UNIVERSIDADE ESTADUAL PAULISTA - UNESP

Faculdade de Engenharia de Ilha Solteira - FEIS- campus de Ilha Solteira

ARIEL GIDEON RAMOS

Previsão de carga residencial incluindo carregamento de veículos elétricos usando Redes Neurais Artificiais

Ilha Solteira

2025

Ariel Gideon Ramos

Previsão de carga residencial incluindo carregamento de veículos elétricos usando Redes Neurais Artificiais

Trabalho de Graduação, apresentada à Universidade Estadual Paulista (UNESP), Faculdade de Engenharia de Ilha Solteira - FEIS, Ilha Solteira, para obtenção do título de Bacharelem Engenharia Elétrica.

Orientador: Profº Dr. Anna Diva Plasencia Lotufo

Ilha Solteira
2025

FICHA CATALOGRÁFICA
Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

Ramos, Ariel Gideon.
R175p Previsão de carga residencial incluindo carregamento de veículos elétricos usando redes neurais artificiais / Ariel Gideon Ramos. -- Ilha Solteira: [s.n.], 2025
68 f. : il.

Trabalho de conclusão de curso (Graduação em Engenharia Elétrica) - Universidade Estadual Paulista (UNESP), Faculdade de Engenharia, Ilha Solteira, 2025

Orientador: Anna Diva Plasencia Lotufo

Inclui bibliografia

1. Previsão de carga elétrica. 2. Redes neurais artificiais. 3. Veículos elétricos. 4. Demanda residencial. 5. Aprendizado de máquina.

Sandra Montibeller -CRB-8/060

UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de Ilha Solteira

ATA DE DEFESA DE TRABALHO DE GRADUAÇÃO

Aos vinte e sete dias do mês de junho do ano de dois mil e vinte e cinco, o discente ***Ariel Gideon Ramos,*** matriculado sob o nº172054265, tendo como banca examinadora a sua orientadora, *a Profª. Drª. Anna Diva Plasencia Lotufo, a Doutoranda Mayla Awana dos Santos e o Doutorando Giovanni Panegossi Formaggio,* apresentou o Trabalho de Graduação intitulado **"Previsão de carga residencial incluindo carregamento de veículos elétricos usando Redes Neurais Artificiais",** obtendo a nota 7,0 (SETE VÍRGULA ZERO) e conceito APROVADO.

Profª. ~~Drª.~~ Anna Diva Plasencia Lotufo
- Orientadora-

Ariel Gideon Ramos
- Discente -

Doutoranda Mayla Awana dos Santos
- Membro da Banca -

Doutorando Giovanni Panegossi Formaggio
- Membro da Banca -

Faculdade de Engenharia – Departamento de Engenharia Elétrica
Avenida Professor José Carlos Rossi, 1370 - CEP 15.385-000, Ilha Solteira - SP - Brasil
Tel (18) 37431150 - Fax (18) 3742-2735 - www.feis.unesp.br

AGRADECIMENTOS

Este trabalho representa não apenas o encerramento de uma importante etapa acadêmica, mas também a realização de um sonho incomum de ser concretizado e muitas vezes inalcançável pelas pessoas que cresceram comigo na minha comunidade.

Agradeço, em primeiro lugar, à minha família, especialmente ao meu irmão e à minha mãe, pelo suporte incondicional, incentivo e compreensão ao longo de toda a jornada acadêmica. Sem o apoio de vocês, tanto emocional quanto financeiro, este caminho teria sido muito mais difícil.

À minha namorada Iris, agradeço pelo suporte emocional, por todo afeto e pela presença constante nos momentos de desafio e superação. "Diante da vastidão do tempo e da imensidão do universo, é um imenso prazer para mim dividir um planeta e uma época com você."(Carl Sagan)

Sou profundamente grato à existência da Universidade Pública, em especial à UNESP, por proporcionar uma formação de qualidade de forma gratuita e por ser um espaço de transformação social. Ao Programa de Permanência Estudantil da faculdade, agradeço pelo apoio por meio da moradia estudantil e do auxílio socioeconômico, que foram essenciais para a continuidade dos meus estudos.

À minha orientadora, Profa. Dra. Anna Diva Plasencia, agradeço por todas as orientações, pela paciência, pelo incentivo ao rigor científico e pela compreensão em todos os momentos de dificuldades, pois foram pontos determinantes para o desenvolvimento deste trabalho.

Ao mestre Giovanni Panegossi, agradeço pela inspiração e pelas contribuições diretas e indiretas que enriqueceram a escrita deste trabalho.

Aos meus colegas de turma, da moradia estudantil e da república que morei, agradeço pela amizade, companheirismo e troca de experiências ao longo desta jornada.

Por fim, agradeço ao Laboratório SINTEL (Sistemas Inteligentes) pelo ambiente de aprendizado, pelo acesso à infraestrutura e pelo estímulo à pesquisa e à inovação.

"A ciência não é apenas compatível com a espiritualidade; ela é uma profunda fonte de espiritualidade."

(Sagan, 1996, p. 48)

RESUMO

A crescente integração de veículos elétricos (VEs) à rede elétrica impõe desafios significativos ao planejamento e operação do setor, particularmente no que tange à demanda residencial de energia. Este trabalho investiga a aplicação e o desempenho de Redes Neurais Artificiais (RNAs) do tipo Perceptron Multicamadas (MLP) com treinamento por Retropopagação para a previsão de curto prazo da demanda elétrica residencial combinada ao carregamento não coordenado de VEs. Utilizou-se um conjunto de dados públicos representativo do consumo residencial e de carregamento de VEs, que foi processado com ferramentas da linguagem Python e suas bibliotecas para ciência de dados e aprendizado de máquina. O estudo envolveu a exploração de diferentes configurações de neurônios nas camadas ocultas, funções de ativação como ReLU e Sigmoide, e técnicas de regularização como *Batch Normalization* e *Dropout*, com a avaliação do desempenho realizada por meio de métricas estatísticas como MAPE, RMSE e R^2. Os resultados demonstraram a capacidade dos modelos MLP em aprender os padrões complexos da demanda combinada do consumo residencial com a introdução de veículos elétricos, identificando-se uma arquitetura otimizada, com 9, 6 e 2 neurônios nas camadas ocultas, utilizando ativação ReLU e *Batch Normalization*, que alcançou um MAPE de aproximadamente 5,76% para o cenário Re+L1. A análise experimental revelou a importância da seleção criteriosa da arquitetura e das técnicas de regularização para a obtenção de um modelo com boa capacidade de generalização. O estudo contribui para a compreensão da aplicabilidade de RNAs na previsão de cenários de demanda residencial com VEs, fornecendo uma análise do processo de refinamento e dos fatores que influenciam o desempenho preditivo, e reforça a relevância de tais abordagens para investigações futuras no contexto da eletrificação do transporte.

Palavras-Chave: previsão de carga elétrica; redes neurais artificiais; veículos elétricos; demanda residencial; aprendizado de máquina.

ABSTRACT

The increasing integration of electric vehicles (EVs) into the power grid poses significant challenges to the sector's planning and operation, particularly concerning residential energy demand. This work investigates the application and performance of Multilayer Perceptron (MLP) Artificial Neural Networks (ANNs), trained with Backpropagation, for the short-term forecasting of residential electricity demand combined with uncoordinated EV charging. A public dataset representative of residential consumption and EV charging was used, which was processed with tools from the Python language and its libraries for data science and machine learning. The study involved exploring different hidden layer neuron configurations, activation functions such as ReLU and Sigmoid, and regularization techniques like Batch Normalization and Dropout, with performance evaluation conducted using statistical metrics such as MAPE, RMSE, and R^2. The results demonstrated the capability of MLP models to learn the complex patterns of the combined demand from residential consumption with the introduction of electric vehicles. An optimized architecture was identified—with 9, 6, and 2 neurons in the hidden layers, using ReLU activation and Batch Normalization—which achieved a MAPE of approximately 5.76% for the Re+L1 scenario. The experimental analysis revealed the importance of a careful selection of the architecture and regularization techniques to obtain a model with good generalization capacity. The study contributes to the understanding of ANN applicability in forecasting residential demand scenarios with EVs, providing an analysis of the refinement process and the factors that influence predictive performance, and reinforces the relevance of such approaches for future investigations in the context of transport electrification.

Keywords: electric load forecasting; artificial neural networks; electric vehicles; residential demand; machine learning.

LISTA DE ILUSTRAÇÕES

Figura 1 Evolução da frota global de veículos elétricos e projeções até 2030. 14
Figura 2 Exemplo de hierarquia entre IA, ML, NLP, Deep Learning e Computer Vision 22
Figura 3 O numero 2 escrito a mão de três formas diferentes. 24
Figura 4 Modelo de um neurônio artificial, ilustrando seus componentes 25
Figura 5 Fluxograma do pipeline de processamento e engenharia de dados. 35
Figura 6 Heatmap do MAPE para as arquiteturas e cenários avaliados com BN/ES . 46
Figura 7 Diagrama da arquitetura da Rede Neural Artificial MLP 9-6-2 selecionada para o estudo 51
Figura 8 Curvas de aprendizado (Loss, MAE, RMSE, R^2) para o modelo 9-6-2 ReLU Com BN, cenário Re+L1, com early stopping. 51
Figura 9 Comparação entre a demanda real e a demanda prevista pelo modelo 9-6-2 ReLU Com BN para uma semana no cenário Re+L1. 52
Figura 10 Dispersão dos valores reais vs. previstos para o modelo 9-6-2 ReLU Com BN, cenário Re+L1. 53
Figura 11 Erro de previsão (Real - Previsto) do modelo 9-6-2 ReLU Com BN para uma semana no cenário Re+L1. 53
Figura 12 Curvas de aprendizado (Loss, MAE, RMSE, R^2) para o modelo 24-12-6 ReLU Sem BN, cenário Re+L1. 63
Figura 13 Curvas de aprendizado para o modelo 32-16-8, cenário Re+L1, sem batch normalization e sem early stopping. 64
Figura 14 Curvas de aprendizado para o modelo 32-16-8, cenário Re+L1, sem batch normalization e sem early stopping. 64
Figura 15 Curvas de aprendizado para o modelo 128-64-32, cenário Re+L1, sem batch normalization e sem early stopping. 65

LISTA DE TABELAS

Tabela 1 – Comparativo internacional de pontos públicos de recarga de VEs (2024) . . 16
Tabela 2 – Relação de features por cenário e total de entradas na RNA 43
Tabela 3 – Resultados da arquitetura 32-16-8 44
Tabela 4 – Resultados da arquitetura 64-32-16 44
Tabela 5 – Resultados da arquitetura 128-64-32 44
Tabela 6 – Desempenho consolidado inicial: Melhor cenário por arquitetura e melhor arquitetura por cenário (MAPE) 45
Tabela 7 – Matriz de Entrada final para treinamento da RNA 46
Tabela 8 – Resultados da Otimização: Desempenho das Arquiteturas com Variação de Função de Ativação e Uso de Batch Normalization no Cenário Re+L1 (com ES) 47
Tabela 9 – Comparativo do Efeito do Batch Normalization por Arquitetura e Função de Ativação (Cenário Re+L1 com ES) 48
Tabela 10 – Top 5 Modelos por Score Combinado (Cenário Re+L1 com ES) 49
Tabela 11 – Desempenho Médio Agregado por Função de Ativação (Cenário Re+L1 com ES) 50
Tabela 12 – Desempenho Médio Agregado por Arquitetura (Cenário Re+L1 com ES, ordenado por MAPE crescente) 50

LISTA DE ABREVIATURAS E SIGLAS

BN	*Batch Normalization* (Normalização em Lote)
ES	*Early Stopping* (Parada Antecipada)
IA	Inteligência Artificial
L1	Nível 1 (referente ao carregamento de VEs)
L2	Nível 2 (referente ao carregamento de VEs)
LSTM	Long Short-Term Memory (Memória de Curto e Longo Prazo)
MAE	Erro Absoluto Médio (*Mean Absolute Error*)
MAPE	Erro Percentual Absoluto Médio (*Mean Absolute Percentage Error*)
MLP	Perceptron Multicamadas (*Multilayer Perceptron*)
MSE	Erro Quadrático Médio (*Mean Squared Error*)
R^2	Coeficiente de Determinação
Re	Cenário Residencial
Re+L1	Cenário Residencial com adição de carregamento de VEs Nível 1
Re+L2	Cenário Residencial com adição de carregamento de VEs Nível 2
Re+L1L2	Cenário Residencial com adição de carregamento de VEs Nível 1 e Nível 2
ReLU	Unidade Linear Retificada (*Rectified Linear Unit*)
RMSE	Raiz do Erro Quadrático Médio (*Root Mean Squared Error*)
RNA	Redes Neurais Artificiais
RNN	Redes Neurais Recorrentes (*Recurrent Neural Networks*)
VE	Veículo Elétrico
VEs	Veículos Elétricos

