,85	0,01802
11	8	0,01	0,7	96,99%	305,25	0,01719
12	8	0,01	0,9	97,02%	307,23	0,01671

os acertos entre as melhores simulações são apresentadas na Figura 66. No Capítulo 7 investiga-se o efeito do aumento da quantidade de neurônios na camada escondida e com um caso comprovado de deficiência.

**Figura 66** – Médias das seis melhores simulações em porcentagens de acertos.



Fonte: Elaboração do próprio autor

Com esses resultados obtiveram-se as melhores combinações para o treinamento da RNA e o tipo de imagem desejada a ser utilizada nas análises posteriores. Os resultados apresentados na Tabela 21 demonstram que a RNA com 4 neurônios na camada escondida, apresentados na Tabela 20, obteve melhores resultados mesmo com um número menor de neurônios do que as simulações da Tabela 21.

### 6.3.2 Segunda Etapa - Aplicação de Filtros

A segunda etapa tem o objetivo de escolher os melhores tipos de filtros e máscaras para o presente trabalho. Os dois tipos de filtros lineares utilizados foram por meio de média aritmética simples e ponderada. A Equação 36 consiste na forma geral da computação dos filtros  $ft(x, y)$  sobre uma imagem  $f(x, y)$  de tamanho  $N \times M$  e máscara  $W(u, v)$ . As constantes  $p$  e  $q$  delimitam a computação dos dados e são encontradas a partir de  $p = (k/2) + 1/2$  e  $q = (k/2) - 1/2$ . Para facilitar a indexação dos dados foram adotadas máscaras de ordens iguais e ímpares. Em todos os filtros tem o produto entre os valores dos píxeis de cada componente da imagem com o valor correspondente à máscara. Os filtros de média simples contêm todos valores da máscara 1. Na média ponderada os valores são decrescentes do centro para as bordas em múltiplos de 2, as diagonais com valores menores que os vizinhos ortogonais ao centro, ou seja, em forma de losango (GONZALEZ; WOODS, 2010).

$$ft(x, y) = \frac{\sum_{r=p}^{N-q} \sum_{c=p}^{M-q} W(r, c) \times f(x + r, y + c)}{\sum_{r=p}^{N-q} \sum_{c=p}^{M-q} W(r, c)} \quad (36)$$

Nos dois casos foram alternados entre vários tamanhos de máscaras, conforme Tabela 22. Para obter os melhores, dentre várias combinações de filtros, utilizou-se a melhor simulação, número 13 da seção 6.3.1 na Tabela 20.

Os testes com os filtros lineares realizados são necessários para a análise em melhorias no ruído da imagem, tanto a imagem desejada como a imagem de entrada para a RNA foram verificadas. Nesta etapa são realizadas três tarefas:

1. Geração das imagens a serem utilizadas como parâmetro de referência da RNA (Imagem Desejada e Entrada), com variação de filtro, máscara, vizinhança e erosão conforme Tabela 68;
2. Treinamento e execução da RNA com a aplicação das melhores combinações de filtros na imagem desejada;
3. Treinamento e execução da RNA com a aplicação das melhores combinações de filtros na imagem original (Entrada da RNA) e desejada (Referência da RNA).

Os tipos de filtros aplicados conforme item (iii) são a Média simples e a Média Ponderada, as máscaras alternam-se em sete tamanhos distintos, são eles: 3x3, 5x5, 7x7, 9x9, 11x11, 15x15, 35x35. A Figura 67 apresenta as máscaras correspondentes aos filtros de média ponderada.

Figura 67 – Máscaras dos filtros de média ponderada: (a) 3x3; (b) 5x5; (c) 7x7; (d) 9x9; (e) 11x11.

1	2	1
2	4	2
1	2	1

(a)

1	1	2	1	1
1	2	4	2	1
2	4	8	4	2
1	2	4	2	1
1	1	2	1	1

(b)

1	1	1	2	1	1	1
1	1	2	4	2	1	1
1	2	4	8	4	2	1
2	4	8	16	8	4	2
1	2	4	8	4	2	1
1	1	2	4	2	1	1
1	1	1	2	1	1	1

(c)

1	1	1	1	2	1	1	1	1
1	1	1	2	4	2	1	1	1
1	1	2	4	8	4	2	1	1
1	2	4	8	16	8	4	2	1
2	4	8	16	32	16	8	4	2
1	2	4	8	16	8	4	2	1
1	1	2	4	8	4	2	1	1
1	1	1	2	4	2	1	1	1
1	1	1	1	2	1	1	1	1

(d)

1	1	1	1	1	2	1	1	1	1	1
1	1	1	1	2	4	2	1	1	1	1
1	1	1	2	4	8	4	2	1	1	1
1	1	2	4	8	16	8	4	2	1	1
1	2	4	8	16	32	16	8	4	2	1
2	4	8	16	32	64	32	16	8	4	2
1	2	4	8	16	32	16	8	4	2	1
1	1	2	4	8	16	8	4	2	1	1
1	1	1	2	4	8	4	2	1	1	1
1	1	1	1	2	4	2	1	1	1	1
1	1	1	1	1	2	1	1	1	1	1

(e)

Fonte: Elaboração do próprio autor

A diferença entre as vizinhanças do píxel são analisadas por três tipos, sem vizinhança (Imagem segmentada apenas com o método de Otsu), vizinhança de 4 e de 8. Por último, o tamanho da máscara da erosão variando em 3x3, 5x5, 11x11 e sem erosão. Na Tabela 22 apresenta-se a organização de todas as combinações com os filtros, as máscaras, as vizinhanças e as erosões investigadas. Nessa preparação da imagem de entrada para a RNA foram geradas 144 imagens, conforme a combinação de todas as possibilidades apresentadas pela Tabela 22. A Figura 68 ilustra as 13 melhores imagens geradas com as combinações realizadas.

**Tabela 22 – Combinação dos filtros aplicados na preparação da imagem desejada para a segmentação, com dois tipos de filtros (média e média ponderada) para ruído impulsivo, máscara ou janela de vários tamanhos ou ordens, a vizinhança com 3 tipos, Sem-Vizinhança (SV), Vizinhança-4 (V4) e Vizinhança-8 (V8) e a erosão em três tamanhos de máscara e Sem Erosão(SE).**

Filtro	Máscara	Vizinhança	Erosão
Média	3x3, 5x5, 7x7, 9x9, 11x11, 15x15, 35x35	SV, V4, V8	SE, 3x3, 5x5, 11x11
Média Ponderada	3x3, 5x5, 7x7, 9x9, 11x11	SV, V4, V8	SE, 3x3, 5x5, 11x11

A primeira tarefa selecionou as 13 melhores imagens dentre as 144 geradas. As simulações com a RNA foram feitas apenas com as 13 imagens escolhidas e estão relacionadas na Figura 68. As 13 imagens foram escolhidas devido a semente estar bem definida em dois pontos de coloração diferenciada na folha, conforme apresentado na Figura 59-(a;b;c). As combinações de tipos de filtros, vizinhança e erosão que foram utilizadas em cada imagem estão descritos na Figura 68, conforme a Tabela 22.

A RNA utilizada na segunda tarefa contém 3 neurônios na camada de entrada (Componentes RGB), 4 neurônios na camada escondida e 1 neurônio na camada de saída. Antes do treinamento e da execução do modo *FeedForward*, para a segmentação, são aplicadas as diferentes combinações selecionadas para a imagem desejada com o intuito de verificar possíveis melhorias no resultado da segmentação pela RNA.

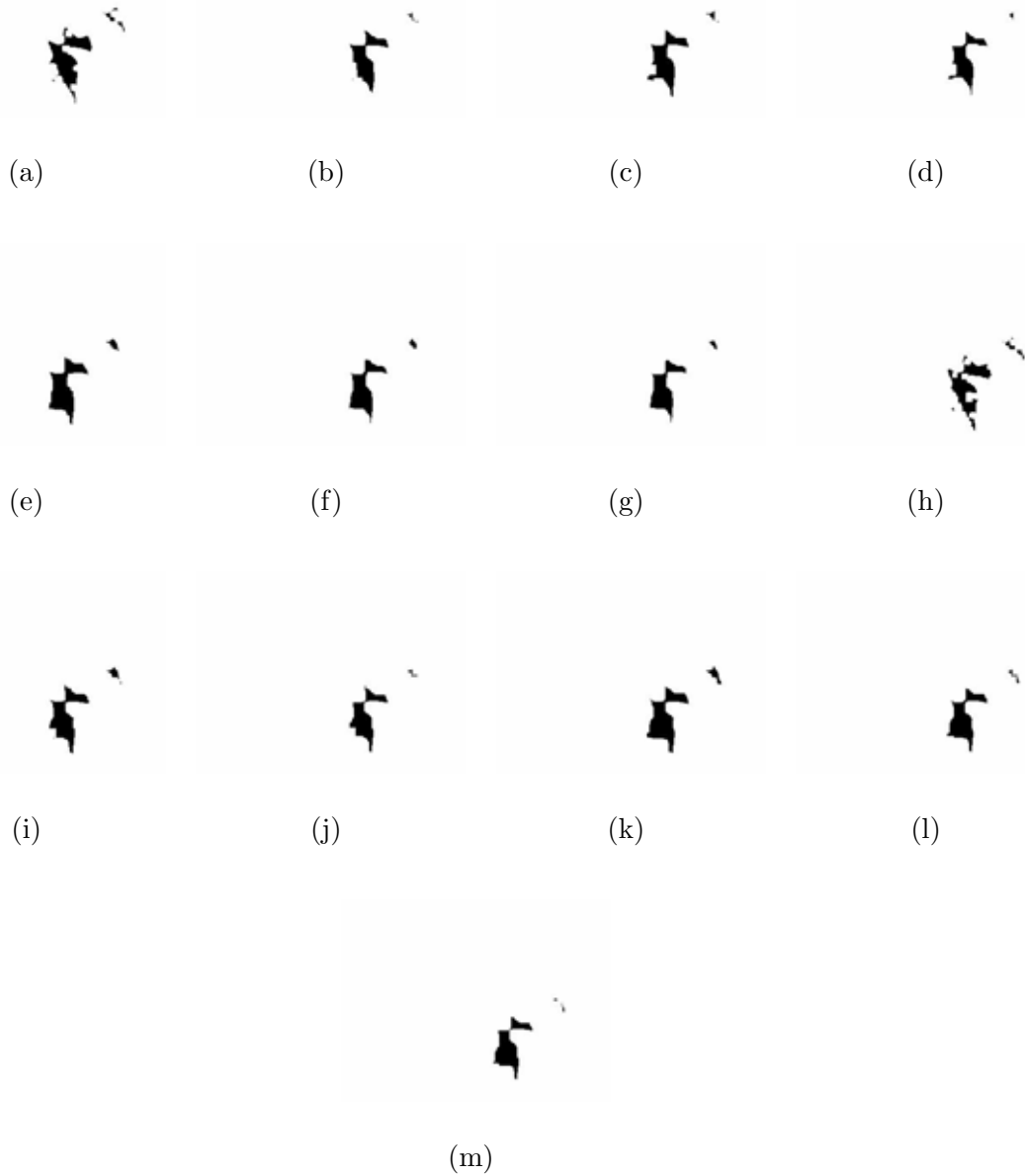
Os resultados obtidos na execução da segunda tarefa desta etapa são apresentados na Tabela 23 com as 13 melhores imagens escolhidas. Os resultados obtidos de 5 simulações são melhores que os apresentados na seção 6.3.1 na Tabela 20 que não contém filtros. As 5 melhores simulações estão em destaque na Tabela 23.

A terceira tarefa contém os mesmos parâmetros apresentados na segunda tarefa, a diferença está na aplicação dos filtros na imagem de entrada e as distintas combinações na imagem desejada. Na tarefa anterior a aplicação do filtro ocorreu apenas na imagem desejada. Nesta tarefa realizaram-se 5 simulações com os melhores resultados apresentadas na Tabela 23 e os resultados obtidos estão apresentados na Tabela 24.

<sup>1</sup>M3 – V8 – E5 ⇒ Mediana3x3 – Vizinhanade8 – Eroso5x5

<sup>2</sup>MP9 – SV – E11 ⇒ MdiaPonderada9x9 – SemVizinhana – Eroso11x11

Figura 68 – As 13 melhores simulações de imagens geradas: (a) M3-V8-E5<sup>1</sup>; (b) M5-SV-E11; (c) M7-SV-E11; (d) M7-V4-E11; (e) M9-SV-E11; (f) M9-V4-E11; (g) M9-V8-E11; (h) MP3-V8-E5; (i) MP9-SV-E11<sup>2</sup>; (j) MP9-V4-E11; (k) MP11-SV-E11; (l) MP11-V4-E11; (m) MP11-V8-E11.



Fonte: Elaboração do próprio autor

Tabela 23 – Resultados da segunda tarefa da segunda etapa com simulações de 1 a 13, a segunda coluna corresponde ao Tipo de Filtro (TF) utilizado.

Simulação	TF	QNCE	TA	CM	PA	Tempo (S)	EMQ
1	<i>Mediana-3x3-viz8-5-Erosao</i>	4	0,01	0,9	97,57%	245,79	0,01356
2	<i>Mediana-5x5-rec-11-Erosao</i>	4	0,01	0,9	97,53%	247,38	0,01692
3	Mediana-7x7-rec-11-Erosao	4	0,01	0,9	97,07%	247,03	0,02508
4	<i>Mediana-7x7-viz4-11-Erosao</i>	4	0,01	0,9	97,48%	247,19	0,02283
5	Mediana-9x9-rec-11-Erosao	4	0,01	0,9	96,48%	245,88	0,02823
6	Mediana-9x9-viz4-11-Erosao	4	0,01	0,9	97,00%	254,24	0,02751
7	<i>Mediana-9x9-viz8-11-Erosao</i>	4	0,01	0,9	97,40%	243,58	0,02715
8	<i>MedPond-3x3-viz8-5-Erosao</i>	4	0,01	0,9	97,41%	242,35	0,01500
9	MedPond-9x9-rec-11-Erosao	4	0,01	0,9	96,88%	242,42	0,02316
10	<i>MedPond-9x9-viz4-11-Erosao</i>	4	0,01	0,9	97,32%	241,58	0,02145
11	MedPond-11x11-rec-11-Erosao	4	0,01	0,9	96,26%	242,25	0,02850
12	MedPond-11x11-viz4-11-Erosao	4	0,01	0,9	97,04%	247,79	0,02778
13	MedPond-11x11-viz8-11-Erosao	4	0,01	0,9	97,19%	244,63	0,02670

Tabela 24 – Resultados da terceira tarefa da segunda etapa com simulações de 1 a 8, a segunda coluna corresponde ao Tipo da Imagem Desejada e a Original (TIDO) utilizada.

Simulação	TIDO	QNCE	TA	CM	PA	Tempo (S)	EMQ
1	Mediana-3x3-viz8-5-Erosao	4	0,01	0,9	97,70%	239,85	0,01257
2	Mediana-5x5-rec-11-Erosao	4	0,01	0,9	97,75%	274,38	0,01530
3	Mediana-7x7-viz4-11-Erosao	4	0,01	0,9	97,48%	349,05	0,02037
4	Mediana-9x9-viz8-11-Erosao	4	0,01	0,9	97,57%	400,55	0,02097
5	MedPond-3x3-viz8-5-Erosao	4	0,01	0,9	97,51%	238,59	0,01410

A porcentagem de acertos na aplicação de filtros na imagem desejada e de entrada (Tarefa 3) é mais significativo que a aplicação apenas na imagem desejada (Tarefa 2). Os resultados apresentados na Tabela 24 mostram melhorias em todos os 5 casos na comparação com a Tabela 23.

Nessa etapa empregou-se filtros lineares, na preparação da imagem antes da segmentação pelo método de Otsu para gerar a imagem desejada a RNA no seu treinamento. A aquisição dessas novas imagens desejadas para o treinamento da RNA não melhora de forma expressiva os resultados da segmentação apresentados pela primeira etapa na seção 6.3.1. No entanto, a etapa determinou as melhores combinações de filtros, máscara, vizinhança e erosão para as próximas análises. Assim, na seção 6.3.4 investiga-se a aplicação das melhores combinações na imagem de entrada quando aplicado um ruído impulsivo, espera-se que com o ruído os resultados da primeira etapa fiquem comprometidos.

### ***6.3.3 Terceira Etapa - Alteração de Entradas***

No sensoriamento remoto por meio de imagens de satélites com segmentação por RNA, geralmente a inserção de novas características da imagem podem influenciar e melhorar os resultados da segmentação (CALDERANO FILHO et al., 2013; PINHEIRO et al., 2013; SILVA et al., 2013).

A utilização de valores estatísticos como parâmetros de entrada em redes neurais artificiais é amplamente utilizado no reconhecimento de padrões ou classificação. Assim, nessa etapa serão analisadas a inserção de diferentes tipos de entradas, elencadas em quatro tarefas:

1. Inserção do histograma como valor adicional a entrada junto aos componentes RGB da imagem, tornando-se a RNA com 4 entradas e 5 neurônios na camada escondida;
2. Definição de 3 entradas para RNA, dois valores estatísticos Intensidade Acumulada e a Variância entre Classes, e a intensidade de níveis de cinza da imagem;
3. RNA com duas entradas: histograma e a intensidade de níveis de cinza da imagem;
4. RNA com apenas uma entrada correspondente a intensidade de níveis de cinza da imagem.

Em todas as tarefas alteram-se as entradas da RNA e os seus parâmetros de taxa de aprendizagem e constante de momento, como feito na seção 6.3.1. Devido ao comportamento diferente da RNA, com as novas entradas, as análises serão feitas com 16

simulações. Nas entradas que combinam valores referentes as imagens e estatísticos, os valores estatísticos adotados são os mesmos contidos no algoritmo de Otsu da seção 6.2.

A partir do histograma são realizados todos os cálculos do método de Otsu, então presume-se que seria uma das características mais adequadas a serem analisadas. Na primeira tarefa é inserido o histograma como entrada adicional a RNA, as outras três entradas são os componentes RGB da imagem. A RNA contém 4 neurônios na camada de entrada, 5 neurônios na camada escondida e 1 neurônio na camada de saída.

Os resultados são apresentados na Tabela 25 e as 6 melhores simulações são identificadas em negrito. As melhores simulações da Tabela 25 correspondem as mesmas apresentadas na seção 6.3.1 com a RNA de 3 entradas. Porém, os resultados alcançados na Tabela 25 são inferiores aos da seção 6.3.1. Assim, a inserção de valores estatísticos adicionais, como o histograma, na entrada da RNA não trazem melhorias, quando comparados aos resultados apresentados na Tabela 20 da seção 6.3.1, com exceção da simulação número 2. Porém, a melhoria foi de 0,07% e não tornou-se um atrativo para utilizar esse tipo de simulação, pelo fato de ter um neurônio a mais na camada de entrada.

**Tabela 25 – Resultados da primeira tarefa da terceira etapa com 16 simulações e RNA de 4 entradas.**

Simulação	TI	QNCE	TA	CM	PA	Tempo (S)	EMQ
<b>1</b>	<b>Semente</b>	<b>4</b>	<b>0,01</b>	<b>0,3</b>	<b>96,99%</b>	<b>246,5</b>	<b>0,01789</b>
<b>2</b>	<b>Semente</b>	<b>4</b>	<b>0,1</b>	<b>0,3</b>	<b>97,36%</b>	<b>246,77</b>	<b>0,01377</b>
3	Semente	4	0,3	0,3	0,03%	245,87	0,04332
4	Semente	4	0,5	0,3	1,87%	246,31	0,20594
<b>5</b>	<b>Semente</b>	<b>4</b>	<b>0,01</b>	<b>0,5</b>	<b>97,05%</b>	<b>244,68</b>	<b>0,01738</b>
<b>6</b>	<b>Semente</b>	<b>4</b>	<b>0,1</b>	<b>0,5</b>	<b>97,07%</b>	<b>247,47</b>	<b>0,01455</b>
7	Semente	4	0,3	0,5	0,00%	243,98	0,10704
8	Semente	4	0,5	0,5	0,00%	244,98	0,19553
<b>9</b>	<b>Semente</b>	<b>4</b>	<b>0,01</b>	<b>0,7</b>	<b>97,10%</b>	<b>246,58</b>	<b>0,01687</b>
10	Semente	4	0,1	0,7	96,74%	244,97	0,01553
11	Semente	4	0,3	0,7	3,41%	244,73	0,15168
12	Semente	4	0,5	0,7	1,08%	244,16	0,27572
<b>13</b>	<b>Semente</b>	<b>4</b>	<b>0,01</b>	<b>0,9</b>	<b>97,11%</b>	<b>265,73</b>	<b>0,01639</b>
14	Semente	4	0,1	0,9	96,37%	258,9	0,01656
15	Semente	4	0,3	0,9	1,87%	252,52	0,18468
16	Semente	4	0,5	0,9	1,08%	250,73	0,31446

Na segunda tarefa desta etapa, as três entradas da RNA são compostas por dois valores estatísticos obtidos a partir das Equações (17) e (21) da seção 2.4.4, e a outra entrada é a intensidade de cinza do píxel correspondente. Os dois valores estatísticos são respectivamente a Intensidade Acumulada e a Variância entre Classes do algoritmo de Otsu, e uma imagem em escala de nível de cinza disponibiliza o píxel correspondente à terceira entrada. A RNA contém 3 neurônios na camada de entrada, 4 neurônios na camada escondida e 1 neurônio na camada de saída.

Na Tabela 25 apresentam-se os resultados das simulações com a variação dos parâmetros de treinamento. Os valores obtidos são menores que os apresentados na Tabela 20 da seção 6.3.1. Porém, os resultados obtidos mostram a capacidade da RNA trabalhar com entradas estatísticas e reduzir a quantidade de memória utilizada no hardware. Já que as duas entradas de intensidade acumulada e a variância entre classes necessitam apenas de vetores de 256 elementos, e apenas uma entrada com matriz do tamanho da imagem, diferentemente da etapa 1 da seção 6.3.1 na qual necessita-se dos 3 componentes RGB com matrizes do tamanho da imagem. Outro ponto que deve ser levantado é a necessidade de um pré-processamento nesta etapa, ou seja, os valores estatísticos (entrada da RNA) são gerados pela execução de parte do algoritmo de Otsu.

**Tabela 26 – Resultados da segunda tarefa da terceira etapa com 16 simulações e RNA de 3 entradas.**

Simulação	TI	QNCE	TA	CM	PA	Tempo (S)	EMQ
1	Semente	4	0,01	0,3	73,83%	225,85	0,042018
2	Semente	4	0,1	0,3	72,45%	222,99	0,0202017
3	Semente	4	0,3	0,3	76,70%	238,18	0,0232404
4	Semente	4	0,5	0,3	0,00%	223,14	0,173427
5	Semente	4	0,01	0,5	77,47%	220	0,038844
6	Semente	4	0,1	0,5	87,52%	214,44	0,0199086
7	Semente	4	0,3	0,5	0,40%	228,17	0,0264741
8	Semente	4	0,5	0,5	1,08%	211,68	0,208839
9	Semente	4	0,01	0,7	81,42%	211	0,033837
10	Semente	4	0,1	0,7	89,85%	231,79	0,0197991
11	Semente	4	0,3	0,7	0,00%	226,12	0,036093
12	Semente	4	0,5	0,7	0,05%	219,35	0,201273
13	Semente	4	0,01	0,9	84,94%	214,36	0,03015
14	Semente	4	0,1	0,9	90,77%	208,86	0,0207594
15	Semente	4	0,3	0,9	0,00%	205,54	0,090645
16	Semente	4	0,5	0,9	1,08%	205,97	0,183255

A terceira e quarta tarefas consistem em efetuar o treinamento da RNA com menor quantidade de entradas e verificar a possibilidade da redução do sistema. Nessas duas tarefas houve a necessidade de efetuar o treinamento com maior quantidade de épocas para estabilização dos resultados. Assim, as simulações apresentadas nas Tabelas 27 e 28 são executadas com 200 épocas no treinamento.

A terceira tarefa contém duas entradas, uma para imagem em escala de cinza e outra para o histograma. O histograma da imagem pode ser obtido com maior facilidade em comparação aos parâmetros estatísticos extraídos pelo método de Otsu. A RNA da terceira tarefa contém 2 neurônios na camada de entrada, 3 neurônios na camada escondida e 1 neurônio na camada de saída. Os resultados são apresentados na Tabela 27 e são inferiores aos resultados da etapa 1 da seção 6.3.1.

A quarta tarefa contém apenas uma entrada correspondente a imagem em escala de cinza. A RNA contém 1 neurônio na camada de entrada, 3 neurônios na camada escondida

**Tabela 27 – Resultados da terceira tarefa da terceira etapa com 16 simulações e RNA de 2 entradas.**

Simulação	TI	QNCE	TA	CM	PA	Tempo (S)	EMQ
1	Semente	3	0,01	0,3	88,87%	271,11	0,02995
2	Semente	3	0,1	0,3	90,36%	270,3	0,02153
3	Semente	3	0,3	0,3	67,63%	274,03	0,02213
4	Semente	3	0,5	0,3	0,00%	267,37	0,07823
5	Semente	3	0,01	0,5	88,71%	267,77	0,02953
6	Semente	3	0,1	0,5	90,16%	265,51	0,02109
7	Semente	3	0,3	0,5	3,41%	263,65	0,02349
8	Semente	3	0,5	0,5	0,00%	265,96	0,14772
9	Semente	3	0,01	0,7	89,12%	266,96	0,02920
10	Semente	3	0,1	0,7	90,39%	263,9	0,02074
11	Semente	3	0,3	0,7	3,41%	265,43	0,02642
12	Semente	3	0,5	0,7	1,87%	264,37	0,19302
13	Semente	3	0,01	0,9	89,06%	265,26	0,02894
14	Semente	3	0,1	0,9	81,77%	263,78	0,02048
15	Semente	3	0,3	0,9	0,18%	256,59	0,03410
16	Semente	3	0,5	0,9	0,00%	263,12	0,21377

e 1 neurônio na camada de saída. Resultados satisfatórios foram alcançados apenas com o mínimo de 3 neurônios na camada escondida. Os resultados são apresentados na Tabela 28.

**Tabela 28 – Resultados da quarta tarefa da terceira etapa com 16 simulações e RNA de 1 entrada.**

Simulação	TI	QNCE	TA	CM	PA	Tempo (S)	EMQ
1	Semente	3	0,01	0,3	92,95%	226,55	0,02638
2	Semente	3	0,1	0,3	68,07%	230,12	0,02072
3	Semente	3	0,3	0,3	68,35%	233,26	0,02156
4	Semente	3	0,5	0,3	0,00%	238,8	0,06119
5	Semente	3	0,01	0,5	94,39%	239,68	0,02489
6	Semente	3	0,1	0,5	68,10%	238,78	0,02055
7	Semente	3	0,3	0,5	68,05%	238,04	0,02285
8	Semente	3	0,5	0,5	0,00%	238,95	0,13617
9	Semente	3	0,01	0,7	94,95%	233,86	0,02370
10	Semente	3	0,1	0,7	68,13%	232,46	0,02037
11	Semente	3	0,3	0,7	3,41%	233,11	0,02442
12	Semente	3	0,5	0,7	3,41%	226,55	0,19414
13	Semente	3	0,01	0,9	95,18%	234,71	0,02331
14	Semente	3	0,1	0,9	68,16%	230,67	0,02020
15	Semente	3	0,3	0,9	0,18%	231,2	0,02744
16	Semente	3	0,5	0,9	1,08%	231,64	0,21780

As quatro combinações de entradas para a RNA obtiveram resultados inferiores aos apresentados na Tabela 20. Porém, os resultados obtidos mostram a capacidade da RNA trabalhar com outros tipos e quantidades de entradas. Todas combinações tiveram resultados acima de 90%, em pelo menos uma configuração. Os melhores resultados estão na primeira e última combinação (Nº de entradas: 4 e 1), respectivamente, alcançando

resultados acima de 97% e 94%.

Nesta etapa, pode-se confirmar que o uso de componentes RGB na entrada conduzem a melhores resultados na segmentação da imagem por RNA. Na seção 6.3.4 será analisado o comportamento da RNA com a inserção de ruído impulsivo, com base na obtenção de parâmetros mais adequados, em vários aspectos, apresentados nas Seções 6.3.1, 6.3.2 e 6.3.3.

#### **6.3.4 Quarta Etapa - Análise de Ruídos**

A última etapa tem como objetivo avaliar o comportamento da RNA e dos filtros com a inserção de ruídos em distintos percentuais e componentes da imagem.

A inserção de ruído impulsivo (Sal e Pimenta) é uma maneira de validar a robustez de um determinado método para esse tipo de anomalia, existe a possibilidade de existência desse tipo de ruído principalmente quando se trabalha com imagens em ambientes externos. O ruído impulsivo consiste na inserção de valores máximos e mínimos de intensidade de forma aleatória, respectivamente preto e branco (255 e 0 com níveis de intensidade de 8 bits) em determinado componente da imagem.

Nesta etapa são comparadas as melhores formas de aplicação de filtro nas imagens para tratar ruídos impulsivos, para isso foram definidas três tarefas:

1. Inserção de ruído impulsivo e aplicação dos melhores filtros e combinações de técnicas para imagem, além do treinamento e de execução da RNA com a função de segmentação da imagem;
2. Inserção de ruído impulsivo, treinamento e execução da RNA com a função de filtro e de segmentação da imagem;
3. Inserção de ruído impulsivo, treinamento e execução da RNA com a função apenas de segmentação da imagem.

A etapa foi organizada com diferentes percentuais de ruído impulsivo (Sal e Pimenta) em 5%, 10%, 15%, 20%, 25% e 30%. A aplicação do ruído foi alternada entre cada um dos três componentes RGB da imagem e com os três juntos. Desta forma, para cada simulação serão gerados 24 tipos distintos de ruído impulsivo na imagem e um sem ruído. O treinamento da RNA será efetuado com 1,82% de cada uma das 25 imagens geradas, 24 imagens com ruído e uma imagem sem ruído.

Os parâmetros da RNA são os mesmos para todas as simulações e correspondem aos valores da melhor simulação na seção 6.3.1, número 13 da Tabela 22. Em cada simulação

são analisados os percentuais de acertos relacionados à parte de filtragem e segmentação da imagem. Na parte da filtragem da imagem de saída é comparada a imagem original nos três componentes (RGB), conforme Equação (37). O limite aceitável para variação do nível de cinza no filtro é menor que 5%. Enquanto que na segmentação esse limite é de 50 % conforme Equações (34) e (35) da seção 6.3.1.

$$PA = \begin{cases} A++ & \text{se } D + ((1/40) * LMax) > Saída > D - ((1/40) * LMax) \\ E++ & \text{se Caso Contrário} \end{cases} \quad (37)$$

A primeira tarefa gera 25 imagens e aplica o filtro correspondente à simulação em todas as imagens. A RNA nessa primeira tarefa terá a função apenas de segmentação e o treinamento será feito com 1,82% das 25 imagens após aplicação do filtro. As imagens desejadas e as cinco melhores combinações com filtros, máscaras, vizinhança e erosão foram encontradas na Tabela 22 da seção 6.3.2. Essas combinações são utilizadas para obtenção dos resultados da primeira tarefa e para comparação com aplicação do filtro por RNA na segunda tarefa.

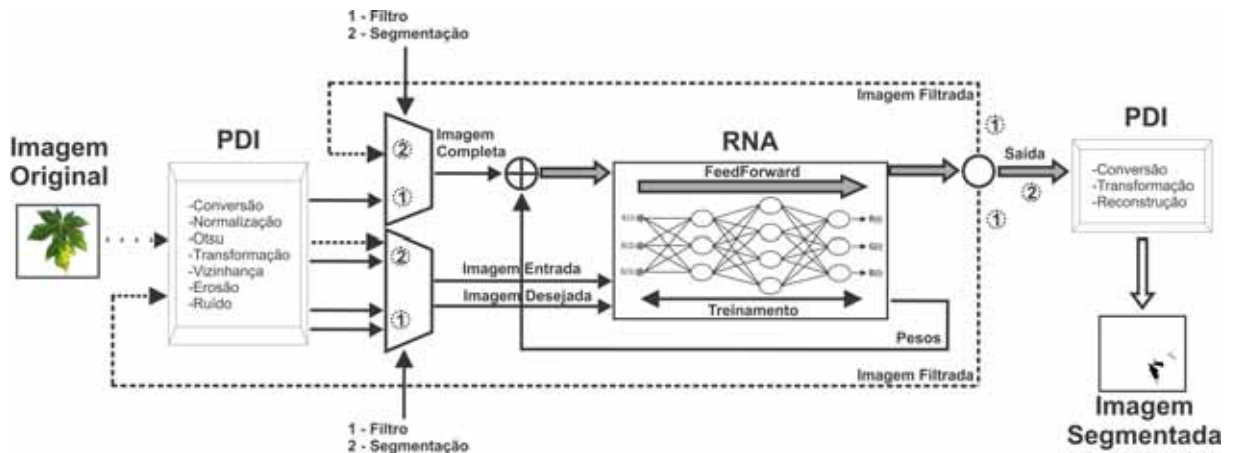
A segunda tarefa tem apenas uma diferença em relação à primeira tarefa, a primeira tarefa aplica diversos tipos de filtros lineares na imagem com ruído para obtenção de melhorias nas imagens antes da segmentação com a RNA. Na segunda tarefa utiliza-se a própria RNA como filtro, as imagens geradas com ruído servem de entrada e a imagem original como alvo para o treinamento da RNA. Após treinamento da RNA, executa-se o modo *Feedforward* para obtenção de cada imagem filtrada. O processo de treinamento e execução do modo *Feedforward* é repetido, mas agora as imagens filtradas adquiridas no treinamento anterior são as entradas e a imagem segmentada é o alvo.

Essa tarefa é apresentada nos resultados como “RNA-Filtro-Segmentação (RFS)” e é ilustrada pela Figura 69. Para a RNA determinar os resultados específicos para cada etapa foram elaboradas duas RNAs, uma para o filtro e outra para a segmentação. No Filtro a RNA é organizada em uma estrutura 3-4-3 e na segmentação 3-4-1.

A “RNA-SFiltro-Segmentação (RSFS)” corresponde à terceira tarefa e não possui o processo de filtragem da imagem, apenas o treinamento com as imagens ruidosas para obtenção da imagem segmentada como alvo, ou seja, apenas o segundo processo da segunda tarefa (Segmentação).

Todas as tabelas de resultados contêm nas cinco primeiras simulações o nome completo corresponde à combinação de filtros lineares utilizados na imagem desejada. A parte antes do segundo hífen é a identificação do tipo de filtro e do tamanho da máscara. Depois sucessivamente apresenta o tipo de vizinhança e erosão. A sexta linha corresponde à execução das duas RNAs, uma com a função de filtro e outra de segmentação, corresponde

Figura 69 – Organização das principais etapas de execução do sistema com a RNA utilizada para filtro.



Fonte: Elaboração do próprio autor

à segunda tarefa. A sétima linha é a execução da RNA apenas como segmentação sem a utilização de filtros e corresponde à terceira tarefa. A relação das siglas com as devidas simulações são listadas a seguir:

1. Mediana-3x3-viz8-5-Erosao (M3V8E5);
2. Mediana-5x5-rec-11-Erosao (M5SVE11);
3. Mediana-7x7-viz4-11-Erosao (M7V4E11);
4. Mediana-9x9-viz8-11-Erosao (M9V8E11);
5. MedPond-3x3-viz8-5-Erosao (MP3V8E5);
6. RNA-Filtro-Segmentação (RFS);
7. RNA-SFiltro-Segmentação (RSFS).

Os resultados dessa etapa são apresentados em 10 tabelas. As cinco primeiras Tabelas 29, 30, 31, 32 e 33 estão relacionadas ao processo de filtro da imagem. A simulação RNA-SFiltro-Segmentação (RSFS) não está presente por não conter o processo de filtragem da imagem. Nas Tabelas 34, 35, 36, 37 e 38 apresentam-se os resultados para a aplicação da RNA com a função de segmentação da imagem e transformação em escala de cinza.

Todos os resultados são apresentados conforme o percentual de acerto correspondente às Equações (37) e (35), respectivamente para o filtro e a segmentação da imagem. Cada grupo de cinco tabelas contém quatro tabelas para os resultados da inserção de ruído em

cada componente RGB e em todos os componentes da imagem, e uma tabela com a média entre as quatro primeiras.

As Tabelas 29, 30 e 31 apresentam os resultados para cada simulação com a inserção de ruído em um dos três componentes da imagem, respectivamente, componente R, G e B. A primeira coluna especifica o tipo de simulação adotado. As sete seguintes colunas são os percentuais de acertos nos ruídos especificados, a primeira sem ruído (0%) e as outras seis com ruídos de 5% a 30% no componente da imagem específico correspondente à tabela.

As três últimas colunas têm, respectivamente, os percentuais de acertos para os componentes sem a inserção de ruído, nas duas primeiras, e a média alcançada por cada simulação na última coluna. Nas duas colunas que não tiveram a inserção de ruído obtiveram valores idênticos na utilização de filtros baseados em máscaras, cinco primeiras simulações. Na sexta simulação, com o filtro baseado em RNA, foi realizada a média aritmética simples entre os resultados para determinar o percentual de acerto médio entre os componentes sem inserção de ruído.

A Tabela 32 contém os resultados das simulações para os percentuais de ruídos em todos os componentes da imagem (RGB). As diferenças para as Tabelas 29, 30 e 31 são a inexistência da coluna sem ruído e dos componentes sem inserção de ruído. Todos os percentuais de acertos da Tabela 32 são apresentados por meio de média aritmética simples entre todos os componentes (RGB) da imagem. A Tabela 33 apresenta as médias gerais de todos os percentuais de ruído entre todos os componentes analisados e a média geral para cada simulação.