SUMÁRIO

1 INTRODUÇÃO 11
1.1 Objetivos do Trabalho 12

2 VEÍCULOS ELÉTRICOS 14
2.1 Comparativo Internacional de Pontos Públicos de Recarga de Veículos Elétricos 15

3 PREVISÃO DE CARGAS ELÉTRICAS 18
3.1 Métodos de Previsão de Cargas Elétricas 19
3.1.1 Métodos Estatísticos 19
3.1.2 Métodos Baseados em Aprendizado de Máquina 19
3.1.3 Métodos Híbridos 20
3.2 Aplicação de Redes Neurais Artificiais na Previsão de Cargas 20

4 MAS AFINAL DE CONTAS, O QUE É UMA REDE NEURAL? 22
4.1 "Hello World"das Redes Neurais 23
4.2 Formalismo matemático 25
4.2.1 O neurônio artificial 25
4.2.2 Funções de Ativação 26
4.2.3 Arquitetura de Redes Neurais 29
4.2.4 O Algoritmo Backpropagation 30

5 METODOLOGIA 33
5.1 Origem dos Dados 33
5.1.1 Características Principais e limitações 33
5.2 Processamento e Engenharia de Dados 34
5.2.1 Pipeline de Scripts 34
5.2.2 Reprodutibilidade dos Experimentos 37
5.2.3 Tratamento de Dados 37
5.3 Arquitetura e Treinamento da Rede Neural 38
5.3.1 Técnincas utilizadas no treinamento 39
5.3.2 Processo de Treinamento 40
5.4 Métricas de Avaliação e Interpretação dos Resultados 40

6 RESULTADOS 43
6.1 Análise Preliminar e Seleção de Arquiteturas MLP Promissoras 43
6.2 Otimização da Arquitetura, Avaliação de Funções de Ativação e Impacto do Batch Normalization no Cenário Re+L1 46

6.2.1 Avaliação do Impacto do Batch Normalization (BN) 47
6.2.2 Seleção do Modelo de Melhor Desempenho 48
6.2.3 Análise de Tendências Gerais 49
6.2.3.1 Desempenho por Função de Ativação 49
6.2.3.2 Desempenho por Complexidade da Arquitetura 50
6.3 Análise Detalhada do Modelo MLP Selecionado (9-6-2 ReLU Com BN) . . 50

7 DISCUSSÃO DOS RESULTADOS 55

8 CONCLUSÃO 57

REFERÊNCIAS 59

APÊNDICE A – CURVAS DE APRENDIZADO ADICIONAIS 63

1 INTRODUÇÃO

A crescente preocupação global com a sustentabilidade e a busca por fontes de energia mais limpas têm instigado uma transformação significativa no setor de mobilidade e, consequentemente, no setor elétrico. A Agência Internacional de Energia (AIE) projeta um aumento expressivo na frota global de veículos elétricos (VEs), que pode passar de 30 milhões em 2022 para mais de 200 milhões até 2030 (International Energy Agency, 2023). Essa transição é fundamental para a redução das emissões de gases de efeito estufa, com um potencial de diminuição de até 50% nas emissões do setor de transportes até 2050 (Intergovernmental Panel on Climate Change, 2023). Contudo, a integração em larga escala desses veículos introduz novos desafios à operação e ao planejamento das redes de distribuição elétrica, exigindo um planejamento robusto para um novo cenário de geração, controle e operação do sistema energético (Lopes; Soares; Almeida, 2011).

O presente trabalho, intitulado “Previsão de carga residencial incluindo carregamento de veículos elétricos usando Redes Neurais Artificiais”, visa estudar a aplicação e o desempenho de Redes Neurais Artificiais (RNAs) – especificamente o modelo Perceptron Multicamadas (MLP) treinado com o algoritmo de BP – na previsão da demanda de energia elétrica residencial ao se considerar o impacto do carregamento de VEs. Para a implementação e análise dos modelos, foram utilizadas ferramentas da linguagem Python, juntamente com suas bibliotecas consolidadas para manipulação de dados e aprendizado de máquina, como Pandas, NumPy, Scikit-learn e TensorFlow/Keras. Esta abordagem metodológica tem se mostrado promissora, com estudos recentes indicando alta precisão em aplicações similares (Rashid; Elfouly; Chen, 2024) quando se utilizam dados de boa qualidade. Destaca-se, nesse contexto, o trabalho de mestrado de Formaggio (2021), intitulado "Previsão de demanda de carregamento para veículos elétricos a curto prazo utilizando rede neural artificial", que empregou RNAs do tipo feedforward com treinamento de regularização Bayesiana para prever a demanda de carregamento de VEs em diferentes ambientes, incluindo o residencial (Formaggio, 2021).

Nas últimas décadas, tem-se observado uma evolução tecnológica acelerada na mobilidade elétrica, impulsionada por investimentos significativos em infraestrutura de carregamento, que ultrapassaram US$ 24 bilhões globalmente em 2023 (Bloomberg NEF, 2023). Essa mudança é fomentada por políticas públicas que visam reduzir a dependência de combustíveis fósseis e mitigar os impactos ambientais. Em paralelo, os avanços no setor de energias renováveis conferem uma nova dinâmica à rede elétrica, demandando a integração de técnicas avançadas para a previsão de demanda. Dentre estas, as RNAs se destacam por sua capacidade de modelar relações complexas e não lineares, frequentemente apresentando desempenho superior aos métodos estatísticos tradicionais em diversos estudos comparativos (Zhang; Xu; Zhou, 2010), uma vantagem também discutida por Formaggio (2021) ao revisar diferentes abordagens de previsão (Formaggio, 2021).

A introdução de VEs na matriz energética, particularmente no contexto residencial, impõe desafios consideráveis ao sistema de distribuição. O carregamento não coordenado de VEs — a prática dos usuários recarregarem seus veículos conforme as próprias necessidades sem considerar o estado da rede elétrica (Muratori, 2018) — pode levar a aumentos significativos na demanda residencial – estudos indicam acréscimos de até 40% (Rashid; Elfouly; Chen, 2024), e em cenários analisados por Formaggio (Formaggio, 2021) com base em Quirós-Tortós, o aumento pode superar 100% (Quirós-Tortós; Ochoa; Lees, 2015). Este cenário exige uma análise metodológica criteriosa que considere tanto as especificidades da carga residencial convencional quanto as do comportamento de carregamento dos VEs. A aplicação de IA, por meio de técnicas de previsão de séries temporais com RNAs, emerge, assim, como uma ferramenta valiosa para o desenvolvimento de modelos preditivos mais confiáveis e robustos, capazes de auxiliar no entendimento e gerenciamento desses novos perfis de demanda. Este trabalho está organizado em seções que detalham a origem dos dados, a metodologia de processamento e engenharia de dados, a arquitetura e o treinamento da rede neural, as métricas de avaliação, a apresentação e discussão dos resultados obtidos e, por fim, as conclusões.

1.1 OBJETIVOS DO TRABALHO

Diante do contexto de crescente adoção de VEs e os consequentes desafios para o sistema elétrico, o presente trabalho tem como objetivo geral investigar e analisar a aplicação de Redes Neurais Artificiais (RNAs) do tipo Perceptron Multicamadas (MLP), treinadas com o algoritmo de backpropagation, para a previsão da carga elétrica residencial quando se considera o impacto do carregamento de VEs. Este estudo busca compreender o desempenho e a viabilidade dessa abordagem de modelagem, utilizando a linguagem Python e suas bibliotecas especializadas em ciência de dados e aprendizado de máquina, como Pandas, NumPy, Scikit-learn, TensorFlow e Keras. A intenção é explorar a capacidade do modelo em capturar os padrões de consumo combinado (residencial e VEs) e obter insights sobre a sua aplicação em estudos de planejamento energético.

Para alcançar este objetivo geral, foram estabelecidos os seguintes objetivos específicos:

1. Analisar o impacto do carregamento não coordenado de VEs sobre a demanda de energia em unidades residenciais, identificando as alterações nos perfis de consumo a partir de dados de referência da literatura, com foco na variabilidade temporal e na intensidade da demanda adicionada.

2. Implementar e treinar modelos de redes neurais MLP, explorando diferentes arquiteturas (número de camadas e neurônios) e otimizando seus hiperparâmetros, incluindo a aplicação de técnicas de regularização (como *Dropout* e *Batch Normalization*) e o uso de conjuntos de validação para monitorar o desempenho durante o treinamento (Goodfellow; Bengio; Courville, 2016).

3. Desenvolver todo o pipeline de processamento de dados e modelagem no ambiente Python, utilizando bibliotecas como `Pandas` para manipulação de dados, `Scikit-learn` para pré-processamento e divisão dos dados, e `TensorFlow` com `Keras` para a construção e treinamento das redes neurais.

4. Avaliar o desempenho dos modelos de previsão desenvolvidos por meio de um conjunto de métricas estatísticas robustas, como o Erro Percentual Absoluto Médio (MAPE), Raiz do Erro Quadrático Médio (RMSE), Erro Absoluto Médio (MAE) e o Coeficiente de Determinação (R^2), comparando os resultados obtidos entre as diferentes configurações testadas.

5. Discutir as implicações dos resultados da previsão para um melhor entendimento do impacto do carregamento não coordenado de VEs no perfil de demanda residencial e a relevância de tais modelos como ferramenta de análise para estudos futuros na área de planejamento e operação de sistemas elétricos.

A condução deste estudo envolve a abordagem de alguns desafios inerentes à modelagem preditiva com RNAs, tais como:

- A dependência da qualidade e das características do conjunto de dados utilizado (perfis sintéticos baseados em dados reais norte-americanos), que influencia diretamente a capacidade de generalização e a acurácia dos modelos preditivos desenvolvidos.

- A necessidade de um processo criterioso de otimização dos hiperparâmetros da rede neural (arquitetura, taxa de aprendizado, funções de ativação), fundamental para alcançar um bom desempenho e evitar problemas como a lenta convergência ou a estagnação em mínimos locais insatisfatórios (Goodfellow; Bengio; Courville, 2016).

- A mitigação do sobreajuste (*overfitting*), empregando técnicas de regularização e o monitoramento do desempenho em um conjunto de validação, para assegurar que o modelo seja capaz de generalizar para dados não vistos durante o treinamento.

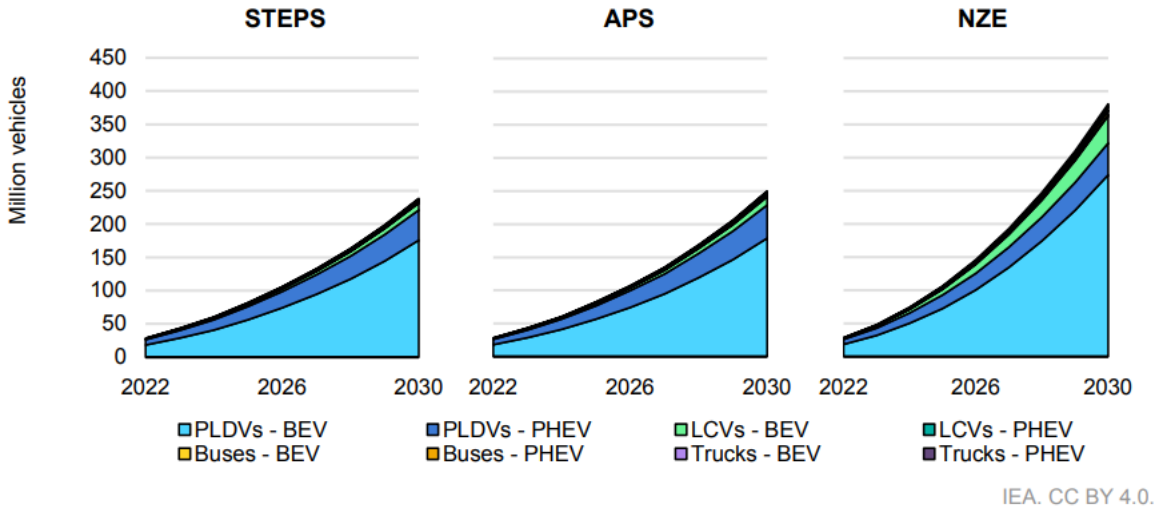
- A inserção de *features* para incorporar variáveis relevantes (como indicadores temporais, características sazonais e a quantidade de VEs em carregamento) que auxiliem o modelo a capturar os diferentes fatores complexos que influenciam o perfil de consumo energético.

Em síntese, este trabalho busca contribuir para a área de previsão de cargas elétricas por meio de uma análise detalhada da aplicação de RNAs MLP no contexto da integração de veículos elétricos à demanda residencial. A abordagem proposta visa não apenas avaliar a precisão das previsões, mas também gerar conhecimento sobre o processo de modelagem e os fatores que influenciam o desempenho, fornecendo uma base para futuras investigações e para a compreensão dos desafios impostos pela eletrificação do transporte ao setor elétrico.

2 VEÍCULOS ELÉTRICOS

Os VEs têm se consolidado como uma das principais soluções para a transição energética global, sendo amplamente promovidos como alternativa sustentável aos veículos movidos a combustíveis fósseis. Essa transição é impulsionada por políticas públicas, avanços tecnológicos e mudanças no comportamento dos consumidores, que buscam reduzir sua pegada de carbono e adotar tecnologias mais eficientes. Segundo a Agência Internacional de Energia (International Energy Agency, 2023), a frota global de VEs ultrapassou 30 milhões de unidades em 2022, com projeções de crescimento para mais de 200 milhões até 2030, caso as metas climáticas globais sejam cumpridas (International Energy Agency, 2023).

Figura 1 – Evolução da frota global de veículos elétricos e projeções até 2030.



Fonte: (International Energy Agency, 2023).
Legenda: Notas: STEPS = Cenário de Políticas Declaradas; APS = Cenário de Compromissos Anunciados; NZE = Cenário de Emissões Líquidas Zero até 2050; BEV = veículo elétrico a bateria; PHEV = veículo elétrico híbrido plug-in; PLDV = veículo leve de passageiros; LCV = veículo comercial leve.

A Figura 1 ilustra a evolução projetada da frota global de VEs entre 2022 e 2030, considerando diferentes cenários de políticas e compromissos climáticos. No cenário de Políticas Declaradas (STEPS), que reflete as políticas e regulamentações já implementadas ou formalmente anunciadas pelos governos, a frota de VEs (excluindo motos e triciclos) cresce de cerca de 30 milhões em 2022 para aproximadamente 240 milhões em 2030, com uma taxa média anual de crescimento de 30%. Nesse cenário, os VEs representam mais de 10% da frota total de veículos rodoviários em 2030 (International Energy Agency, 2023).

A figura também compara o STEPS com o cenário de Compromissos Anunciados (APS), que assume o cumprimento integral e pontual de todas as metas e promessas de eletrificação e neutralidade de carbono anunciadas pelos governos, mesmo que ainda não estejam legisladas. O

APS projeta uma frota ainda maior de VEs, evidenciando o “implementation gap” — a diferença entre as políticas já implementadas e as ações necessárias para atingir as metas anunciadas. O cenário Net Zero Emissions by 2050 (NZE) apresenta um caminho normativo para zerar as emissões líquidas de CO_2 até 2050, compatível com o limite de aquecimento global de 1,5°C (International Energy Agency, 2023). A diferença entre o APS e o NZE destaca o “ambition gap”, ou seja, o esforço adicional necessário para alinhar as políticas globais às metas do Acordo de Paris (International Energy Agency, 2023).

Os VEs utilizam motores elétricos alimentados por baterias recarregáveis, geralmente compostas por íons de lítio, que oferecem alta densidade energética e eficiência. A eficiência energética dos VEs é significativamente superior à dos veículos a combustão interna, com taxas de conversão de energia que podem ultrapassar 90%, em comparação com os 25% a 30% típicos dos motores a combustão (Ehsani; Gao; Longo, 2018). No entanto, a adoção em larga escala desses veículos apresenta desafios significativos para os sistemas de distribuição de energia, exigindo adaptações tecnológicas, regulatórias e comportamentais.

A União Europeia estabeleceu que, a partir de 2035, todos os novos veículos vendidos deverão ser livres de emissões de carbono (European Commission, 2023). Nos Estados Unidos, o governo federal oferece créditos fiscais de até US$ 7.500 para a compra de VEs, enquanto estados como a Califórnia têm metas ambiciosas para a eletrificação do transporte (U.S. Department of Energy, 2023). No Brasil, a adoção de VEs ainda é tímida, mas vem crescendo rapidamente. Segundo a Associação Brasileira do Veículo Elétrico (ABVE), o número de veículos eletrificados (híbridos e elétricos puros) em circulação no país ultrapassou 150 mil unidades em 2023, representando um aumento de 40% em relação ao ano anterior (Associação Brasileira do Veículo Elétrico, 2023). Apesar desse crescimento, desafios como o alto custo inicial dos veículos e a falta de infraestrutura de carregamento ainda limitam a expansão do mercado nacional.

2.1 COMPARATIVO INTERNACIONAL DE PONTOS PÚBLICOS DE RECARGA DE VEÍCULOS ELÉTRICOS

A infraestrutura de carregamento é um dos principais fatores que influenciam a adoção de VEs. A disponibilidade de estações de carregamento públicas e privadas é essencial para garantir a conveniência e a confiabilidade do uso desses veículos. Países como Noruega, China e Estados Unidos têm investido massivamente na expansão dessa infraestrutura. Na Noruega, por exemplo, há uma estação de carregamento para cada 10 veículos elétricos, enquanto na China o número de estações públicas ultrapassa 1,8 milhão (China Electric Vehicle Association, 2023). Entretanto, a expansão da infraestrutura de carregamento apresenta desafios técnicos e econômicos. A instalação de estações de carregamento rápido (DC) requer investimentos significativos em equipamentos e reforço da rede elétrica local, enquanto o carregamento residencial depende da capacidade elétrica das residências e da coordenação com as distribuidoras de energia (Verma; Tripathi; Naqvi, 2021).

Tabela 1 – Comparativo internacional de pontos públicos de recarga de VEs (2024)

País/Região	Pontos	Área (km^2)	Pop. (mi)	Pts/km^2	Pts/mi hab.
Brasil	11.000	8.515.767	203	0,0013	54,19
China	3.600.000	9.596.961	1.425	0,3752	2.526
EUA	61.000	9.371.180	339	0,0066	180
Europa (UE+RU)	882.000	4.324.782	448	0,2043	1.968
Reino Unido	75.000	243.610	67	0,3079	1.119
Alemanha	130.000	357.386	84	0,3637	1.548
França	127.000	640.695	65	0,1982	1.954
Noruega	26.000	325.207	5,5	0,0799	4.727

Fontes: Adaptado de IEA (International Energy Agency, 2023; Agency, 2024; Agency, 2022), ABVE (Elétrico, 2024), European Alternative Fuels Observatory (Observatory, 2024).

Países como China, Estados Unidos e membros da União Europeia têm investido fortemente na expansão de pontos de recarga, enquanto o Brasil ainda apresenta uma densidade significativamente inferior. A Tabela 1 apresenta um comparativo entre o Brasil e outros países ou blocos econômicos, considerando o número absoluto de pontos públicos de recarga, área territorial, população e densidade de pontos por área e por habitante, com dados projetados para o ano de 2024.

Observa-se que a China lidera mundialmente, concentrando cerca de 70% dos pontos públicos de recarga globais, seguida pela Europa e pelos Estados Unidos. O Brasil, apesar de avanços recentes, ainda possui uma infraestrutura muito pequena, o que representa um desafio para a expansão da mobilidade elétrica no país. A baixa densidade de pontos de recarga, tanto em relação à área territorial quanto à população, evidencia a necessidade de políticas públicas e investimentos para acelerar a implantação dessa infraestrutura, especialmente em regiões fora dos grandes centros urbanos.

O comportamento dos consumidores também desempenha papel crucial na integração dos VEs à sociedade. Estudos mostram que a maioria dos usuários prefere carregar seus veículos em casa, durante a noite, quando as tarifas de energia são mais baixas (Institute, 2021). No entanto, essa preferência pode gerar picos de demanda em horários específicos, sobrecarregando a rede elétrica (Lopes; Soares; Almeida, 2011). Além disso, a falta de conhecimento sobre práticas de carregamento eficiente e a percepção de que a infraestrutura de carregamento público é insuficiente podem limitar a adoção de VEs (Institute, 2021).

A integração dos VEs ao sistema de distribuição de energia apresenta desafios significativos, que exigem adaptações na infraestrutura, no planejamento e na operação das redes elétricas. O carregamento de VEs adiciona uma carga significativa à demanda de energia elétrica, especialmente em áreas residenciais. Um único veículo elétrico pode consumir entre 7 kWh e 20 kWh por dia, dependendo do modelo e do padrão de uso (Tesla Motors, 2023). Em um cenário onde múltiplos veículos são carregados simultaneamente em uma mesma área, a demanda

total pode exceder a capacidade instalada da rede, resultando em sobrecargas e interrupções no fornecimento de energia (Muratori, 2018).

Além do aumento da demanda, o comportamento de carregamento dos usuários de VEs é altamente variável e depende de fatores como horários de trabalho, disponibilidade de estações de carregamento e tarifas de energia. Em geral, os horários de pico de carregamento coincidem com os horários de maior consumo residencial, como o período noturno, agravando os desafios de gerenciamento da demanda (Lopes; Soares; Almeida, 2011). A infraestrutura de distribuição elétrica, incluindo transformadores, cabos e subestações, foi projetada para atender a padrões de consumo convencionais. A introdução de VEs altera significativamente esses padrões, aumentando a probabilidade de falhas e reduzindo a vida útil dos equipamentos. Além disso, o carregamento descoordenado pode causar desequilíbrios de tensão e perdas técnicas adicionais, comprometendo a eficiência do sistema (Verma; Tripathi; Naqvi, 2021).

Diante desses desafios, diversas estratégias têm sido propostas para mitigar os impactos dos VEs na rede elétrica. O carregamento coordenado, por exemplo, visa distribuir a demanda de energia ao longo do tempo, evitando picos de consumo e reduzindo a sobrecarga na infraestrutura de distribuição. Essa estratégia pode ser implementada por meio de incentivos tarifários, que encorajam os usuários a carregarem seus veículos em horários de menor demanda, ou por meio de sistemas automatizados que controlam o carregamento de forma centralizada (Rashid; Elfouly; Chen, 2024). A integração de fontes renováveis, como energia solar e eólica, pode reduzir a dependência de fontes convencionais e mitigar os impactos ambientais do carregamento de VEs. No entanto, a intermitência dessas fontes exige o uso de sistemas de armazenamento de energia e técnicas avançadas de previsão para garantir o equilíbrio entre oferta e demanda (Rashid; Elfouly; Chen, 2024). Por fim, as redes inteligentes (smart grids) utilizam tecnologias de comunicação e automação para monitorar e controlar o fluxo de energia em tempo real, permitindo a integração eficiente dos VEs, otimizando o uso da infraestrutura existente e melhorando a confiabilidade do sistema elétrico (Zhang; Xu; Zhou, 2010).

Em síntese, os VEs representam uma tendência irreversível na mobilidade global, trazendo muitas vezes benefícios ambientais e econômicos relevantes, mas também desafios significativos para o setor elétrico. A compreensão desses desafios e das estratégias para sua mitigação é fundamental para o planejamento e a operação eficiente dos sistemas de energia do futuro, especialmente em cenários de alta penetração de VEs, como o que se projeta para as próximas décadas (International Energy Agency, 2023).

3 PREVISÃO DE CARGAS ELÉTRICAS

A previsão de cargas elétricas constitui uma atividade fundamental no planejamento e na operação eficiente dos sistemas de energia elétrica. Esta prática permite que os diversos agentes do setor, desde operadores de sistema até concessionárias e planejadores, antecipem a demanda futura de energia. Tal antecipação é crucial para assegurar a continuidade do fornecimento, evitando problemas como sobrecargas ou interrupções, ao mesmo tempo em que se otimiza a alocação de recursos energéticos e se minimizam os custos operacionais (Rashid; Elfouly; Chen, 2024; Hyndman; Athanasopoulos, 2021).