**Tabela 29 – Resultados do filtro da imagem RGB para o componente R.**

Simulação	0%	5%	10%	15%	20%	25%	30%	G	B	Média
M3V8E5	89,46%	62,22%	45,60%	34,91%	27,42%	23,02%	19,84%	89,49%	88,73%	48,90%
M5SVE11	82,92%	59,61%	44,74%	35,16%	29,03%	24,72%	21,76%	82,90%	81,28%	47,40%
M7V4E11	77,79%	61,91%	47,68%	37,98%	31,67%	28,11%	23,56%	77,59%	75,85%	48,05%
M9V8E11	73,43%	62,00%	49,20%	40,08%	34,54%	29,26%	25,18%	72,93%	71,51%	48,09%
MP3V8E5	91,56%	66,23%	47,81%	36,54%	28,48%	22,98%	19,34%	91,60%	91,03%	50,50%
RFS	78,92%	76,65%	74,33%	72,26%	69,90%	67,61%	65,63%	62,61%	60,66%	69,84%

**Tabela 30 – Resultados do filtro da imagem RGB para o componente G.**

Simulação	0%	5%	10%	15%	20%	25%	30%	R	B	Média
M3V8E5	89,49%	61,55%	42,60%	30,91%	22,67%	18,20%	15,65%	89,46%	88,73%	46,22%
M5SVE11	82,90%	59,27%	42,61%	32,17%	25,90%	20,60%	17,37%	82,92%	81,28%	45,27%
M7V4E11	77,59%	60,41%	42,58%	32,73%	23,97%	17,62%	12,70%	77,79%	75,85%	42,96%
M9V8E11	72,93%	59,12%	40,21%	28,97%	19,47%	13,29%	10,56%	73,43%	71,51%	39,57%
MP3V8E5	91,60%	66,53%	48,45%	37,59%	28,24%	22,94%	19,11%	91,56%	91,03%	50,68%
RFS	69,49%	69,40%	69,30%	69,26%	69,11%	69,02%	68,90%	78,80%	65,73%	69,89%

Os resultados do processo de filtragem mostram que a RNA é melhor, do que as

**Tabela 31 – Resultados do filtro da imagem RGB para o componente B.**

Simulação	0%	5%	10%	15%	20%	25%	30%	R	G	Média
M3V8E5	88,73%	68,25%	52,51%	38,70%	28,36%	19,85%	13,91%	89,46%	89,49%	50,07%
M5SVE11	81,28%	51,35%	35,26%	25,26%	17,50%	11,11%	7,84%	82,92%	82,90%	39,27%
M7V4E11	75,85%	49,82%	30,89%	16,78%	9,52%	5,57%	3,33%	77,59%	77,79%	33,91%
M9V8E11	71,51%	50,66%	26,84%	12,53%	6,52%	2,57%	1,28%	73,43%	72,93%	30,85%
MP3V8E5	91,03%	70,42%	53,91%	41,48%	30,68%	24,05%	18,55%	91,56%	91,60%	52,78%
RFS	65,75%	65,67%	65,47%	65,47%	65,40%	65,20%	65,26%	77,72%	68,06%	67,11%

**Tabela 32 – Resultados do filtro da imagem RGB para os componentes RGB.**

Simulação	5%	10%	15%	20%	25%	30%	Média
M3V8E5	63,72%	46,41%	34,51%	26,14%	20,62%	16,41%	34,64%
M5SVE11	56,13%	41,01%	30,89%	24,06%	18,76%	15,68%	31,09%
M7V4E11	57,48%	40,72%	29,51%	21,47%	17,06%	13,58%	29,97%
M9V8E11	57,25%	39,74%	27,40%	19,70%	15,15%	11,89%	28,52%
MP3V8E5	67,25%	50,14%	38,25%	29,19%	23,27%	19,18%	37,88%
RFS	69,02%	66,57%	63,81%	61,16%	57,83%	54,87%	62,21%

**Tabela 33 – Média dos resultados do filtro das imagens.**

Simulação	0%	5%	10%	15%	20%	25%	30%	R G	G B	Média
M3V8E5	89,23%	63,87%	46,66%	34,67%	36,06%	32,14%	16,85%	89,47%	88,99%	38,37%
M5SVE11	82,37%	56,44%	40,94%	30,88%	36,76%	32,75%	17,22%	82,91%	81,82%	35,83%
M7V4E11	77,08%	57,43%	40,55%	29,34%	36,09%	32,47%	15,51%	77,66%	76,50%	35,23%
M9V8E11	72,62%	57,25%	39,25%	27,29%	35,22%	31,10%	14,26%	73,27%	71,98%	34,06%
MP3V8E5	91,40%	67,49%	50,10%	38,39%	38,25%	33,92%	19,16%	91,57%	91,22%	41,22%
RFS	71,39%	69,80%	68,13%	66,40%	68,84%	67,21%	62,40%	73,04%	64,81%	67,13%

demais simulações, quando existe a inserção de ruído. Nos componentes que não têm ruído o percentual de acerto é um dos menores. Na imagem sem ruído a mesma situação é comprovada. Porém, com os melhores resultados quando existe ruído torna a média geral da RNA melhor do que qualquer outro tipo de filtro. O filtro de média ponderada com máscara 3x3 foi o que obteve os melhores resultados dentre os filtros comuns.

As máscaras com tamanhos maiores foram as piores na maiorias dos resultados apresentados. A diferença entre os percentuais de ruídos podem ser observados de forma expressiva em alguns casos, mesmo com um percentual de 15% de ruído. O ruído de 30% reduz drasticamente o percentual de acertos na maioria dos casos ou componentes, com exceção das simulações com a RNA.

Nas Tabelas 34, 35, 36 e 37 apresentam-se os percentuais de acerto em uma imagem em escala de cinza com os percentuais de ruído entre 5% e 30%, em cada componente e em todos juntos. Os percentuais de acertos foram encontrados pelas Equações (34) e (35). Na Tabela 38 apresenta as médias encontradas dentre todos os resultados, o percentual de acerto para a imagem sem ruído também é apresentado.

**Tabela 34 – Resultados da segmentação da imagem RGB para o componente R.**

Simulação	5%	10%	15%	20%	25%	30%	Média
M3V8E5	93,72%	91,60%	90,22%	89,27%	88,95%	88,80%	90,43%
M5SVE11	98,44%	98,28%	97,78%	97,35%	96,76%	95,51%	97,36%
M7V4E11	97,63%	97,27%	96,95%	96,64%	96,13%	94,81%	96,57%
M9V8E11	70,22%	71,17%	71,98%	73,09%	74,68%	76,09%	72,87%
MP3V8E5	95,62%	95,42%	95,43%	95,29%	95,33%	95,27%	95,39%
RFS	95,81%	94,81%	93,82%	92,78%	91,81%	90,77%	93,30%
RSFS	98,29%	98,29%	98,28%	98,29%	98,29%	98,24%	98,28%

**Tabela 35 – Resultados da segmentação da imagem RGB para o componente G.**

Simulação	5%	10%	15%	20%	25%	30%	Média
M3V8E5	97,01%	97,04%	96,98%	97,03%	96,96%	97,09%	97,02%
M5SVE11	98,45%	98,47%	98,45%	98,45%	98,49%	98,36%	98,45%
M7V4E11	97,76%	97,79%	97,80%	97,57%	97,49%	97,29%	97,62%
M9V8E11	70,23%	70,34%	70,72%	71,17%	71,54%	72,46%	71,08%
MP3V8E5	95,38%	95,14%	94,77%	94,47%	94,18%	93,72%	94,61%
RFS	96,87%	96,84%	96,85%	96,85%	96,80%	96,80%	96,84%
RSFS	98,31%	98,30%	98,30%	98,29%	98,29%	98,29%	98,30%

Os resultados apresentados nas Tabelas 34, 35, 36, 37 e 38 mostram a robustez e generalização no emprego de RNA para a segmentação de imagens. Em todos os casos, com exceção do número 4 (M9V8E11), obtiveram resultados acima de 90% mesmo com a inserção de ruídos de 30%. Nas melhores simulações a maior diferença entre a segmentação da imagem sem ruído e com ruído de 30% foi menor que 2,5%.

**Tabela 36 – Resultados da segmentação da imagem RGB para o componente B.**

Simulação	5%	10%	15%	20%	25%	30%	Média
M3V8E5	96,98%	97,00%	96,95%	96,99%	97,04%	97,02%	97,00%
M5SVE11	98,26%	98,28%	98,16%	98,11%	98,25%	98,11%	98,20%
M7V4E11	98,14%	97,97%	98,31%	98,38%	98,47%	98,58%	98,31%
M9V8E11	71,84%	76,08%	80,48%	84,98%	88,58%	91,06%	82,17%
MP3V8E5	96,09%	96,22%	96,25%	96,25%	96,26%	96,06%	96,19%
RFS	96,74%	96,63%	96,53%	96,43%	96,31%	96,20%	96,47%
RSFS	98,04%	97,83%	97,62%	97,37%	97,12%	96,91%	97,48%

**Tabela 37 – Resultados da segmentação da imagem RGB para o componente RGB.**

Simulação	5%	10%	15%	20%	25%	30%	Média
M3V8E5	93,86%	92,80%	92,33%	92,09%	91,74%	91,49%	92,38%
M5SVE11	98,17%	97,99%	97,49%	97,27%	96,91%	96,65%	97,42%
M7V4E11	97,51%	97,84%	97,68%	97,88%	97,76%	97,86%	97,76%
M9V8E11	73,66%	80,08%	86,62%	90,45%	91,68%	91,88%	85,73%
MP3V8E5	95,03%	93,88%	93,09%	92,41%	91,98%	91,37%	92,96%
RFS	95,77%	94,78%	93,90%	93,02%	92,19%	91,51%	93,53%
RSFS	98,06%	97,85%	97,73%	97,42%	97,35%	97,18%	97,60%

**Tabela 38 – Média dos resultados da segmentação da imagem RGB.**

Simulação	0%	5%	10%	15%	20%	25%	30%	Média
M3V8E5	96,96%	95,39%	94,61%	94,12%	93,84%	93,67%	93,60%	94,60%
M5SVE11	98,51%	98,33%	98,26%	97,97%	97,80%	97,60%	97,16%	97,95%
M7V4E11	97,85%	97,76%	97,72%	97,68%	97,61%	97,46%	97,14%	97,60%
M9V8E11	69,80%	71,49%	74,42%	77,45%	79,92%	81,62%	82,87%	76,80%
MP3V8E5	95,81%	95,53%	95,16%	94,89%	94,61%	94,43%	94,10%	94,93%
RFS	96,86%	96,30%	95,77%	95,28%	94,77%	94,28%	93,82%	95,30%
RSFS	98,31%	98,17%	98,07%	97,98%	97,84%	97,76%	97,66%	97,97%

A simulação de número 4 (M9V8E11) obtiveram-se os piores resultados provavelmente devido à maior máscara (9x9) dentre todas as outras com filtros lineares. Na coluna sem ruído pode-se observar que é o pior valor encontrado nessa simulação, esse tipo de máscara retorna melhores resultados com índices de ruído maiores, como pode ser verificado nas colunas com 25% e 30%.

A diferença entre os dois melhores resultados, número 2 (M5SVE11) e 7 (RSFS), é muito baixa. A número 2 tem melhores resultados com baixa inserção de ruído e a número 7 ao contrário. A vantagem da número 7 está na construção do hardware, que apenas tem a necessidade do sistema da RNA. A simulação de número 2 tem a necessidade de um módulo extra para fazer o filtro da imagem antes de executar o sistema da RNA. Assim, o melhor caso é a simulação de número 7 (RSFS).

A Figura 70 apresenta o resultado da segmentação de imagem pela RNA com maiores níveis de ruído, ou seja, nas imagens com inserção de ruído em todos os 3 componentes RGB. Os resultados mostram a capacidade adequada da RNA para o objetivo proposto.

A Figura 70-(c) ilustra a segmentação realizada com o treinamento da primeira etapa, número 13 da Tabela 20, sem inserção de ruído. Na Figura 70-(d;f;h;j;l;n) são imagens geradas a partir da segmentação com RNA e treinamento com a inserção de ruído. Na comparação da imagem alvo, Figura 70-(b), com a imagem encontrada pela RNA com e sem ruído, Figura 70-(c;d), a segmentação pela RNA gerou resultados visuais melhores do que o método de Otsu. Desta forma, as imagens desejadas posteriores devem ser geradas de outras formas como por um editor gráfico, e não só pelo método de Otsu.

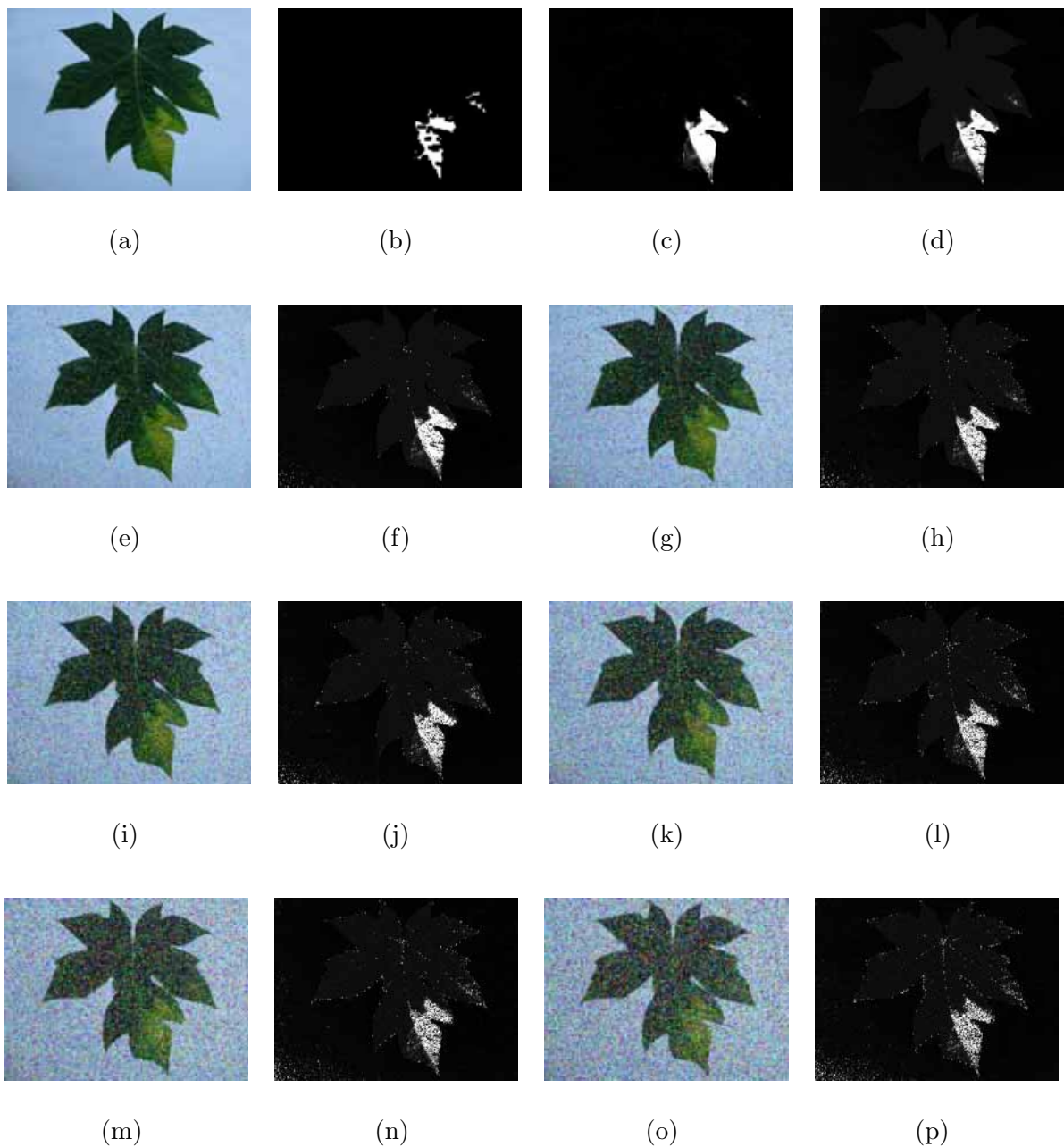
## 6.4 Conclusão deste Capítulo

Neste estudo de caso investigaram-se diversos métodos no processamento digital de imagens na aplicação de filtros, identificação de objetos e segmentação em imagens no domínio de tempo discreto.

A seção 6.1 apresentou a utilização de limiarização global por histograma para a segmentação de imagens. Nessa técnica determina-se um limiar global único com base na diferenciação entre picos do histograma. Este método torna-se ineficaz quando o histograma da imagem não contém os picos bem definidos.

A limiarização utilizando o método de Otsu pode encontrar um limiar ótimo para a imagem, mesmo na inexistência de picos bem definidos no histograma. A utilização de limiares múltiplos pode selecionar diferentes classes dentro de uma imagem por meio de cálculos estatísticos, probabilidades e variância nos níveis de intensidade. Os resultados alcançados com esse método na seção 6.2 evidenciou a sua robustez na segmentação de imagens, especificamente quando existe a variação de níveis de intensidade de cor

Figura 70 – Resultados da segmentação de imagens com a RNA: (a) Imagem original sem ruído; (b) Imagem desejada (alvo); (c) Segmentação de (a) com RNA e treinamento sem ruído (Tabela 20 número 13); (d) Segmentação de (a) e treinamento com ruído; (e) Imagem original com ruído de 5%; (f) Segmentação de (e) com RNA; (g) Ruído de 10%; (h) Segmentação de (g) com RNA; (i) Ruído de 15%; (j) Segmentação de (i) com RNA; (k) Ruído de 20%; (l) Segmentação de (k) com RNA; (m) Ruído de 25%; (n) Segmentação de (m) com RNA; (o) Ruído de 30%; (p) Segmentação de (o) com RNA.



Fonte: Elaboração do próprio autor

mostrando a deficiência nutrientes na folha e na planta. Na Figura 58-(e,f) apresentou-se uma folha da Caferana com baixa variação de cor na folha e o método conseguiu definir a classe a ser segmentada.

Na seção 6.3 foi apresentada a investigação das melhores configurações para redes neurais artificiais, com o objetivo de filtragem e segmentação de imagens quanto a deficiência de nutrientes na folha do Mamão. Diversos parâmetros foram analisados como: taxa de aprendizagem, constante de momento, quantidade de neurônios na camada escondida, tipos e quantidades de entradas, vários tipos de filtros e inserção de ruídos. Uma ampla variação de parâmetros foi analisada e foi definida uma estrutura adequada para a RNA. Principalmente na determinação do tipo e da quantidade de entradas na RNA, a camada de saída, a não utilização de filtros, e os parâmetros de aprendizagem e neurônios na camada escondida devem ser novamente investigados para aplicação alvo do trabalho na cultura da soja.

Todos os parâmetros analisados determinaram que a melhor escolha para a RNA tem a seguinte organização: 3 entradas RGB, 4 neurônios na camada escondida, taxa de aprendizagem de 0,01 e constante de momento de 0,9. A comparação da RNA com outro sistema com duas RNAs, uma com função de filtro e outra de segmentação, mostrou que os resultados alcançados com apenas uma RNA (otimizada) é melhor que com duas para o tratamento de ruído impulsivo (Sal e pimenta). Além disso, a RNA otimizada obteve melhores resultados que com filtros lineares comuns por média simples e ponderada. O método de Otsu foi escolhido para gerar as imagens utilizadas no treinamento da RNA, mas a RNA obteve melhores resultados visualmente do que o método de Otsu. A RNA mesmo com poucos neurônios pode obter resultados satisfatórios para a segmentação de imagens no reconhecimento de deficiência de nutrientes. No Capítulo 7 investiga-se a segmentação com RNA em comparação com o método de Otsu para a cultura do algodão.

## 7 ANÁLISE NA CULTURA DO ALGODÃO

Neste capítulo utiliza-se o sistema da RNA descrito em alto nível de abstração com casos reais no cultivo do algodão. O sistema modelado em alto nível de abstração foi apresentado no Capítulo 6, diferentemente, na abordagem deste capítulo são realizados experimentos reais e com comprovação de deficiência de nutrientes.

Neste capítulo são analisados dois experimentos com o algodão e cultivar FMT 709 <sup>1</sup>. Os experimentos no cultivo do algodão são implementados com uma RNA e comparados ao método de Otsu. No primeiro experimento, apresentado na seção 7.1, verifica-se o comportamento da cultura do algodão com deficiência do macronutriente de fósforo. No segundo experimento, apresentado na seção 7.2, contém a deficiência do macronutriente K na cultura do algodão. A organização e os parâmetros da RNA nos experimentos são baseados no melhor resultado adquirido na seção 6.3 e investigados alguns aspectos neste caso. Para comprovar a análise da seção 6.3 foram realizadas diversas investigações para possibilitar a análise de qualidade nas segmentações realizadas a partir da RNA.

Os dois experimentos analisados são da safra 2012/13, esse foi o último ano da safra que iniciou em 2009/10. A estação experimental está situada em Itiquira-MT, a 80 km de Rondonópolis, às margens da BR-163 (-17°5'52.03", -54°45'34.55"). A estação experimental pertence à Fundação MT que disponibilizou as informações e imagens vinculadas à mesma área de cultivo. O solo é um Latossolo Vermelho distrófico e muito argiloso (em torno de 65% de argila) em todos os ensaios nos dois experimentos.

### 7.1 Experimento 1: Análise de Deficiência do Fósforo

No primeiro experimento os ensaios com o algodão foram realizados com diferentes variações de dois macronutrientes, nitrogênio (N) e fósforo (P). O nitrogênio variou entre doses de 0, 20, 45 e 70 Kg/ha, e o fósforo em 0, 50, 100 Kg/ha de P<sub>2</sub>O<sub>5</sub><sup>2</sup>. Os ensaios

---

<sup>1</sup>Tipo de cultivar do algodão modificado geneticamente pelo programa de Melhoramento Genético de Algodão da Fundação de Apoio à Pesquisa Agropecuária de Mato Grosso, Fundação MT. Dependendo do tipo de cultivar pode-se modificar diversas características no plantio como a forma e coloração das folhas, região ou estado de atuação, tipo de solo recomendado e tempo de desenvolvimento da planta.

<sup>2</sup>Tipo de fertilizante utilizado para adubação por fósforo no plantio. O tipo de composto químico utilizado é uma característica muito importante para se obter os mesmos resultados na adubação do

correspondem a 36 variações, em 9 tipos de tratamentos com 4 amostras em cada um e 4 diferentes blocos analisados. Este experimento está localizado nas mesmas áreas e estação experimental, ao lado do segundo experimento.

A análise laboratorial do primeiro experimento é apresentado na Tabela 39 e pode ser relacionado à cor e à deficiência observada nas folhas do algodão. A aplicação dos tratamentos sem o macronutriente P apresentaram resultados de deficiências nas folhas do algodão. Os três primeiros tratamentos não contêm adubação de fósforo e as suas variações são as mais baixas dentre todos os outros. As imagens selecionadas foram com o grau mais alto de severidade analisado, ou seja, sem adubação de fósforo (Tratamento Nº 1, 2 e 3 na Tabela 39). O déficit do macronutriente P apresentou em suas folhas a pigmentação roxa. Mesmo com uma pequena diferença em relação aos demais, foram nesses tratamentos que ocorreram as diferenças de coloração nas folhas do algodão. As imagens contendo as deficiências do nutriente K estão apresentadas nas Figuras 73-(a;e;i).

**Tabela 39 – Ensaio com 9 tipos de tratamentos e média entre 4 amostras de 4 blocos diferentes.**

Cutura	Tratamento (Kg/ha)			Resultados Foliare (g/kg)					
	Nº	N	K	N	P	K	Ca	Mg	S
Algodão	1	0	0	26,95	1,65	6,70	34,40	6,60	2,80
Algodão	2	20	0	26,60	1,45	6,10	31,65	5,90	3,10
Algodão	3	70	0	27,30	1,40	7,80	25,50	5,35	2,05
Algodão	4	20	50	29,75	1,90	6,40	55,05	6,75	2,85
Algodão	5	45	50	29,75	1,95	6,00	43,05	6,80	3,40
Algodão	6	70	50	31,15	1,75	6,70	29,90	6,00	3,25
Algodão	7	0	100	26,60	1,70	6,80	48,45	6,30	2,65
Algodão	8	20	100	27,30	1,70	6,60	36,45	6,25	2,70
Algodão	9	70	100	31,50	1,80	7,70	30,10	6,15	3,25

Os menores valores de P foram adquiridos nesses tratamentos, dentre todos os tratamentos analisados, mesmo com a aplicação de 20 e 70 Kg/ha de nitrogênio. Os valores ficaram em 1,4, 1,45 e 1,65 definidos na faixa “muito baixo”. Os outros tratamentos ficaram acima de 1,7 e mais próximos a faixa “baixo”. O recomendado para o algodão, em solos acima de 65% de argila é de 4,1 a 6,0 (CARVALHO; FERREIRA, 2006).

Os casos com deficiência estão com valores considerados “muito baixo”, já os tratamentos no intervalo “baixo”, próximos a 2,0, não apresentaram deficiências de nutrientes nas folhas. Na Tabela 39 mostram-se os resultados da análise laboratorial completa em todos os tratamentos com os macronutrientes: Nitrogênio (N), Fósforo (P), Potássio (K), Cálcio (Ca), Manganês (Mg) e Enxofre (S).

Nos dois métodos, RNA e Otsu, foram utilizadas as mesmas imagens de entrada. Em todas as imagens foram retirados os planos de fundo (Solo) com editor gráfico. Todas as

---

plantio.

imagens foram treinadas na RNA de acordo com sua respectiva imagem segmentada pelo editor gráfico. Essas imagens foram utilizadas como novas amostras para o treinamento da RNA. Em todas as etapas, a RNA foi treinada com branco (255) no fundo e área segmentada preta ou próximo à escala em nível de cinza 0. O método de Otsu usa uma imagem em escala de cinza para análise dos resultados com deficiência de fósforo, diferentemente da RNA, mais detalhes sobre essas folhas são discutidas na seção 7.1.1.3.

As análises foram realizadas a partir de 6 imagens, 3 sem deficiência e 3 com deficiência de fósforo. Com o editor gráfico foram geradas mais 6 imagens sem o plano de fundo e 6 imagens desejadas, ou seja, com a segmentação da parte com deficiência de nutriente. Assim, as entradas da RNA são baseadas em 12 imagens com e sem plano de fundo e a cada duas imagens corresponde a mesma imagem desejada ou segmentada.

Neste experimento foram analisados diversos parâmetros de qualidade e as variações nos principais parâmetros da RNA para definir com propriedade a melhor organização da RNA para imagens reais de entrada e com deficiência comprovada no plantio. A RNA e o método de Otsu foram comparados para determinação da incapacidade do método de Otsu para determinadas situações, como visto na seção 6.3.

As imagens foram adquiridas por uma câmera digital Sony DSC-W530 de 14,1 MegaPíxel e redefinidas em uma resolução de 320x300 com editor gráfico. Na validação cruzada utilizaram-se 12 imagens e o treinamento da RNA com 2% de cada imagem, ou seja, 1920 píxeis para as primeiras análises na seção 7.1.1.2. Os outros 98% foram definidos para validação do teste da RNA e análise dos resultados. O percentual baixo definido em cada imagem tem o objetivo de obter as melhores combinações para a RNA de Taxa de Aprendizagem (TA), Constante de Momento (CM) e Quantidade de Neurônios na Camada Escondida (QNCE).

### ***7.1.1 Segmentação com RNA e deficiência de fósforo (P)***

Diversos treinamentos foram realizados para determinar parâmetros adequados à RNA, visando uma estrutura mais otimizada. A taxa de aprendizagem e constante de momento foram alternadas em (0,01, 0,05, 0,1) e (0,3, 0,5, 0,7 e 0,9), respectivamente, conforme limites recomendados e simulações iniciais da seção 6.3 (HAYKIN, 2001).

A quantidade de camadas da RNA são três, sendo uma escondida, e a quantidade de neurônios nessa camada variou em 3, 4, 5, 6, 7, 8, 12 e 16. A função de ativação (Tangente Hiperbólica) é utilizada em todos os neurônios da RNA. A quantidade de épocas tem ampla variação de quantidade para definição dos parâmetros da RNA e validação dos resultados com as melhores combinações encontradas.

Os resultados deste trabalho mostraram que a segmentação com a RNA MLP

é adequada para identificação de deficiência do macronutriente P. Os resultados são apresentados por meio de diversos parâmetros para análise de qualidade da segmentação e as imagens geradas pelos métodos.

### 7.1.1.1 Parâmetros de Qualidade

Para analisar os resultados encontrados nas segmentações com maior qualidade foram definidos quatro parâmetros: o PHM (*Percentage of Hit-or-Miss*), o erro médio quadrático (MSE), PSNR (*Peak Signal-to-Noise Ratio*) e o Índice de Qualidade (QI).

O PHM mede a precisão de cada píxel com base nos níveis de cinza na saída da RNA. A avaliação é realizada pela quantidade de acertos em cada simulação, conforme intervalo definido nas Equações (38) e (39). O cálculo é baseado no incremento dos acertos e erros conforme um limite adequado de nível de cinza para o resultado. As Equações foram redefinidas a partir das Equações (34) e (35).

A camada de saída da RNA contém um neurônio para determinar o padrão de saída, ou seja, a imagem segmentada. Portanto, a imagem gerada (segmentada) na saída da RNA e do método de Otsu estão em escala de nível de cinza. Na Equação (38), o incremento da variável acerto ( $H++$ ) é realizado conforme a relação a saída  $Y$  da RNA no intervalo  $L$  de níveis de cinza possíveis.  $D$  é o valor desejado do píxel para comparação da saída da RNA. Os valores de  $Lz$ ,  $Lx$  e  $Ly$  são encontrados por  $Lz = (1/4) * LMax$ ,  $Lx = LMin + 1/4$  e  $Ly = LMax - 1/4$ . Caso nenhuma das três possibilidades de incremento de  $H$  ocorra, é incrementado o valor de erro  $M$ . A precisão em porcentagem é obtida por meio da Equação (39).

$$H|M = \begin{cases} H+ = 1 & \text{se } D - Lz \leq Y \leq D + Lz, \quad \forall Y \in L | Lx < L < Ly \\ H+ = 1 & \text{se } Lmin \leq Y \leq D + Lz, \quad \forall Y \in L | Lmin < L < Ly \\ H+ = 1 & \text{se } D - Lz \leq Y \leq Lmax, \quad \forall Y \in L | Ly < L < Lmax \\ M+ = 1 & \text{caso contrário} \end{cases} \quad (38)$$

$$PHM = 100 \times \left[ \frac{\sum_{i=1}^{Pn} H}{\sum_{i=1}^{Pn} H + \sum_{i=1}^{Pn} M} \right] \quad (39)$$

Dois parâmetros foram avaliados com base no erro dos resultados na comparação das imagens de saída e desejada. O MSE é adquirido a partir da Equação (40), onde  $Pn$  é o número de píxeis da imagem utilizados na validação dos dados de teste. Como a saída

da RNA é normalizada, realizou-se o produto da resolução da imagem com MSE, para melhor análise dos valores em relação aos níveis de cinza. O PSNR traz a medida do pico de sinal em decibéis (dB) pela Equação (41). Essa medida está relacionada com a energia da imagem e o ruído, de forma semelhante ao MSE (JUCA; MELLO, 2011). Com valores baixos de PSNR (0 a 10) a imagem contém altos valores de ruído.

$$MSE = \frac{1}{P_n} \left( \sum (Y_i - D_i) \right) \quad (40)$$

$$PSNR = 10 \times \log_{10} \left( \frac{MSE}{S^2} \right) \quad (41)$$

O quarto parâmetro, o QI, está relacionado à perda de correlação, à distorção de luminância e ao contraste entre as duas imagens avaliadas, saída e desejada. A correlação entre as imagens é calculada pela covariância entre as imagens por meio de análise estatística dos dados, conforme Equação (42) (WANG; BOVIK, 2002). O índice de qualidade tem valores no intervalo [-1:1], o valor é 1 quando as imagens são iguais.

$$QI = \frac{4\sigma_{YD}\bar{Y}\bar{D}}{(\sigma_Y^2 + \sigma_D^2) \left[ (\bar{Y})^2 + (\bar{D})^2 \right]} \quad (42)$$

Para facilitar a análise dos resultados foi definido um parâmetro de fator de qualidade (FQ) que calcula a média ponderada entre os quatro parâmetros definidos, conforme Equação (43). O valor de FQ está no intervalo [0:10], sendo 10 o melhor resultado. Os Valores de PSNR e QI foram normalizados para manter o mesmo intervalo dos demais. Os parâmetros relacionados com os erros (MSE e PSNR) não são adequados para análise dos resultados visuais da segmentação, mas para a análise de energia na imagem (WANG; BOVIK, 2002; JUCA; MELLO, 2011). Assim, os parâmetros relacionados ao erro foram atribuídos valores baixos na ponderação para não comprometer a análise visual, principal objetivo do trabalho.

$$FQ = [(PHM/100) \times 3,5] + [(1 - MSE) \times 1,5] + (PSNR \times 1,5) + (QI \times 3,5) \quad (43)$$

Para obter a melhor combinação da organização da RNA de forma otimizada, foram realizadas 27 simulações. Para cada cenário ou treinamento foram executados o modo *feedforward* para as 12 imagens e a média dos resultados são apresentadas nas Tabelas 40

41 e 42. As variações estão nos neurônios da camada escondida, na taxa de aprendizagem, na constante de momento e no percentual da imagem para o treinamento. Nas Tabelas 40, 41, 42 e 43 apresentam-se os quatro parâmetros de qualidade (PHM, MSE, PSNR e QI) e o FQ.

### 7.1.1.2 Análise na Qualidade das Segmentações

No experimento com a deficiência de fósforo foram definidas quatro etapas. A primeira etapa executa 12 treinamentos com o objetivo de encontrar os melhores valores de taxa de aprendizagem e constante de momento. A segunda etapa visa encontrar a melhor quantidade de neurônios na camada escondida. A terceira avalia os resultados com a variação do percentual do conjunto de imagens utilizadas para o treinamento. Por último, a validação dos resultados com a melhor combinação em comparação com o método de Otsu.

Na primeira etapa são realizadas 12 simulações para definir a melhor combinação com a variação da TA e a CM, conforme Tabela 40. Pelo FQ das segmentações, pode-se observar que a maior influência nos resultados estão relacionadas a TA. Os melhores resultados estão com o menor valor de TA em todos os parâmetros, com exceção da simulação 6. Com valores maiores de TA obtém-se uma variação maior entre os resultados em comparação com os melhores. O parâmetro média corresponde ao fator de qualidade, Equação (43), mas com pesos iguais (2,5) em todos os quatro parâmetros. A Tabela 40 apresenta os resultados com todos os parâmetros de qualidade com 700 épocas. Nesse primeiro estágio foram executados os treinamentos com 200, 500 e 700 épocas e os melhores valores encontrados foram com 700.

**Tabela 40 – Primeiro estágio para encontrar valores adequados de TA e CM.**

Simulações	TA	CM	PHM(%)	MSE	PSNR	QI	Média	FQ
1	0,01	0,3	88,59	10,74	63,32	0,900	7,90	8,41
2	0,05	0,3	65,37	71,09	56,85	0,645	6,02	6,56
3	0,1	0,3	63,74	72,04	56,77	0,643	5,96	6,49
4	0,01	0,5	88,93	11,30	63,20	0,877	7,86	8,37
5	0,05	0,5	65,51	71,47	56,84	0,643	6,02	6,56
6	0,1	0,5	83,44	10,33	63,71	0,901	7,80	8,25
7	0,01	0,7	88,01	11,13	63,04	0,920	7,89	8,41
8	0,05	0,7	64,40	71,68	56,83	0,633	5,97	6,51
9	0,1	0,7	63,68	72,37	56,74	0,643	5,96	6,49
10	0,01	0,9	88,38	11,66	62,93	0,889	7,85	8,36
11	0,05	0,9	63,98	71,75	56,82	0,650	5,98	6,52
12	0,1	0,9	63,67	72,48	56,75	0,647	5,96	6,50

Na verificação da média e do fator de qualidade, as duas melhores simulações foram a 1 e 7 com resultados semelhantes. Como na primeira simulação obtiveram-se os melhores

resultados na Tabela 40 em comparação aos dois resultados, média e FQ, a primeira simulação foi escolhida para servir de parâmetro de análise na seção 7.2 e no Capítulo 8.

A partir da simulação 1, outras 8 simulações foram realizadas alterando a quantidade de neurônios na camada escondida. A quantidade de neurônios na camada escondida depende do tipo de conjunto de dados e da estrutura da rede neural artificial. Na literatura existe divergência entre os autores sobre a quantidade ótima desses neurônios (HAYKIN, 2001; KARSOLIYA, 2012; SHAHAMIRI; SALIM, 2014; SHEELA; DEEPA, 2013). Considerando a literatura, definiu-se valores conforme a seguinte regra  $Entradas \leq QNCE \leq 2 \times Entradas$ , inicialmente o valor máximo foi ultrapassado para analisar os resultados e as dúvidas em questão. Conforme Haykin (2001) define-se o valor mínimo e aumentam-se gradativamente os valores de QNCE, geralmente com quantidades menores obtêm-se melhores valores. A segunda etapa corresponde a avaliação da QNCE, procurou-se investigar os menores valores de QNCE para obter uma RNA mais otimizada e adequada para implementação em hardware.

Existe grande semelhança nos resultados entre todas as simulações analisadas utilizando-se a Tabela 41. Os resultados mostram os parâmetros de média e FQ com diferenças máximas de 0,1, se retirar o melhor e o pior resultado, essa diferença diminui para 0,05. Nas simulações, 7 e 8, com as maiores quantidades de neurônios obtiveram-se os piores resultados. Os resultados alcançados são bem semelhantes, assim escolheu-se o menor QNCE analisado, correspondente a simulação 1, para obter o sistema da RNA mais otimizado no transporte ao FPGA. Nos treinamentos foram utilizadas 200 épocas para alcançar os resultados da Tabela 41.

**Tabela 41 – Segundo estágio com alteração na quantidade de neurônios na camada escondida.**

Simulações	QNCE	PHM(%)	MSE	PSNR	QI	Média	FQ
1	3	87,67	11,77	62,63	0,886	7,80	8,32
2	4	88,33	12,22	63,03	0,899	7,86	8,38
3	5	87,57	12,46	62,72	0,896	7,81	8,33
4	6	87,62	12,69	62,63	0,895	7,81	8,33
5	7	87,58	12,99	62,50	0,906	7,81	8,34
6	8	87,86	12,84	62,52	0,897	7,81	8,34
7	12	87,45	13,82	62,07	0,888	7,75	8,29
8	16	87,13	13,60	62,08	0,887	7,74	8,27

O terceiro estágio avalia a influência da porcentagem de imagem no treinamento. Segundo Haykin (2001) as porcentagens de dados para o treinamento podem chegar a 70%. As simulações foram realizadas com porcentagens abaixo do recomendado, mesmo assim obtiveram-se resultados satisfatórios. As 6 simulações e as diferenças nos resultados das segmentações estão relacionadas na Tabela 42. Na simulação 2, com 2% das imagens para o treinamento, foram obtidos os melhores resultados, e na primeira simulação, com

1%, o pior resultado.