O cenário energético contemporâneo tem intensificado os desafios associados à previsão de cargas. Dois fatores principais contribuem para essa complexidade: a crescente integração de fontes de energia renovável, caracterizadas por sua intermitência (como a solar e a eólica), e a emergência de novas categorias de carga, com destaque para os VEs. A IEA sublinha que a previsão de cargas é uma ferramenta indispensável para manter a estabilidade e a confiabilidade do sistema elétrico durante a transição energética global (International Energy Agency, 2023).

A expansão da frota de VEs, em particular, introduz uma nova dinâmica ao consumo de energia residencial. O carregamento desses veículos, especialmente se não coordenado, pode gerar picos de demanda significativos e impor uma variabilidade considerável aos perfis de consumo diários (Muratori, 2018). Nesse contexto, uma previsão de cargas precisa e adaptativa desempenha papéis essenciais, tais como:

- **Planejamento da Expansão e Operação da Rede:** Auxiliar no dimensionamento adequado da infraestrutura elétrica (transformadores, linhas, subestações) para suportar o aumento da demanda agregada, conforme detalhado por estudos sobre o impacto residencial dos VEs (Formaggio, 2021; Muratori, 2018).

- **Gerenciamento da Demanda:** Permitir a antecipação de períodos de consumo elevado, viabilizando a aplicação de estratégias para mitigar picos, como programas de resposta à demanda ou incentivos para o carregamento em horários de menor carga (Rashid; Elfouly; Chen, 2024; Lopes; Soares; Almeida, 2011).

- **Integração de Recursos Energéticos Distribuídos:** Facilitar o balanço entre oferta e demanda em redes com alta penetração de fontes renováveis e VEs, que podem atuar tanto como cargas quanto, futuramente, como recursos de armazenamento (vehicle-to-grid) (Lopes; Soares; Almeida, 2011).

- **Otimização de Custos:** Contribuir para a redução de custos operacionais ao otimizar o despacho de unidades geradoras e a compra de energia (Rashid; Elfouly; Chen, 2024; Hyndman; Athanasopoulos, 2021).

Historicamente, a previsão de carga apoiava-se em métodos estatísticos clássicos, como médias móveis, regressão linear e modelos de séries temporais da família ARIMA (Hyndman; Athanasopoulos, 2021). Embora eficazes para padrões de consumo relativamente estáveis, esses métodos encontram limitações diante da não linearidade e da complexidade introduzidas pelas novas dinâmicas do sistema elétrico, incluindo o comportamento de carregamento dos VEs (Formaggio, 2021). O estudo de Formaggio (2021) também destaca a dificuldade adicional na previsão em estações de carregamento devido à maior imprevisibilidade em comparação com o carregamento residencial (Formaggio, 2021).

3.1 MÉTODOS DE PREVISÃO DE CARGAS ELÉTRICAS

Os métodos de previsão de carga podem ser categorizados em três grandes grupos, cada um com suas particularidades e âmbitos de aplicação:

3.1.1 Métodos Estatísticos

Esses métodos baseiam-se em modelos matemáticos que buscam identificar padrões históricos e tendências nos dados de consumo. Exemplos incluem (Hyndman; Athanasopoulos, 2021):

- **Média Móvel e Suavização Exponencial:** Técnicas que utilizam valores passados da série temporal para prever valores futuros, adequadas para séries com certa estabilidade.
- **Regressão Linear Múltipla:** Modela a relação entre a carga e variáveis exógenas influentes, como temperatura, dia da semana, e indicadores econômicos.
- **Modelos ARIMA (Autoregressive Integrated Moving Average) e SARIMA (Seasonal ARIMA):** Amplamente utilizados para séries temporais com componentes de tendência e sazonalidade.

Apesar de sua utilidade, os métodos estatísticos podem ter dificuldade em capturar relações não lineares complexas presentes nos dados de consumo modernos, especialmente com a influência de VEs (Formaggio, 2021).

3.1.2 Métodos Baseados em Aprendizado de Máquina

Com o avanço da inteligência artificial, os métodos de aprendizado de máquina (AM) têm ganhado destaque, dada a sua capacidade de aprender padrões complexos diretamente dos dados (Goodfellow; Bengio; Courville, 2016). Alguns dos mais utilizados são:

- **Redes Neurais Artificiais (RNAs):** Modelos inspirados na estrutura do cérebro humano, capazes de modelar relações não lineares e dinâmicas complexas, tornando-se uma escolha robusta para a previsão de carga com VEs (Haykin, 2001; Goodfellow; Bengio; Courville, 2016; Kim et al., 2024).

- **Máquinas de Vetores de Suporte (SVM - Support Vector Machines):** Algoritmos de aprendizado supervisionado eficazes tanto para classificação quanto para regressão, incluindo a previsão de séries temporais (Pedregosa et al., 2018).
- **Florestas Aleatórias (Random Forests) e Árvores de Decisão:** Métodos de conjunto que combinam múltiplas árvores de decisão para melhorar a precisão e robustez da previsão (Pedregosa et al., 2018)..

Estes métodos são implementados em diversas bibliotecas de software, como Scikit-learn (Pedregosa et al., 2018).

3.1.3 Métodos Híbridos

Os métodos híbridos buscam combinar as fortalezas de diferentes abordagens. Por exemplo, um modelo estatístico pode ser usado para capturar a tendência e a sazonalidade base, enquanto um modelo de AM modela a componente residual não linear. Essa combinação pode levar a um desempenho superior em cenários desafiadores (Formaggio, 2021).

3.2 APLICAÇÃO DE REDES NEURAIS ARTIFICIAIS NA PREVISÃO DE CARGAS

Conforme detalhado na metodologia deste trabalho (Capítulo 5), optou-se pela utilização de Redes Neurais Artificiais do tipo Perceptron Multicamadas (MLP). As MLPs são um tipo de RNA feedforward que consiste em, no mínimo, uma camada de entrada, uma ou mais camadas ocultas de neurônios, e uma camada de saída (Haykin, 2001). Os neurônios de uma camada são conectados aos neurônios da camada subsequente através de pesos sinápticos, e cada neurônio aplica uma função de ativação à soma ponderada de suas entradas para produzir sua saída (Haykin, 2001).

O processo de aprendizado em uma MLP é comumente realizado pelo algoritmo de backpropagation (retropropagação do erro) (Werbos, 1974). Este algoritmo ajusta iterativamente os pesos da rede de forma a minimizar uma função de custo, que tipicamente mede a diferença entre os valores previstos pela rede e os valores reais (Goodfellow; Bengio; Courville, 2016; Haykin, 2001).

As RNAs oferecem vantagens significativas para a previsão de cargas elétricas, como:

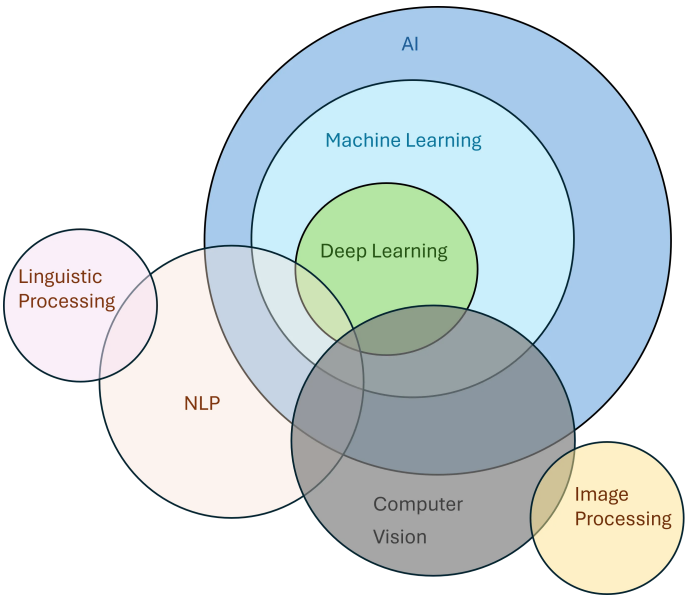
- **Modelagem de Não Linearidades:** Capacidade intrínseca de aprender e representar relações não lineares complexas entre as variáveis de entrada (como dados históricos de carga, variáveis temporais, número de VEs) e a carga futura (Haykin, 2001).
- **Adaptabilidade:** Habilidade de se adaptar a mudanças nos padrões de consumo ao longo do tempo, especialmente se re-treinadas periodicamente com novos dados (Haykin, 2001).
- **Robustez a Ruído:** Capacidade de lidar com um certo grau de ruído e incompletude nos dados de entrada (Haykin, 2001).

Contudo, o desenvolvimento de modelos de RNA eficazes também envolve desafios. A qualidade da previsão é altamente dependente da disponibilidade, representatividade e pré-processamento dos dados históricos (Formaggio, 2021). A definição da arquitetura da rede (número de camadas e neurônios) e a otimização de outros hiperparâmetros (como taxa de aprendizado e funções de ativação) são cruciais e frequentemente exigem um processo experimental iterativo (Goodfellow; Bengio; Courville, 2016). Portanto, é fundamental empregar técnicas para evitar o sobreajuste (*overfitting*), como a regularização (e.g., *Dropout*, *Batch Normalization*) e o uso de um conjunto de validação, garantindo que o modelo generalize bem para dados não vistos (Srivastava et al., 2014; Ioffe; Szegedy, 2015; Goodfellow; Bengio; Courville, 2016). A avaliação final do desempenho do modelo deve ser realizada utilizando métricas estatísticas consolidadas, como o Erro Percentual Absoluto Médio (MAPE), que oferece uma medida relativa da acurácia da previsão (Proceffa, 2023; Kim et al., 2024).

4 MAS AFINAL DE CONTAS, O QUE É UMA REDE NEURAL?

Nos tempos atuais, quando pensamos em inteligência artificial, vêm à mente exemplos como o ChatGPT, Gemini e outros sistemas avançados de processamento de linguagem natural ou geração de imagens. Essas ferramentas já fazem parte do nosso cotidiano e estão presentes no imaginário coletivo, muitas vezes levando a uma compreensão superficial ou até mesmo equivocada do que realmente é inteligência artificial (IA).

Figura 2 – Exemplo de hierarquia entre IA, ML, NLP, Deep Learning e Computer Vision



Fonte: Adaptado de (Jain, 2023).

Como ilustrado na Figura 2, a inteligência artificial é um campo amplo, que engloba diversas subáreas, como o aprendizado de máquina (*machine learning*), o processamento de linguagem natural (NLP), o aprendizado profundo (*deep learning*) e a visão computacional (*computer vision*) (Jain, 2023). Essas áreas se sobrepõem em diferentes níveis de especialização e complexidade, o que evidencia que reduzir a IA a um único segmento de toda essa dimensionalidade é, no mínimo, reducionista.

No entanto, é fundamental entender os princípios que fundamentam essas tecnologias. Entre os diversos métodos e algoritmos que compõem o universo da IA, as RNAs ocupam um papel central, especialmente no contexto do aprendizado profundo. As RNAs são inspiradas, de forma bastante simplificada, no funcionamento do cérebro humano, buscando replicar a capacidade de aprender a partir de exemplos, identificar padrões e tomar decisões com base em dados (Haykin, 2001).

A motivação para o estudo das redes neurais vai além do fascínio tecnológico. Elas representam uma das ferramentas mais poderosas para resolver problemas complexos, nos quais métodos tradicionais de programação se mostram ineficazes. Por exemplo, tarefas como reconhecimento

de voz, tradução automática, diagnóstico médico por imagem e previsão de séries temporais são áreas em que as RNAs têm apresentado resultados notáveis (Goodfellow; Bengio; Courville, 2016).

De forma intuitiva, uma rede neural pode ser vista como um sistema composto por unidades simples, chamadas de neurônios artificiais, organizadas em camadas. Cada neurônio realiza operações matemáticas básicas, mas, quando conectados em grande quantidade e treinados adequadamente, são capazes de modelar relações altamente complexas entre entradas e saídas. A série de vídeos e textos de Grant Sanderson, do projeto 3Blue1Brown, apresenta uma abordagem visual e acessível para entender como essas redes funcionam, mostrando que, apesar da complexidade aparente, o conceito fundamental por trás das RNAs é bastante intuitivo: ajustar parâmetros internos para que a rede aprenda a associar padrões de entrada a respostas desejadas (Sanderson, 2017).

Portanto, antes de adentrarmos nos detalhes técnicos e matemáticos das RNAs, é importante construir uma compreensão intuitiva de seu funcionamento e de sua relevância no contexto da inteligência artificial moderna. Essa abordagem facilita o entendimento dos capítulos seguintes, nos quais serão explorados os fundamentos, a modelagem e a aplicação das RNAs na previsão de cargas residenciais, incluindo o carregamento de veículos elétricos.

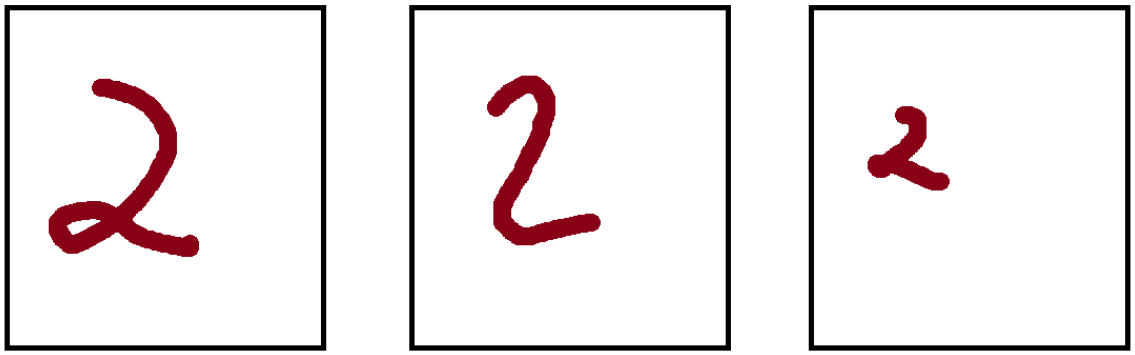
4.1 "HELLO WORLD"DAS REDES NEURAIS

O termo "Hello World"tornou-se um marco tradicional no ensino de qualquer linguagem de programação. Ele representa o primeiro passo de um iniciante ao aprender uma nova linguagem, geralmente consistindo em escrever um código simples que exibe a mensagem "Hello World"na tela ou na interface utilizada. Esse exercício inicial tem como objetivo proporcionar ao aprendiz um contato direto com a sintaxe básica da linguagem, além de introduzir conceitos fundamentais como estrutura de comandos, compilação ou execução do código. Além disso, esse costume é amplamente difundido entre programadores, funcionando quase como um ritual de iniciação no universo da programação (Sanderson, 2017).

No contexto das RNAs, existe um conceito análogo ao "Hello World", que consiste em compreender o funcionamento de uma rede neural simples aplicada à identificação de dígitos manuscritos, de 0 a 9. Esse problema, conhecido como reconhecimento de dígitos manuscritos, é frequentemente utilizado como ponto de partida para quem está começando a estudar redes neurais, pois envolve uma arquitetura suficientemente simples para que seja possível acompanhar e entender cada etapa do processamento, desde a entrada dos dados até a saída da rede (Sanderson, 2017). Se o estudante fosse introduzido diretamente a redes neurais mais modernas e complexas, como as redes profundas (*deep learning*), provavelmente encontraria uma barreira significativa de compreensão, especialmente sem o devido embasamento teórico e prático (Sanderson, 2017).

A tarefa de identificar dígitos escritos à mão, apesar de parecer trivial para um ser humano, é um excelente exemplo de desafio para sistemas computacionais. Isso ocorre porque, ao longo

Figura 3 – O numero 2 escrito a mão de três formas diferentes.



Fonte: Elaborado pelo autor.

da vida, as pessoas se familiarizam com inúmeras variações de escrita de um mesmo número, desenvolvendo uma habilidade quase automática de reconhecer padrões, mesmo quando o traço, o tamanho ou a inclinação variam consideravelmente. Por exemplo, como ilustrado na Figura 3, o dígito "2"pode ser escrito de formas bastante distintas, mas o cérebro humano, graças à sua capacidade de processamento visual e reconhecimento de padrões, identifica rapidamente que todas aquelas representações correspondem ao mesmo número. Esse processo ocorre de maneira tão natural que raramente nos damos conta da complexidade envolvida (Sanderson, 2017).

No entanto, ao tentar descrever em detalhes como o cérebro realiza essa tarefa, percebemos que se trata de um problema extremamente complexo. Se tentássemos programar um computador para reconhecer dígitos manuscritos utilizando apenas métodos tradicionais de programação — como estruturas condicionais (*if, else*), laços de repetição (*loops*), classes, objetos e funções —, rapidamente nos depararíamos com limitações, pois seria necessário prever e codificar todas as possíveis variações de escrita, o que é praticamente inviável (Sanderson, 2017).

A ideia central por trás das RNAs é mimetizar, de forma simplificada, a estrutura e o funcionamento do cérebro humano, permitindo que o computador aprenda a partir da exposição a múltiplos exemplos (Haykin, 2001). Ao "ver"diversas variações de um mesmo dígito durante o treinamento, a rede neural é capaz de extrair características relevantes e generalizar esse conhecimento para reconhecer novas variações que nunca foram apresentadas anteriormente (Goodfellow; Bengio; Courville, 2016). Dessa forma, as RNAs tornam possível resolver problemas que desafiam os métodos tradicionais de programação, aproximando o processamento computacional da flexibilidade e adaptabilidade do cérebro humano (Sanderson, 2017).

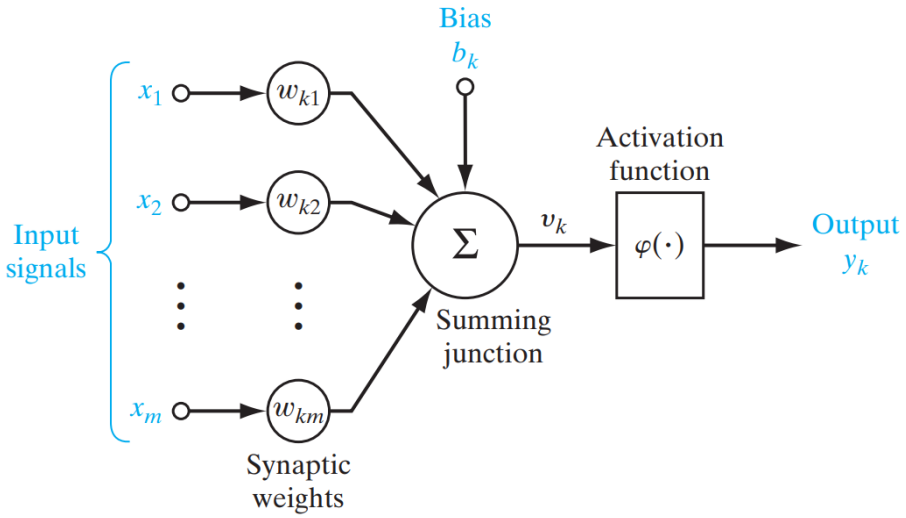
4.2 FORMALISMO MATEMÁTICO

Como vimos, as RNAs são modelos computacionais inspirados na estrutura e no funcionamento do cérebro humano. Elas são projetadas para processar informações de maneira análoga à forma como os neurônios biológicos operam, permitindo que aprendam padrões complexos a partir de dados. A seguir, exploraremos os conceitos e as formulações matemáticas que fundamentam a aplicabilidade e a eficiência desses modelos (Haykin, 2001; Goodfellow; Bengio; Courville, 2016).

4.2.1 O neurônio artificial

O neurônio artificial, também conhecido como unidade de processamento, é o componente fundamental de uma RNA (Haykin, 2001). Sua concepção é uma simplificação do neurônio biológico, que recebe múltiplos sinais de entrada (dendritos), processa essas informações em seu corpo celular e, caso um limiar de excitação seja atingido, transmite um sinal de saída através do axônio. No neurônio artificial, esse processo é mimetizado através de uma combinação linear ponderada das entradas, seguida pela aplicação de uma função de ativação para determinar a saída do neurônio (Haykin, 2001; Goodfellow; Bengio; Courville, 2016), conforme ilustra o modelo conceitual apresentado na Figura 4 onde é possível observar os sinais de entrada ($x_1, x_2, \ldots, x_m$), os pesos sinápticos ($w_{k1}, w_{k2}, \ldots, w_{km}$), o termo de polarização (*peso bias* b_k), a junção aditiva (somador), a saída linear (v_k) do neurônio k a função de ativação ($\phi(\cdot)$) e o sinal de saída não linear (y_k).

Figura 4 – Modelo de um neurônio artificial, ilustrando seus componentes



Fonte: (Haykin, 2001).

Matematicamente, a operação de um neurônio artificial pode ser descrita em duas etapas principais (Haykin, 2001):

1. **Combinação Linear (ou Soma Ponderada)**: O neurônio calcula uma soma ponderada de suas n entradas x_i, onde cada entrada é multiplicada por um peso sináptico w_i correspondente. Um termo de polarização (viés ou *bias*) b também é adicionado a essa soma. O resultado, frequentemente denominado potencial de ativação ou entrada líquida z, é dado pela Equação 1 (Haykin, 2001; Goodfellow; Bengio; Courville, 2016):

$$z = \sum_{i=1}^{n} w_i x_i + b \tag{1}$$

   onde:

   - x_i representa o valor da i-ésima entrada do neurônio.
   - w_i é o peso sináptico associado à i-ésima entrada, indicando a força ou importância dessa conexão.
   - b é o termo de viés, que permite ajustar o limiar de ativação do neurônio, deslocando a função de ativação.
   - z é o resultado da combinação linear das entradas ponderadas pelos respectivos pesos, mais o viés.

2. **Função de Ativação**: Após o cálculo de z, o neurônio aplica uma função de ativação $f(\cdot)$ para introduzir não linearidade ao modelo e determinar sua saída a, conforme a Equação 2:

$$a = f(z) = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{2}$$

   onde a é a saída (ou ativação) não linear do neurônio. A escolha da função de ativação é crucial e depende da natureza do problema e da camada da rede (Haykin, 2001; Goodfellow; Bengio; Courville, 2016).

4.2.2 Funções de Ativação

As funções de ativação desempenham um papel crítico nas redes neurais, sendo responsáveis por introduzir comportamentos não lineares no modelo (Haykin, 2001). Se as redes neurais utilizassem apenas transformações lineares (ou seja, sem funções de ativação não lineares), uma RNA com múltiplas camadas seria matematicamente equivalente a uma única camada linear, limitando severamente sua capacidade de aprender mapeamentos complexos entre entradas e saídas (Goodfellow; Bengio; Courville, 2016).

Algumas das principais funções de ativação utilizadas, com destaque para Sigmoide e ReLU que são relevantes para este trabalho, são:

- **Função Sigmoide (Logística)**: A função sigmoide logística é definida de acordo com a equação 3:

$$f(z) = \sigma(z) = \frac{1}{1 + e^{-z}} \tag{3}$$

Esta função mapeia qualquer valor real de entrada z para o intervalo $(0, 1)$ (Haykin, 2001). Sua forma em "S"(sigmoidal) a torna suave e diferenciável em todos os pontos, o que é uma propriedade importante para o algoritmo de *backpropagation*. **Vantagens:** A saída entre 0 e 1 pode ser interpretada como uma probabilidade, tornando-a útil em camadas de saída para problemas de classificação binária ou como uma forma de normalizar ativações (Goodfellow; Bengio; Courville, 2016). **Desvantagens:** Uma das principais desvantagens é o problema da *saturação* e do *gradiente evanescente* (*vanishing gradient*) (Goodfellow; Bengio; Courville, 2016). Quando a entrada z assume valores muito grandes (positivos ou negativos), a saída da sigmoide se aproxima de 1 ou 0, respectivamente. Nessas regiões de saturação, o gradiente (derivada) da função se torna muito próximo de zero. Em redes profundas, se muitos neurônios operam nessas regiões saturadas, os gradientes multiplicados durante o *backpropagation* podem se tornar excessivamente pequenos, dificultando ou paralisando o aprendizado das camadas iniciais da rede (Haykin, 2001; Goodfellow; Bengio; Courville, 2016). Além disso, suas saídas não são centradas em zero, o que pode, em alguns casos, tornar o treinamento menos eficiente.