**Tabela 42 – Terceiro estágio com alteração na porcentagem de imagem utilizada para o treinamento.**

Simulações	Porcentagem(%)	PHM(%)	MSE	PSNR	QI	Média	FQ
1	1	71,93	29,95	57,99	0,836	6,89	7,41
2	2	87,67	11,77	62,63	0,886	7,80	8,32
3	4	86,83	12,93	62,02	0,880	7,73	8,25
4	8	87,93	14,24	61,87	0,836	7,68	8,20
5	16	89,06	10,09	63,39	0,747	7,73	8,16
6	32	90,88	7,94	64,70	0,491	7,55	7,84

O último estágio avalia a melhor combinação em comparação com a técnica tradicional de segmentação em PDI, o método de Otsu. A cada três simulações da Tabela 43 corresponde, respectivamente, as imagens com e sem plano de fundo. Os resultados na RNA são semelhantes na segmentação das imagens, mesmo com diferentes planos de fundo.

A Tabela 43 apresenta os resultados com o mesmo conjunto de imagens para a RNA e o algoritmo de Otsu. Na comparação entre os métodos na Tabela 43, a RNA alcança resultados melhores que o método de Otsu nas médias e FQ. A razão está diretamente relacionada com o parâmetro PHM, que permite a análise da precisão em cada pixel. O parâmetro QI nos dois métodos obtiveram médias similares.

**Tabela 43 – Quarto estágio, comparação dos resultados com todas as imagens entre a RNA e o método de Otsu.**

Imagens	PHM(%)		MSE		PSNR		QI		Média		FQ	
	RNA	Otsu	RNA	Otsu	RNA	Otsu	RNA	Otsu	RNA	Otsu	RNA	Otsu
1	92,42	91,37	5,84	10,48	64,57	62,02	0,982	0,916	8,22	7,91	8,76	8,49
2	86,89	68,61	12,83	20,50	61,15	59,11	0,947	0,840	7,76	6,98	8,34	7,40
3	95,04	85,16	3,42	14,89	66,89	60,50	0,992	0,912	8,46	7,62	8,97	8,19
4	93,40	71,75	4,92	17,68	65,31	59,76	0,985	0,853	8,30	7,14	8,83	7,57
5	87,90	87,39	11,80	11,86	61,51	61,49	0,953	0,909	7,83	7,76	8,41	8,31
6	96,46	71,11	2,32	14,08	68,58	60,74	0,995	0,870	8,61	7,24	9,09	7,63
7	87,51	71,54	8,56	19,64	62,91	59,30	0,890	0,872	7,85	7,11	8,35	7,57
8	77,10	85,61	28,91	7,28	57,62	63,61	0,727	0,917	6,88	7,89	7,39	8,37
9	83,87	68,84	13,84	19,86	60,82	59,25	0,777	0,876	7,45	7,04	7,92	7,48
10	88,33	81,10	7,85	9,11	63,28	62,64	0,885	0,907	7,90	7,69	8,39	8,14
11	78,17	59,55	28,09	22,91	57,74	58,63	0,719	0,853	6,91	6,71	7,43	7,07
12	84,95	82,11	12,90	7,27	61,13	63,61	0,778	0,913	7,50	7,80	7,98	8,24
Média	87,67	77,01	11,77	14,63	62,63	60,89	0,89	0,89	7,74	7,39	8,28	7,86

A combinação escolhida entre todas as simulações corresponde à seguinte organização:  $QNCE = 3$ ,  $TA = 0,01$  e  $CM = 0,3$ . Essa organização foi empregada nos resultados das Tabelas 42 e 43, e na seção 7.2 e no sistema descrito no Capítulo 8.

### 7.1.1.3 *Análise Visual da Segmentação de imagens*

Os resultados visuais na segmentação das imagens sem deficiência deveriam gerar imagens totalmente brancas ou com apenas as bordas. As imagens apresentadas nas Figuras 71, 72, 73 e 74 mostram que a segmentação pelo método de Otsu é muito deficiente em relação a imagem ideal. Somente nas imagens de entrada sem plano de fundo e com deficiência de fósforo, Figura 73, o método de Otsu alcançou bons resultados. Para obter esses resultados houve a necessidade de inversão das classes da segmentação, mas preservou a forma de aquisição da medida de separabilidade com base na variância entre as classes.

A RNA alcançou resultados visuais adequados para a segmentação da imagem com deficiência do macronutriente P em todas as imagens. A RNA otimizada com poucos neurônios pode obter segmentação de imagem adequada mesmo com plano de fundo.

Todas as etapas e seus respectivos treinamentos são executados de forma independente e com o mesmo conjunto de imagens. Para cada tipo de imagem são gerados três tipos de segmentações: segmentação com o método de Otsu, segmentação com a RNA e segmentação com editor gráfico. As Figuras 71, 72, 73 e 74 apresentam 54 imagens, divididas em dois grupos de 12 imagens e dois grupos de 15 imagens.

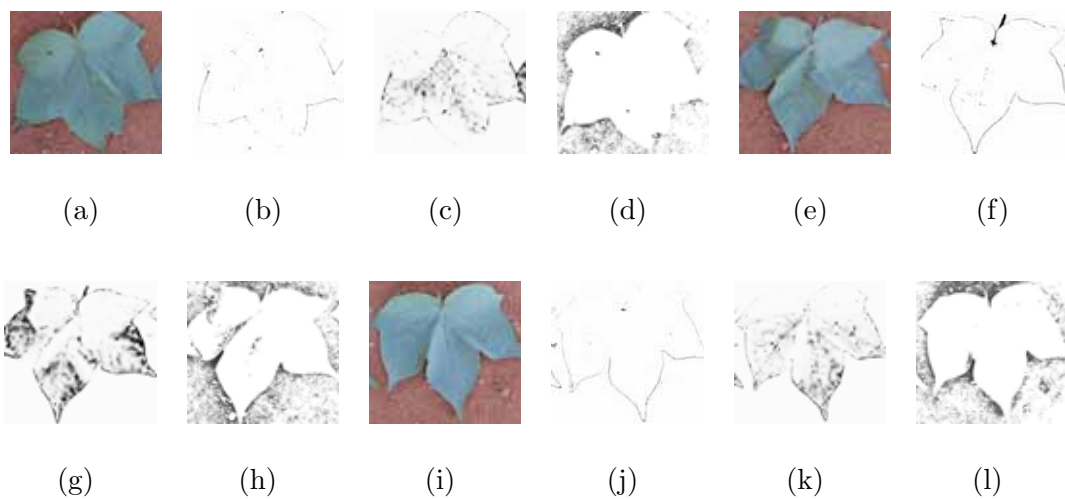
Os dois primeiros grupos contêm imagens sem deficiência, Figuras 71 e 72, e os dois outros grupos com deficiência severa de fósforo, Figuras 73 e 74. As Figuras 71 e 72 contêm 3 amostras de imagens e cada amostra tem uma imagem original e três segmentações, editor gráfico (ideal), ANN e Otsu.

Nas Figuras 73 e 74 tem três amostras de imagens, mas tem duas segmentações de imagens com editor gráfico (ideal), uma para ANN e outra para o método de Otsu, para analisar de forma adequada as segmentações.

As segmentações estão relacionadas nas Figuras 71 e 72 em: (a;e;i) imagem original; (b;f;j) Segmentação com editor gráfico (ideal para os dois tipos de métodos de segmentações); (c;g;k) Segmentação com a RNA e sua melhor combinação; (d;h;i) Segmentação com o método de Otsu. Nas Figuras 73 e 74 contém a adição de mais uma coluna antes da segmentação pelo método de Otsu, que corresponde à segmentação com editor gráfico (ideal) para esse método. O ideal para a RNA continuou com o seu item anterior.

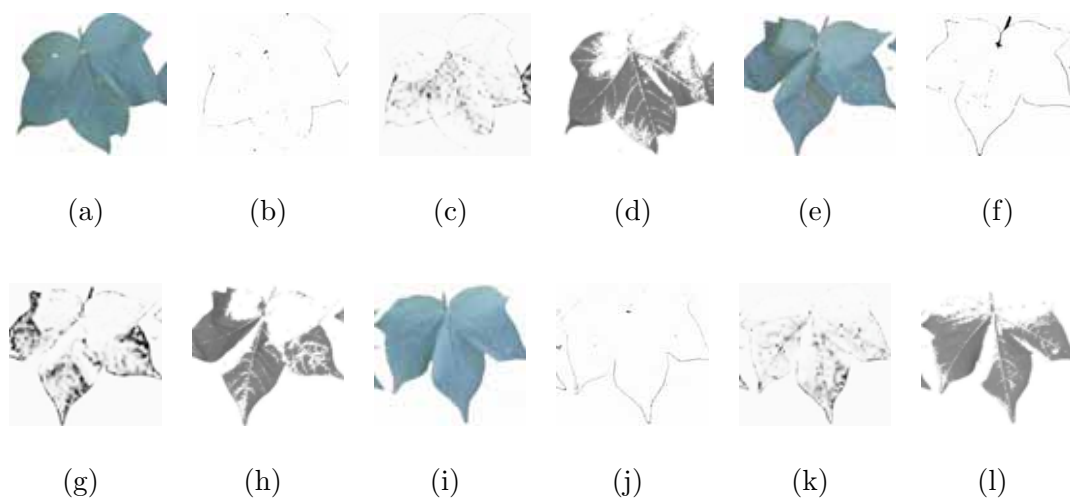
Nos resultados apresentados nas Figuras 72 e 73 mostram a segmentação com o algoritmo de Otsu inadequada, quando comparado ao ideal (segmentação com editor gráfico). A segmentação com RNA apresenta resultados melhores que o algoritmo de Otsu e próximo aos ideais (segmentação com editor gráfico), ou seja, as regiões da folha com coloração roxa definida pelo agrônomo.

**Figura 71 – Imagens sem sintomas de deficiência de fósforo (P) e com plano de fundo.**



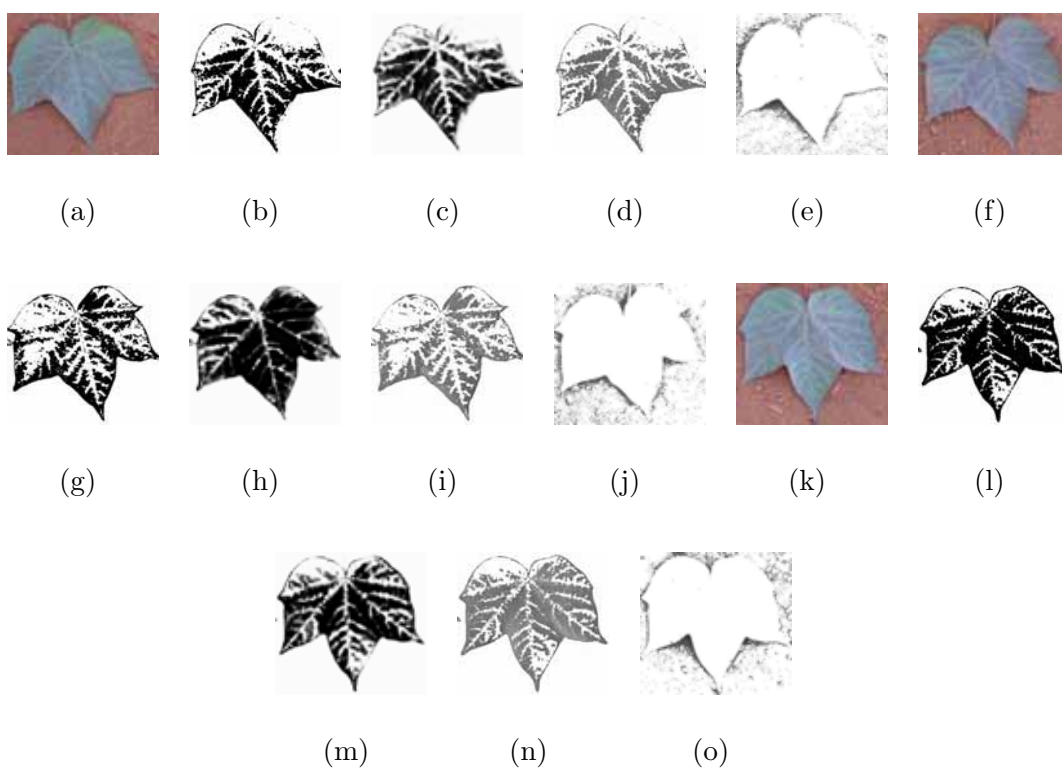
Fonte: Elaboração do próprio autor

**Figura 72 – Imagens sem sintomas de deficiência de fósforo (P) e sem plano de fundo.**



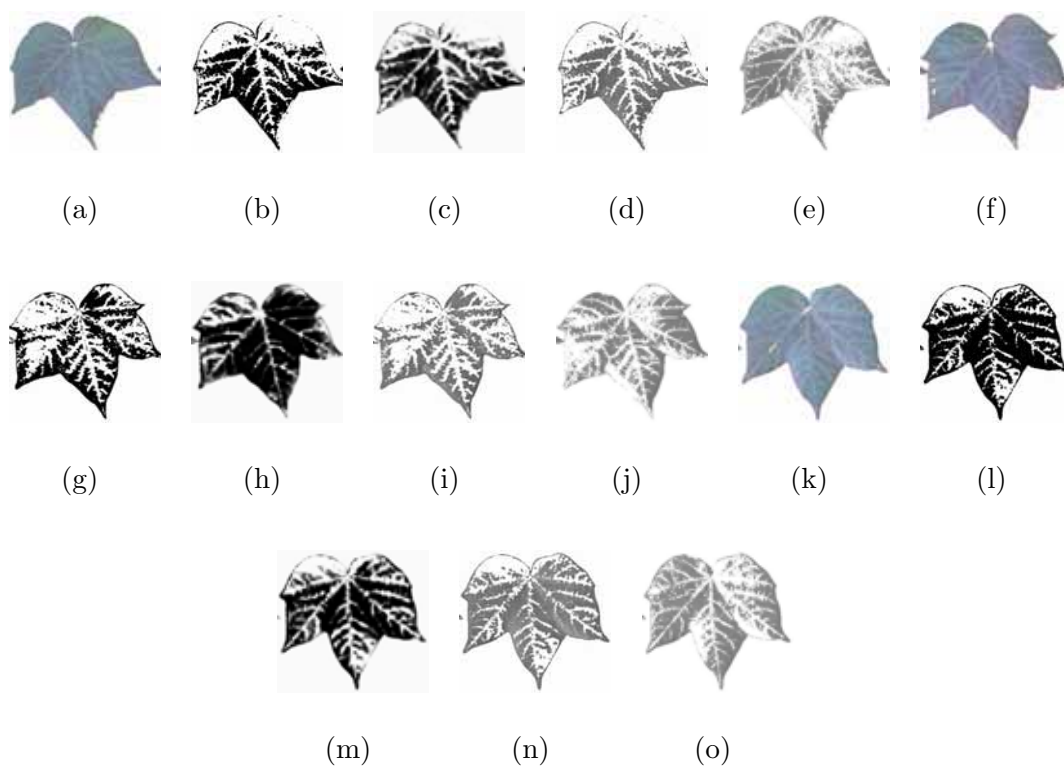
Fonte: Elaboração do próprio autor

Figura 73 – Imagens com sintomas de deficiência de fósforo (P) e com plano de fundo.



Fonte: Elaboração do próprio autor

Figura 74 – Imagens com sintomas de deficiência de fósforo (P) e sem plano de fundo.



Fonte: Elaboração do próprio autor

## 7.2 Experimento 2: Análise de Deficiência do Potássio

No segundo experimento o conjunto de ensaios tem a variação de nitrogênio (N) e potássio (K). Os ensaios analisados foram de 82 variações com 20 tipos de tratamentos divididos em 4 blocos. No N foram utilizadas doses de 0, 30, 50, 60, 70, 90, 110 Kg/ha e no K doses de 0, 40, 80, 120 e 160. A Tabela 44 apresenta a média dos resultados da análise laboratorial da foliar do algodão.

Em cada tratamento foram coletadas 4 amostras uma de cada bloco, a média aritmética simples foi realizada entre as amostras para melhor apresentação na Tabela 44. Os dois tratamentos com menor valor de K são os números 1 e 4. No primeiro, as quatro amostras variaram em 12,0, 13,0, 13,8 e 14,6, com valores abaixo de todos os outros tratamentos. Já, o tratamento número 4 contém valores em 12,0, 12,2, 15,2 e 18,0, ou seja, teve duas amostras irregulares, por isso tornou a média do tratamento baixa.

O único tratamento que apresentou resultados da deficiência nas folhas do algodão foi o número 1, que contém os menores valores e média. Esse tratamento é o único que não adicionou-se adubação dos macronutrientes N e K, conforme terceira e quarta coluna da Tabela 44. As imagens coletadas de deficiência do nutriente K do segundo experimento foram adquiridas a partir do primeiro tratamento.

**Tabela 44 – Ensaios com 20 tratamentos e média entre 4 amostras de 4 blocos diferentes.**

Cutura	Tratamento(Kg/ha)			Resultados Foliares (g/kg)					
	Nº	N	K	N	P	K	Ca	Mg	S
Algodão	1	0	0	38,50	2,75	13,40	28,88	7,13	3,73
Algodão	2	50	40	36,40	2,65	16,90	27,23	6,60	4,04
Algodão	3	70	40	37,45	2,95	16,60	28,58	5,70	3,42
Algodão	4	90	40	37,45	3,10	13,70	28,80	6,75	3,64
Algodão	5	110	40	39,20	2,90	15,40	27,75	6,68	3,88
Algodão	6	50	80	38,15	2,95	21,20	24,90	5,93	4,26
Algodão	7	70	80	37,80	2,95	20,40	27,23	6,38	3,52
Algodão	8	90	80	37,80	2,85	19,40	25,20	6,45	3,87
Algodão	9	110	80	37,80	2,80	17,40	27,30	5,85	3,37
Algodão	10	50	120	36,40	2,65	21,90	27,75	5,70	3,47
Algodão	11	70	120	38,50	2,95	22,10	24,08	5,63	4,27
Algodão	12	90	120	39,20	3,00	22,10	26,93	6,30	3,70
Algodão	13	110	120	37,80	3,15	21,00	26,93	6,00	4,35
Algodão	14	50	160	39,20	2,95	21,50	26,33	6,00	3,49
Algodão	15	70	160	39,20	2,65	23,30	25,95	6,08	3,48
Algodão	16	90	160	41,30	3,10	26,40	26,48	5,40	4,08
Algodão	17	110	160	38,50	3,25	28,60	24,90	5,70	4,02
Algodão	18	30	120	37,10	3,10	22,40	26,03	5,78	3,86
Algodão	19	60	120	36,05	3,00	19,90	27,38	6,60	3,66
Algodão	20	90	120	37,10	2,70	19,00	27,30	6,75	3,44

A deficiência de K tem variação específica nas folhas do algodão. Nas Figuras

79-(a;e;i;m;q;u) e 80-(a;e) ilustram as folhas com variação de amarelo, passando pelo alaranjado, e em alguns casos com uma pigmentação mais escura ou semelhante ao vermelho. Com incidência da luz solar, o solo e as folhas com deficiência mais severa são semelhantes, o que causou problemas na segmentação.

### **7.2.1 Segmentação com RNA e deficiência de potássio (K)**

O experimento com a cultura do algodão e deficiência do macronutriente K foi analisado por meio de um total de 16 imagens, 9 sem deficiência e 7 com deficiência do macronutriente K. Com editor gráfico, em todas as imagens foram retiradas o plano de fundo (Solo) para servir como uma nova amostra para o treinamento da RNA. Cada imagem original contém sua respectiva imagem sem o plano de fundo, com isso foram analisadas um total de 32 imagens. As imagens foram adquiridas no mesmo local e equipamento do experimento 1 na seção 7.1.

A RNA utilizada na segmentação das imagens do segundo experimento é a melhor simulação do primeiro experimento, seção 7.1. A RNA consiste de uma estrutura 3-3-1, ou seja, três neurônios na camada de entrada, três na camada escondida e 1 na camada de saída. A taxa de aprendizagem é de 0,01 e a constante de momento de 0,3.

No experimento com a deficiência de K foram realizados diversos treinamentos: Imagens sem plano de fundo e Algoritmo de Otsu, Imagens com plano de fundo e Algoritmo de Otsu e Imagens com plano de fundo e editor gráfico. Os resultados apresentados são com as imagens desejadas geradas por editor gráfico com e sem plano de fundo, devido à grande dificuldade de se obter imagens desejadas de qualidade pelo método de Otsu.

O treinamento da RNA foi realizado com 200 e 500 épocas e todas as 32 imagens com e sem plano de fundo. As mesmas imagens desejadas foram utilizadas para as imagens com e sem plano de fundo no treinamento. Os resultados foram divididos em três partes, duas sem deficiência e uma com deficiência de K. Nas Tabelas 45, 46 e 47 apresentam-se dados sobre todas as 32 imagens analisadas. Todas as imagens foram avaliadas com base nos parâmetros de qualidade descritos na seção 7.1.1.1.

#### **7.2.1.1 Resultados**

A primeira etapa corresponde às imagens sem deficiência de K, com e sem plano de fundo e baixa iluminação. Os valores alcançados em todos os parâmetros de qualidade nas 10 primeiras imagens são apresentados na Tabela 45. Os 5 primeiros resultados correspondem às imagens com plano de fundo e as outras 5 com imagens sem o plano

de fundo. As diferenças nos resultados entre as imagens com e sem plano de fundo são evidentes, o FQ nas cinco primeiras estão entre 5 e 6, já nas cinco últimas os valores estão em 9. Na coluna média, percebe-se a RNA com piores resultados nas imagens com plano de fundo e melhores resultados sem plano de fundo, e o método de Otsu tem essa relação invertida. Essa relação ocorre em todos os resultados das Tabelas 45, 46 e 47.

A semelhança do solo com a deficiência nesse tipo de cultura dificultou o aprendizado da RNA. Diante disso, os resultados de forma geral, na cultura do algodão com deficiência do macronutriente K, obtiveram-se resultados ruins nas imagens com plano de fundo. No entanto, os resultados nas imagens sem plano de fundo foram muito superiores ao método de Otsu e na análise visual foram melhores até que as imagens desejadas, principalmente nas imagens com deficiência do macronutriente.

**Tabela 45 – Comparação da RNA e o método de Otsu, nos resultados com as 10 primeiras imagens sem deficiência do macronutriente K.**

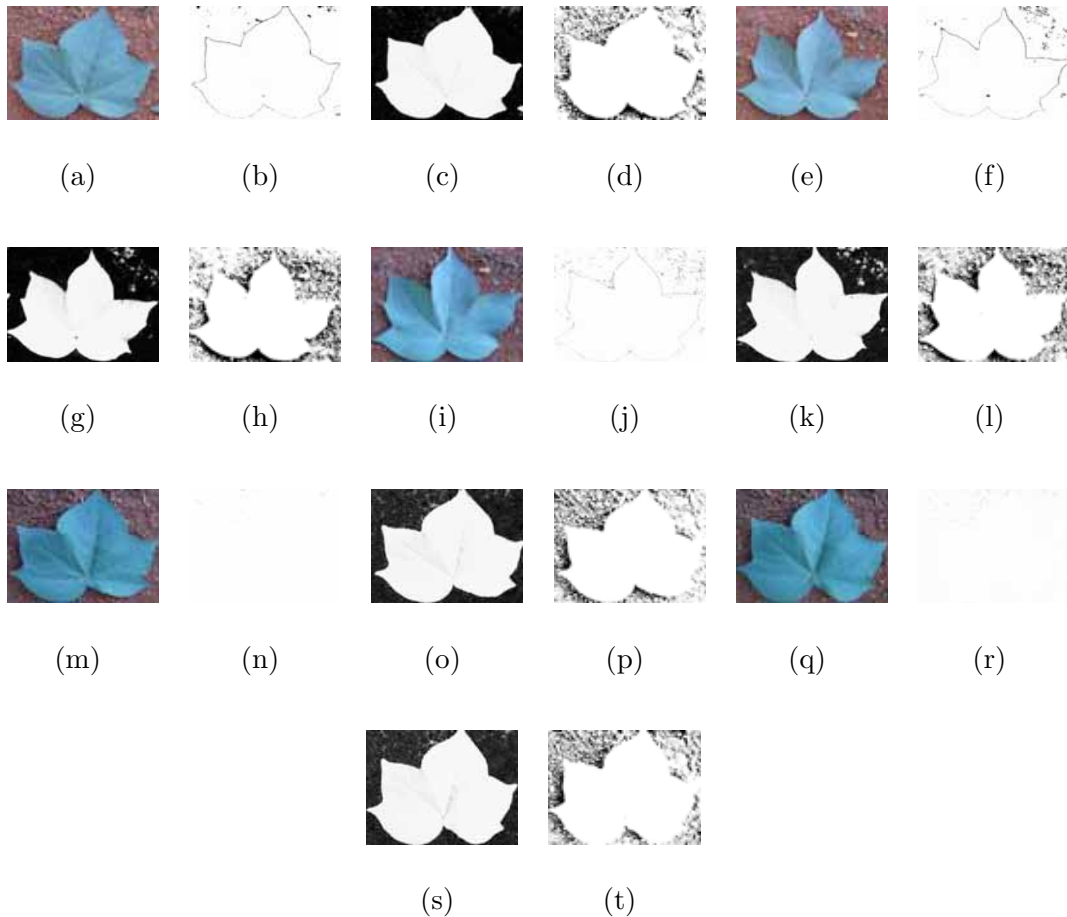
Imagens	PHM(%)		MSE		PSNR		QI		Média		FQ	
	RNA	Otsu	RNA	Otsu	RNA	Otsu	RNA	Otsu	RNA	Otsu	RNA	Otsu
1	56,93	87,87	99,77	27,76	52,24	57,80	0,612	0,950	5,21	7,45	5,88	8,17
2	51,13	82,83	111,74	38,91	51,75	56,33	0,556	0,914	4,85	7,08	5,49	7,82
3	55,60	83,47	87,34	38,62	52,82	56,36	0,630	0,912	5,36	7,10	5,96	7,84
4	57,70	88,70	91,93	28,70	52,60	57,65	0,656	0,941	5,38	7,44	6,04	8,18
5	59,50	87,91	82,77	30,01	53,05	57,46	0,682	0,939	5,58	7,39	6,22	8,13
6	98,94	77,91	0,77	52,50	73,38	55,03	0,999	0,811	8,97	6,62	9,36	7,34
7	97,74	79,98	1,33	45,56	70,98	55,64	0,999	0,840	8,80	6,81	9,23	7,52
8	98,20	79,32	1,36	49,38	70,89	55,30	0,998	0,828	8,80	6,72	9,24	7,45
9	98,71	74,66	1,25	64,55	71,26	54,13	0,997	0,770	8,84	6,31	9,27	7,05
10	97,90	69,56	1,51	76,34	70,43	53,40	0,997	0,711	8,76	5,95	9,21	6,67
Média	77,24	81,22	47,98	45,23	61,94	55,91	0,813	0,862	7,06	6,88	7,59	7,62

Os resultados visuais das 10 primeiras imagens são apresentados nas Figuras 75 e 76, na mesma sequência da Tabela 45. Para cada imagem original (colorida) há três imagens na sequência: a imagem desejada (editor gráfico), imagem segmentada pela RNA e por último a imagem segmentada pelo método de Otsu. O algoritmo de Otsu foi implementado e investigado em dois tipos de resultados, com imagens binárias e em escala de cinza. As imagens apresentadas são binárias, cujos resultados visuais foram melhores.

Na comparação entre o método de Otsu e a RNA, a análise visual e dos parâmetros de qualidade apresentam os melhores valores para a RNA para imagens sem plano de fundo. Nas imagens com plano de fundo o método de Otsu apresentou os melhores valores. Porém, o método de Otsu, nas imagens com plano de fundo, não teve exatamente resultados excelentes, mas com uma intensidade menor que a RNA. Por exemplo, o método de Otsu apresentado nas Figuras 75-(d;h;l;p;t) segmentaram-se as partes das imagens relacionadas aos planos de fundo. Os resultados do método de Otsu, nesses casos, foram melhores que a RNA devido à intensidade de segmentação ser menor, embora não tenham produzido a segmentação correta das imagens.

Nas Figuras 75-(d;h;l;p;t) e 76-(d;h;l;p;t) apresentam-se a segmentação inadequada com o algoritmo de Otsu. A RNA obteve os resultados esperados, sem nenhuma segmentação nas imagens sem plano de fundo, o que é ilustrado pelas Figuras 76-(c;g;k;o;s). Algumas pequenas folhas dispostas no solo, não retiradas pelo editor gráfico, também foram reconhecidas pela RNA devido a sua pigmentação semelhante ao da deficiência de nutriente. Nas Figuras 75-(c;g;k;o;s), com a iluminação diferenciada e o plano de fundo complexo, a implementação com a RNA não se comportou como deveria.

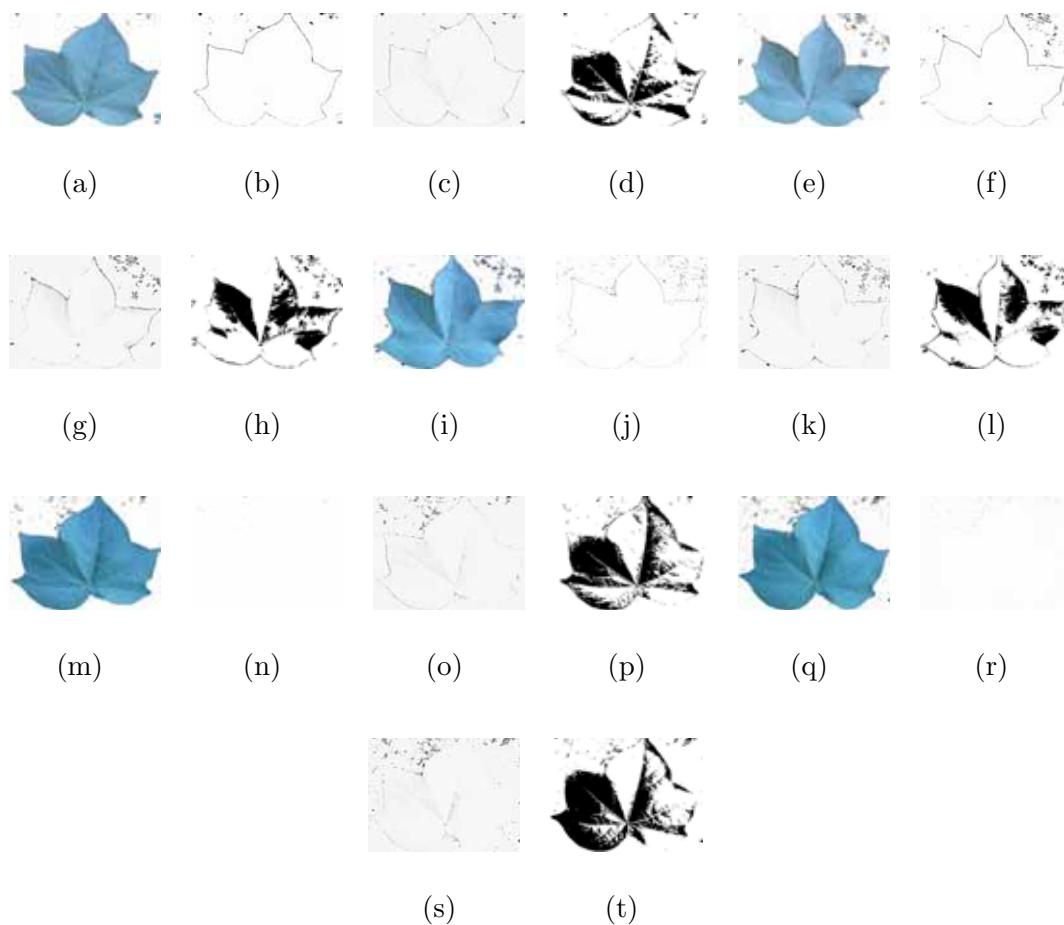
**Figura 75 – Imagens sem deficiência do macronutriente K, com baixa luminosidade e com plano de fundo. Em cada conjunto contém quatro tipos de imagens em sequência: original, desejada, segmentação pela RNA (SRNA) e segmentação pelo método de Otsu (SMO).**



Fonte: Elaboração do próprio autor

A segunda parte analisada têm 8 imagens, sem deficiência de K, quatro com plano de fundo e quatro sem plano de fundo. Na iluminação, a tonalidade do verde e o plano de fundo são distintos das imagens da primeira etapa. Com essas diferenças, os resultados alcançados foram piores do que da primeira etapa. Na Tabela 46 apresenta-se os resultados dos parâmetros de qualidade. Percebe-se a média geral dessas imagens e o fator de qualidade com valores menores do que os apresentados na primeira etapa. Na imagem 13

**Figura 76** – Imagens sem deficiência do macronutriente K, com baixa luminosidade e sem plano de fundo. Em cada conjunto contém quatro tipos de imagens em sequência: original, desejada, segmentação pela RNA (SRNA) e segmentação pelo método de Otsu (SMO).



Fonte: Elaboração do próprio autor

obteve-se os piores resultados dentre todos os outros. Nas imagens sem plano de fundo a RNA alcançou resultados satisfatórios e melhores que o método de Otsu. Os resultados visuais podem ser observados nas Figuras 77 e 78.

As Figuras 75 e 76 contêm uma baixa luminosidade, o que é mais adequado para a segmentação. As Figuras 77 e 78 contêm alta luminosidade solar, tornando imagem imprópria para a segmentação.

A variação dos resultados mostram que a luminosidade deve ser evitada durante a obtenção de imagens da cultura do algodão, principalmente com a deficiência do macronutriente K. Uma solução simples é coletar as imagens no início da manhã ou no fim da tarde conforme descrito por Wang et al. (2013). O bloqueio da luz solar no ângulo de visão da câmera com algum objeto pode ser uma solução viável e fácil.

**Tabela 46 – Comparação da RNA e o método de Otsu, nos resultados com as imagens (11 a 18) sem deficiência do macronutriente K.**

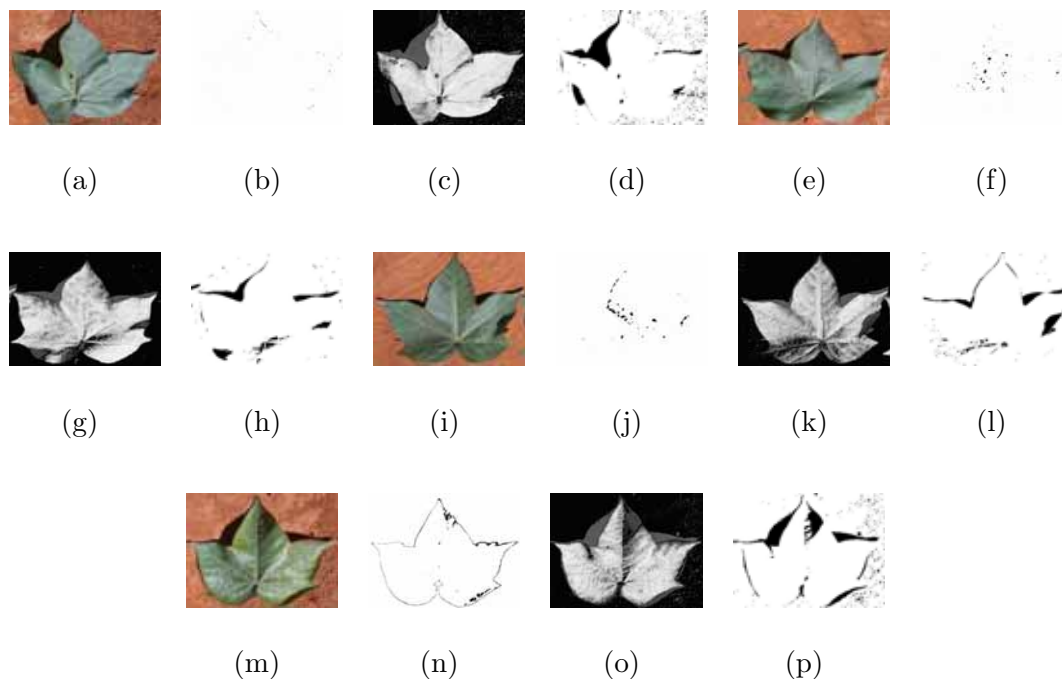
Imagens	PHM(%)		MSE		PSNR		QI		Média		FQ	
	RNA	Otsu	RNA	Otsu	RNA	Otsu	RNA	Otsu	RNA	Otsu	RNA	Otsu
11	33,98	92,20	167,93	19,31	49,98	59,37	0,450	0,943	3,63	7,72	4,31	8,42
12	31,20	94,45	156,60	13,42	50,28	60,95	0,453	0,959	3,70	7,95	4,30	8,62
13	17,94	94,95	168,40	11,71	49,97	61,54	0,385	0,968	3,15	8,03	3,63	8,68
14	26,43	89,74	173,87	21,71	49,83	58,86	0,410	0,935	3,33	7,60	3,94	8,29
15	90,98	88,50	6,64	28,70	64,01	57,65	0,972	0,897	8,13	7,38	8,67	8,09
16	83,73	81,16	15,33	46,74	60,37	55,53	0,940	0,829	7,60	6,81	8,18	7,54
17	72,70	77,16	20,52	56,25	59,11	54,73	0,908	0,786	7,16	6,51	7,66	7,24
18	81,23	83,67	19,66	36,78	59,29	56,57	0,925	0,867	7,42	7,08	8,00	7,79
Média	54,77	87,73	91,12	29,33	55,36	58,15	0,680	0,898	5,51	7,38	6,08	8,08

A terceira parte corresponde às imagens com deficiência do macronutriente K na folha do algodão. As 14 imagens analisadas mostram as mesmas diferenças apresentadas na primeira e na segunda parte, em relação ao plano de fundo. Na Tabela 47 apresenta-se os resultados das últimas 14 imagens.