- **Função Tangente Hiperbólica (tanh)**: Relacionada à sigmoide, a função tanh é definida de acordo com a equação 4:

$$f(z) = \tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} = 2\sigma(2z) - 1 \tag{4}$$

Vantagens: Similar à sigmoide, também é sigmoidal e diferenciável, mas mapeia as entradas para o intervalo $(-1, 1)$ (Haykin, 2001). O fato de sua saída ser centrada em zero (média próxima de zero) pode, por vezes, ajudar a acelerar a convergência durante o treinamento em comparação com a sigmoide logística (Goodfellow; Bengio; Courville, 2016). **Desvantagens:** Assim como a sigmoide, a função tanh também sofre com o problema do gradiente evanescente nas regiões de saturação (Goodfellow; Bengio; Courville, 2016).

- **Função ReLU (Rectified Linear Unit - Unidade Linear Retificada)**: A função ReLU tornou-se uma das funções de ativação mais populares, especialmente em redes profundas, devido à sua simplicidade e eficácia. É definida segundo a equação 5:

$$f(z) = \max(0, z) = \begin{cases} z & \text{se } z > 0 \\ 0 & \text{se } z \leq 0 \end{cases} \tag{5}$$

Vantagens:

- *Eficiência Computacional:* É muito simples de calcular (uma comparação e, possivelmente, uma atribuição a zero) (Goodfellow; Bengio; Courville, 2016).
- *Mitigação do Gradiente Evanescente:* Para entradas positivas ($z > 0$), o gradiente da ReLU é constante e igual a 1. Isso significa que, enquanto o neurônio estiver ativo, o gradiente não diminui à medida que se propaga para trás, ajudando a evitar o problema do gradiente evanescente que afeta as funções sigmoide e tanh (Goodfellow; Bengio; Courville, 2016).
- *Esparsidade da Ativação:* Como a ReLU resulta em zero para todas as entradas negativas, ela pode levar a ativações esparsas na rede (nem todos os neurônios disparam simultaneamente). Essa esparsidade pode ser computacionalmente eficiente e ajudar na representação de características (Goodfellow; Bengio; Courville, 2016).

Desvantagens:

- *Problema do "Neurônio Morto"(Dying ReLU):* Se um neurônio ReLU recebe uma entrada que o faz ter uma saída zero, e se o gradiente que passa por ele também é zero (o que acontece para $z \leq 0$), esse neurônio pode parar de aprender completamente. Isso ocorre se um grande gradiente fluindo por ele durante o *backpropagation* causar uma atualização nos pesos que leve o neurônio a nunca mais ser ativado por qualquer ponto de dados do conjunto de treinamento (Goodfellow; Bengio; Courville, 2016). Variações como Leaky ReLU, Parametric ReLU (PReLU) ou Exponential Linear Unit (ELU) foram propostas para mitigar esse problema (Goodfellow; Bengio; Courville, 2016).
- *Não centrada em zero:* Assim como a sigmoide, suas ativações não são centradas em zero.

Apesar das desvantagens, a ReLU e suas variantes são frequentemente a escolha padrão para camadas ocultas em muitas arquiteturas de redes neurais profundas devido ao seu bom desempenho empírico e eficiência (Srivastava et al., 2014; Ioffe; Szegedy, 2015).

- **Função Softmax**: Geralmente utilizada na camada de saída de redes neurais para problemas de classificação multiclasse, a função Softmax transforma um vetor z de K valores reais em um vetor de K probabilidades que somam 1. Para o i-ésimo elemento do vetor de entrada z é definida pela Equação 6:

$$f(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{6}$$

Vantagens: A saída representa uma distribuição de probabilidade sobre K classes distintas, o que é diretamente interpretável para tarefas de classificação (Goodfellow; Bengio; Courville, 2016).

4.2.3 Arquitetura de Redes Neurais

Uma RNA é tipicamente estruturada em camadas de neurônios. As arquiteturas mais comuns, como a Perceptron Multicamadas (MLP) utilizada neste trabalho, consistem em três tipos principais de camadas (Haykin, 2001):

- **Camada de Entrada (*Input Layer*)**: Esta camada não realiza cálculos; seus neurônios simplesmente recebem os dados brutos de entrada (as *features* do problema) e os transmitem para a primeira camada oculta. O número de neurônios na camada de entrada corresponde ao número de variáveis de entrada do conjunto de dados (Haykin, 2001).

- **Camadas Ocultas (*Hidden Layers*)**: São as camadas intermediárias entre a camada de entrada e a camada de saída. É aqui que a maior parte do processamento e da extração de características complexas ocorre. Os neurônios nas camadas ocultas aplicam as transformações lineares ponderadas seguidas pelas funções de ativação não lineares (Haykin, 2001). Uma RNA pode ter uma ou múltiplas camadas ocultas; redes com muitas camadas ocultas são frequentemente chamadas de redes neurais profundas (*deep neural networks*) (Goodfellow; Bengio; Courville, 2016). A escolha do número de camadas ocultas e do número de neurônios em cada uma delas é uma decisão de projeto crucial que afeta a capacidade de aprendizado da rede (Goodfellow; Bengio; Courville, 2016; Haykin, 2001).

- **Camada de Saída (*Output Layer*)**: Esta camada produz o resultado final da rede, que pode ser uma predição de valor contínuo (regressão), uma classificação de classe (binária ou multiclasse) ou outra forma de saída, dependendo da tarefa. O número de neurônios na camada de saída e a escolha da função de ativação para esses neurônios dependem diretamente do problema que a rede se propõe a resolver (Haykin, 2001). Por exemplo, para regressão, um único neurônio com ativação linear pode ser usado; para classificação binária, um neurônio com ativação sigmoide; e para classificação multiclasse, K neurônios com ativação Softmax (Goodfellow; Bengio; Courville, 2016).

Em uma MLP, a informação flui em uma única direção, da camada de entrada, através das camadas ocultas, até a camada de saída – um processo conhecido como propagação para frente (*forward propagation*) (Haykin, 2001). Considerando uma notação onde $a^{(l)}$ representa o vetor de ativações da camada l, $W^{(l)}$ a matriz de pesos conectando a camada $l-1$ à camada l, e $b^{(l)}$ o vetor de vieses da camada l, o cálculo da ativação da camada l a partir da camada anterior $a^{(l-1)}$ ocorre em duas etapas (Goodfellow; Bengio; Courville, 2016).

Primeiro, calcula-se a entrada líquida (ou potencial de ativação) $z^{(l)}$, que é a combinação linear ponderada das ativações da camada anterior, conforme a Equação 7:

$$z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)} \tag{7}$$

Em seguida, a função de ativação $f^{(l)}$ é aplicada sobre $z^{(l)}$ para produzir a ativação (saída) $a^{(l)}$ da camada, como mostra a Equação 8:

$$a^{(l)} = f^{(l)}(z^{(l)}) \tag{8}$$

onde $f^{(l)}$ é a função de ativação da camada l, e $a^{(0)}$ é o próprio vetor de entrada (Goodfellow; Bengio; Courville, 2016). A saída final da rede, $a^{(L)}$ (onde L é o índice da última camada, a de saída), é obtida aplicando essas transformações sequencialmente através de todas as camadas (Goodfellow; Bengio; Courville, 2016; Haykin, 2001).

4.2.4 O Algoritmo Backpropagation

O treinamento de uma rede neural consiste em ajustar seus pesos (W) e vieses (b) de forma iterativa para que ela aprenda a mapear corretamente as entradas às saídas desejadas. Esse aprendizado é geralmente supervisionado, o que significa que a rede é alimentada com um conjunto de dados de treinamento contendo pares de entrada-saída conhecidos (Haykin, 2001). O objetivo é minimizar uma *função de custo* (ou função de perda, *loss function*), E, que quantifica a discrepância entre as previsões da rede ($\hat{y}$) e os valores reais (y) para o conjunto de treinamento (Goodfellow; Bengio; Courville, 2016).

O algoritmo de *backpropagation* (retropropagação do erro) é o método mais comum e eficiente para calcular os gradientes da função de custo em relação a cada peso e viés da rede (Haykin, 2001; Goodfellow; Bengio; Courville, 2016). Esses gradientes indicam como cada parâmetro deve ser ajustado para reduzir o erro. O algoritmo é fundamentalmente uma aplicação da regra da cadeia da derivação para calcular esses gradientes de forma eficiente, camada por camada, da saída para a entrada (Haykin, 2001; Goodfellow; Bengio; Courville, 2016; Sanderson, 2017).

O processo de treinamento utilizando *backpropagation* geralmente envolve os seguintes passos, repetidos por várias épocas (iterações sobre todo o conjunto de treinamento) (Goodfellow; Bengio; Courville, 2016; Sanderson, 2017):

1. **Inicialização dos Parâmetros**: Os pesos da rede são inicializados com valores pequenos, geralmente aleatórios, e os vieses podem ser inicializados valores pequenos como 1 ou -1 (Goodfellow; Bengio; Courville, 2016).

2. **Propagação para Frente (*Forward Propagation*)**: Para uma amostra de treinamento (ou um lote de amostras), os dados de entrada são propagados através da rede, da camada de entrada até a camada de saída, utilizando as equações de ativação (7) para calcular as ativações de cada camada e, finalmente, a previsão da rede $\hat{y}$ (Haykin, 2001).

3. **Cálculo do Erro (Função de Custo)**: O erro entre a saída prevista $\hat{y}$ e o valor alvo real y é calculado usando uma função de custo apropriada. Para problemas de regressão, como a

previsão de carga, o Erro Quadrático Médio (MSE) descrito na Equação 9 é uma escolha comum:

$$E = \frac{1}{N}\sum_{k=1}^{N}(y_k - \hat{y}_k)^2 \tag{9}$$

onde N é o número de amostras no lote (ou no conjunto de dados) e k indexa as amostras.

4. **Propagação para Trás (*Backward Propagation*)**: Esta é a etapa central do algoritmo. O gradiente da função de custo E em relação aos parâmetros (pesos e vieses) de cada camada é calculado. Isso começa com o cálculo do gradiente do erro em relação às ativações da camada de saída. Usando a regra da cadeia, esses gradientes são então propagados para trás, camada por camada, permitindo o cálculo dos gradientes do erro em relação às ativações das camadas ocultas, e subsequentemente, em relação aos pesos e vieses de cada camada. Essencialmente, calcula-se $\frac{\partial E}{\partial w_{ij}^{(l)}}$ e $\frac{\partial E}{\partial b_j^{(l)}}$ para todos os pesos e vieses da rede (Haykin, 2001; Goodfellow; Bengio; Courville, 2016).

5. **Atualização dos Pesos e Vieses (Gradiente Descendente)**: Os pesos e vieses da rede são atualizados na direção oposta ao seu respectivo gradiente, com o objetivo de minimizar a função de custo. A regra de atualização mais básica é a do gradiente descendente, expressa na Equação 10:

$$w^{(t+1)} = w^{(t)} - \eta\frac{\partial E}{\partial w^{(t)}} \tag{10}$$

onde:

- $w^{(t+1)}$ representa o valor do peso (ou um vetor de pesos) na próxima iteração de treinamento (o peso atualizado).
- $w^{(t)}$ é o valor atual do peso antes da atualização (o peso na iteração corrente t).
- η (eta) é a *taxa de aprendizado* (*learning rate*), um hiperparâmetro escalar positivo que controla o tamanho do passo dado na direção do gradiente negativo. Uma taxa de aprendizado muito pequena pode levar a uma convergência lenta, enquanto uma taxa muito grande pode fazer com que o algoritmo oscile ou divirja (Haykin, 2001; Goodfellow; Bengio; Courville, 2016).
- $\frac{\partial E}{\partial w^{(t)}}$ é o gradiente da função de custo E em relação ao peso w na iteração t. Este termo, conhecido como "gradiente do peso local"em algumas terminologias, indica a sensibilidade da função de erro a uma pequena mudança no peso w; ou seja, a direção em que o erro aumenta mais rapidamente. Ao subtrair este gradiente (multiplicado pela taxa de aprendizado), o peso é ajustado na direção que mais reduz o erro (Haykin, 2001).

Uma regra análoga é aplicada para atualizar os vieses. Existem muitas variantes do gradiente descendente, como o gradiente descendente estocástico (SGD), mini-batch gradient descent, e algoritmos mais sofisticados como Adam, RMSprop, que adaptam

a taxa de aprendizado ou utilizam momento para acelerar a convergência e melhorar o desempenho (Goodfellow; Bengio; Courville, 2016).

Este ciclo de propagação para frente, cálculo do erro, propagação para trás e atualização dos parâmetros é repetido por muitas épocas, até que a função de custo convirja para um valor mínimo aceitável ou outro critério de parada seja atingido. A capacidade do *backpropagation*, juntamente com o gradiente descendente, de treinar eficientemente redes com múltiplas camadas é uma das principais razões para o sucesso das redes neurais em uma vasta gama de aplicações (Sanderson, 2017; Goodfellow; Bengio; Courville, 2016).

5 METODOLOGIA

5.1 ORIGEM DOS DADOS

Os perfis de carga residencial e de carregamento de veículos elétricos utilizados neste trabalho foram baseados no artigo publicado na revista *Nature* por Muratori (2018), intitulado "Impact of uncoordinated plug-in electric vehicle charging on residential power demand" (Muratori, 2018). Neste estudo, o autor propôs um modelo para avaliar o impacto do carregamento não coordenado de veículos elétricos plug-in (VEPs) na demanda residencial, utilizando dados estatísticos e estocásticos para simular cenários realistas de consumo energético (Muratori, 2017).

A geração dos perfis de consumo foi fundamentada em dados da pesquisa *Residential Energy Consumption Survey* (RECS), conduzida pela *U.S. Energy Information Administration* (EIA) (U.S. Energy Information Administration, 2013), e em informações disponibilizadas pelo *National Renewable Energy Laboratory* (NREL) (Muratori, 2017). O conjunto de dados original está disponível publicamente no repositório do NREL[1] e foi utilizado em pesquisas recentes para simulação de perfis de consumo residencial e de carregamento de VEs.

5.1.1 Características Principais e limitações

O banco de dados empregado compreende perfis de consumo de energia elétrica de 200 residências localizadas na região Centro-Oeste dos Estados Unidos, abrangendo um período de um ano completo, com resolução temporal de 10 minutos. A composição das residências inclui 502 indivíduos e 348 veículos de passeio, números extraídos de amostras representativas da pesquisa RECS de 2009 (U.S. Energy Information Administration, 2013). Os perfis de consumo residencial refletem padrões reais de uso de eletricidade, validados por dados medidos, e são expressos em termos de potência elétrica (Watts) (Muratori, 2017).

Os perfis de carregamento dos veículos elétricos foram gerados a partir de modelos estocásticos baseados em cadeias de Markov, utilizando dados do *Bureau of Labor Statistics* dos EUA sobre padrões de uso de veículos e hábitos de consumo energético. Esses modelos permitiram a simulação de cenários de carregamento descoordenado, nos quais os horários de recarga dos VEs não são restritos a períodos específicos, podendo ocorrer tanto dentro quanto fora dos horários de pico (Muratori et al., 2013).

É importante ressaltar que os dados utilizados são sintéticos, gerados por meio de simulações estatísticas e estocásticas, ainda que fundamentados em dados reais e validados por medições. Uma das principais limitações refere-se à representatividade dos perfis para a realidade brasileira, uma vez que os dados originais correspondem ao contexto norte-americano, tanto em termos de hábitos de consumo quanto de infraestrutura residencial e de transporte (Muratori et al., 2013).

[1] https://data.nrel.gov/submissions/69

Além disso, o modelo assume hipóteses específicas quanto ao comportamento de carregamento dos VEs, como a possibilidade de recarga em qualquer horário do dia, refletindo um cenário de carregamento descoordenado. A penetração de VEs e a composição das residências também seguem as proporções observadas na amostra da RECS de 2009, o que pode não corresponder à realidade de outros países, períodos ou regiões (Muratori, 2017).

A utilização deste banco de dados, no entanto, é justificada pela sua aceitação na literatura internacional como uma modelagem confiável e pela ausência de conjuntos de dados equivalentes para o contexto brasileiro. Trabalhos anteriores, como o de Formaggio (2021), também utilizaram abordagens semelhantes para a avaliação do impacto do carregamento de VEs na demanda residencial, reforçando a relevância e a aplicabilidade da metodologia adotada neste estudo (Formaggio, 2021).

5.2 PROCESSAMENTO E ENGENHARIA DE DADOS

O processamento dos dados foi realizado integralmente em Python, utilizando um conjunto de bibliotecas amplamente reconhecidas e consolidadas na comunidade científica. O **Pandas** (McKinney, 2010; PandasLib, 2024) foi empregado para manipulação, limpeza e análise de dados tabulares, oferecendo estruturas de dados eficientes e flexíveis para operações como filtragem, agregação e transformação. O **NumPy** (Harris et al., 2020) foi utilizado para operações matemáticas de baixo nível e manipulação eficiente de arrays multidimensionais, sendo a base para diversas operações vetorizadas e cálculos numéricos de alto desempenho.

Para o pré-processamento, normalização e divisão dos conjuntos de dados, utilizou-se o **scikit-learn** (Pedregosa et al., 2018; learnLib, 2024), que fornece uma ampla gama de ferramentas para aprendizado de máquina, incluindo escalonamento de variáveis, seleção de atributos e particionamento dos dados em conjuntos de treino, validação e teste. O treinamento e avaliação das Redes Neurais Artificiais (RNAs) foi realizado com **TensorFlow** (Abadi et al., 2016; TensorFlowLib, 2024) e sua interface de alto nível **Keras** (Keras Ref, 2024), devido à sua robustez, flexibilidade e suporte a técnicas modernas de aprendizado profundo, como regularização, callbacks e métricas customizadas.

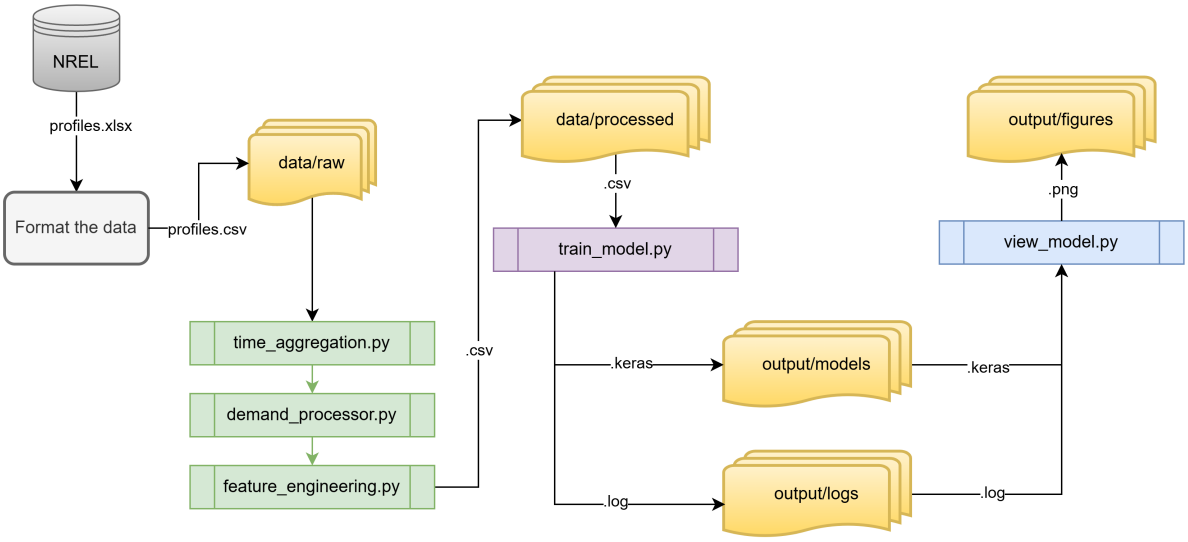
A visualização gráfica dos dados e dos resultados foi realizada com as bibliotecas **Matplotlib** e **Seaborn**, que permitem a criação de gráficos estatísticos de alta qualidade, facilitando a análise exploratória e a comunicação dos achados. A escolha dessas ferramentas se justifica por sua robustez, integração e ampla aceitação em projetos de ciência de dados e aprendizado de máquina, além de sua documentação abrangente e comunidade ativa de desenvolvimento (Hunter, 2007; Waskom, 2021).

5.2.1 Pipeline de Scripts

O pipeline de processamento foi estruturado em etapas sequenciais, cada uma implementada em um script Python dedicado, garantindo modularidade, clareza e facilidade de manutenção do

código. Esse fluxo de trabalho, representado no fluxograma da Figura 5, abrange desde a ingestão dos dados brutos até a disponibilização dos conjuntos de dados prontos para o treinamento e avaliação do modelo de Rede Neural Artificial (RNA). Cada etapa do pipeline foi projetada para tratar aspectos específicos do problema, promovendo a qualidade dos dados e a robustez do processo de modelagem.

Figura 5 – Fluxograma do pipeline de processamento e engenharia de dados.



Fonte: Elaborado pelo autor.

Inicialmente, a etapa de **agregação temporal** (`time_aggregation.py`) realiza a conversão dos perfis de consumo e carregamento de veículos elétricos, originalmente registrados em intervalos de 10 minutos, para uma resolução horária. Essa transformação é fundamental para alinhar a granularidade dos dados ao objetivo de previsão horária, além de reduzir a dimensionalidade (Brownlee, 2018).

Input:

- `Residential-Profiles.csv`
- `PEV-Profiles-L1.csv`
- `PEV-Profiles-L2.csv`

Output:

- `Residential-Profiles-By-Hours.csv`
- `PEV-Profiles-L1-By-Hours.csv`
- `PEV-Profiles-L2-By-Hours.csv`

Em seguida, o **processamento da demanda** (`demand_processor.py`) consolida as informações de consumo residencial e de veículos elétricos, calculando as demandas totais para

cada cenário analisado (apenas residencial, residencial + L1, etc.). Nessa etapa, também são extraídas variáveis temporais relevantes, como data, hora e indicadores de fim de semana, que enriquecem o conjunto de dados para a modelagem.

Input:

- Arquivos agregados por hora (.csv)

Output:

- `Total-Demand.csv`

A etapa de **engenharia de features** (`feature_engineering.py`) é responsável pela criação de variáveis adicionais que capturam padrões sazonais, comportamentais e operacionais, tais como período do dia, estação do ano, número de veículos plugados, entre outras. Essas features são essenciais para aumentar o poder preditivo do modelo, permitindo que a RNA aprenda relações mais complexas entre as variáveis de entrada e a demanda futura (Brownlee, 2018).

Input:

- `Total-Demand.csv`

Output:

- `Total-Demand-Features.csv`

Posteriormente, a **preparação para o modelo** (`train_model.py`) envolve a normalização dos dados, garantindo que todas as variáveis estejam na mesma escala, e a divisão dos dados em conjuntos de treino, validação e teste, de modo a possibilitar a avaliação imparcial do desempenho do modelo e evitar vazamento de informações.

Input:

- `Total-Demand-Features.csv`

Output:

- Modelos treinados (.keras, .csv, .npy)
- Arquivos de histórico (.csv, .log)
- Métricas (.txt)

Por fim, a etapa de **visualização e análise** (`view_model.py`) compreende a geração de gráficos e relatórios que permitem a interpretação dos resultados do modelo, incluindo curvas de aprendizado, gráficos de dispersão, análises de sensibilidade e comparações entre cenários. Essa etapa é fundamental para a validação dos experimentos e para a comunicação dos achados de forma clara e objetiva.

Input:

- Modelos treinados (.keras, .csv, .npy)
- Arquivos de histórico (.csv, .log)

Output:

- Gráficos (.png)
- Relatórios de avaliação (.txt, .csv)

Esse pipeline modular e sequencial assegura a rastreabilidade de cada transformação aplicada aos dados, facilita a reprodutibilidade dos experimentos e permite ajustes pontuais em cada etapa sem comprometer o fluxo global do projeto.

5.2.2 Reprodutibilidade dos Experimentos

A reprodutibilidade dos experimentos foi assegurada por meio de um conjunto de boas práticas. O versionamento de todos os scripts utilizados no pipeline — incluindo ***time_aggregation.py, demand_processor.py, feature_engineering.py, train_model.py e view_model.py*** — foi realizado em um repositório Git, o que permite o rastreamento detalhado de alterações, facilita a colaboração entre pesquisadores e garante a integridade do histórico de desenvolvimento do projeto e estão disponibilizados publicamente no meu repositório no GitHub[2].