A análise visual pode ser verificada nas Figuras 79, 80 e 81. A segmentação com o método de Otsu foi melhor que a RNA nas imagens com deficiência e com plano de fundo, mas também apresentou irregularidades para essas imagens. Nas Figuras 79-(d;h;l;p;t) mostram-se o método de Otsu segmentando partes do plano de fundo das imagens, o que não deveria ocorrer, da mesma forma como nas etapas anteriores.

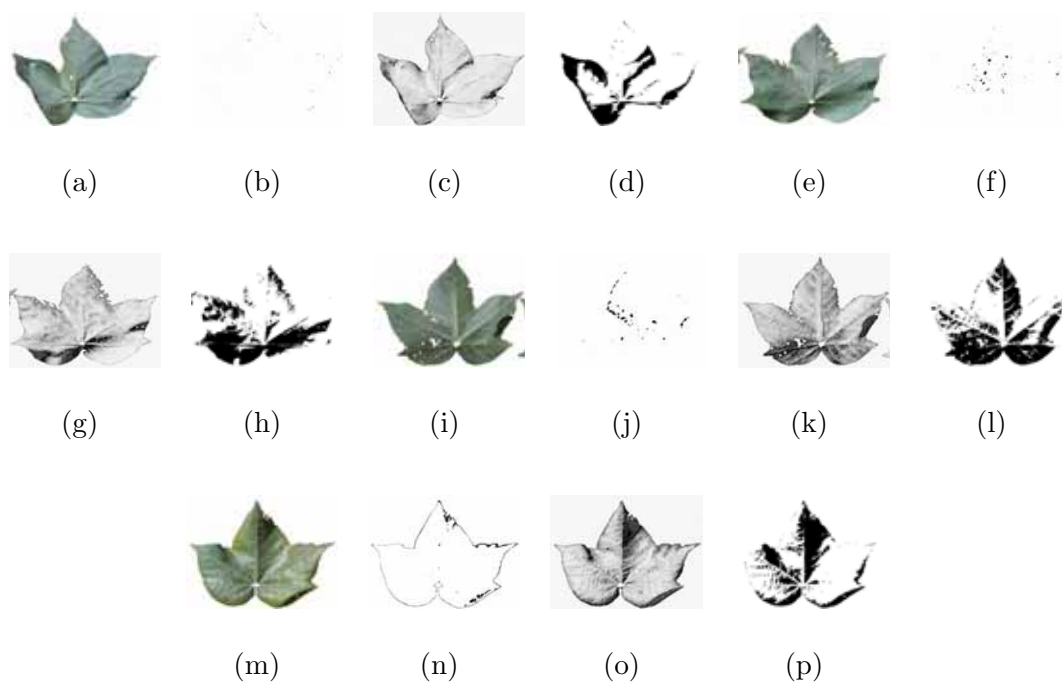
A deficiência de K pode ser observada de forma clara nas folhas com a pigmentação alaranjada, conforme é apresentado nas Figuras 79-(i;m;q;u) e 80-(a). Nas Figuras 79-(a;e;i;m;q) apresentam-se as imagens com deficiências de K grave e baixa luminosidade. Nas Figuras 79-(q;u) apresentam-se as imagens com deficiências de K severa e alta luminosidade.

Figura 77 – Imagens sem deficiência do macronutriente K, com alta luminosidade e com plano de fundo. Em cada conjunto contém quatro tipos de imagens em sequência: original, desejada, SRNA e SMO.



Fonte: Elaboração do próprio autor

Figura 78 – Imagens sem deficiência do macronutriente K, com alta luminosidade e sem plano de fundo. Em cada conjunto contém quatro tipos de imagens em sequência: original, desejada, SRNA e SMO.

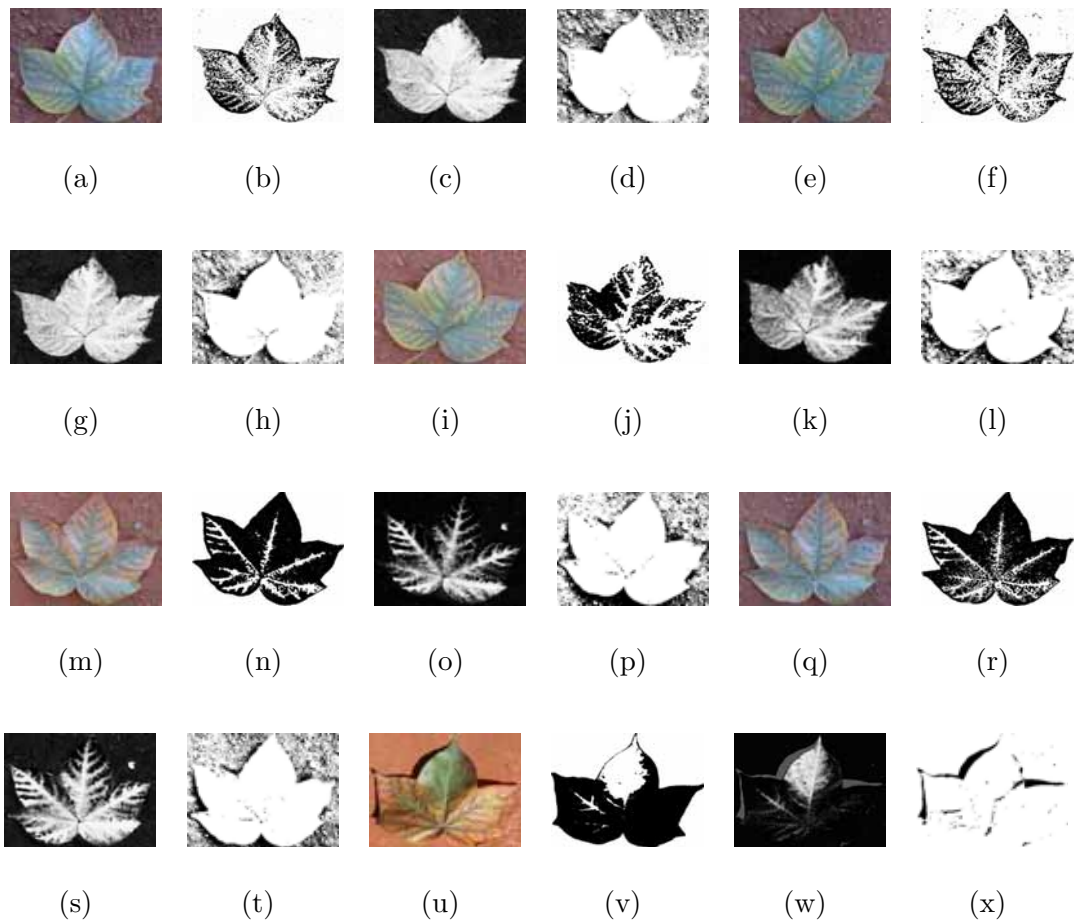


Fonte: Elaboração do próprio autor

Tabela 47 – Comparação da RNA e o método de Otsu, nos resultados com todas imagens com deficiência do macronutriente K.

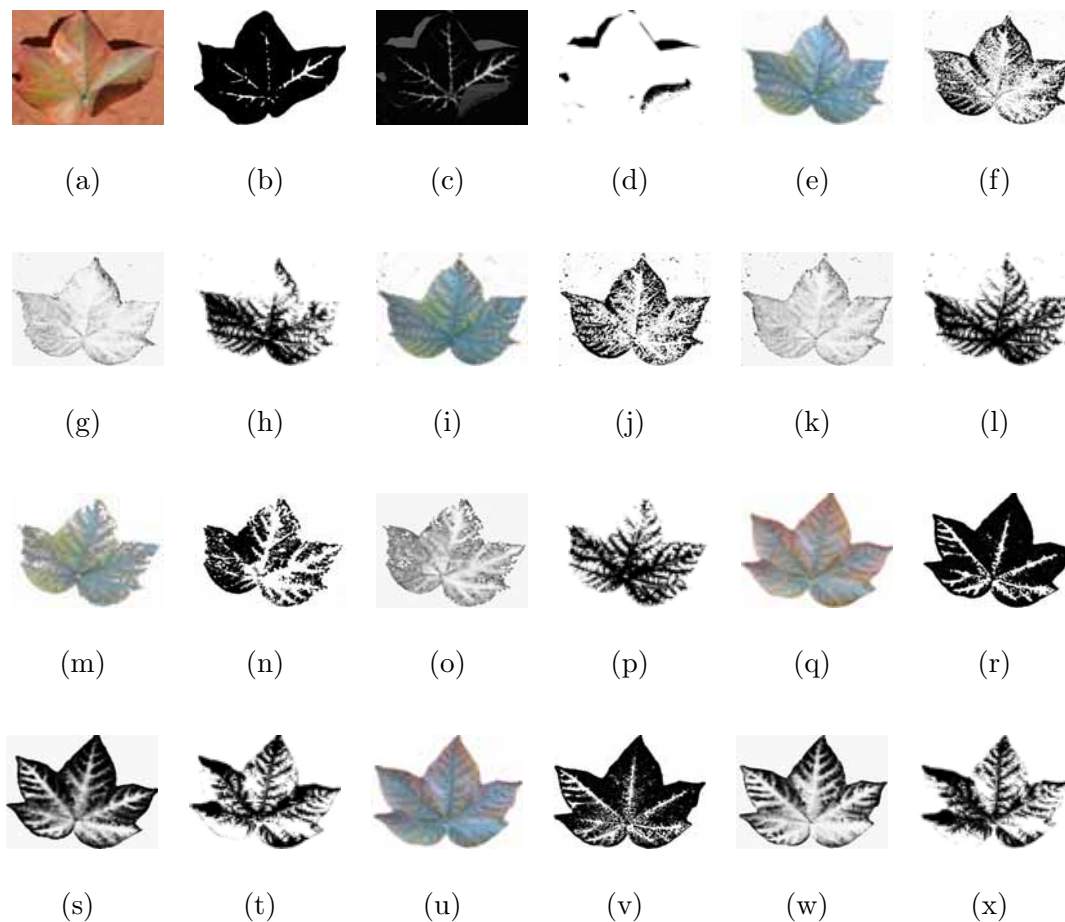
Imagens	PHM(%)		MSE		PSNR		QI		Média		FQ	
	RNA	Otsu	RNA	Otsu	RNA	Otsu	RNA	Otsu	RNA	Otsu	RNA	Otsu
19	28,71	55,42	133,04	88,25	50,99	52,77	0,457	0,733	3,91	5,47	4,38	6,12
20	27,52	58,62	128,55	84,20	51,14	52,98	0,453	0,752	3,93	5,63	4,36	6,30
21	18,49	54,01	146,64	102,17	50,57	52,14	0,349	0,648	3,36	5,15	3,74	5,82
22	33,44	36,88	123,99	149,07	51,30	50,50	0,211	0,456	3,83	3,93	4,18	4,55
23	25,80	37,33	122,44	137,24	51,35	50,86	0,246	0,493	3,70	4,12	3,98	4,71
24	44,78	54,74	135,33	110,24	50,92	51,81	0,160	0,542	3,92	4,94	4,41	5,60
25	50,62	44,46	129,02	136,65	51,12	50,87	0,139	0,421	4,11	4,22	4,62	4,84
26	78,22	63,05	24,58	56,97	58,33	54,67	0,891	0,766	7,19	6,12	7,77	6,70
27	76,29	62,48	26,92	59,26	57,93	54,50	0,884	0,755	7,09	6,06	7,66	6,64
28	70,41	62,21	24,46	67,24	58,35	53,95	0,819	0,713	6,91	5,89	7,37	6,49
29	78,39	59,47	14,71	63,31	60,56	54,22	0,666	0,578	7,15	5,71	7,52	6,19
30	66,66	55,62	34,08	65,16	56,90	54,09	0,643	0,628	6,42	5,65	6,83	6,13
31	88,90	66,83	13,24	73,44	61,01	53,57	0,654	0,550	7,43	5,72	7,89	6,32
32	94,92	58,08	3,07	97,20	67,36	52,35	0,579	0,682	7,97	5,36	8,26	6,06
Média	55,94	54,94	75,72	92,17	55,56	52,81	0,511	0,623	5,49	5,28	5,93	5,89

Figura 79 – Imagens com deficiência do macronutriente K, com distintos planos de fundo e luminosidade. Em cada conjunto contém quatro tipos de imagens em sequência: original, desejada, SRN e SMO.



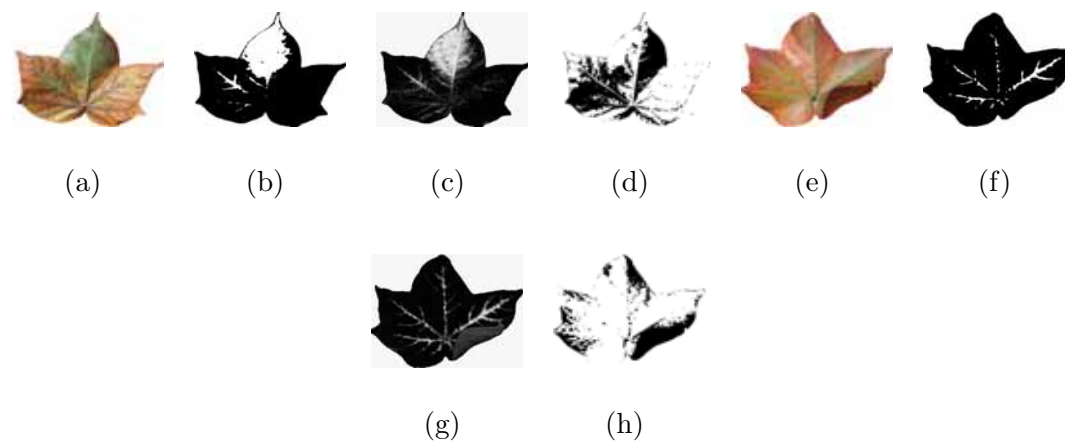
Fonte: Elaboração do próprio autor

Figura 80 – Imagens com deficiência do macronutriente K, com distintos planos de fundo e luminosidade. Em cada conjunto contém quatro tipos de imagens em sequência: original, desejada, SRN e SMO.



Fonte: Elaboração do próprio autor

Figura 81 – Imagens com deficiência do macronutriente K, com distintos planos de fundo e luminosidade. Em cada conjunto contém quatro tipos de imagens em sequência: original, desejada, SRN e SMO.



Fonte: Elaboração do próprio autor

### 7.3 Conclusão deste Capítulo

Neste estudo mostrou-se que uma RNA otimizada pode ser usada na segmentação de imagem dos macronutrientes P e K pela folha do algodão. O sistema mostra-se robusto mesmo com poucas unidades de neurônios em aplicações não lineares como a segmentação de imagem.

Diversas simulações foram realizadas para encontrar as melhores combinações e organização da RNA para uma aplicação real na folha do algodão. Nas combinações foram analisados vários parâmetros para determinar um fator de qualidade para a segmentação. A qualidade da segmentação da imagem foi comparada com o método de Otsu. Os parâmetros relacionados ao erro (MSE e PSNR) não são adequados para a análise de qualidade da segmentação.

Nos dois experimentos foram comparados a RNA com o métodos de Otsu. No primeiro experimento utilizou-se o método de Otsu em escala de cinza e com imagem desejada no mesmo sistema de cor. No segundo experimento os resultados do método de Otsu foram obtidos a partir de imagens binárias.

No primeiro experimento a superioridade da RNA é visível pelos parâmetros de qualidade e resultados visuais das segmentações. No segundo experimento, a RNA não teve bons resultados com as imagens que possuíam plano de fundo. Os melhores resultados foram obtidos com as imagens sem o plano de fundo, devido a sua semelhança com o tipo de coloração da deficiência de K na cultura do algodão.

Em algumas imagens, como as das Figuras 79-(q;u), têm-se alta semelhança na coloração do solo com a deficiência do macronutriente K na folha do algodão. Com isso, o aprendizado da RNA foi confundido e ponderou de maneira favorável o solo para a segmentação, atrapalhando os resultados da mesma. No segundo experimento mostra-se que os dois métodos, RNA e Otsu, não obtiveram resultados satisfatórios em todas as imagens. Para esse caso, seria ideal analisar métodos ou técnicas para auxiliarem na segmentação e extração de características. A segmentação pela folha da soja será realizada em abstrações de níveis, alto e baixo, descrito no Capítulo 8.

## 8 SISTEMA RNA PARA CULTURA DA SOJA

Neste capítulo apresenta-se o sistema da RNA em abstrações de níveis, alto e baixo, com experimento real e comprovação de deficiência de Potássio (K) no cultivo da soja. O experimento compara os dois níveis de abstrações, pelo Matlab e FPGA, além da segmentação realizada de forma manual com editor gráfico<sup>1</sup>. Na seção 8.1 apresentam-se as informações relacionadas aos materiais e métodos utilizados no experimento, na seção 8.2 apresenta-se o sistema utilizado nos seus respectivos níveis de abstrações. Na seção 8.3 discutem-se os resultados encontrados nos dois níveis de abstrações. As conclusões do capítulo são apresentadas na seção 8.4.

### 8.1 Análise de Deficiência na Soja

O experimento realizado no cultivo da soja foi na estação experimental em Itiquira-MT e pertence a Fundação MT. O experimento foi realizado em três blocos com o cultivar de soja TMG 1176 RR (tipo de cultivar), grupo de maturação 7,6 e resistente a aplicação do herbicida glifosato -“Roundup Read” - RR. Os três blocos têm o tamanho de 324 metros de comprimento por 91 metros de largura, ou seja, 2,95 hectares. O plantio é em linha com espaçamento de 45 cm entre cada linha nos três blocos e a deficiência do macronutriente K está presente em dois blocos. As características dos três blocos são:

1. Bloco 1: Bloco com tratamento comercial, forma padrão realizada por produtores do estado de MT (-17°9'7.2", -54°45'7.2")<sup>2</sup>;
2. Bloco 2: Bloco que contém o primeiro ano de deficiência de K (-17°9'8.9994", -54°45'7.2");
3. Bloco 3: Bloco no sexto ano de deficiência de K (-17°9'10.7994", -54°45'7.2").

---

<sup>1</sup>Houve auxílio de um especialista (agrônomo da FMT) para a identificação da coloração correta na folha da soja, que identifica a presença da falta do macronutriente K na planta.

<sup>2</sup>Posição GPS dos Blocos, os valores podem ser consultados pelo Google Maps.

A análise laboratorial das plantas foram realizadas entre os estádios<sup>3</sup> R1 e R2, ou seja, no início do estágio reprodutivo da planta conforme indicado por SFREDO (2008). A coleta das folhas foi de forma aleatória em cada bloco, pelo terceiro trifólio<sup>4</sup> da soja e respeitando a secagem adequada conforme recomendação. A análise foliar de teores de nutrientes foi realizada pela UNESP de Ilha Solteira no Laboratório de Nutrição de Plantas. Na Tabela 48 apresentam-se os resultados de macronutrientes nos três blocos<sup>5</sup>.

**Tabela 48 – Análise foliar da soja de macronutrientes nos três blocos.**

Amostras	$N_{g/kg}$ Nitrogênio	$P_{g/kg}$ Fósforo	$K_{g/kg}$ Potássio	$Ca_{g/kg}$ Cálcio	$Mg_{g/kg}$ Magnésio	$S_{g/kg}$ Enxofre
Bloco 1	54,2	3,9	20	9,5	5,5	2,0
Bloco 2	56,4	3,7	15	6,7	4,3	2,3
Bloco 3	58,0	3,0	10	6,9	6,0	2,5

As deficiências de K são encontradas nos blocos 2 e 3 com níveis menores do que os considerados baixos. A recomendação para o macronutriente K é classificada em “Baixo” para valores menores que 17,6, “Suficiente” para valores entre 17,6 e 26,3 e “Alto” para valores maiores que 26,3 conforme SFREDO (2008). O bloco 2 está próximo do intervalo entre as classificações “Baixo” e “Suficiente”, por isso não teve o aparecimento de deficiência nas folhas. Apenas o bloco 3 com valores bem menores que a classificação “Baixo” apresentou as deficiências investigadas nas seções seguintes.

As imagens foram coletadas em três visitas realizadas na estação experimental, safra 2013/14. As visitas foram alternadas entre os estádios, V2 e V3, R1 e R2, R4 e R5, conforme indicado por Fang, Ma e Chen (2010), Vollmann et al. (2011), Li-shu (2010). O aparecimento da deficiência do macronutriente K na folha da soja foi constatado apenas no estágio R4 e R5 com início do desenvolvimento das vagens. Após esse estágio, a planta inicia o processo de envelhecimento das folhas e vagens com desenvolvimento pleno, tornando-se irrelevante o monitoramento e reconhecimento da deficiência.

As imagens foram adquiridas com uma câmera de baixo custo, Samsung de 14 MegaPixel e resolução de 4320x3240. Nos três períodos de coletas de imagens foram obtidas mais de 800 imagens. Porém, foram selecionadas 72 imagens com as melhores qualidades e com maiores diferenças entre características como cor, iluminação, distância e posição angular.

<sup>3</sup>Os estádios são as etapas de desenvolvimento da planta e são divididos em Vegetativo (V) e Reprodutivo (R). Na soja os estádios vegetativos podem ser de V1 até Vn, sendo que V2 é o estágio cujo primeiro trifólio está completamente desenvolvido, e V3 o segundo trifólio, assim sucessivamente. Os estádios reprodutivos são divididos em 8 partes (R1 até R8). Em cada par de estádios é relacionado na sequência para a formação e o desenvolvimento completo das flores, vagem, semente e maturação (FARIAS; NEPOMUCENO; NEUMAIER, 2007).

<sup>4</sup>Trifólio é o conjunto de três folhas em uma única haste completamente separadas. O terceiro trifólio corresponde a terceira haste de cima (topo) para baixo (raiz) na planta.

<sup>5</sup>Relatório da Análise foliar consta no Anexo A na Figura 114.

Diversas formas de coleta das imagens foram avaliadas e selecionadas para investigação da aplicação. A posição angular da câmera foi utilizada de três modos em 90, 60 e 45 graus. A distância entre a lente da máquina e as folhas tiveram três definições com 30, 60 e 290 cm. As formas de captura das imagens foram duas em laboratório, com luz interna para os trifólios e as folhas, com distância de 30 cm e posição angular de 90 graus. As imagens externas foram realizadas em campo com as três posições angulares e distâncias.

## 8.2 Implementação em Diferentes Níveis de Abstrações

O experimento real com a cultura da soja e deficiência do macronutriente K foi investigado com dois sistemas em níveis de abstrações distintos. O sistema em alto nível de abstração foi desenvolvido e definido na seção 6.3 e no Capítulo 7. O sistema investigado para a cultura da soja contém a mesma organização utilizada no primeiro experimento avaliado na Tabela 43 e do segundo experimento, ambos descritos no Capítulo 7.

O sistema de hardware (baixo nível de abstração) foi aprimorado a partir dos quatro estudos nos Capítulos 4, 5, 6 e 7. O primeiro com o intuito de consolidar o aprendizado em dispositivos reconfiguráveis, principalmente no uso de IP Cores de ponto flutuante. No segundo contém várias investigações importantes ao sistema final, pode-se elencar as principais como o hardware para a função de ativação Tangente Hiperbólica, uma compreensão abrangente sobre o controle com FSM, o armazenamento em memória e a organização geral do sistema da RNA em dispositivo reconfigurável. O terceiro estudo de caso compreende da investigação dos parâmetros mais adequados à filtragem e à segmentação da imagem com o uso de ferramentais padrões no processamento digital de imagem e a RNA. O quarto estudo foi investigado a melhor combinação da RNA com uma aplicação real no cultivo do algodão.

### 8.2.1 Arquitetura do Sistema

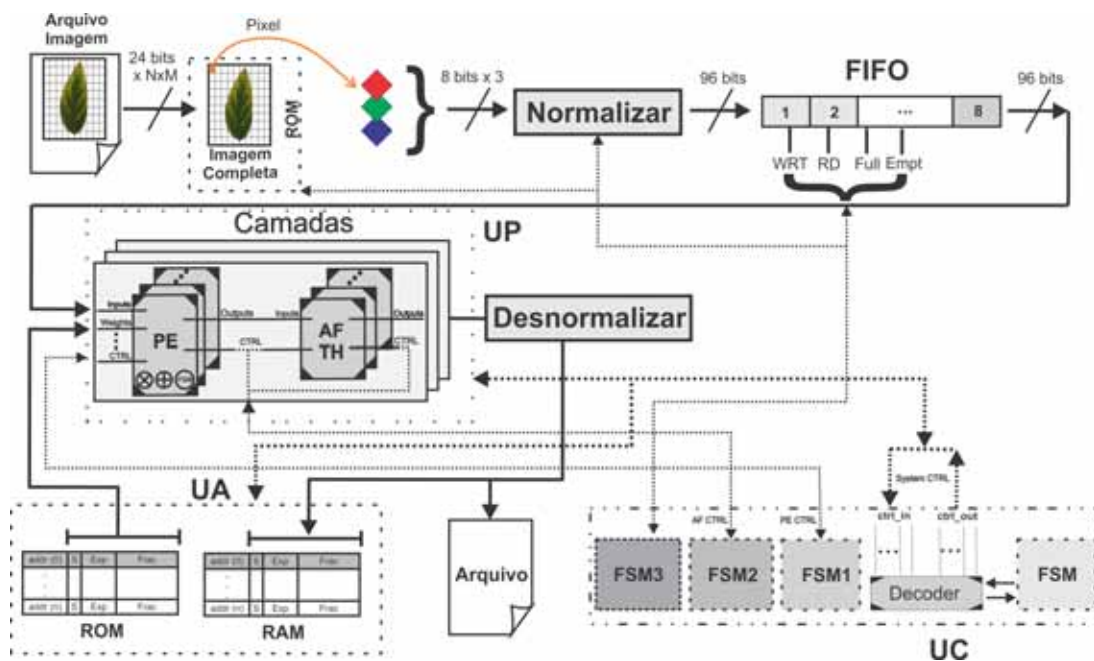
O sistema da RNA definido no Capítulo 5 foi aprimorado com inserção de novos módulos para incorporar as funcionalidades de aquisição da imagem por simulação funcional, conversão e normalização dos dados e a segmentação de imagem. A expansão do sistema de baixo nível foi determinada pela mesma estrutura da RNA em alto nível consolidada no Capítulo 7. A execução geral do sistema em alto nível é ilustrada pela Figura 63 da seção 6.3, já o sistema em baixo nível de abstração está ilustrado pela Figura 82.

As principais modificações em relação ao anterior, Capítulo 5, pode-se citar a topologia da RNA, atualmente é 3-3-1 e na anterior era de 1-2-1. Como a RNA executa um tipo de aplicação, segmentação de imagens, a RNA contém 3 entradas que correspondem aos três

componentes (RGB) de um píxel e com uma quantidade de 60.000 e 105.000 amostras para cada imagem dependendo do tipo, o controle do fluxo das entradas é discutido na seção 8.2.1.1. A unidade de processamento foi expandida por meio dos elementos de processamento (Figura 26) e AF (Figura 36), as alterações ocorreram em todas as camadas da RNA.

A unidade de armazenamento foi alterada para a nova arquitetura do sistema. As entradas são carregadas a partir de um arquivo de texto na simulação funcional e não carregadas a partir de uma BRAM como anteriormente. No sistema desenvolvido no Capítulo 5 a quantidade de entradas utilizadas eram apenas 128, no sistema para a segmentação de imagens, trabalha-se com 60.000 ou 105.000 amostras de entrada (ou píxeis) com três valores. Com essa grande quantidade de amostras houve a necessidade de fazer uma memória temporária (*Buffer*) por meio de uma FIFO (*First In, First Out*) para organizar melhor as entradas de dados, sem perdê-los. A nova organização do sistema é ilustrado por meio do diagrama apresentado na Figura 82.

**Figura 82 – Organização geral do sistema em baixo nível de abstração.**



Fonte: Elaboração do próprio autor

A normalização dos dados de entrada é outra característica adicionada ao sistema. Os píxeis de entrada (componentes RGB) são convertidos em representação de dados de ponto flutuante, normalizados no intervalo  $[0:1]$  como em alto nível. O módulo “normalizar” desenvolvido é executado em paralelo, um para cada componente RGB. A normalização recebe os valores do píxel RGB em tipo de dado inteiro ( $3 \times 8bits$ ) e faz a conversão em ponto flutuante normalizado por meio do método HPC, mesmo método utilizado para aproximação da AF TH, seção 5.2.3.

O módulo é executado com 10 valores de expoentes específicos para a variação de intensidades de 8 bits de resolução. Os intervalos são divididos em 8 partes iguais de 32 elementos, por ter uma resposta linear. Na representação de ponto flutuante utilizou-se de 5 bits de fração, para as amostras de passos com precisão de  $2^{-8}$ . As saídas são determinadas por expressões booleanas simplificadas (cenários HPC) específicas para este módulo.

Um módulo para a desnormalização e decodificação dos dados também foi desenvolvido, para fazer o sentido inverso do módulo de normalização. Os mesmos valores de fração, expoente, precisão e intervalos foram adotados. Porém, o módulo é totalmente distinto na decodificação dos dados, expoentes e nas expressões booleanas utilizadas. Os módulos para “normalizar” e “desnormalizar” os dados estão ilustrados na Figura 82. A normalização ocorre antes da inserção das entradas na FIFO, e a desnormalização executa-se após a obtenção da saída da RNA para armazenar os resultados em memória e arquivo.

#### 8.2.1.1 Unidade de Controle

A unidade de controle foi a parte do sistema com maiores alterações. A FSM do sistema base da RNA é ilustrado pela Figura 25 no Capítulo 5, algumas alterações foram realizadas para obtenção das novas funcionalidades da RNA. Na Figura 83 apresenta-se a nova FSM geral do sistema com a retirada do estado “Load DataIN”.

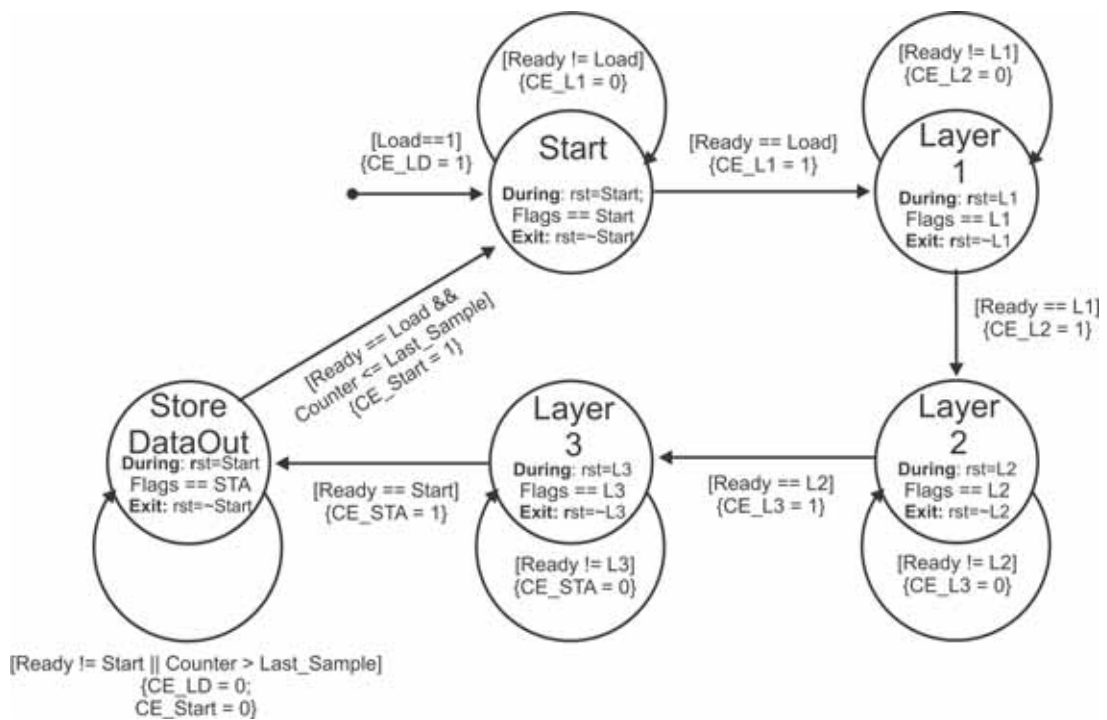
O estado “Load DataIN” era responsável por carregar os dados de entrada a partir de uma memória dedicada BRAM. O estado “Start” tinha como funcionalidade carregar os pesos de uma memória ROM para sinais internos do sistema. Agora, os pesos são carregados a partir de uma memória ROM contida em um pacote disponibilizado por meio de uma biblioteca (*package*) nas três camadas da RNA. O estado “Start” ficou responsável pela entrada de dados, normalização e conversão em ponto flutuante. Para isso, foi necessária uma FSM para organizar essas etapas no estado “Start”, conforme apresentado na Figura 84.

O estado “Start” é iniciado após ocorrer um sinal de *reset* no sistema, assim a FSM fica no estado de “Wait” aguardando a habilitação do sinal “Load” da FSM geral. Então, o “ROM-En” habilita a entrada do píxel contido na ROM, cuja leitura foi realizada por meio do arquivo de imagem, conforme apresentado na Figura 82. O estado “Input” divide a entrada de 24 bits e armazena em três de registradores (RGB) de 8 bits. No estado “Convert” contém três módulos de conversão em paralelo, que coletam os valores contidos nos registradores, e fazem a normalização dos dados e sua conversão em ponto flutuante.

A normalização e a conversão seguem o mesmo padrão adotado para a aproximação da Tangente Hiperbólica baseado em HPC, visto no Capítulo 5. Após, gerar as novas

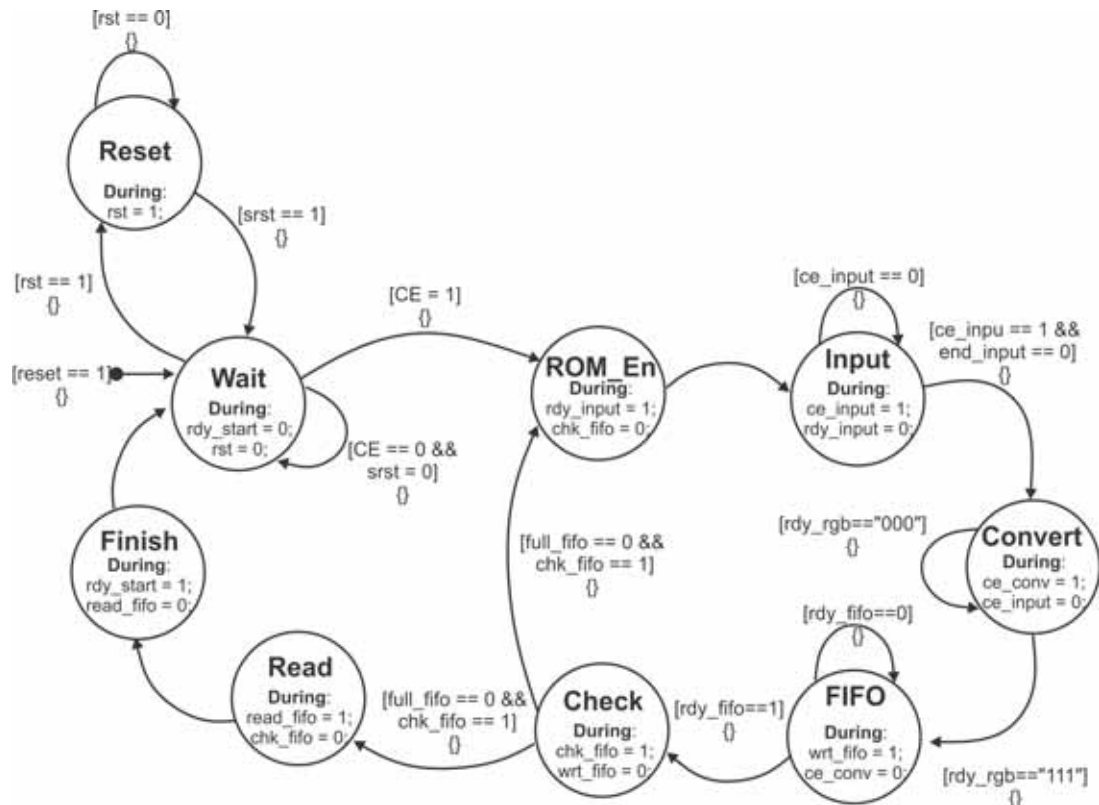
representações das entradas no estado “Convert”, os resultados são agrupados em 96 bits de dados em ponto flutuante na primeira posição da FIFO, no estado com o mesmo nome. O estado “Check” verifica se a FIFO está cheia, senão estiver, retorna ao estado “ROM-En” para repetir o processo na leitura do próximo píxel, caso contrário o estado “Read” fará a leitura do primeiro píxel armazenado na FIFO e disponibilizará em um registrador de saída. O estado “Finish” habilita um sinal interno para avisar a FSM geral que os dados de entrada para a primeira camada da RNA estão disponíveis.

**Figura 83 – Máquina de estados finitos do controle geral da rede neural artificial.**



Fonte: Elaboração do próprio autor

Figura 84 – Máquina de estados finitos do estado Load.



### 8.3 Resultados

O sistema completo, nos dois níveis de abstração, para a segmentação de imagens na identificação de deficiência do macronutriente K na folha da soja são comparados por meio de resultados visuais e por fatores de qualidade. Os resultados foram divididos em duas partes, a análise da utilização de área na FPGA, apresentado na seção 8.3.1, e os resultados visuais junto com os parâmetros de qualidade, apresentados na seção 8.3.2.

#### 8.3.1 Utilização de área na FPGA

Os resultados de utilização de área na FPGA foram implementados em três plataformas Nexys2 - Spartan-3E, Nexys4 - Artix 7 e Zedboard, pertencentes aos fabricantes Xilinx e Digilent. Apenas a Spartan-3E está presente para os testes, as outras serão adquiridas, mas foram verificadas para comprovar que tais plataformas irão trazer suporte adequado para o sistema.

A quantidade de área utilizada dos principais itens nas plataformas são apresentadas na Tabela 49 e comprovam que as novas plataformas FPGA comportam grandes quantidades de unidades de ponto flutuante. Na Spartan-3E, a implementação do sistema

ultrapassa os valores ideais da plataforma. Os valores das plataformas Nexys4 e Zedboard são iguais pelo fato de que o modelo do chip da FPGA é o mesmo a Artix 7, com capacidades diferentes.

**Tabela 49 – Utilização de área na FPGA para o sistema da RNA.**

Cenários	Slices Reg	Slices	LUT	BRAM(Mem)	Multip.(18x18)	DSP-48
Nexys2 - Spartan-3E	22.967(246%)	15.216(205%)	19.159(326%)	1(5%)	84(420%)	-
Nexys4 - Artix 7	19.022(54%)	15.124(85%)	11.567(51%)	1(1%)	-	42(52%)
ZedBoard	19.022(17%)	15.124(28%)	11.567(51%)	1(0%)	-	42(19%)

### 8.3.2 Resultados Visuais e Parâmetros de Qualidade

Os resultados visuais foram divididos em três grupos, dois com imagens das folhas e trifólios da soja em áreas internas (laboratório) e o outro grupo com imagens de campo, ou seja, área externa. Os primeiros resultados são apresentados por imagens nos três estádios da soja, em sua maioria por imagens com distância de 30 cm e ângulo de 90 graus. Apenas as imagens correspondentes às Figuras 91-(g;m;p;s;v) e 92-(a) estão com distâncias e ângulos diferentes. As imagens extras das Figuras 93, 94 e 95 contêm todas as variações de distância, ângulo, deficiência e blocos, mas as suas avaliações são feitas apenas de forma visual em alto nível de abstração.

Os índices de qualidade das segmentações são os mesmos utilizados e definidos na seção 7.1.1.1. Nas Tabelas 50, 51 e 52 apresentam-se os resultados em níveis de abstrações, alto e baixo, com todos os índices de qualidade determinados e imagens desejadas definidas a partir de editor gráfico. As médias das três Tabelas 50, 51 e 52 são apresentadas pela Tabela 53.

As imagens de folhas individuais coletadas da cultura da soja, todas adquiridas em laboratório, têm seus resultados apresentados na Tabela 50. Nas Figuras 85, 86 e 87 apresentam-se as imagens que correspondem aos resultados da Tabela 50.

A sequência de apresentação das imagens são de agrupamentos de quatro imagens, sendo a primeira a imagem original, a segunda imagem desejada adquirida por editor gráfico, a terceira imagem gerada pelo sistema de alto nível e a quarta imagem gerada pelo sistema de baixo nível. Assim, isso é determinado sucessivamente para todas as imagens das Figuras 85, 86, 87, 88, 89, 90, 91 e 92.

Na Tabela 50 pode-se observar que as imagens sem deficiência (1 a 7) e com pouca ou sem a existência de segmentação alcançaram os melhores resultados com fator de qualidade acima de 9. As imagens com deficiência obtiveram resultados promissores nos dois principais parâmetros, com porcentagem de acerto (PHM) acima de 80%, chegando a quase 95%, e índice de qualidade com a maioria dos valores acima de 0,6. Pelos

resultados visuais, pode-se analisar as imagens consideradas, ideais ou desejadas, são muito semelhantes às encontradas em todos os casos das Figuras 85, 86, 87, 88, 89, 90, 91 e 92.

**Tabela 50 – Resultados encontrados nas imagens de folhas independentes adquiridas em laboratório para implementação em níveis de abstrações, alto (M-Matlab) e baixo (F-FPGA).**

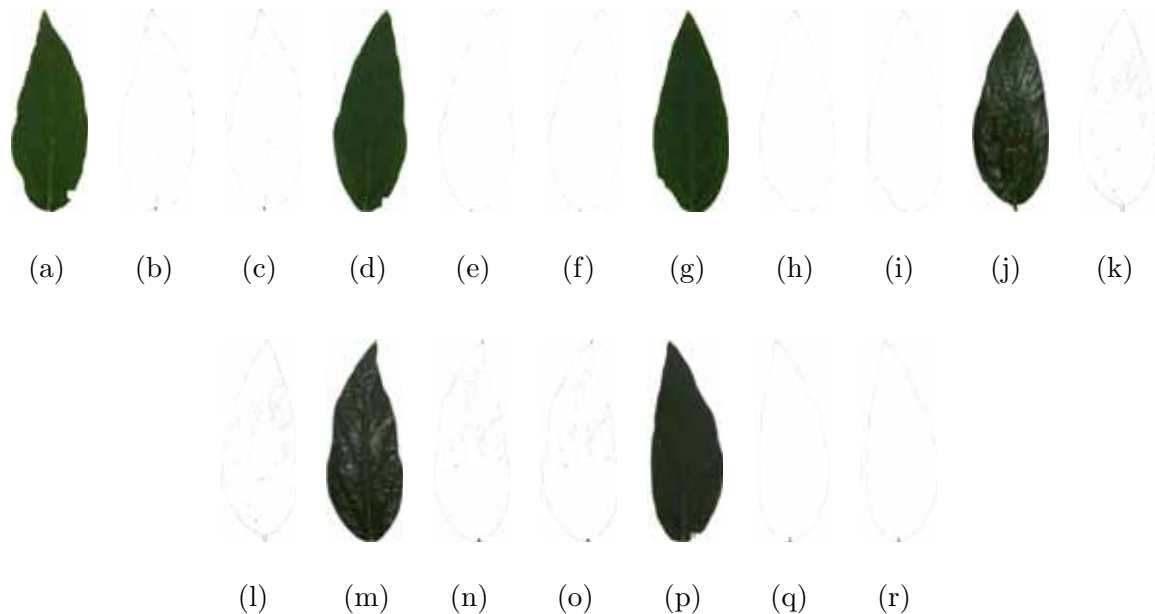
IMGs	PHM(%)		MSE		PSNR		QI		Média		FQ	
	M	F	M	F	M	F	M	F	M	F	M	F
1	99,91	99,92	0,08	0,07	83,12	83,68	0,931	0,931	9,50	9,53	9,63	9,65
2	99,97	99,98	0,04	0,04	85,81	86,30	0,931	0,931	9,66	9,69	9,73	9,75
3	99,97	99,98	0,05	0,04	85,31	85,76	0,931	0,931	9,63	9,66	9,71	9,73
4	99,86	99,88	0,13	0,12	81,10	81,61	0,931	0,931	9,38	9,41	9,56	9,57
5	99,86	99,88	0,15	0,14	80,44	80,90	0,931	0,931	9,34	9,37	9,53	9,55
6	99,92	99,93	0,09	0,08	82,85	83,16	0,931	0,931	9,49	9,50	9,62	9,63
7	98,44	97,59	1,20	1,84	71,44	69,58	0,930	0,929	8,76	8,62	9,15	9,05
8	93,37	87,37	5,78	15,63	64,61	60,29	0,611	0,595	7,78	7,26	8,15	7,70
9	91,88	87,94	10,08	15,82	62,20	60,24	0,723	0,719	7,70	7,42	8,18	7,93
10	87,68	84,49	12,78	19,10	61,17	59,42	0,741	0,742	7,53	7,28	8,01	7,80
11	94,80	87,28	5,60	14,29	64,75	60,68	0,496	0,494	7,68	7,16	8,00	7,54
12	88,31	81,23	12,40	23,48	61,30	58,52	0,680	0,663	7,48	7,01	7,93	7,49
13	91,26	87,69	7,85	14,95	63,28	60,48	0,863	0,846	7,94	7,60	8,45	8,16
14	90,89	85,68	9,43	19,17	62,48	59,40	0,840	0,806	7,84	7,39	8,36	7,95
15	94,90	89,26	3,82	14,76	66,41	60,54	0,803	0,782	8,18	7,56	8,61	8,10
16	81,42	77,93	25,42	32,96	58,18	57,05	0,482	0,459	6,74	6,49	7,16	6,91

Os resultados visuais, na segmentação de imagens em distintos níveis de abstrações, foram melhores que as segmentações com o próprio editor gráfico, em alguns casos. A eficiência da RNA mostra a sua robustez para tal aplicação. Na Figura 85 ilustra-se as imagens sem deficiência de K, assim todas as segmentações não devem conter nada (imagem na cor branca), isso ocorre em todas as imagens analisadas nos resultados visuais. Nas Figuras 86-(m;n;o;p) e 86-(u;v;w;x) ilustram-se na parte inferior direita da folha, uma linha a mais na segmentação com RNA e inexistente na segmentação com o editor gráfico em ambos os casos. Na Figura 87-(m;n;o;p) apresenta-se a segmentação pelo editor gráfico com buracos na segmentação, o que não ocorre pela segmentação com a RNA.

Na Tabela 51 apresentam-se os resultados com os trifólios da soja e as Figuras 88, 89 e 90 os resultados visuais das imagens dos trifólios. Nos trifólios não é possível fazer a diferenciação das imagens com e sem deficiência K. Em todos os casos obtiveram-se excelentes resultados com fatores de qualidade acima de 8,5, e os principais parâmetros também com excelentes resultados, o PHM com valores acima de 92% e o QI com valores acima de 0,85.

Os resultados das imagens externas pela avaliação de qualidade definida na Tabela 52 e os resultados visuais nas Figuras 91 e 92, também são investigados nos três blocos da plantação de soja. Os valores alcançados contêm excelentes resultados conforme fator de

Figura 85 – Resultados visuais das imagens das folhas independentes adquiridas em laboratório sem deficiência de K, com agrupamentos de três imagens: original, segmentação com RNA no Matlab e segmentação com RNA no FPGA. As imagens não têm deficiência de K, a segmentação no editor gráfico é uma imagem totalmente branca.

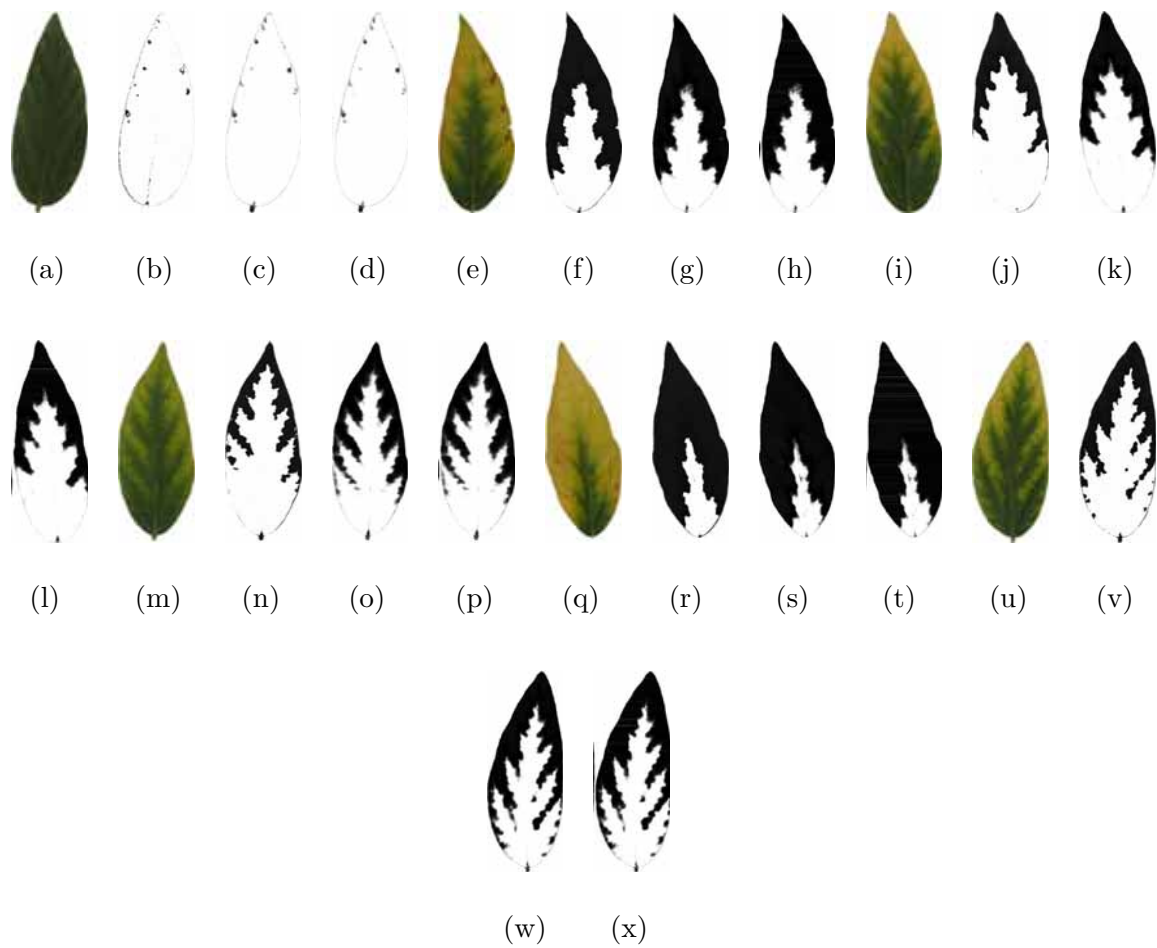


Fonte: Elaboração do próprio autor

Tabela 51 – Resultados encontrados nas imagens dos trifólios adquiridas em laboratório para implementação em níveis de abstrações, alto (M-Matlab) e baixo (F-FPGA).

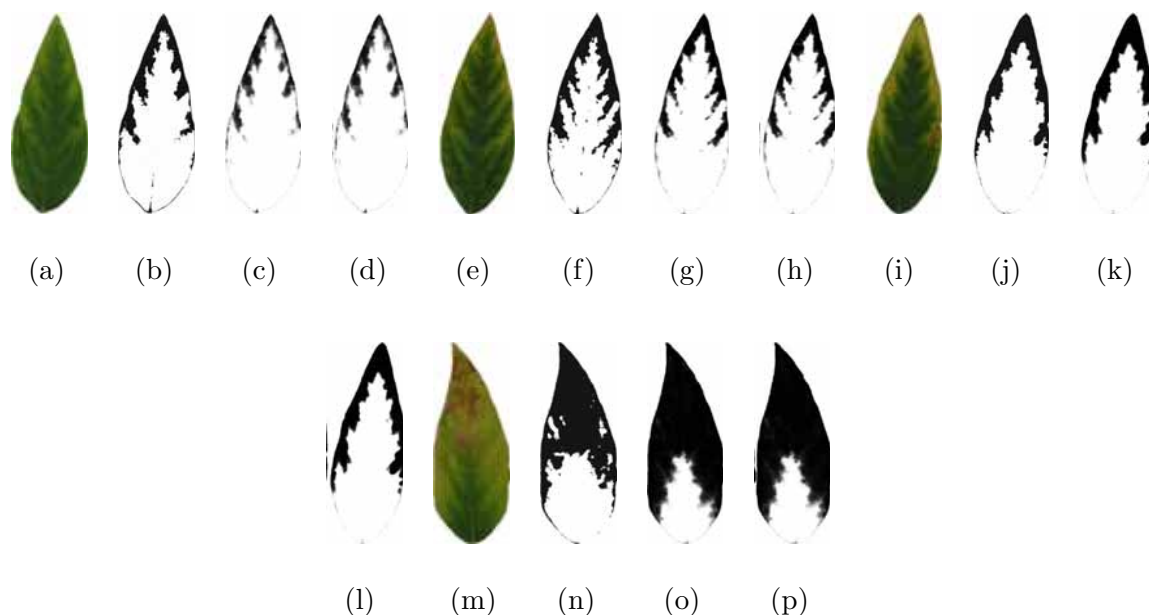
IMGs	PHM(%)		MSE		PSNR		QI		Média		FQ	
	M	F	M	F	M	F	M	F	M	F	M	F
1	99,75	99,75	0,18	0,17	79,68	79,90	0,937	0,937	9,30	9,31	9,51	9,52
2	99,77	99,78	0,15	0,14	80,56	80,74	0,937	0,937	9,35	9,36	9,54	9,55
3	99,73	99,74	0,22	0,21	78,84	79,04	0,937	0,937	9,25	9,26	9,48	9,49
4	99,98	99,98	0,04	0,03	86,51	87,15	0,937	0,937	9,71	9,75	9,76	9,79
5	99,91	99,93	0,06	0,05	84,60	84,88	0,937	0,937	9,60	9,61	9,69	9,70
6	99,96	99,97	0,05	0,04	85,68	86,24	0,937	0,937	9,66	9,70	9,73	9,75
7	98,76	98,79	1,24	1,21	71,29	71,40	0,934	0,934	8,76	8,77	9,17	9,17
8	92,03	92,30	6,14	5,88	64,35	64,54	0,917	0,918	8,11	8,13	8,62	8,64
9	98,69	98,72	1,49	1,46	70,50	70,60	0,933	0,933	8,71	8,72	9,13	9,14
10	95,84	93,50	2,78	6,00	67,79	64,45	0,921	0,913	8,45	8,15	8,91	8,67
11	97,26	92,73	1,76	8,27	69,77	63,05	0,883	0,851	8,56	7,95	8,97	8,47
12	94,99	92,34	3,40	8,22	66,91	63,08	0,832	0,825	8,26	7,91	8,69	8,42
13	95,95	92,04	2,92	9,31	67,58	62,54	0,859	0,831	8,36	7,86	8,79	8,39
14	96,63	93,84	3,06	6,82	67,37	63,90	0,921	0,908	8,44	8,11	8,92	8,65
15	95,88	91,95	3,13	9,51	67,27	62,45	0,857	0,838	8,33	7,86	8,78	8,39
16	97,24	93,33	1,75	7,21	69,80	63,65	0,921	0,901	8,61	8,07	9,03	8,61

Figura 86 – Resultados visuais das imagens das folhas independentes adquiridas em laboratório com deficiência de K, com agrupamentos de quatro imagens: original, segmentação com editor gráfico, segmentação com RNA no Matlab e segmentação com RNA no FPGA.



Fonte: Elaboração do próprio autor

**Figura 87 – Resultados visuais das imagens das folhas independentes adquiridas em laboratório sem deficiência de K, com agrupamentos de quatro imagens: original, segmentação com editor gráfico, segmentação com RNA no Matlab e segmentação com RNA no FPGA.**



Fonte: Elaboração do próprio autor

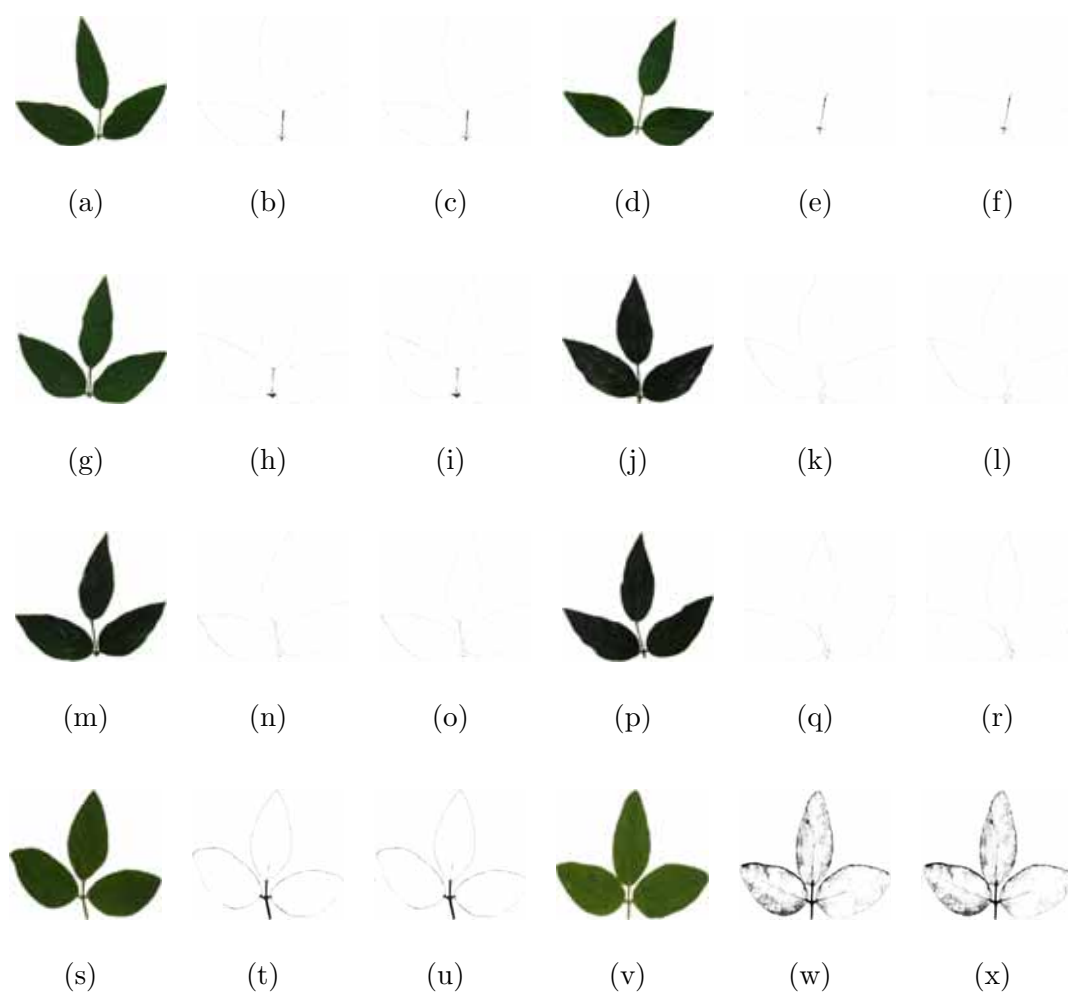
qualidade acima de 8,2, com exceção da imagem número 12 que contém uma complexidade maior da segmentação, mas com resultados satisfatórios de 7,7 no fator de qualidade.

As imagens extras possuem todas as variações analisadas de ângulos e distâncias. Os resultados mostram a capacidade da RNA na segmentação em uma larga abrangência de cenários nas Figuras 93, 94 e 95. Nestas figuras foram analisados apenas os resultados visuais em alto nível de abstração.

Em algumas imagens foram deixados de propósito alguns objetos e até uma mão, para verificar o comportamento dos resultados. Nas Figuras 93-(g;m;s) e 94-(a;g;m;s), os objetos são identificados de forma visível, principalmente a trena em amarelo fica em destaque pela similaridade com a deficiência das folhas. O reconhecimento da mão e do tripé podem-se tornar um exemplo de uma RNA com uma maior quantidade de saídas para classificar outros tipos de objetos.

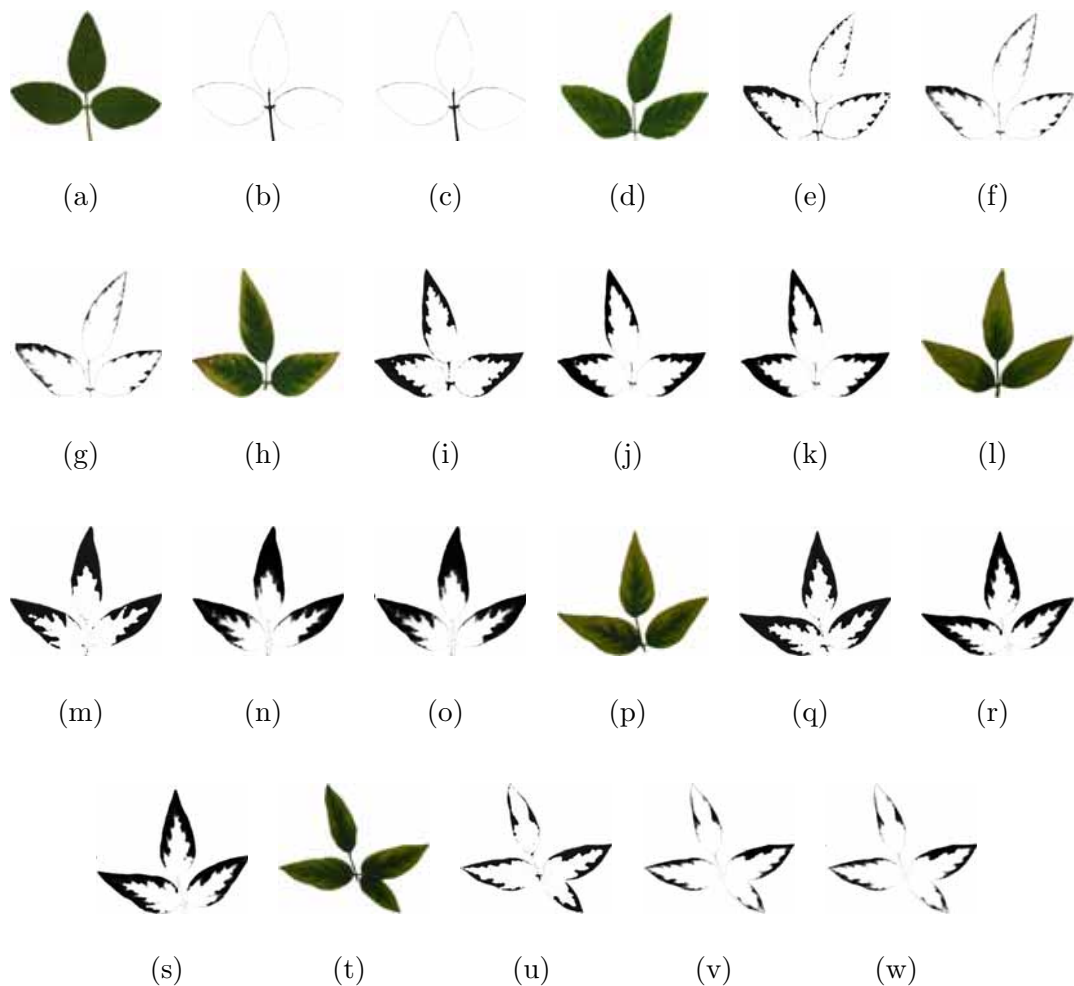
Nas Figuras 91, 92, 93, 94 e 95, referentes às imagens em campo (externas), obtiveram-se também excelentes resultados. A diversidade de tipos de imagens nestas figuras trazem algumas importantes definições. As Figuras 93 e 94-(a-l) sem deficiência de K não ocorreram a identificação de píxeis, com exceção dos objetos explicados, apenas a Figura 93-(k;q) que a segmentação foi maior por causa da coloração diferenciada em cinza e roxo nas folhas. Já, nas imagens com deficiência, a segmentação está bem definida

Figura 88 – Resultados visuais das imagens dos trifólios adquiridas em laboratório sem deficiência de K, com agrupamentos de três imagens: original, segmentação com RNA no Matlab e segmentação com RNA no FPGA. As imagens não têm deficiência de K, a segmentação no editor gráfico é uma imagem totalmente branca.



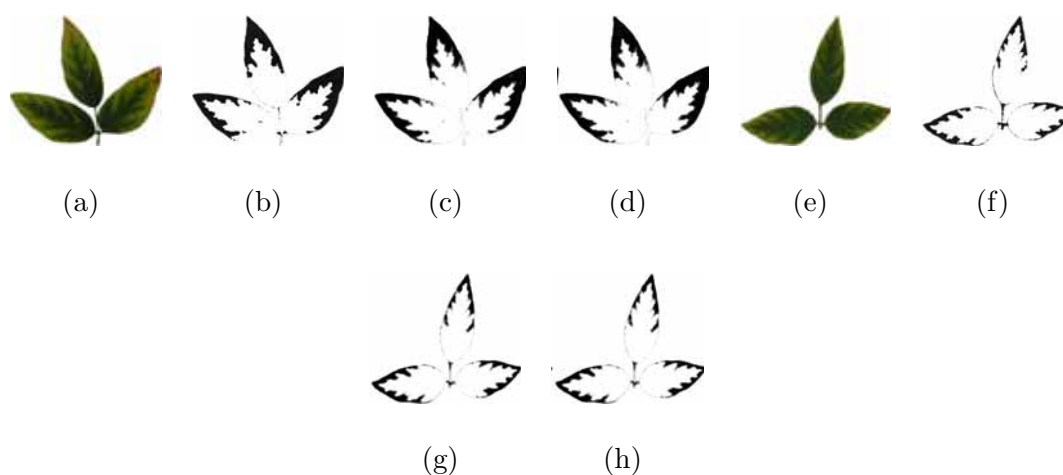
Fonte: Elaboração do próprio autor

Figura 89 – Resultados visuais das imagens dos trifólios adquiridas em laboratório sem deficiência de K, com agrupamentos de quatro imagens: original, segmentação com editor gráfico, segmentação com RNA no Matlab e segmentação com RNA no FPGA. Apenas o primeiro agrupamento contém três imagens, porque não tem deficiência de K e a imagem da segmentação com editor gráfico foi retirada.



Fonte: Elaboração do próprio autor

Figura 90 – Resultados visuais das imagens dos trifólios adquiridas em laboratório sem deficiência de K, com agrupamentos de quatro imagens: original, segmentação com editor gráfico, segmentação com RNA no Matlab e segmentação com RNA no FPGA.



Fonte: Elaboração do próprio autor

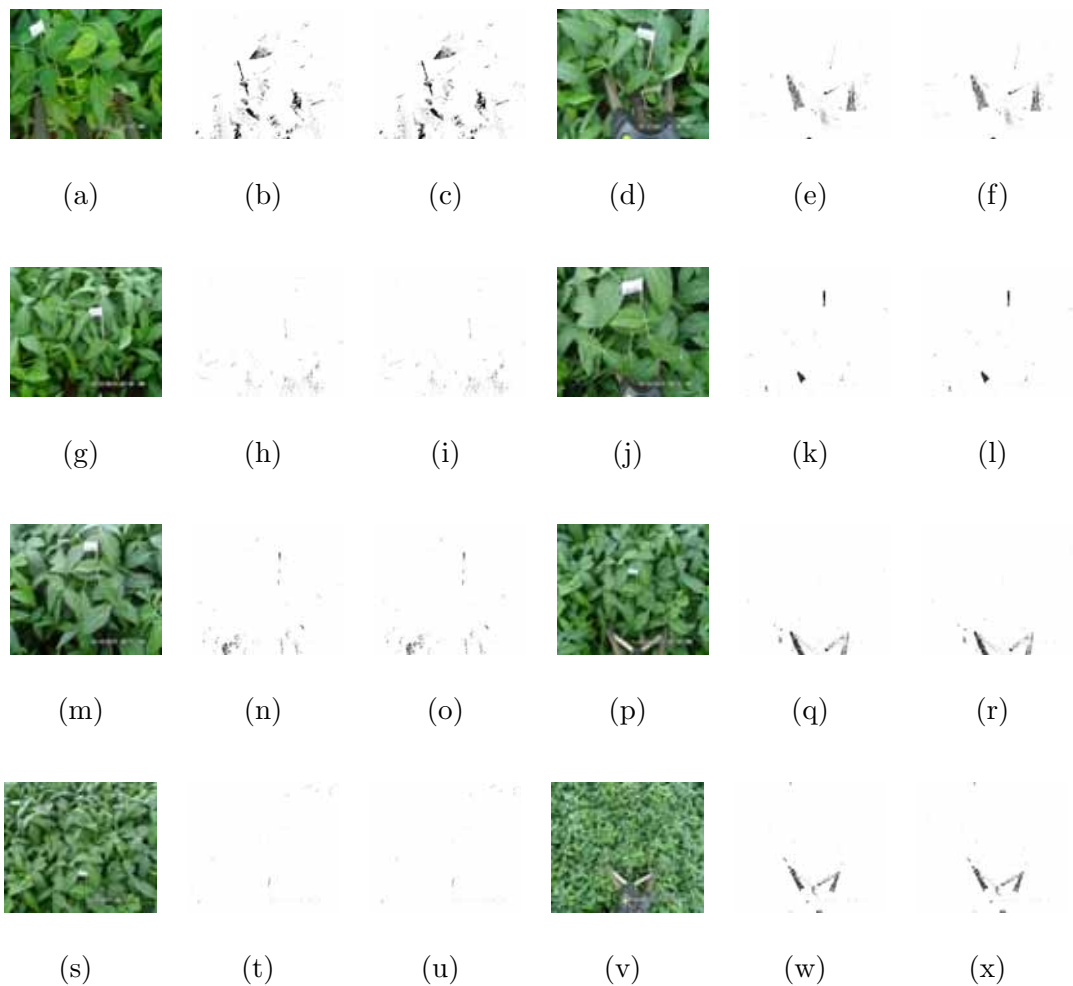
Tabela 52 – Resultados encontrados nas imagens externas em níveis de abstrações, alto (M-Matlab) e baixo (F-FPGA).

IMGs	PHM(%)		MSE		PSNR		QI		Média		FQ	
	M	F	M	F	M	F	M	F	M	F	M	F
1	96,04	96,08	4,60	3,94	65,60	66,28	0,926	0,926	8,31	8,36	8,83	8,86
2	97,62	97,63	1,39	1,37	70,79	70,85	0,931	0,931	8,70	8,70	9,10	9,10
3	99,55	99,59	0,32	0,30	77,13	77,46	0,937	0,936	9,14	9,16	9,41	9,42
4	99,44	99,45	0,66	0,65	74,03	74,11	0,935	0,935	8,95	8,95	9,29	9,30
5	99,11	99,16	0,63	0,59	74,26	74,54	0,936	0,936	8,95	8,97	9,29	9,30
6	98,59	98,60	1,12	1,10	71,74	71,80	0,934	0,934	8,78	8,79	9,18	9,18
7	99,94	99,96	0,04	0,04	86,28	86,53	0,937	0,937	9,70	9,71	9,75	9,76
8	98,44	98,46	1,13	1,11	71,71	71,76	0,933	0,933	8,78	8,78	9,17	9,17
9	99,96	99,97	0,03	0,03	87,25	87,46	0,937	0,937	9,76	9,77	9,79	9,80
10	90,01	90,14	10,67	10,44	61,95	62,04	0,904	0,904	7,86	7,87	8,42	8,43
11	88,63	88,77	11,23	11,05	61,72	61,80	0,910	0,910	7,81	7,82	8,37	8,38
12	79,54	72,75	18,39	30,35	59,58	57,41	0,833	0,811	7,29	6,85	7,80	7,37
13	94,37	94,36	9,21	8,57	62,59	62,90	0,911	0,911	8,03	8,05	8,61	8,63

Tabela 53 – Média dos Resultados com os três tipos de imagens em níveis de abstrações, alto (M-Matlab) e baixo (F-FPGA).

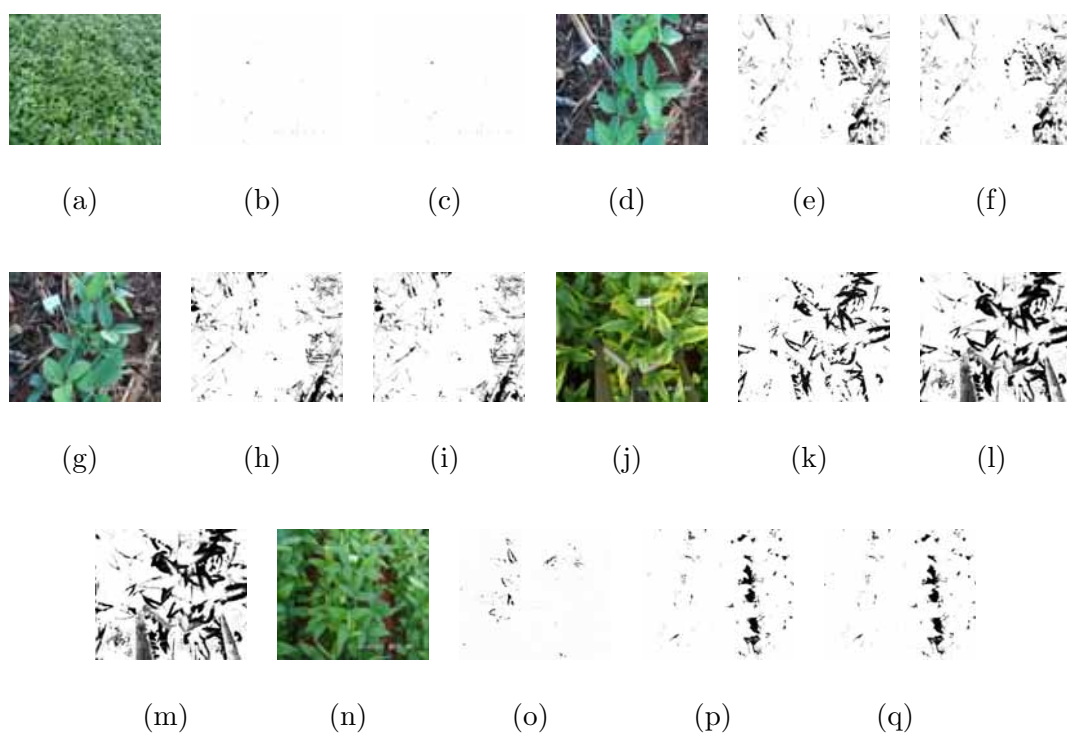
IMGs	PHM(%)		MSE		PSNR		QI		Média		FQ	
	M	F	M	F	M	F	M	F	M	F	M	F
Folha	94,53	91,63	5,93	10,78	70,90	69,23	0,797	0,789	8,41	8,18	8,74	8,53
Trifólio	97,65	96,17	1,77	4,03	73,66	71,73	0,913	0,905	8,84	8,66	9,17	9,02
Externa	95,48	94,99	4,57	5,35	71,13	71,15	0,920	0,919	8,62	8,60	9,00	8,98

Figura 91 – Resultados visuais das imagens externas com e sem sintomas de deficiência de K na soja, com agrupamentos de três imagens: original, segmentação com RNA no Matlab e segmentação com RNA no FPGA. As imagens não têm deficiência de K, a segmentação no editor gráfico é uma imagem totalmente branca.



Fonte: Elaboração do próprio autor

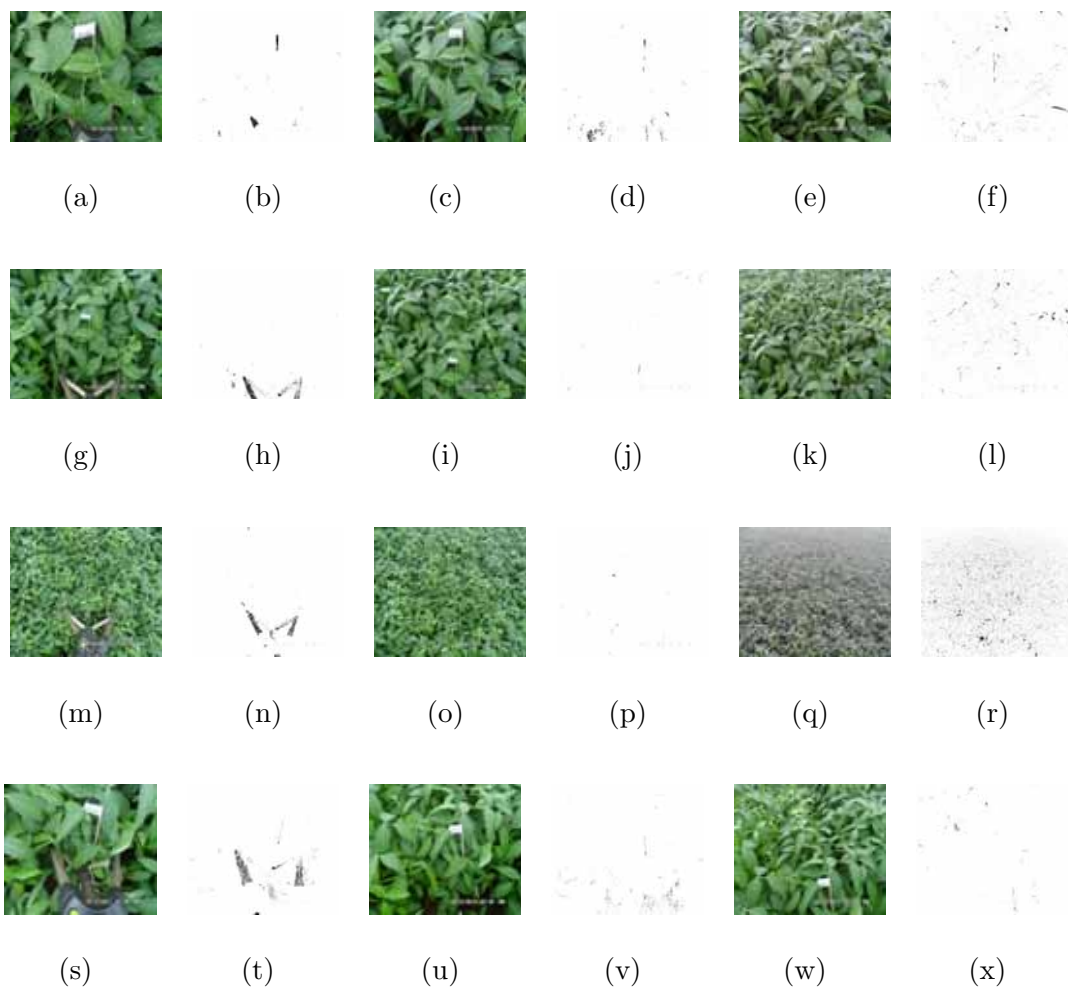
Figura 92 – Resultados visuais das imagens externas com e sem sintomas de deficiência de K na soja, com agrupamentos de quatro imagens: original, segmentação com editor gráfico, segmentação com RNA no Matlab e segmentação com RNA no FPGA. Apenas os dois últimos agrupamentos contém as quatro imagens, os outros foram retiradas a segmentação com editor gráfico por ser uma imagem sem deficiência, totalmente em branco.



Fonte: Elaboração do próprio autor

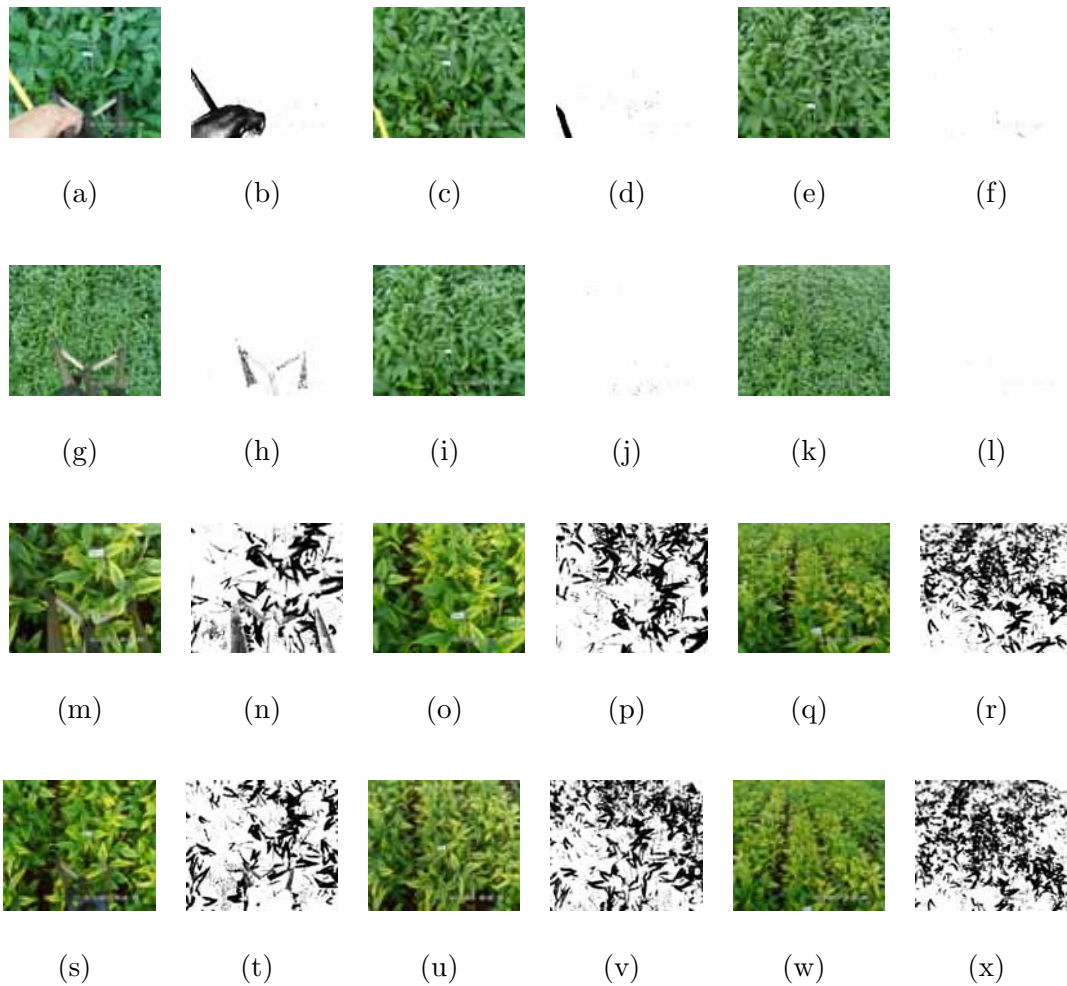
independentemente da distância ou ângulo, conforme apresentado nas Figuras 94-(m-w) e 95. A Figura 94-(m-w) contém imagens com distâncias de 30 e 60 cm, e ângulos dos três tipos. Na Figura 95-(a) observa-se uma segmentação de parte do solo, não esperado, na parte superior direita da imagem de 90 graus com distância de 290 cm, isso pode ser contornado com agregação de outros métodos junto com a RNA em trabalhos futuros. As imagens da Figura 95-(c;e) contêm resultados melhores com ângulos de 45 e 60 graus e a mesma distância. No lado esquerdo das imagens, verifica-se de forma clara, que não houve segmentação da parte mais verde das imagens.

**Figura 93 – Resultados visuais das imagens externas com e sem deficiência de K, com agrupamentos de duas imagens: Original e segmentação com RNA no Matlab.**



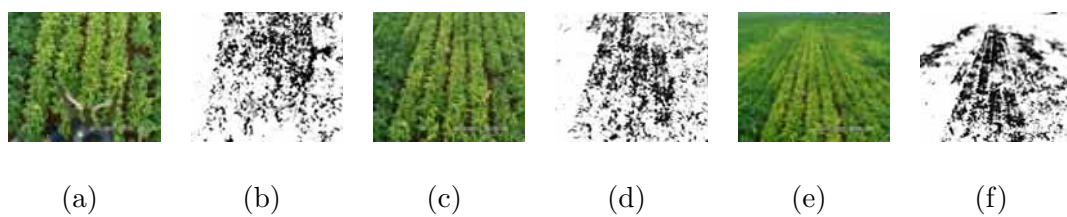
Fonte: Elaboração do próprio autor

**Figura 94 – Resultados visuais das imagens externas com e sem deficiência de K, com agrupamentos de duas imagens: Original e segmentação com RNA no Matlab.**



Fonte: Elaboração do próprio autor

**Figura 95 – Resultados visuais das imagens externas com e sem deficiência de K, com agrupamentos de duas imagens: Original e segmentação com RNA no Matlab.**



Fonte: Elaboração do próprio autor

## 8.4 Conclusão deste Capítulo

Neste estudo foi comprovada a utilização da RNA para a segmentação de imagens na identificação de deficiência do macronutriente K pela folha da soja. O sistema mostra-se robusto mesmo com diversos tipos de imagens analisadas em laboratório e campo. A comparação entre os sistemas em níveis de abstrações, alto e baixo, foram investigados em vários aspectos de qualidade na segmentação de imagens.

As imagens adquiridas em laboratório, sem plano de fundo das folhas individuais, obtiveram resultados adequados na identificação da deficiência do macronutriente K, mas piores que os trifólios e as imagens externas. As folhas individuais contêm um nível de detalhes maior do que os outros tipos de imagens. A análise principalmente das folhas com deficiência pode-se observar em alguns casos uma segmentação com a RNA melhor do que a própria imagem desejada, feita com editor gráfico. Assim, todas as imagens desejadas utilizadas no treinamento e validação dos dados geradas por editor gráfico, não são exatamente um modelo ideal, por ter sido feito de forma manual baseado na inferência do software com substituição de cores, conforme um determinado limite pelo modelo de cores HSI, adotado na ferramenta do próprio editor gráfico. Em vários momentos houve a necessidade de limpezas manuais, com ferramentas de edição, das partes não encontradas pelo software tornando o processo lento e trabalhoso. A deficiência encontrada em cada planta tem a cor amarelada de forma semelhante, mas nunca igual.

## 9 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho apresentou-se um novo sistema em baixo nível de abstração para a identificação da falta do macronutriente K pela folha da soja. O sistema desenvolvido em dispositivo reconfigurável foi comparado ao Matlab com resultados semelhantes em vários parâmetros de qualidade. O sistema da RNA em FPGA apresentou resultados com alta precisão nos dados, a média da porcentagem de acerto nas imagens das folhas, trifólios e ambiente externo são de 92%, 96% e 95%, respectivamente. O fator de qualidade nas médias dos três tipos de imagens foram acima de 8,5 e as principais características investigadas para alcançar o objetivo foram:

1. A utilização de ponto flutuante no desenvolvimento de RNA em hardware;
2. A composição entre diversos métodos para a implementação da função de ativação não linear Tangente Hiperbólica em hardware;
3. A construção de uma arquitetura de hardware para a RNA MLP em dispositivo reconfigurável;
4. A investigação do processamento digital de imagens e seus métodos para a preparação e segmentação das imagens;
5. O desenvolvimento dos estudos em distintos níveis de abstrações, alto e baixo;
6. Pesquisa de ferramentas, métodos e processos em referências que correspondem ao estado da arte nos temas abordados;
7. A análise dos resultados em todos os estudos de casos com parâmetros de desempenho, área utilizada na FPGA, erros, fator de qualidade para a segmentação, entre outros.

Os estudos proporcionaram uma investigação profunda das técnicas e métodos para o desenvolvimento de RNA em dispositivo reconfigurável. O primeiro estudo, apresentado no Capítulo 4, teve grande relevância na verificação das inúmeras possibilidades de organização do sistema e de utilização de IP *Cores*. Com esse experimento, a

identificação de uma estrutura adequada ao sistema foi avaliado para efetuar otimização nos componentes corretos e ao mesmo tempo garantir a estabilidade do sistema.

O segundo estudo, apresentado no Capítulo 5, disponibilizou uma arquitetura adequada para execução de RNA MLP em hardware com função de ativação não linear. O sistema foi definido em partes funcionais e determinantes, como a unidade de controle, armazenamento e processamento.

A disponibilização da arquitetura do sistema em vários níveis proporcionou uma estrutura mais organizada e controlada. Por exemplo, o sistema do primeiro estudo tem uma arquitetura mais simples, com AF linear, um neurônio e uma camada a menos. Porém, o uso do IP *Core* da Xilinx não foi possível ser prototipado com multiplicadores dedicados, diferente do sistema do segundo experimento, que contém AF não linear, mais neurônios e mais camadas, e todos os cenários utilizaram tais multiplicadores. Todos os resultados mostrados no segundo estudo foram com a mesma plataforma. Isso indica a evolução do sistema na implementação da RNA.

A complexidade na aproximação da AF Tangente Hiperbólica é uma característica de extrema importância no sistema e foi verificada com uma ampla quantidade de métodos possíveis para o segundo estudo. Na aproximação de uma função não linear, sistema completo com a RNA, e da AF Tangente Hiperbólica, obtiveram-se precisões adequadas nos dois casos. Isso mostrou, a capacidade do sistema em atuar com qualidade na aplicação proposta neste trabalho.

O terceiro estudo, apresentado no Capítulo 6, contribuiu na investigação de métodos para a segmentação de imagens com a limiarização global, e múltiplos limiares com método de Otsu e a RNA. Diversas definições promissoras junto ao sistema para processamento de imagem foram abordadas. As investigações na segmentação com a RNA foram destaque com o uso de filtros lineares de média simples e ponderada com várias máscaras, a aplicação de um sistema com duas RNAs, o comportamento dos filtros e RNAs com a inserção de ruídos, e a variação de quantidade e tipos de entradas. Nesse estudo foi possível consolidar o uso de RNA, independentemente da existência de ruídos nas imagens, e suas principais características de entradas, neurônios e imagens analisadas.

O quarto estudo, apresentado no Capítulo 7, visou fortalecer os parâmetros ideais para a segmentação de imagens com RNA em outra cultura, a do algodão. O uso de amostras reais e confirmadas com as deficiências de potássio e fósforo foram analisadas na cultura do algodão. O estudo apresentou diversos parâmetros de qualidade para analisar de forma mais adequada as segmentações de imagens.

No último estudo, apresentado no Capítulo 8 tem-se o sistema final em níveis de abstrações, alto e baixo, para a segmentação de imagens com RNA. As imagens foram adquiridas na cultura da soja com a deficiência de potássio. Os resultados alcançados

foram promissores e com valores excelentes nos fatores de qualidade apresentados.

O sistema em alto nível de abstração para a segmentação de imagens na identificação da deficiência do macronutriente de potássio pela folha da soja é exclusivo na literatura atual. O desenvolvimento em baixo nível de abstração em dispositivo reconfigurável, com ponto flutuante e função de ativação por método híbrido (HPC), são características únicas e inexistentes em trabalhos relatados na literatura. A grande abrangência de trabalhos relacionados analisados no Capítulo 3 mostra o impacto significativo deste trabalho nas pesquisas atuais.

### 9.1 Sugestões para Trabalhos Futuros

A continuidade desta pesquisa pode ser determinada em vários caminhos em níveis de abstrações, alto e baixo. Atualmente, os pensamentos estão voltados ao aprimoramento da plataforma de baixo nível como:

1. o estudo de outros métodos de processamento de imagens para auxiliar a segmentação com a RNA;
2. A atualização do sistema de baixo nível para uma plataforma FPGA com mais recursos;
3. O desenvolvimento de um módulo de baixo nível para a utilização de memória DDR;
4. O desenvolvimento de um módulo de baixo nível para a aquisição de imagens em tempo real;
5. O acoplamento entre memória, aquisição de imagens e o sistema da RNA em baixo nível;
6. Pesquisa de ferramentas, métodos e processos em referências que correspondem ao estado da arte nos temas abordados;
7. A análise dos resultados em todos os estudos de casos com parâmetros de desempenho, área utilizada na FPGA, erros, fator de qualidade para a segmentação, entre outros.

Existem diversos estudos a serem evidenciados em alto nível de abstração, o principal seria o desenvolvimento de um sistema com outro tipo de RNA, como a ArtMap, e posteriormente o seu transporte para um dispositivo reconfigurável.

## REFERÊNCIAS

- ABDULLAH, N. et al. Classification of rubber tree leaf diseases using multilayer perceptron neural network. In: STUDENT CONFERENCE ON RESEARCH AND DEVELOPMENT - SCORED, 5., 2007, Selangor. **Proceedings...** Selangor: IEEE, 2007. p. 1–6.
- AL-KAZZAZ, S.; KHALIL, R. FPGA implementation of artificial neurons: comparison study. In: INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGIES: FROM THEORY TO APPLICATIONS - ICTTA, 3., 2008, Damascus. **Proceedings...** Damascus: IEEE, 2008. p. 1–6.
- ALI, H. K.; MOHAMMED, E. Z. Design artificial neural network using FPGA. **International Journal of Computer Science and Network Security**, Seoul, v. 10, n. 8, p. 88–92, 2010.
- ALIPPI, C.; STORTI-GAJANI, G. Simple approximation of sigmoidal functions: realistic design of digital neural networks capable of learning. In: INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS - ISCAS, 11., 1991, Singapore. **Proceedings...** Singapore: IEEE, 1991. v. 3, p. 1505–1508.
- ALTERA. **Floating-point megafunctions**: user guide. San Jose: Altera, 2011.
- ALTERA. **Clock control block (ALTCLKCTRL) megafunction**: user guide. San Jose: Altera, 2012.
- AMIN, H.; CURTIS, K.; HAYES-GILL, B. Piecewise linear approximation applied to nonlinear function of a neural network. **IEEE Proceedings Circuits, Devices and Systems**, Stevenage, v. 144, n. 6, p. 313–317, dez. 1997. ISSN 1350-2409.
- BAHOURA, M.; PARK, C.-W. FPGA-implementation of high-speed MLP neural network. In: INTERNATIONAL CONFERENCE ON ELECTRONICS, CIRCUITS AND SYSTEMS - ICECS, 18., 2011, Beirut. **Proceedings...** Beirut: IEEE, 2011. p. 426–429.
- BAJGER, M.; OMONDI, A. Low-error, high-speed approximation of the sigmoid function for large FPGA implementations. **Journal of Signal Processing Systems**, Hingham, v. 52, n. 2, p. 137–151, ago. 2008. ISSN 1939-8018.
- BEAUCHAMP, M. J.; HAUCK, S. Architectural modifications to improve floating-point unit efficiency in FPGAs. In: SYMPOSIUM ON FIELD-PROGRAMMABLE CUSTOM COMPUTING MACHINES - FCCM, 14., 2006, Napa Valley. **Proceedings...** Napa Valley: IEEE, 2006. p. 515–520.
- BENJEMMAA, A. et al. Implementations approaches of neural networks lane following system. In: MEDITERRANEAN ELECTROTECHNICAL CONFERENCE - MELECON, 16., 2012, Yasmine Hammamet. **Proceedings...** Yasmine Hammamet: IEEE, 2012. p. 515–518. ISSN 2158-8473.

- BERNAL, M. et al. Foliar and root cu supply affect differently fe- and zn-uptake and photosynthetic activity in soybean plants. **Environmental and Experimental Botany**, Amsterdam, v. 60, n. 2, p. 145 – 150, 2007. ISSN 0098-8472.
- BOCCO, M. et al. Estimating soybean ground cover from satellite images using neural-networks models. **International Journal of Remote Sensing**, United Kingdom, v. 33, n. 6, p. 1717–1728, mar. 2012.
- BONALDO, M. S.; RIEDO, I. C.; LIMA, A. R. Monitoramento e diagnóstico de doenças foliares da cultura da soja na região COMCAM na safra 2007/2008. **Campo Digital**, Campo Mourão, v. 4, n. 1, p. 127–136, 2009.
- BONNICI, M. et al. Artificial neural network optimization for FPGA. In: INTERNATIONAL CONFERENCE ON ELECTRONICS, CIRCUITS AND SYSTEMS - ICECS, 13., 2006, Nice. **Proceedings...** Nice: IEEE, 2006. p. 1340–1343.
- BURGOS-ARTIZZU, X. P. et al. Real-time image processing for crop/weed discrimination in maize fields. **Computers and Electronics in Agriculture**, Amsterdam, v. 75, n. 2, p. 337 – 346, 2011. ISSN 0168-1699.
- CALDERANO FILHO, B. et al. Predição de classes de solos de paisagens montanhosas da serra do mar, com o uso de redes neurais artificiais (RNAs). In: SIMPÓSIO BRASILEIRO DE SENSORIAMENTO REMOTO - SBSR, 16., 2013, Foz do Iguaçu. **Anais...** Foz do Iguaçu: Inpe, 2013. p. 3916–3923.
- CARVALHO, M. C. S.; FERREIRA, G. B. **Calagem e adubação do algodoeiro no cerrado**. Campina Grande: Embrapa, 2006.
- CHRISTY, N. A.; KARTHIGAIAKUMAR, P. FPGA implementation of AES algorithm using composite field arithmetic. In: INTERNATIONAL CONFERENCE ON DEVICES, CIRCUITS AND SYSTEMS - ICDCS, 32., 2012, Macau. **Proceedings...** Macau: IEEE, 2012. p. 713–717. Borkar.
- COFER, R. C.; HARDING, B. **Rapid system prototyping with FPGAs: accelerating the design process**. Burlington: Elsevier, 2006.
- COMPANHIA NACIONAL DE ABASTECIMENTO - CONAB. **Acompanhamento de safra brasileira: grãos. Décimo levantamento**. Brasília: Conab, 2013.
- CRUVINEL, P. E. **Agronegócio e oportunidades para o desenvolvimento sustentável do Brasil**. São Carlos: Embrapa, 2009.
- CUI, D. et al. Image processing methods for quantitatively detecting soybean rust from multispectral images. **Biosystems Engineering**, London, v. 107, n. 3, p. 186–193, 2010. ISSN 1537-5110.
- DAGENAIS, M.; AGARWAL, V.; RUMIN, N. Mcboole: A new procedure for exact logic minimization. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, New York, v. 5, n. 1, p. 229–238, January 1986. ISSN 0278-0070.
- DAMAK, A.; KRID, M.; MASMOUDI, D. Neural network based edge detection with pulse mode operations and floating point format precision. In: INTERNATIONAL

CONFERENCE ON DESIGN AND TECHNOLOGY OF INTEGRATED SYSTEMS IN NANOSCALE ERA - DTIS, 3., 2008, Tozeur. **Proceedings...** Tozeur: IEEE, 2008. p. 1–5.

DELAMEA, R. B. C. **Eficiência da adubação à taxa variável em áreas manejadas com agricultura de precisão no Rio Grande do Sul.** 2008. 161 f. Dissertação (Mestrado) — Universidade Federal de Santa Maria, Centro de Ciências Rurais, Programa de Pós-Graduação em Ciência do Solo, Santa Maria, 2008.

DOSSA, D. et al. **Brasil: projeções do agronegócio 2011/2012 a 2021/2022.** Brasília: MAPA, 2012.

EDDY, P. et al. Hybrid segmentation – artificial neural network classification of high resolution hyperspectral imagery for site-specific herbicide management in agriculture. **Photogrammetric Engineering & Remote Sensing**, Cambridge, v. 74, n. 10, p. 1249–1257, out. 2008.

EINSTEIN, A. **Ideas and opinions.** New York: Crown Publishers, 1954.

FANG, J.; MA, L.; CHEN, Y. Color analysis of soybean leaves based on computer vision. In: SECOND WORLD CONGRESS ON SOFTWARE ENGINEERING - WCSE, 4., 2010, Dubai. **Proceedings...** Dubai: IEEE, 2010. v. 1, p. 221–224.

FARIAS, J. R. B.; NEPOMUCENO, A. L.; NEUMAIER, N. **Ecofisiologia da Soja.** Londrina: Embrapa, 2007. (Circular Técnica, 48).

FERREIRA, A. do A.; BARROS, E. da S. A high performance full pipelined architecture of MLP neural networks in FPGA. In: INTERNATIONAL CONFERENCE ON ELECTRONICS, CIRCUITS, AND SYSTEMS - ICECS, 17., 2010, Athens. **Proceedings...** Athens: IEEE, 2010. p. 742–745.

FU, H.; CHI, Z. Combined thresholding and neural network approach for vein pattern extraction from leaf images. **IEE Proceedings - Vision, Image and Signal Processing**, Stevenage, v. 153, n. 6, p. 881–892, Dec 2006. ISSN 1350-245X.

GONZALEZ, R. C.; WOODS, R. C. **Processamento digital de imagens.** 3. ed. São Paulo: Pearson Prentice Hall, 2010.

GU, Y.; VANCOURT, T.; HERBORDT, M. C. Explicit design of FPGA-based coprocessors for short-range force computations in molecular dynamics simulations. **Parallel Computing**, Amsterdam, v. 34, n. 4-5, p. 261–277, mai. 2008. ISSN 0167-8191.

GUILLERMO, M. **Floating point adder and multiplier:** Overview. [S.l.], 2010. Disponível em: <<http://opencores.org>>. Acesso em: 18 Ago. 2012.

HARTENSTEIN, R. A decade of reconfigurable computer: A visionary retrospective. In: CONFERENCE ON DESIGN, AUTOMATION AND TEST IN EUROPE - DATE, 4., 2001, Munich. **Proceedings...** Munich: IEEE, 2001. p. 642–649.

HAVEL, V.; VLCEK, K. Computation of a nonlinear squashing function in digital neural networks. In: WORKSHOP ON DESIGN AND DIAGNOSTICS OF ELECTRONIC CIRCUITS AND SYSTEMS - DDECS, 11., 2008, Bratislava. **Proceedings...** Bratislava: IEEE, 2008. p. 1–4.

- HAYKIN, S. S. **Redes neurais: princípio e prática**. Singapore: Bookman, 2001.
- HENNING, A. A. Manejo de doenças da soja(*glycine max l. merrill*). **Informativo ABRATES**, Londrina, v. 19, n. 3, p. 09–12, 2009.
- HERBORDT, M. C. et al. Achieving high performance with fpga-based computing. **Computer**, Los Alamitos, v. 40, n. 3, p. 50–57, mar. 2007. ISSN 0018-9162.
- HOFFMAN, M. et al. Hardware synthesis of artificial neural networks using field programmable gate arrays and fixed-point numbers. In: IEEE REGION 5 CONFERENCE, 2006, San Antonio. **Proceedings...** San Antonio: IEEE, 2006. p. 324–328.
- HUSIN, Z. et al. Embedded portable device for herb leaves recognition using image processing techniques and neural network algorithm. **Computers and Electronics in Agriculture**, Amsterdam, v. 89, n. 0, p. 18 – 29, 2012. ISSN 0168-1699.
- IEEE COMPUTER SOCIETY. **IEEE standard for floating-point arithmetic**. New York, ago. 2008. 1-58 p. Disponível em: <<http://dx.doi.org/10.1109/IEEESTD.2008.4610935>>. Acesso em: 17 Fev. 2013.
- INAMASU, R. Y. et al. **Agricultura de precisão: um novo olhar**. São Carlos: Embrapa Instrumentação, 2011. 334 p.
- JIDAN, A. E. **FPU: overview**. [S.l.: s.n.], 2011. Disponível em: <<http://opencores.org>>. Acesso em: 18 Ago. 2012.
- JUCA, V.; MELLO, C. Texture-based segmentation using image fidelity indexes. **IEEE (Revista IEEE America Latina) Latin America Transactions**, Piscataway, v. 9, n. 3, p. 423–428, June 2011. ISSN 1548-0992.
- KARSOLIYA, S. Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture. **International Journal of Engineering Trends and Technology**, India, v. 3, n. 6, p. 714–717, 2012. ISSN 2231-5381.
- KHODJA, D. E.; KHELDOUN, A.; REFOUFI, L. Sigmoid function approximation for ANN implementation in FPGA devices. In: WSEAS INTERNATIONAL CONFERENCE ON CIRCUITS, SYSTEMS, ELECTRONICS, CONTROL; SIGNAL PROCESSING - CSECS, 9., 2010, Athens. **Proceedings...** Athens: World Scientific and Engineering Academy and Society - WSEAS, 2010. p. 112–116. ISBN 978-960-474-262-2.
- KHRAISHA, R.; LEE, J. A scalable H.264/AVC deblocking filter architecture using dynamic partial reconfiguration. In: INTERNATIONAL CONFERENCE ON ACOUSTICS SPEECH AND SIGNAL PROCESSING - ICASSP, 3., 2010, Dallas. **Proceedings...** Dallas: IEEE, 2010. p. 1566–1569. ISSN 1520-6149.
- KLOCK, C. E. et al. Karma: um ambiente para o aprendizado de síntese de funções booleanas. **Revista Brasileira de Informática na Educação**, Porto Alegre, v. 18, n. 2, p. 33–42, 2010.
- KOSAKA, N. et al. Monitoring of bacterial pustule on soybean by neural network using hyperspectral data. In: INTERNATIONAL CONFERENCE ON GEOSCIENCE AND REMOTE SENSING SYMPOSIUM - IGARSS, 3., 2006, Denver. **Proceedings...** Denver: IEEE, 2006. p. 3879–3882.

KULKARNI, A. et al. A leaf recognition technique for plant classification using rbpnn and zernike moments. **International Journal of Advanced Research in Computer and Communication Engineering**, India, v. 2, n. 1, p. 984–988, 2013.

LEBOEUF, K.; MUSCEDERE, R.; AHMADI, M. Performance analysis of table-based approximations of the hyperbolic tangent activation function. In: INTERNATIONAL MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS - MWSCAS, 54., 2011, Seoul. **Proceedings...** Seoul: IEEE, 2011. p. 1–4. ISSN 1548-3746.

LEBOEUF, K. et al. High speed VLSI implementation of the hyperbolic tangent sigmoid function. In: INTERNATIONAL CONFERENCE ON CONVERGENCE AND HYBRID INFORMATION TECHNOLOGY - ICCIT, 3., 2008, Busan. **Proceedings...** Busan: IEEE, 2008. v. 1, p. 1070–1073.

LEE, Y.; KO, S.-B. FPGA implementation of a face detector using neural networks. In: CANADIAN CONFERENCE ON ELECTRICAL AND COMPUTER ENGINEERING - CCECE, 19., 2006, Ottawa. **Proceedings...** Ottawa: IEEE, 2006. p. 1914–1917.

LI-SHU, W. Soybean leaves nitrogen elements information collected preprocessing based on model identification. In: INTERNATIONAL CONFERENCE ON ADVANCED COMPUTER THEORY AND ENGINEERING - ICACTE, 3., 2010, Chengdu. **Proceedings...** Chengdu: IEEE, 2010. v. 6, p. 222–226. ISSN 2154-7491.

LIN, C.-W.; WANG, J.-S. A digital circuit design of hyperbolic tangent sigmoid function for neural networks. In: INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS - ISCAS, 35., 2008, Beijing. **Proceedings...** Beijing: IEEE, 2008. p. 856–859.

LISHU, W. Identification based on color and texture of the soybean leaf nitrogen diagnostic model. In: CHINESE CONTROL CONFERENCE - CCC, 29., 2010, Beijing. **Proceedings...** Beijing: IEEE, 2010. p. 2849–2852.

LITWILLER, D. **CCD vs. CMOS: facts and fiction**. Waterloo: DALSA, 2001. Disponível em: <<https://www.teledynedalsa.com>>. Acesso em: 15 out. 2012.

LIU, J.; PATTEY, E. Retrieval of leaf area index from top-of-canopy digital photography over agricultural crops. **Agricultural and Forest Meteorology**, Amsterdam, v. 150, n. 11, p. 1485 – 1490, 2010. ISSN 0168-1923.

MA, X.; GUAN, H.; TAN, F. Investigation on the extraction of soybean brown spot based on improved genetic algorithm. In: INTERNATIONAL CONFERENCE OF INFORMATION SCIENCE AND MANAGEMENT ENGINEERING - ICMSE, 2., 2010, Melbourne. **Proceedings...** Melbourne: IEEE Computer Society, 2010. v. 1, p. 14–17. ISBN 978-0-7695-4132-7.

MARCUS, G. et al. A fully synthesizable single-precision, floating-point adder/subtractor and multiplier in VHDL for general and educational use. In: INTERNATIONAL CARACAS CONFERENCE ON DEVICES, CIRCUITS AND SYSTEMS - ICCDCS, 5., 2004, Punta Cana. **Proceedings...** Punta Cana: IEEE, 2004. v. 1, p. 319–323.

MARQUES FILHO, O.; VIEIRA NETO, H. **Processamento digital de imagens**. Rio de Janeiro: Pearson Prentice Hall, 1999.

MCCULLOCH, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, Chicago, v. 5, p. 115–133, 1943. ISSN 0007-4985.

MEHER, P. An optimized lookup-table for the evaluation of sigmoid function for artificial neural networks. In: VLSI SYSTEM ON CHIP CONFERENCE - VLSI-SOC, 18., 2010, Madrid. **Proceedings...** Madrid: IEEE/IFIP, 2010. p. 91–95.

MELLO, B. A.; CAIMI, L. L. Simulação na validação de sistemas computacionais para a agricultura de precisão. **Revista Brasileira de Engenharia Agrícola e Ambiental**, Campina Grande, v. 12, n. 6, p. 666–675, abr. 2008.

MISHRA, A.; ZAHEERUDDIN; RAJ, K. Implementation of a digital neuron with nonlinear activation function using piecewise linear approximation technique. In: INTERNATONAL CONFERENCE ON MICROELECTRONICS - ICM, 19., 2007, Cairo. **Proceedings...** Cairo: IEEE, 2007. p. 69–72.

MOGHADDAM, P. A.; DERAFSHI, M. H.; SHIRZAD, V. Estimation of single leaf chlorophyll content in sugar beet using machine vision. **Turkish Journal of Agriculture and Forestry**, Ankara, v. 35, n. 6, p. 563–568, dec. 2011.

MORAES, P. V. D. de et al. Agricultura de precisão no controle de plantas daninhas. **Revista da FZVA**, Uruguaiana, v. 15, n. 1, p. 01–14, 2008.

MYERS, D.; HUTCHINSON, R. Efficient implementation of piecewise linear activation function for digital vlsi neural networks. **Electronics Letters**, Stevenage, v. 25, n. 24, p. 1662 –1663, nov. 1989. ISSN 0013-5194.

NAMIN, A. et al. Efficient hardware implementation of the hyperbolic tangent sigmoid function. In: INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS - ISCAS, 36., 2009, Taipei. **Proceedings...** Taipei: IEEE, 2009. p. 2117–2120.

NEDJAH, N.; SILVA, R. M. da; MOURELLE, L. de M. Compact yet efficient hardware implementation of artificial neural networks with customized topology. **Expert Systems with Applications**, Elmsford, v. 39, n. 10, p. 9191–9206, ago. 2012.

NICHOLS, K. R. **A Reconfigurable Computing Architecture For Implementing Artificial Neural Networks on FPGA**. 2003. 221 f. Dissertação (Mestrado) — The Faculty of Graduate Studies, The University of Guelph, Guelph, 2003.

OLIVEIRA, A. M. G. et al. **Nutrição, calagem e adubação do mamoeiro irrigado**. Cruz das Almas: EMBRAPA, 2004. 1-10 p.

OTSU, N. A threshold selection method from gray-level histograms. **IEEE Transactions on Systems, Man and Cybernetics**, New York, v. 9, n. 1, p. 62–66, jan. 1979. ISSN 0018-9472.

PACUMBABA JUNIOR, R.; BEYL, C. Changes in hyperspectral reflectance signatures of lettuce leaves in response to macronutrient deficiencies. **Advances in Space Research**, Kidlington, v. 48, n. 1, p. 32–42, 2011. ISSN 0273-1177.

PAGOLA, M. et al. New method to assess barley nitrogen nutrition status based on image colour analysis: Comparison with spad-502. **Computers and Electronics in Agriculture**, Amsterdam, v. 65, n. 2, p. 213 – 218, 2009. ISSN 0168-1699.

PAN, J.; HUANG, M.; HE, Y. Crop and weed image recognition by morphological operations and ann model. In: INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE - IMTC, 24., 2007, Warsaw. **Proceedings...** Warsaw: IEEE, 2007. p. 1 –4. ISSN 1091-5281.

PEDRINI, H.; SCHWARTZ, W. R. **Análise de Imagens Digitais: Princípios, algoritmos e aplicações.** São Paulo: Cengage Learning, 2007.

PEDRONI, V. A. **Digital electronics and design with VHDL.** Hong Kong: Morgan Kaufmann, 2008.

PINHEIRO, H. S. K. et al. Uso de dados de sensoriamento remoto em mapeamento digital de solos por redes neurais artificiais. In: SIMPÓSIO BRASILEIRO DE SENSORIAMENTO REMOTO - SBSR, 16., 2013, Foz do Iguaçu. **Anais...** Foz do Iguaçu: Inpe, 2013. p. 9240–9247.

PRADO, R. M.; FRANCO, C. F.; PUGA, A. P. Deficiência de macronutrientes em plantas de soja cv. BRSMG 68 (vencedora) cultivada em solução nutritiva. **Comunicata Scientiae**, Bom Jesus, v. 1, n. 2, p. 114–119, 2010.

PRATAP, R.; SUBADRA, M. Efficient digital implementation of the sigmoidal function for artificial neural network. **AIP Conf. Proc.**, Melville, v. 1391, n. 1, p. 477–479, out. 2011.

RAKVIC, R. N. et al. FPGA or cell for an image processing application. In: INTERNATIONAL CONFERENCE ON ADVANCES IN COMPUTING, CONTROL, AND TELECOMMUNICATION TECHNOLOGIES - ACT, 1., 2009, Kerala. **Proceedings...** Kerala: IEEE Computer Society, 2009. p. 271–274. ISBN 978-0-7695-3915-7.

RICHETTI, A. **Viabilidade econômica da cultura da soja na safra 2011/2012, em mato grosso do sul.** Dourados: Embrapa, 2011. (Comunicado Técnico, 168).

RIGLER, S.; BISHOP, W.; KENNINGS, A. FPGA-based lossless data compression using Huffman and LZ77 algorithms. In: CANADIAN CONFERENCE ON ELECTRICAL AND COMPUTER ENGINEERING - CCECE, 20., 2007, Vancouver. **Proceedings...** Vancouver: IEEE Computer Society, 2007. p. 1235–1238. ISSN 0840-7789.

SAICHAND, V. et al. FPGA realization of activation function for artificial neural networks. In: INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS DESIGN AND APPLICATIONS - ISDA, 8., 2008, Kaohsiung. **Proceedings...** Kaohsiung: IEEE, 2008. v. 3, p. 159–164.

SANTOS, P. et al. Artificial neural network acceleration on FPGA using custom instruction. In: CANADIAN CONFERENCE ON ELECTRICAL AND COMPUTER ENGINEERING - CCECE, 24., 2011, Ontario. **Proceedings...** Ontario: IEEE Computer Society, 2011. p. 450–455. ISSN 0840-7789.

SARAIVA, A. M.; CUGNASCA, C. E. Redes de comunicação serial em máquinas agrícolas: Uma revisão. **Revista Brasileira de Agroinformática**, São Paulo, v. 8, n. 1, p. 17–35, jul. 2006. ISSN 1517-3267.

SAVICH, A.; MOUSSA, M.; AREIBI, S. The impact of arithmetic representation on implementing MLP-BP on FPGAs: A study. **IEEE Transactions on Neural Networks**, New York, v. 18, n. 1, p. 240–252, jan. 2007.

SCROFANO, R. et al. Accelerating molecular dynamics simulations with reconfigurable computers. **IEEE Transactions on Parallel and Distributed Systems**, Los Alamitos, v. 19, n. 6, p. 764–778, jun. 2008. ISSN 1045-9219.

SFREDO, G. J. **Soja no Brasil**: calagem, adubação e nutrição mineral. Londrina: Embrapa, 2008. (Documentos, 305).

SHAHAMIRI, S. R.; SALIM, S. S. B. Real-time frequency-based noise-robust automatic speech recognition using multi-nets artificial neural networks: A multi-views multi-learners approach. **Neurocomputing**, Netherlands, v. 129, n. 0, p. 199–207, 2014. ISSN 0925-2312.

SHEELA, K. G.; DEEPA, S. N. Review on methods to fix number of hidden neurons in neural networks. **Mathematical Problems in Engineering**, USA, v. 2013, n. 425740, p. 199–207, 2013. ISSN 0925-2312.

SILVA, A. C. de O. et al. Discriminação de áreas cafeeiras de imagens multiespectrais da região de três pontas – mg através de redes neurais artificiais e extratores de textura. In: SIMPÓSIO BRASILEIRO DE SENSORIAMENTO REMOTO - SBSR, 16., 2013, Foz do Iguaçu. **Anais...** Foz do Iguaçu: Inpe, 2013. p. 3795–3802.

SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. **Redes neurais artificiais**: para engenharia e ciências aplicadas. São Paulo: Artiliber, 2010.

STEPHENSON, J. et al. **Understanding metastability in FPGAs**. San Jose: Altera, 2009. White Paper. Acesso em: 05 Jul. 2012.

STRENSKI, D. **FPGA floating point performance**: a pencil and paper evaluation. [S.l.: s.n.], 2007. Disponível em: <<http://www.hpcwire.com/hpcwire/2007-01-12/fpga-floating-point-performance.html>>. Acesso em: 20 Ago. 2012.

TISAN, A. et al. Digital implementation of the sigmoid function for FPGA circuits. **ACTA TECHNICA NAPOCENSIS Electronics and Telecommunications**, Cluj-Napoca, v. 50, n. 2, p. 15–20, jun. 2009.

TOMMISKA, M. Efficient digital implementation of the sigmoid function for reprogrammable logic. **IEE Proceedings - Computers and Digital Techniques**, Stevenage, v. 150, n. 6, p. 403–411, nov. 2003. ISSN 1350-2387.

VOLLMANN, J. et al. Digital image analysis and chlorophyll metering for phenotyping the effects of nodulation in soybean. **Computers and Electronics in Agriculture**, Amsterdam, v. 75, n. 1, p. 190–195, 2011. ISSN 0168-1699.

WANG, J. et al. An adaptive thresholding algorithm of field leaf image. **Computers and Electronics in Agriculture**, Amsterdam, v. 96, n. 0, p. 23 – 39, 2013. ISSN 0168-1699.

WANG, Z.; BOVIK, A. A universal image quality index. **IEEE Signal Processing Letters**, Piscataway, v. 9, n. 3, p. 81–84, 2002. ISSN 1070-9908.

XILINX. **Using digital clock managers (DCMs) in spartan-3 FPGAs**. [S.l.: s.n.], 2006.

XILINX. **Spartan-3E FPGA family: Data sheet**. [S.l.: s.n.], 2009.

XILINX. **Logicore IP floating-point operator v5.0**. [S.l.: s.n.], 2011.

XILINX. **7 series FPGAs configurable logic block**. [S.l.: s.n.], 2012.

YAO, H. et al. Using vegetation index and modified derivative for early detection of soybean plant injury from glyphosate. **Computers and Electronics in Agriculture**, Elsevier Science Publishers B. V., Amsterdam, v. 89, p. 145–157, nov. 2012. ISSN 0168-1699.

ZHANG, J.; CHEN, L. Hardware implementation of neural networks based on DSP48 slice of Virtex-4 FPGA. **Energy Procedia**, Amsterdam, v. 13, p. 9548–9555, 2011.

## APÊNDICE A - RESULTADOS ADJACENTES DE ERROS

Os resultados individuais de cada cenário com três tipos de erros são apresentados neste apêndice. Os erros analisados são o absoluto, relativo e quadrático, respectivamente conforme Equações (27), (28) e (29) da seção 5.3.1.2.

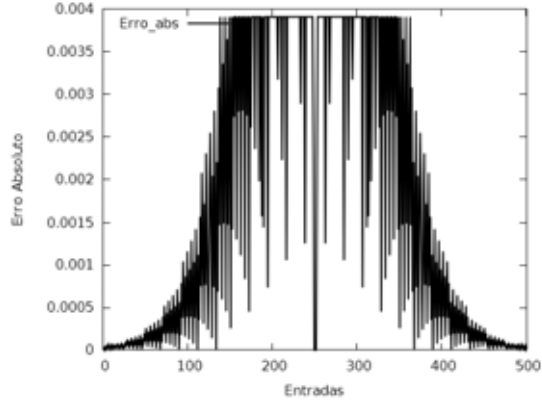
As Figuras 96, 97 e 98 mostram os erros absolutos, relativos e quadráticos do primeiro grupo de cenários analisados no estudo de caso 2 da seção 5. Em cada figura contém seis itens que representam cada cenário do primeiro grupo analisado para a aproximação da função de ativação Tangente Hiperbólica. As Figuras 99, 100 e 101 correspondem à análise dos mesmos erros, porém para a aproximação da senoide pela execução da RNA completa.

As Figuras 96, 99, 97, 100, 98 e 101 correspondem aos erros individuais das implementações para possibilitar a investigação em todos os momentos no intervalo e amostras definidas. Assim, os erros absolutos em todos os cenários do primeiro grupo estão nas Figuras 96 e 99. Os erros relativos ilustrados nas Figuras 97 e 100, e os erros médios quadráticos nas Figuras 98 e 101. Os erros absoluto, relativo e médio quadrático nos gráficos demonstram a conduta de cada implementação e determinados pontos de suavização dos métodos implementados, como o exemplo visível da Figura 96-(a).

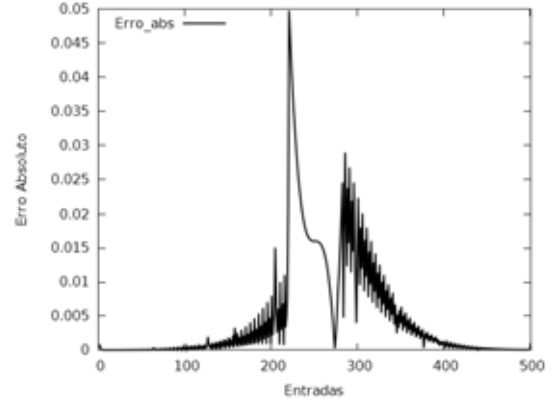
As Figuras 102, 103, 104, 105, 106 e 107 apresentam os erros analisados para o segundo grupo de cenários do estudo de caso 2 da seção 5. Os erros absolutos, relativos e quadráticos são apresentados em cada par de figuras, correspondendo aos 9 cenários analisados nesse grupo. As Figuras 108, 109, 110, 111, 112 e 113 apresentam os erros para aproximação da senoide pela RNA completa, como feito anteriormente na AF.

Os erros obtidos na aproximação da AF estão apresentadas nas Figuras 102, 103, 104, 105, 106 e 107. Os erros da aproximação da Equação (24) pela implementação da RNA são apresentados pelas Figuras 108, 109, 110, 111, 112 e 113.

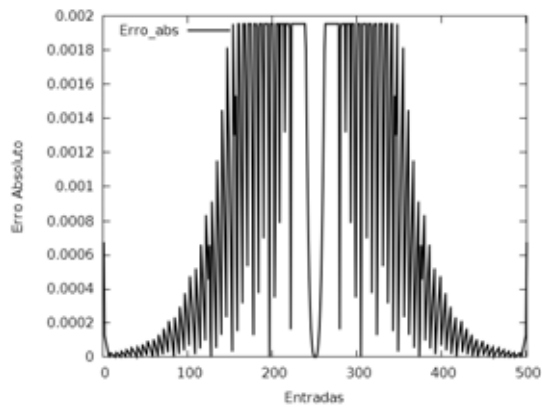
Figura 96 – Avaliação dos erros absolutos nas implementações da AF para o primeiro grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6.



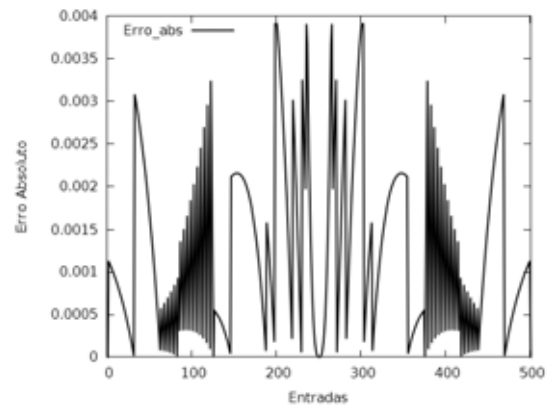
(a)



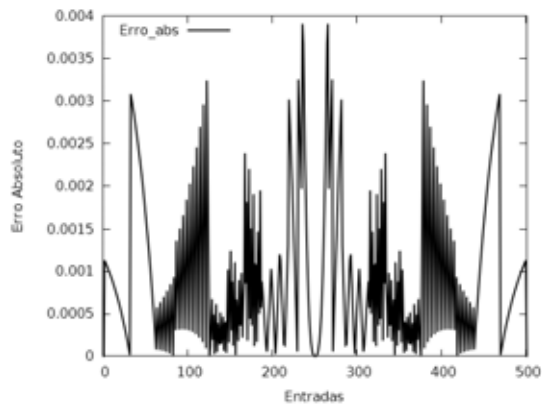
(b)



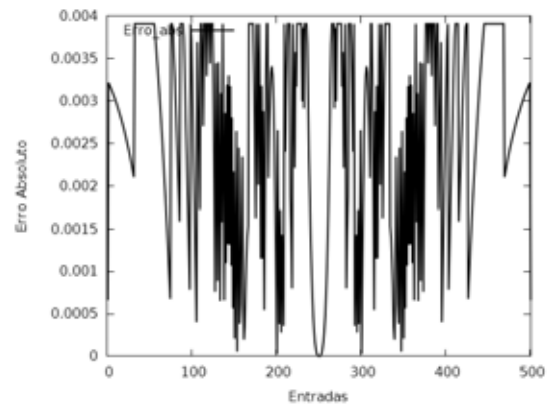
(c)



(d)



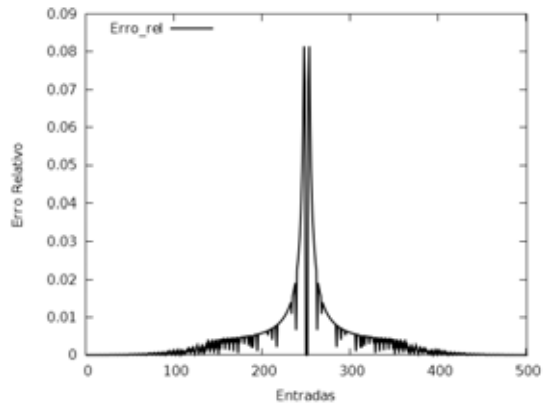
(e)



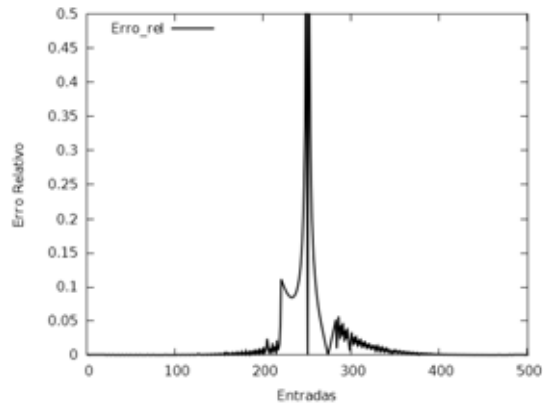
(f)

Fonte: Elaboração do próprio autor

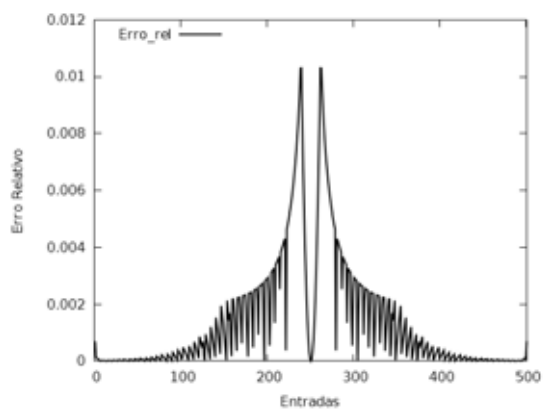
Figura 97 – Avaliação dos erros relativos nas implementações da AF para o primeiro grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6.



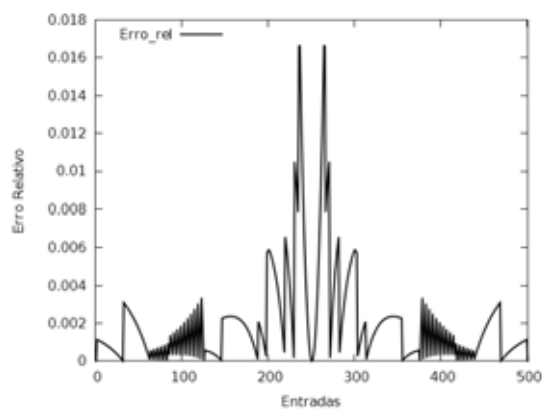
(a)



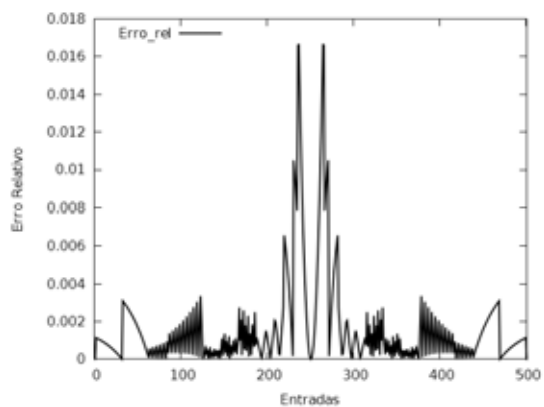
(b)



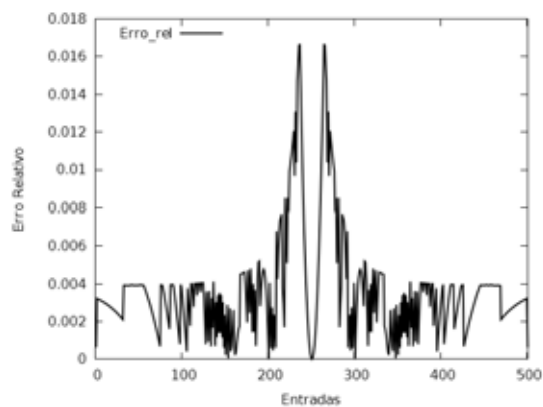
(c)



(d)



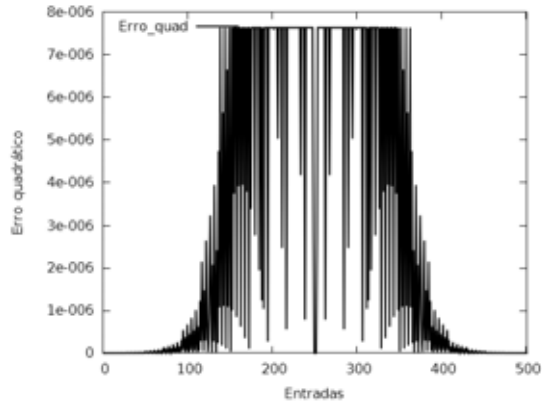
(e)



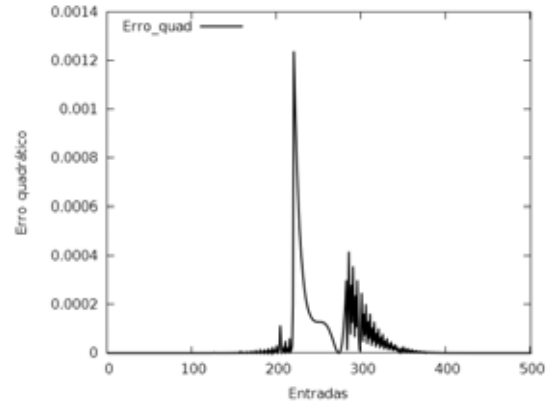
(f)

Fonte: Elaboração do próprio autor

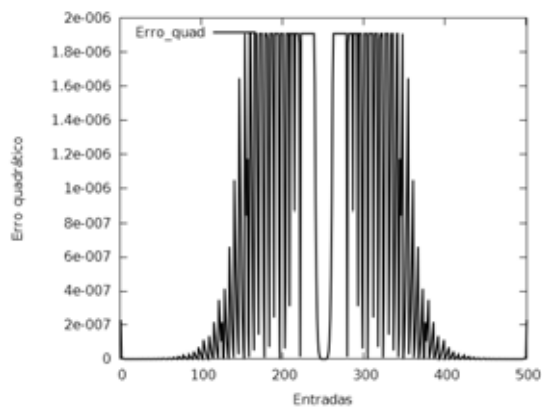
Figura 98 – Avaliação dos erros médios quadráticos nas implementações da AF para o primeiro grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6.



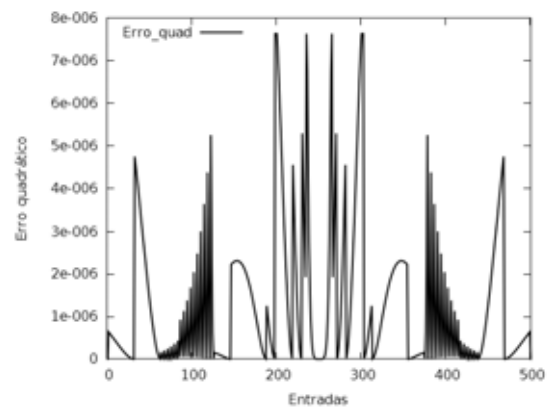
(a)



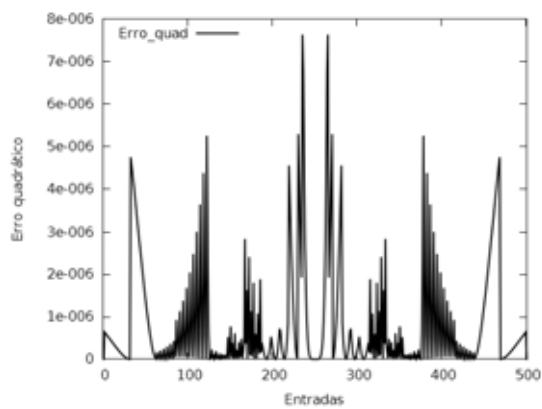
(b)



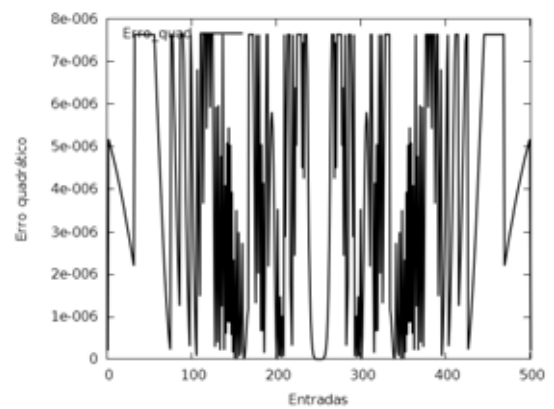
(c)



(d)



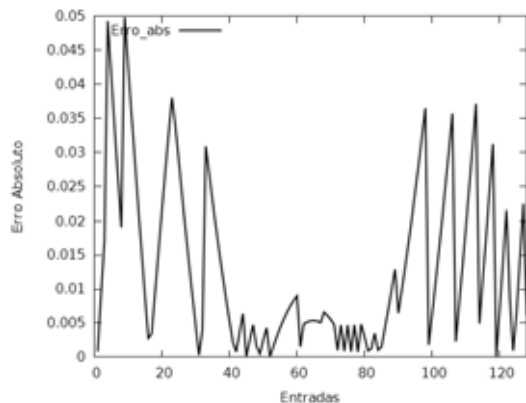
(e)



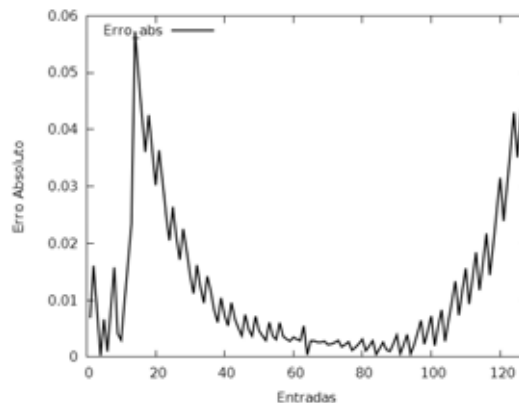
(f)

Fonte: Elaboração do próprio autor

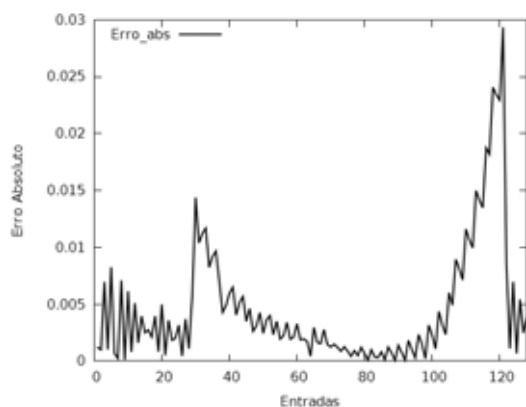
Figura 99 – Avaliação dos erros absolutos nas implementações da RNA para o primeiro grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6.



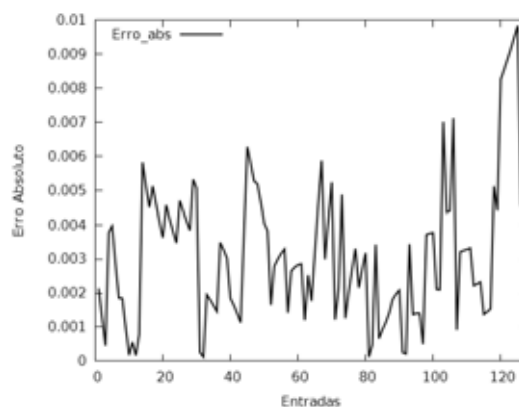
(a)



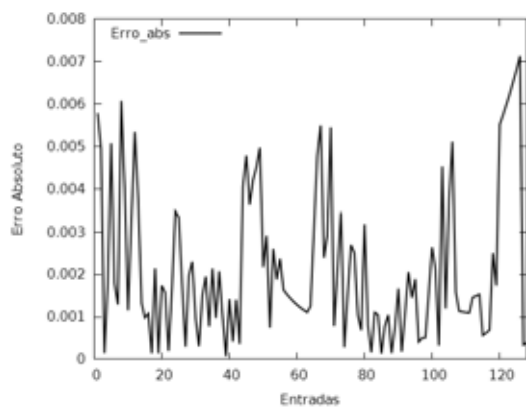
(b)



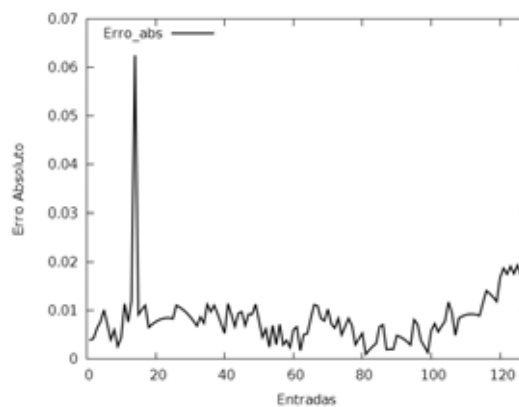
(c)



(d)



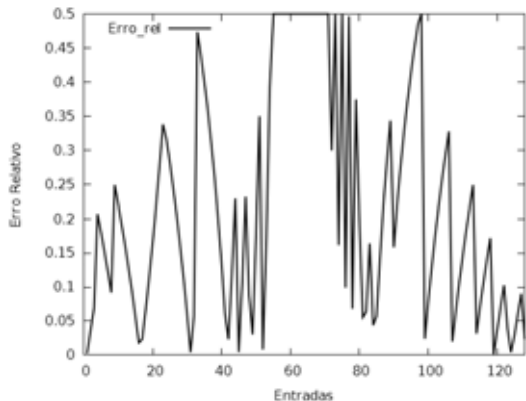
(e)



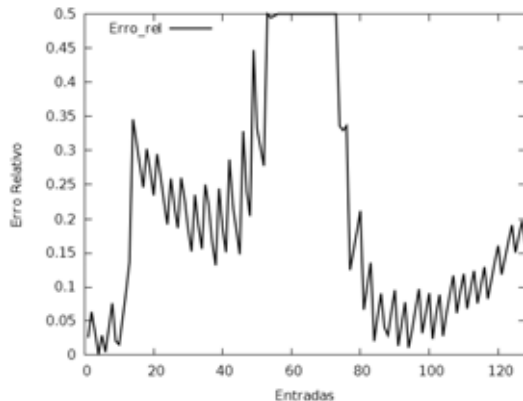
(f)

Fonte: Elaboração do próprio autor

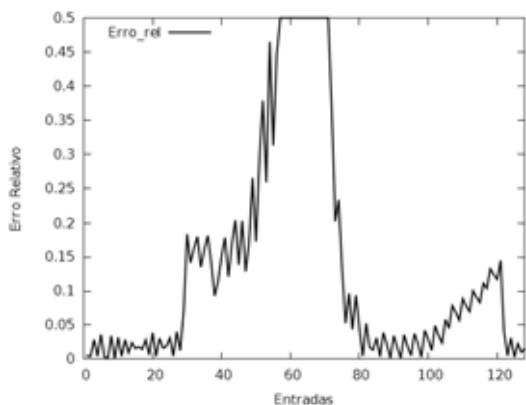
Figura 100 – Avaliação dos erros relativos nas implementações da RNA para o primeiro grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6.



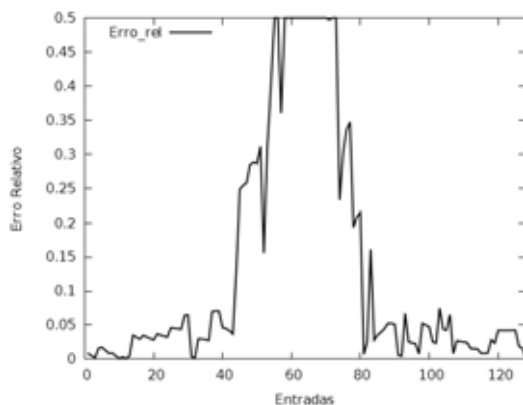
(a)



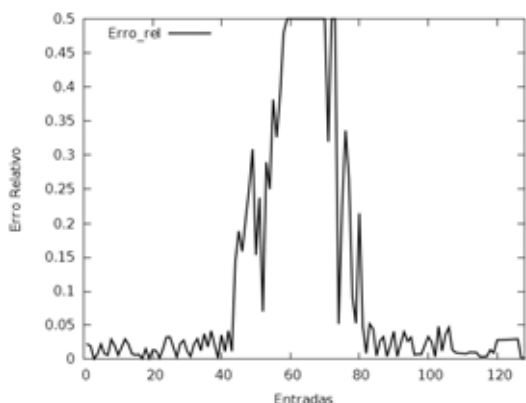
(b)



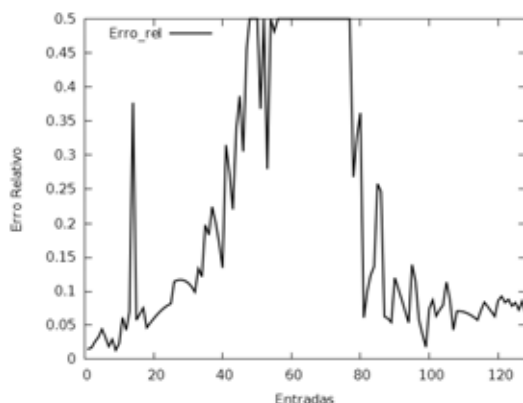
(c)



(d)



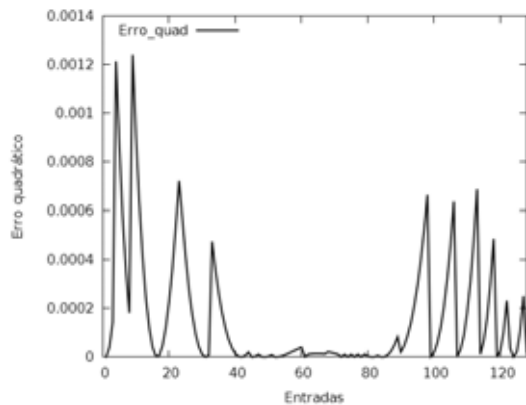
(e)



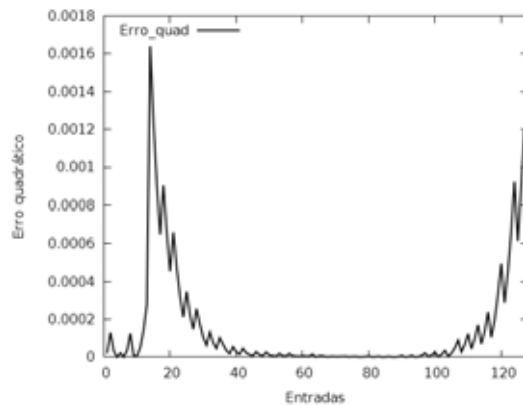
(f)

Fonte: Elaboração do próprio autor

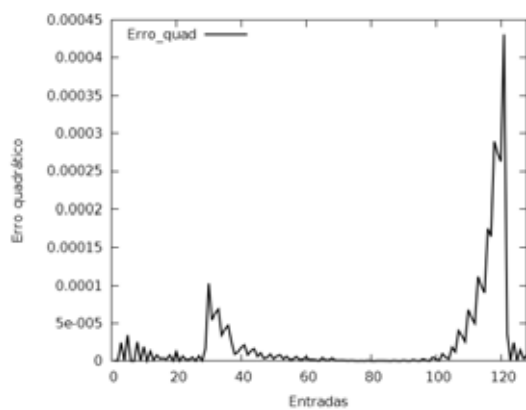
Figura 101 – Avaliação dos erros médios quadráticos nas implementações da RNA para o primeiro grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6.



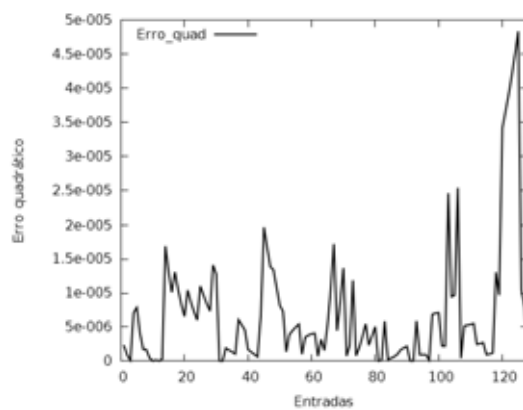
(a)



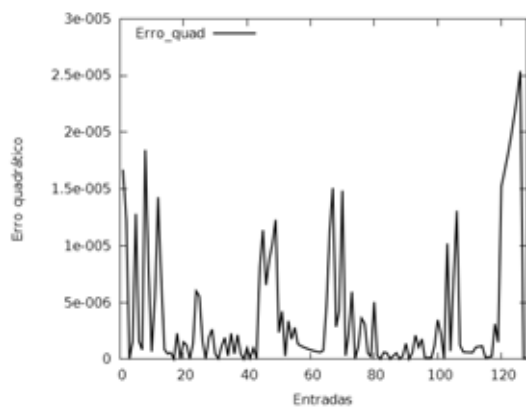
(b)



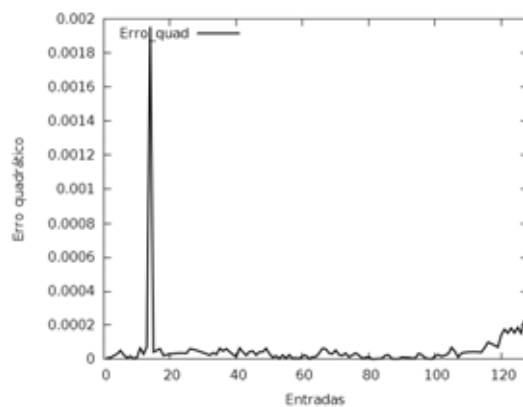
(c)



(d)



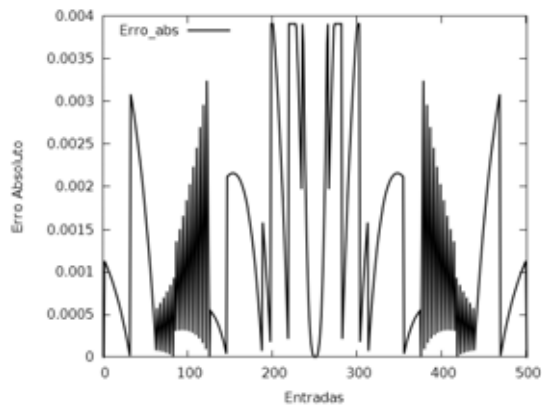
(e)



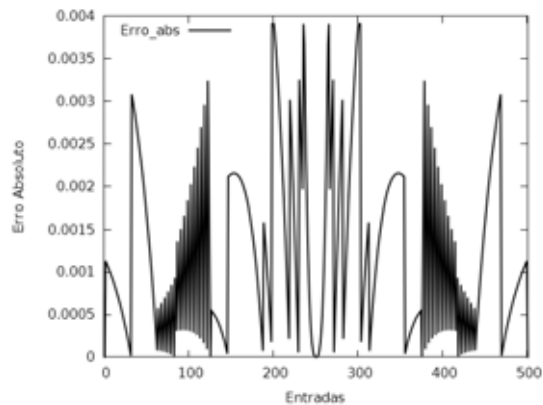
(f)

Fonte: Elaboração do próprio autor

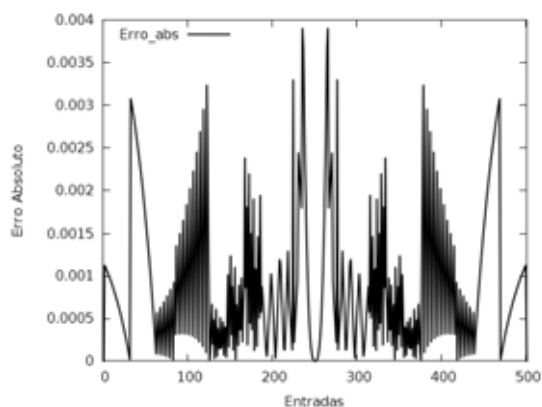
Figura 102 – Avaliação dos erros absolutos nas implementações da AF para o segundo grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6.



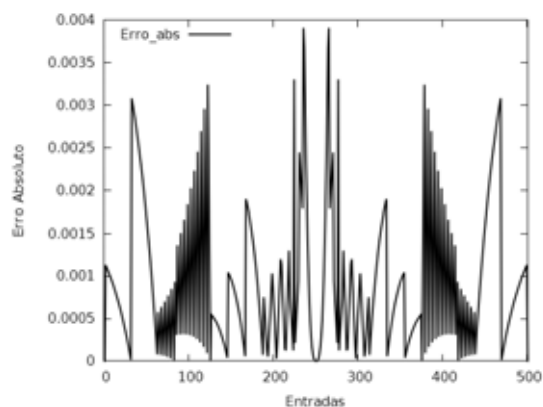
(a)



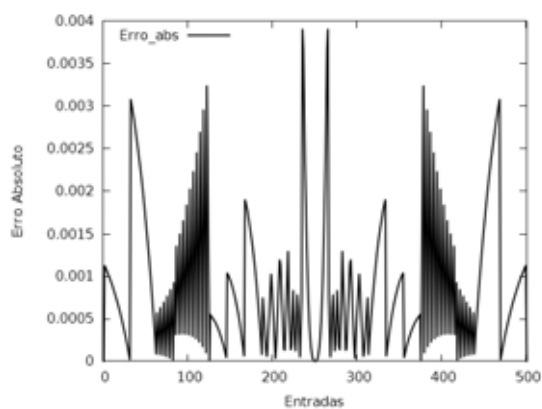
(b)



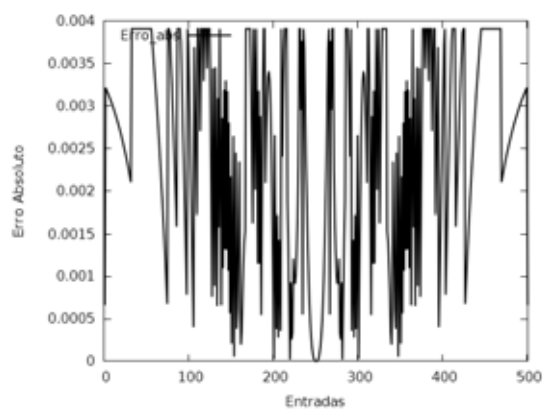
(c)



(d)



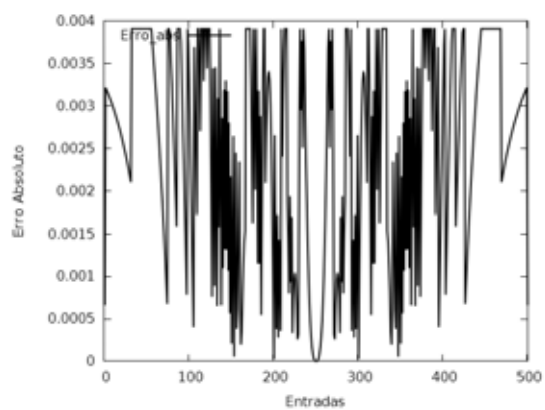
(e)



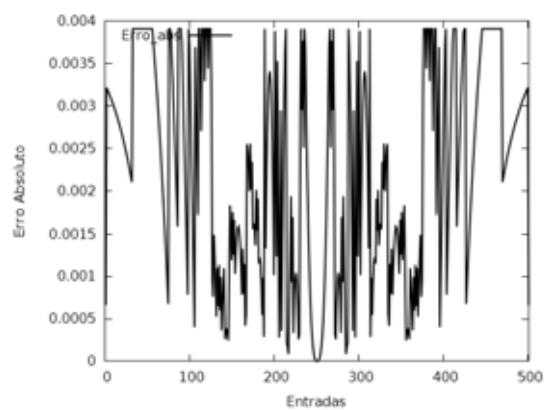
(f)

Fonte: Elaboração do próprio autor

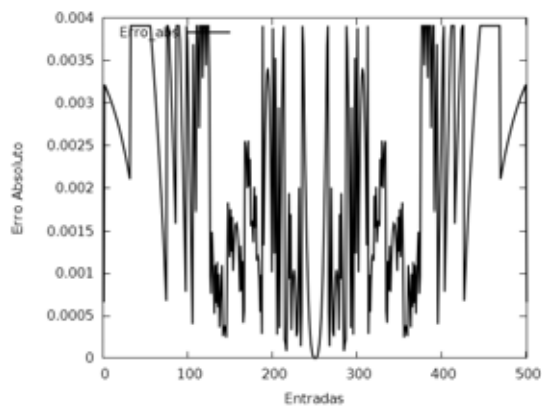
Figura 103 – Avaliação dos erros absolutos nas implementações da AF para o segundo grupo de cenários: (a) Cenário 7; (b) Cenário 8; (c) Cenário 9



(a)



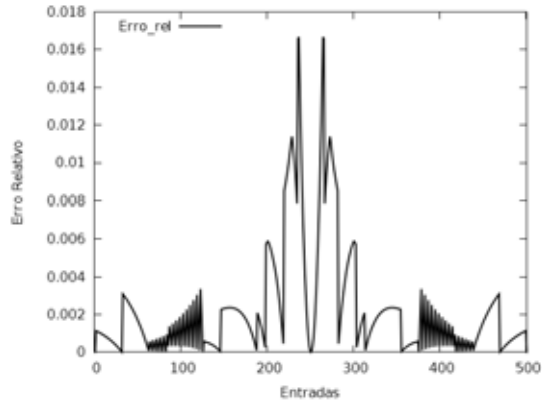
(b)



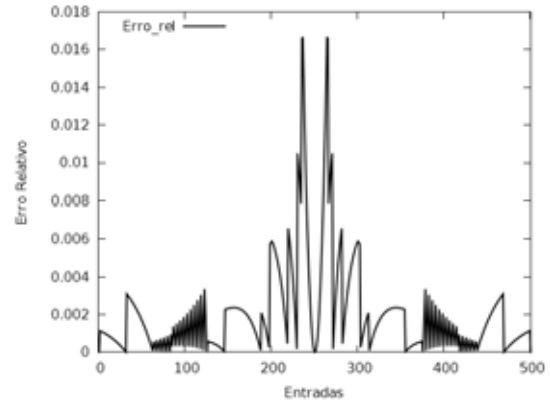
(c)

Fonte: Elaboração do próprio autor

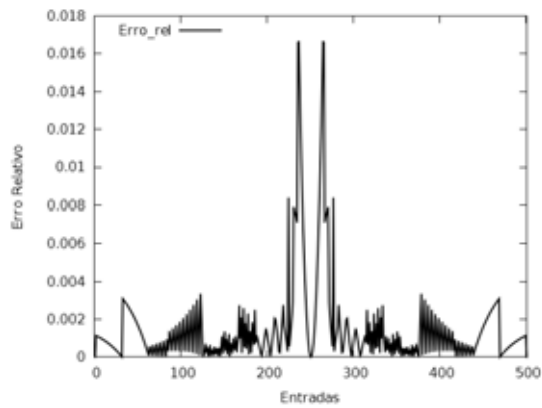
Figura 104 – Avaliação dos erros relativos nas implementações da AF para o segundo grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6.



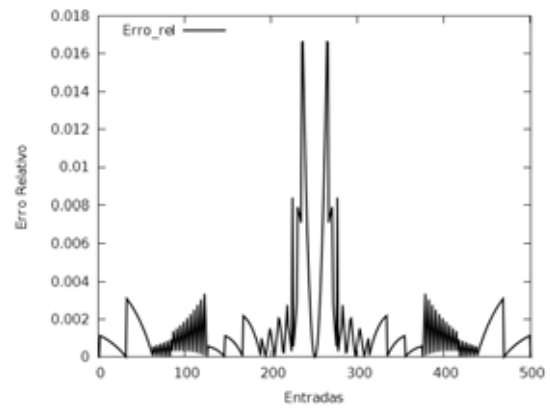
(a)



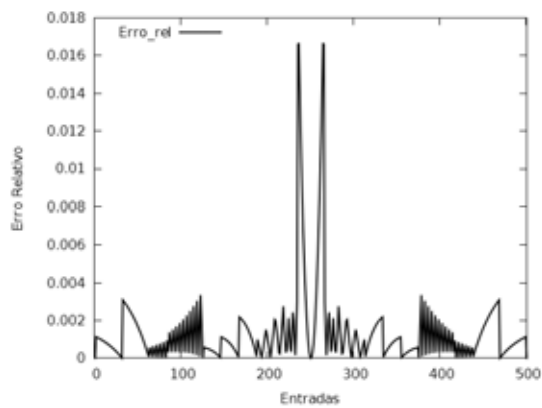
(b)



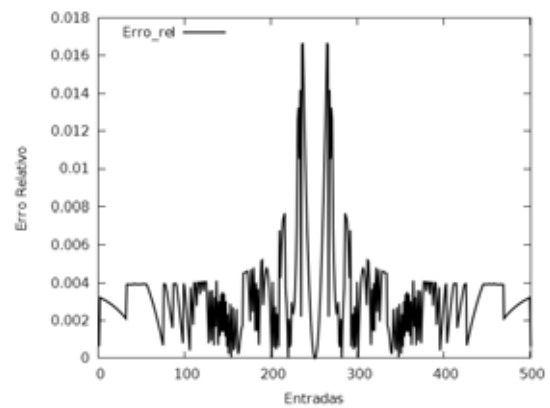
(c)



(d)



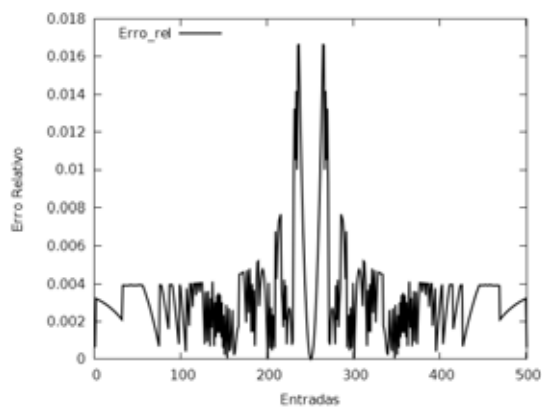
(e)



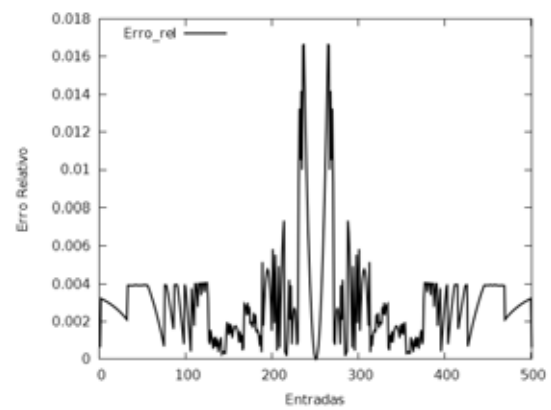
(f)

Fonte: Elaboração do próprio autor

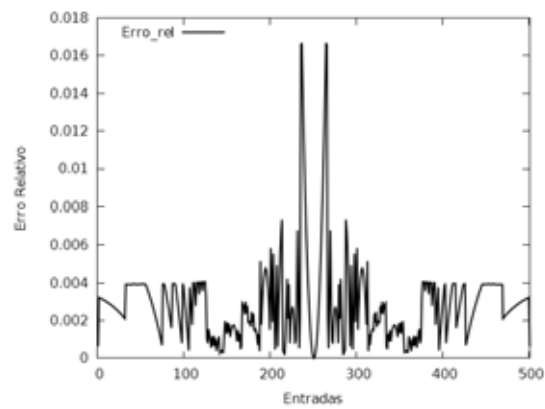
Figura 105 – Avaliação dos erros relativos nas implementações da AF para o segundo grupo de cenários: (a) Cenário 7; (b) Cenário 8; (c) Cenário 9.



(a)



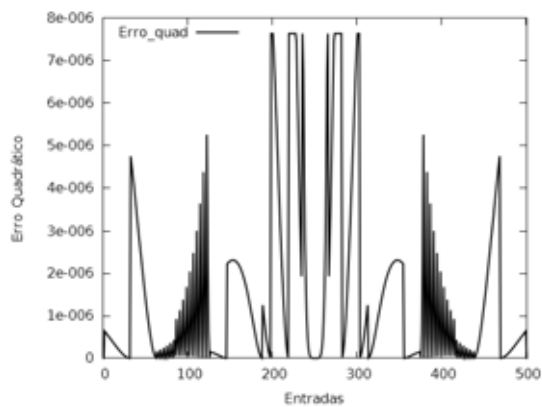
(b)



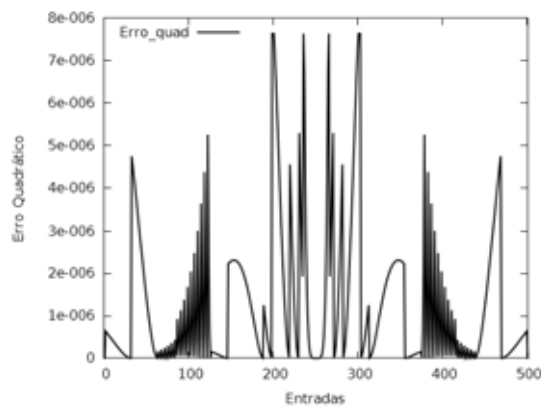
(c)

Fonte: Elaboração do próprio autor

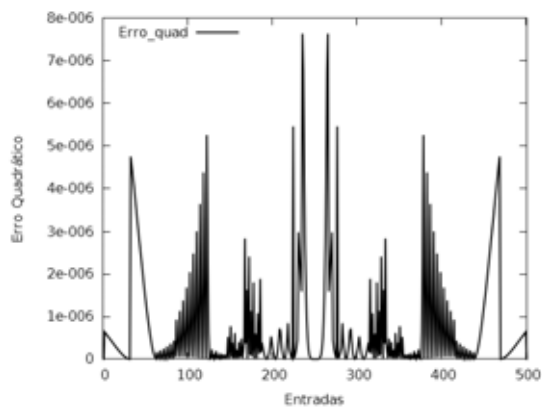
Figura 106 – Avaliação dos erros médios quadráticos nas implementações da AF para o segundo grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6.



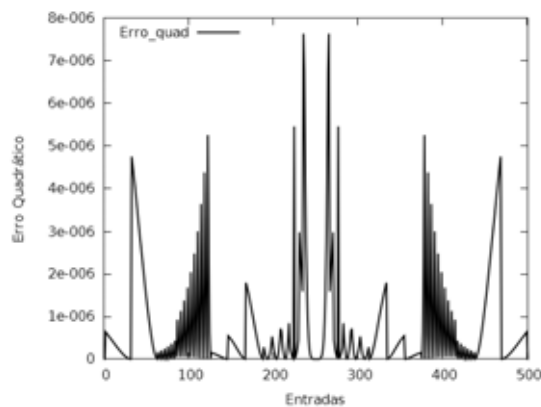
(a)



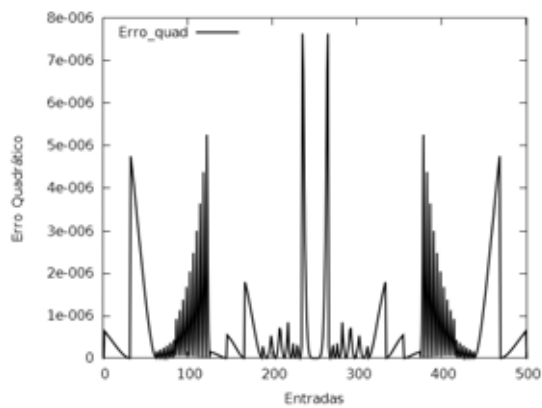
(b)



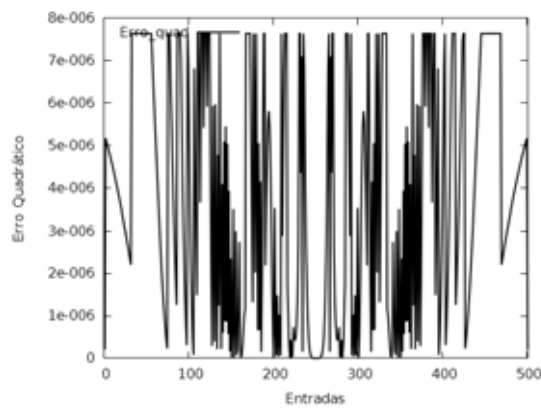
(c)



(d)



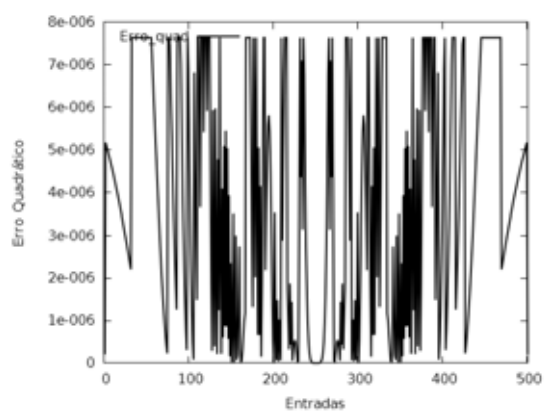
(e)



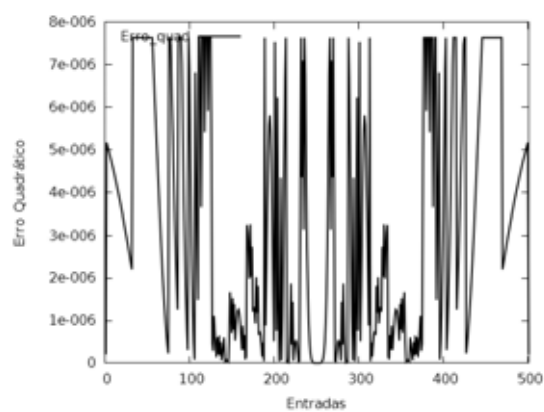
(f)

Fonte: Elaboração do próprio autor

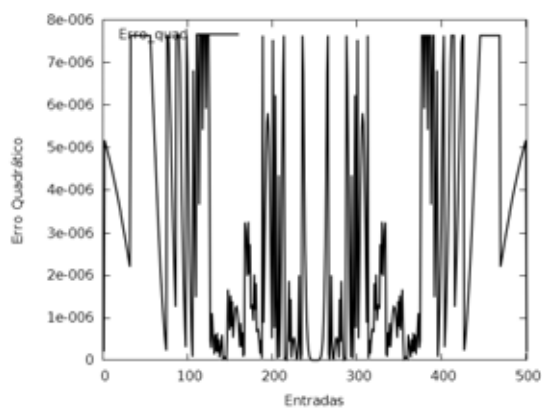
Figura 107 – Avaliação dos erros médios quadráticos nas implementações da AF para o segundo grupo de cenários: (a) Cenário 7; (b) Cenário 8; (c) Cenário 9.



(a)



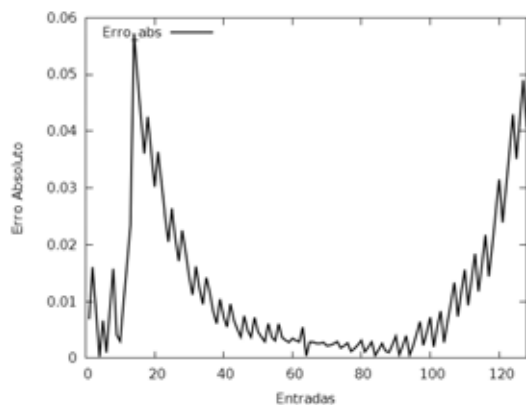
(b)



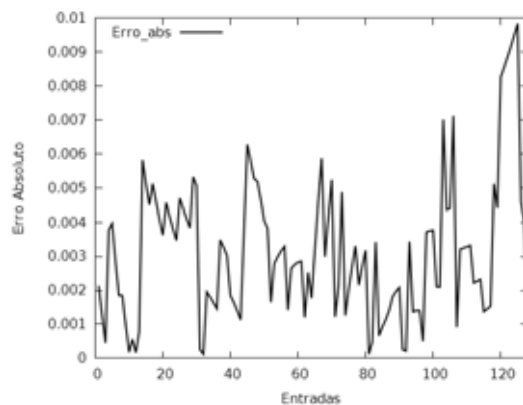
(c)

Fonte: Elaboração do próprio autor

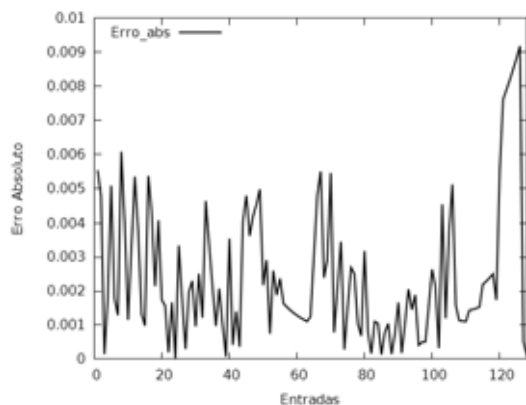
Figura 108 – Avaliação dos erros absolutos nas implementações da RNA para o segundo grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6.



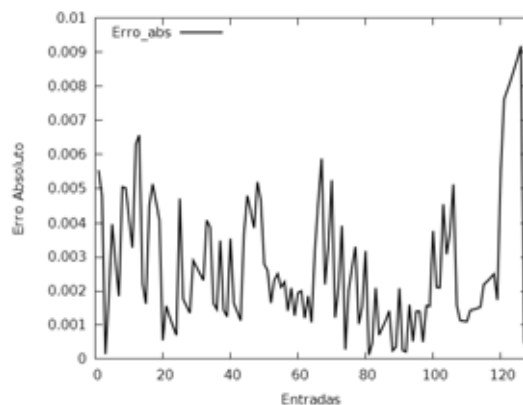
(a)



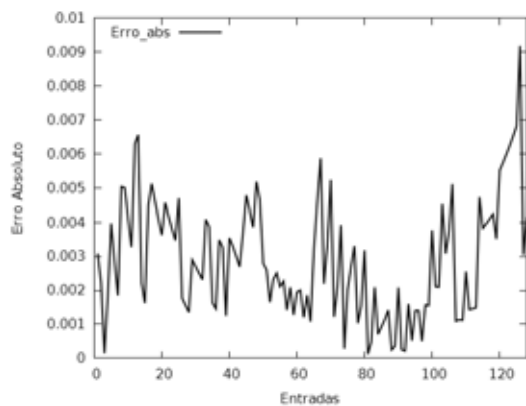
(b)



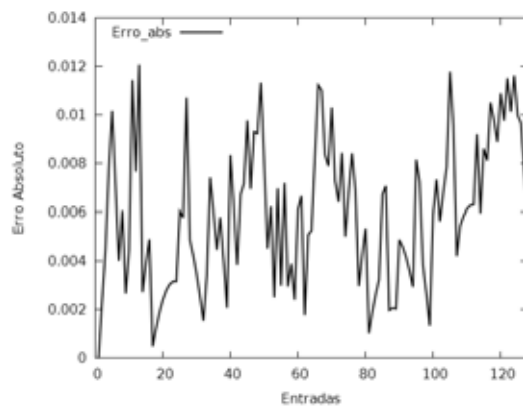
(c)



(d)



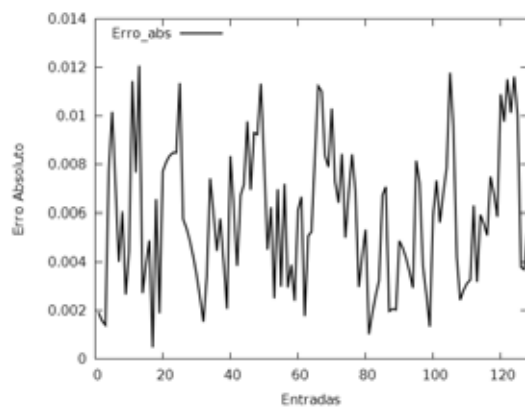
(e)



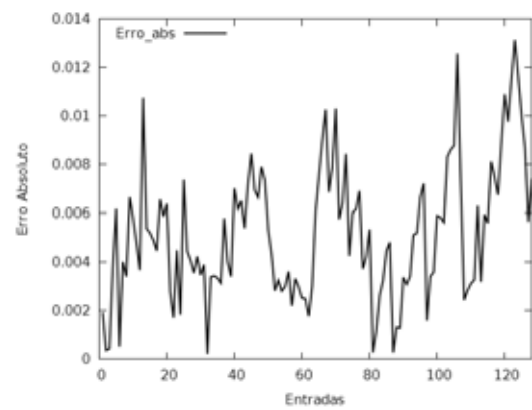
(f)

Fonte: Elaboração do próprio autor

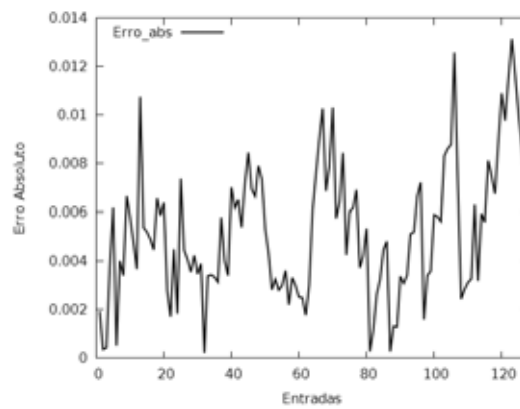
Figura 109 – Avaliação dos erros absolutos nas implementações da RNA para o segundo grupo de cenários: (a) Cenário 7; (b) Cenário 8; (c) Cenário 9.



(a)



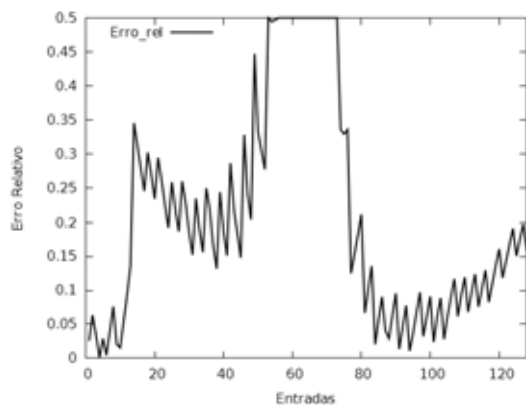
(b)



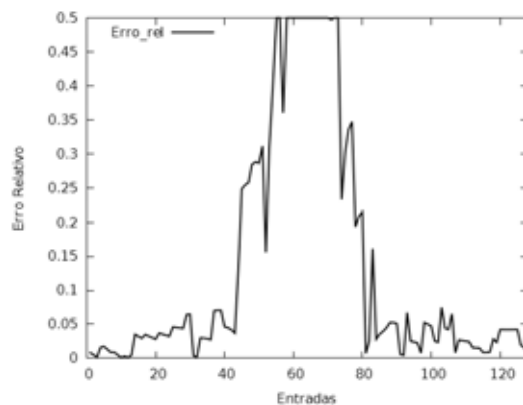
(c)

Fonte: Elaboração do próprio autor

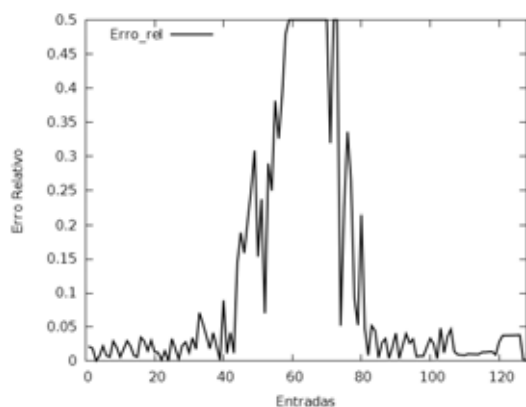
Figura 110 – Avaliação dos erros relativos nas implementações da RNA para o segundo grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6.



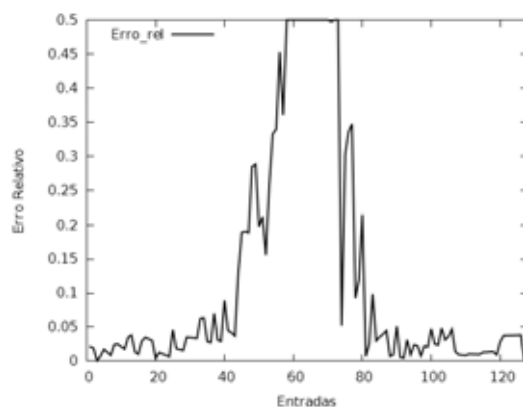
(a)



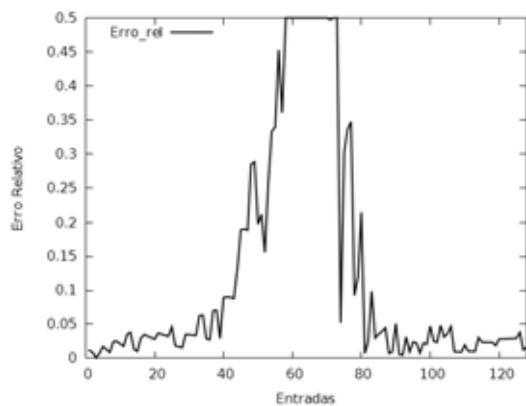
(b)



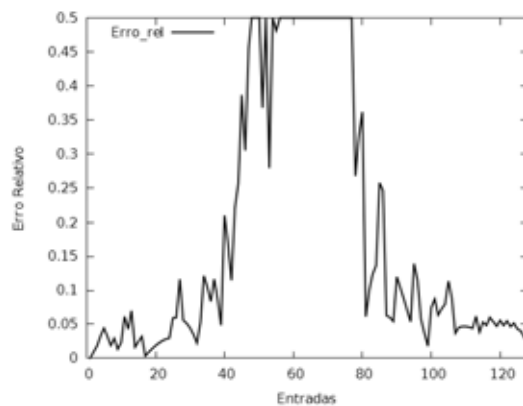
(c)



(d)



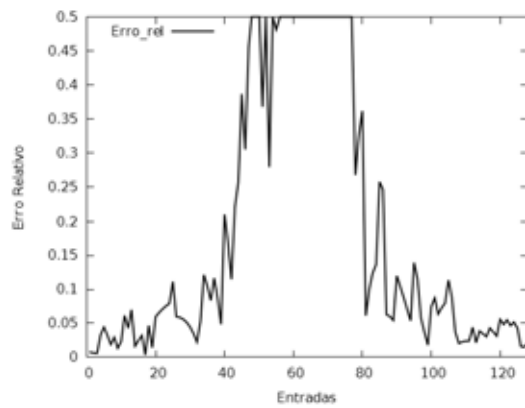
(e)



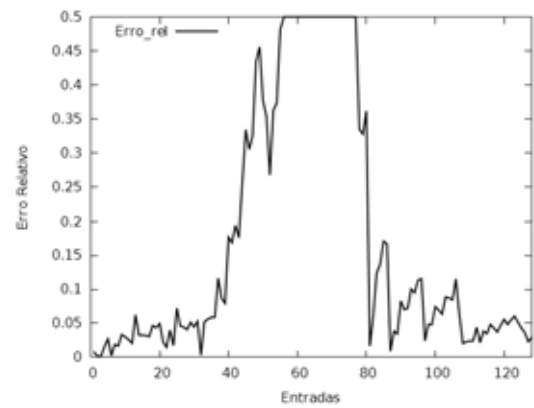
(f)

Fonte: Elaboração do próprio autor

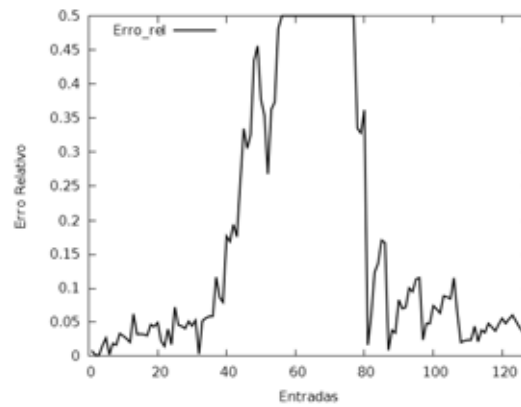
Figura 111 – Avaliação dos erros relativos nas implementações da RNA para o segundo grupo de cenários: (a) Cenário 7; (b) Cenário 8; (c) Cenário 9.



(a)



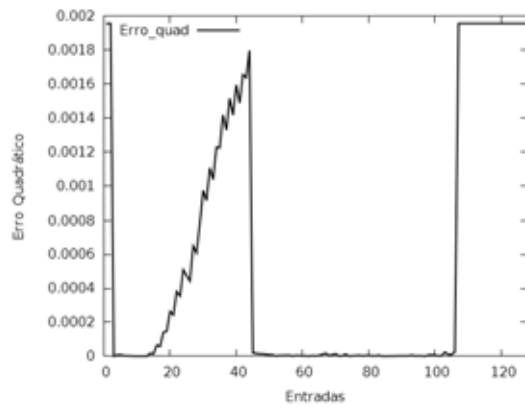
(b)



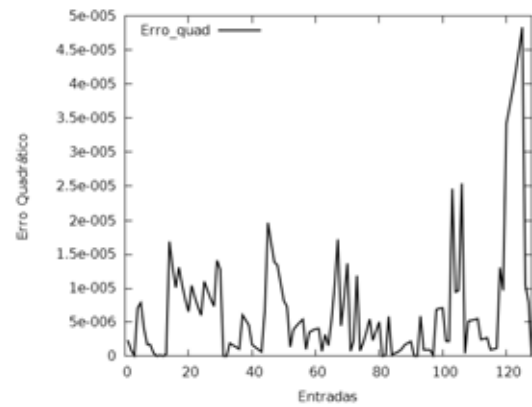
(c)

Fonte: Elaboração do próprio autor

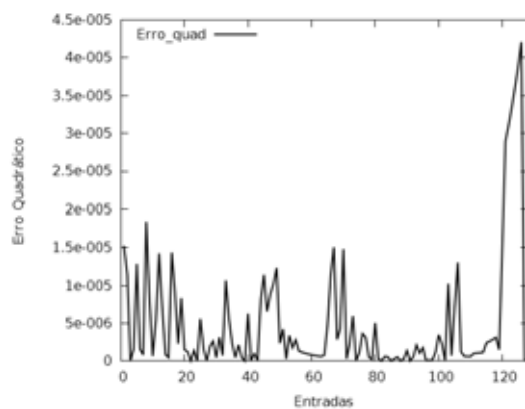
Figura 112 – Avaliação dos erros médios quadráticos nas implementações da RNA para o segundo grupo de cenários: (a) Cenário 1; (b) Cenário 2; (c) Cenário 3; (d) Cenário 4; (e) Cenário 5; (f) Cenário 6.



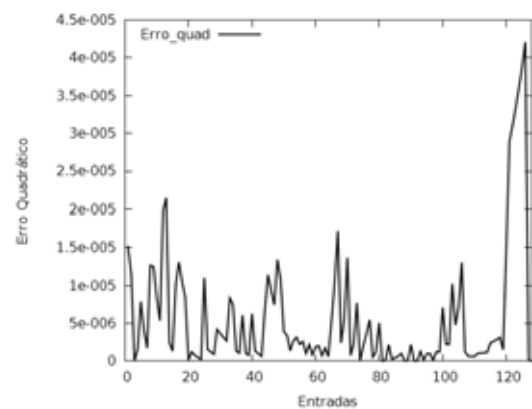
(a)



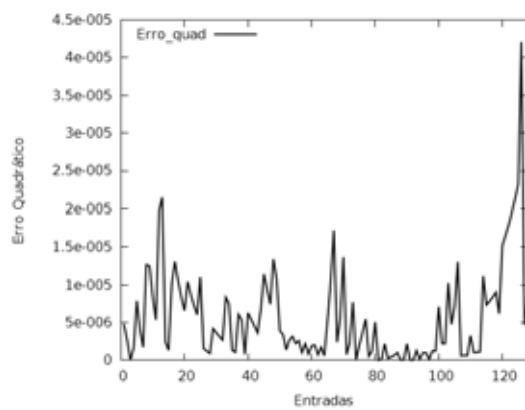
(b)



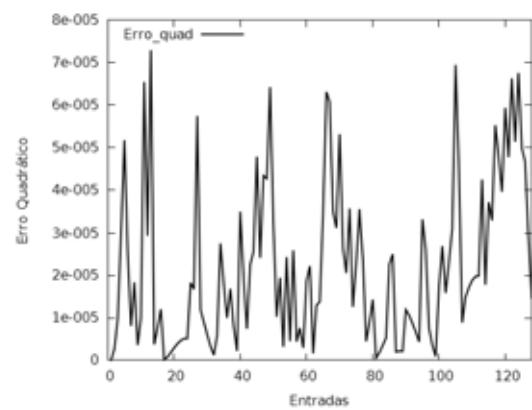
(c)



(d)



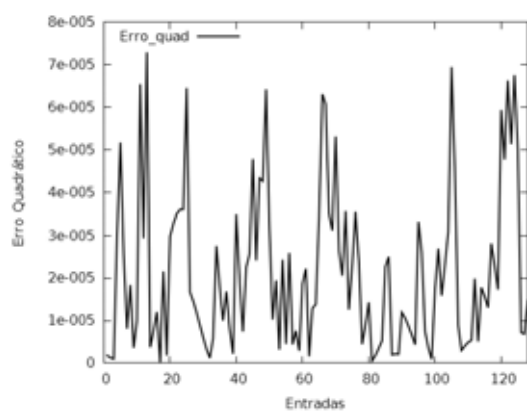
(e)



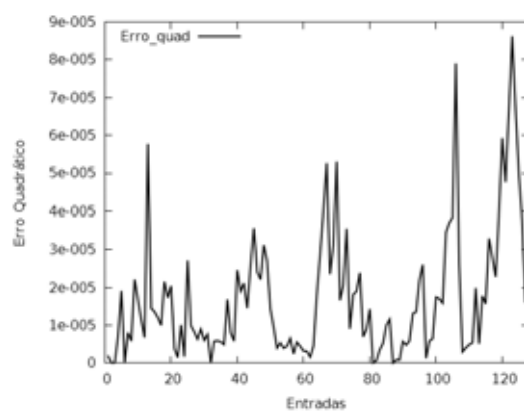
(f)

Fonte: Elaboração do próprio autor

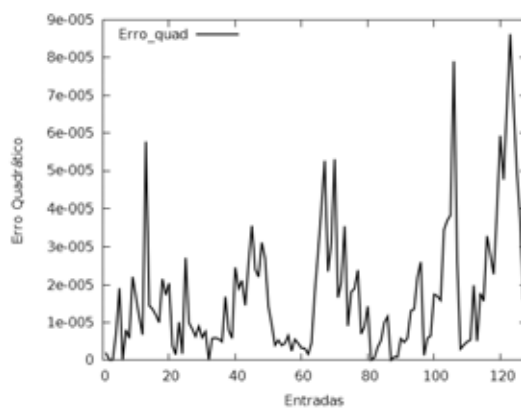
Figura 113 – Avaliação dos erros médios quadráticos nas implementações da RNA para o segundo grupo de cenários: (a) Cenário 7; (b) Cenário 8; (c) Cenário 9.



(a)



(b)



(c)

Fonte: Elaboração do próprio autor

# ANEXO A - RELATÓRIO DE ANÁLISE FOLIAR DA SOJA

Figura 114 – Relatório de análise foliar da soja.

**unesp** LABORATÓRIO DE NUTRIÇÃO DE PLANTAS  
 Faculdade de Engenharia de Ilha Solteira  
 e-mail: moliva@adm.fes.unesp.br Tel: (18) 3743-1142 Ramal: 1748


**TEORES DE NUTRIENTES**

Nome: Matson	Orientador:					
Cultura: Soja	Análise: Macro nutrientes					
Data: 12/12/2013	Nº amostras: 3					

**Resultados Analíticos**

Amostra	N g/kg Nitrogênio	P g/kg Fósforo	K g/kg Potássio	Ca g/kg Cálcio	Mg g/kg Magnésio	S g/kg Enxofre
B1	54,2	3,9	20,0	9,5	5,5	2,0
B2	56,4	3,7	18,0	6,7	4,3	2,3
B3	58,0	3,0	10,0	6,9	6,0	2,5

PROGRAMA  
 INTERNACIONAL  
**2013**  
 ANÁLISE FOLIAR  
 LA FOLIA ANALYSIS

Enviado em: 12/12/2013  Prof. Dr. Salaber Buzetti Responsável pelo laboratório	Rua Monte Alegre, 225 Ilha Solteira - SP CEP: 13385-000 - Caixa Postal 31 Fone: (18) 3743 - 1142 Ramal: 1748 Fax: (18) 3743 - 1176
--	---

Fonte: Elaboração do próprio autor