Além disso, o ambiente computacional foi documentado por meio de arquivos descritivos `README.md` e do arquivo `requirements.txt`, no qual constam as versões exatas das bibliotecas e dependências utilizadas. Essa prática possibilita a replicação fiel do ambiente de execução em diferentes máquinas ou plataformas, minimizando problemas de incompatibilidade e assegurando que os resultados obtidos possam ser reproduzidos por outros pesquisadores.

5.2.3 Tratamento de Dados

A base de dados utilizada, proveniente do NREL, não apresenta valores faltantes, o que elimina a necessidade de imputação ou interpolação. Da mesma forma, não foram identificados outliers extremos que justificassem remoção ou substituição, visto que os dados são sintéticos e validados estatisticamente conforme o artigo de origem (Muratori, 2018). O tratamento concentrou-se, portanto, na agregação temporal, padronização de formatos e verificação de consistência dos registros.

A normalização dos dados foi realizada utilizando o método *MinMaxScaler* da biblioteca scikit-learn, que transforma os valores das variáveis para o intervalo [0, 1]. Esta etapa é recomendada para o treinamento de RNAs, pois garante que todas as variáveis de entrada contribuam de forma equilibrada para o processo de aprendizado, evitando que variáveis com escalas maiores dominem o gradiente durante a retropropagação (Haykin, 2001).

[2] https://github.com/arigideon/nre-data-analysis

O script `feature_engineering.py` foi responsável pela criação de variáveis adicionais, detalhadas a seguir:

- **Estação do ano (`Estacao`)**: Variável categórica numérica, definida conforme as datas astronômicas do hemisfério norte. Os valores possíveis são: 1 = Primavera (21/03 a 20/06), 2 = Verão (21/06 a 20/09), 3 = Outono (21/09 a 20/12), 4 = Inverno (21/12 a 20/03).
- **Período do dia (`Periodo_Dia`)**: Variável categórica numérica, segmentando o dia em quatro faixas: 0 = Madrugada (22h-5h), 1 = Manhã (6h-11h), 2 = Tarde (12h-17h), 3 = Noite (18h-21h).
- **Flag de fim de semana (`Fim_Semana`)**: Variável binária, onde 0 indica dia útil e 1 indica sábado ou domingo.
- **Quantidade de VEs plugados (`VEs_Plugados_L1` e `VEs_Plugados_L2`)**: Variáveis numéricas inteiras, representando o número de veículos elétricos conectados à rede em cada instante, para carregadores de nível 1 (L1) e nível 2 (L2), respectivamente. O valor é obtido pela contagem de perfis de veículos com demanda superior a zero em cada horário.

Essas variáveis foram escolhidas para compreender a sua relevância para a previsão de carga. Por exemplo, o comportamento de consumo varia entre estações do ano, períodos do dia e dias da semana, enquanto a quantidade de VEs plugados pode ser determinante para o perfil de consumo em um determinado periodo.

A divisão dos dados em conjuntos de treino, validação e teste foi realizada de forma sequencial, respeitando a ordem temporal dos registros, para evitar vazamento de informação entre os conjuntos. O script `train_model.py` utilizou a função `train_test_split` do scikit-learn, com 80% dos dados para treino e 20% para teste. Internamente, durante o treinamento, 20% do conjunto de treino foi utilizado para validação, resultando em aproximadamente 64% treino, 16% validação e 20% teste.

5.3 ARQUITETURA E TREINAMENTO DA REDE NEURAL

A arquitetura da rede neural foi projetada e avaliada considerando a natureza do problema de previsão de demanda elétrica horária, que exige a modelagem de relações não-lineares e dependências temporais entre as variáveis de entrada. Foi implementada uma estrutura sequencial do tipo Perceptron Multicamadas (MLP), composta por camadas densamente conectadas (*fully connected*), com uma topologia decrescente no número de neurônios por camada oculta. O processo experimental para definição da arquitetura foi conduzido em duas fases principais.

Inicialmente, para uma primeira exploração da complexidade da rede, **foram avaliadas** três configurações para as camadas ocultas, com um número de neurônios progressivamente menor:

128-64-32, 64-32-16 e 32-16-8. Nestes testes iniciais, a função de ativação prioritariamente investigada para as camadas ocultas foi a ReLU (*Rectified Linear Unit*), devido à sua reconhecida capacidade de modelar não-linearidades e sua eficiência computacional em mitigar o problema do desaparecimento do gradiente (Goodfellow; Bengio; Courville, 2016; Haykin, 2001). Todas as configurações propostas culminarão em uma camada de saída com um único neurônio, utilizando uma função de ativação linear, para realizar a previsão da demanda horária futura.

Posteriormente, com base nos resultados obtidos nesta fase exploratória, **foi executado uma segunda fase de otimização**. Esta etapa **focou** em arquiteturas potencialmente mais compactas – incluindo aquelas que se mostraram promissoras na fase inicial (como a `32-16-8`, por exemplo) e outras configurações ainda menores (tais como `24-12-6`, `16-10-4` e `9-6-2`). Um aspecto central desta segunda fase foi a **investigação comparativa** de diferentes funções de ativação para as camadas ocultas, notadamente a ReLU e a Sigmoid, visando identificar a combinação que ofereça o melhor equilíbrio entre precisão preditiva, simplicidade do modelo e capacidade de generalização. Os detalhes e resultados desta abordagem experimental serão apresentados no Capítulo de resultados.

A estrutura geral da rede pode ser descrita da seguinte forma:

- **Camada de entrada**: Número de neurônios igual ao número de variáveis de entrada (features), variando conforme o cenário (Re, Re+L1, etc).
- **Camadas ocultas**: Três camadas densas, com 128-64-32, 64-32-16 ou 32-16-8 neurônios, cada uma seguida por BatchNormalization, Dropout (taxa de 0,1 a 0,3) e taxa de aprendizado de 0,001.
- **Camada de saída**: 1 neurônio, ativação linear, responsável pela previsão da demanda horária da próxima hora.

5.3.1 Técnincas utilizadas no treinamento

Para estudar o comportamento dessas técnicas de regularização a fim de melhorar a generalização e prevenir overfitting, foram empregadas técnicas como BatchNormalization e Dropout após as principais camadas ocultas. O **Dropout** consiste em desligar aleatoriamente uma fração dos neurônios durante o treinamento (tipicamente 0,2 ou 0,3), forçando a rede a não depender excessivamente de caminhos específicos e promovendo a robustez do modelo (Srivastava et al., 2014). O **BatchNormalization** normaliza as ativações de cada camada para média zero e variância unitária, acelerando o treinamento e reduzindo a sensibilidade à inicialização dos pesos (Ioffe; Szegedy, 2015). Ambas as técnicas são reconhecidas por sua eficácia em evitar o sobreajuste (overfitting) e melhorar a estabilidade do treinamento.

Durante o treinamento, foram utilizados também os callbacks **EarlyStopping** e **ModelCheckpoint**:

- **EarlyStopping**: Interrompe o treinamento caso a perda de validação não apresente melhora após 10 épocas consecutivas, evitando o sobreajuste e economizando recursos computacionais.
- **ModelCheckpoint**: Salva automaticamente o modelo com melhor desempenho na validação, garantindo que o modelo final corresponda ao melhor ponto do treinamento.

Essas estratégias são recomendadas para garantir a robustez do modelo e evitar o ajuste excessivo aos dados de treino (Goodfellow; Bengio; Courville, 2016).

5.3.2 Processo de Treinamento

O treinamento da rede neural foi realizado utilizando o algoritmo de (*BP*), que ajusta iterativamente os pesos das conexões da rede de modo a minimizar a função de erro (MSE) entre a previsão da demanda e o valor real observado. O treinamento foi realizado com até 100 épocas, batch size de 32, taxa de aprendizado de 0,001 e validação em 20% do conjunto de treino. O alvo da rede (*target*) é a demanda horária prevista para a próxima hora (*hora* + 1), ou seja, a rede aprende a prever a carga futura com base nas informações atuais (Kim et al., 2024; Scikit-learn developers, 2024).

5.4 MÉTRICAS DE AVALIAÇÃO E INTERPRETAÇÃO DOS RESULTADOS

A avaliação do desempenho do modelo de previsão de carga foi conduzida por meio de um conjunto de métricas estatísticas, com ênfase no **MAPE (Mean Absolute Percentage Error)** como principal indicador da qualidade das previsões. Embora o MAPE seja o foco central da análise, as demais métricas (MSE, RMSE, MAE e R^2) foram utilizadas em conjunto para fornecer uma perspectiva abrangente e robusta do desempenho do modelo (Kim et al., 2024; Scikit-learn developers, 2024).

O MAPE, definido pela Equação 11, mede o erro percentual absoluto médio entre os valores previstos e os valores reais, permitindo avaliar o desempenho relativo do modelo independentemente da escala dos dados. Sua interpretabilidade e ampla aceitação na literatura especializada o tornam uma métrica importante para comparar modelos em diferentes contextos e para comunicar os resultados de forma clara e intuitiva (Kim et al., 2024).

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100 \tag{11}$$

onde y_i representa o valor real da carga elétrica, $\hat{y}_i$ o valor calculado pela rede neural, e N o número de amostras. Valores de MAPE inferiores a 10% são geralmente considerados excelentes para problemas de previsão temporal, enquanto valores entre 10% e 20% indicam desempenho aceitável (Proceffa, 2023).

Além do MAPE, foram utilizadas para complementar a análise as métricas MSE, RMSE, MAE e R^2 que serão apresentadas a seguir e suas definições foram baseadas em Kim et al. (2025) (Kim et al., 2024):

- **MSE (Mean Squared Error)**: Mede o erro quadrático médio entre os valores previstos e os valores reais, sendo sensível a grandes desvios e penalizando fortemente previsões distantes do valor observado. Valores menores de MSE indicam melhor desempenho do modelo (Kim et al., 2024; Scikit-learn developers, 2024). A métrica é calculada pela Equação 12:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{12}$$

  onde y_i é o valor real, $\hat{y}_i$ o valor previsto, e N o número total de amostras. É uma das funções de custo e métricas mais utilizadas em problemas de regressão e previsão (Formaggio, 2021; Rashid; Elfouly; Chen, 2024).

- **RMSE (Root Mean Squared Error)**: Corresponde à raiz quadrada do MSE, retornando o erro para a mesma unidade da variável alvo, o que facilita a interpretação. Assim como o MSE, valores mais baixos de RMSE refletem maior precisão nas previsões (Kim et al., 2024). Sua fórmula é dada pela Equação 13:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{13}$$

- **MAE (Mean Absolute Error)**: Representa o erro absoluto médio entre as previsões e os valores reais. Por não elevar o erro ao quadrado, é menos sensível a outliers do que o MSE. Valores reduzidos de MAE indicam que as previsões estão, em média, próximas dos valores reais (Kim et al., 2024; Scikit-learn developers, 2024). É calculada conforme a Equação 14:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{14}$$

- **R^2 (Coeficiente de Determinação)**: Indica a proporção da variância dos dados que é explicada pelo modelo. Valores próximos de 1 sugerem que o modelo explica bem a variabilidade dos dados, enquanto valores negativos ou próximos de zero indicam baixo poder explicativo (Scikit-learn developers, 2024). É definida pela Equação 15:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{15}$$

  onde $\bar{y}$ é o valor médio dos dados reais (y_i).

A interpretação conjunta dessas métricas é fundamental para uma análise crítica do desempenho do modelo. Conforme destacado por Kim et al. (Kim et al., 2024), embora MSE e MAE sejam amplamente utilizados devido à sua simplicidade e intuitividade, a análise de séries temporais requer uma avaliação mais abrangente. MSE, RMSE e MAE fornecem uma avaliação quantitativa do erro absoluto e quadrático, enquanto o MAPE permite uma análise relativa e comparativa entre diferentes cenários, capturando padrões sazonais e periódicos. O coeficiente de determinação R^2, por sua vez, oferece uma medida global da qualidade do ajuste do modelo aos dados observados, útil para examinar tendências de longo prazo. Dessa forma, a análise integrada dessas métricas possibilita identificar não apenas a precisão das previsões, mas também a robustez e a capacidade de generalização do modelo em diferentes condições de operação (Kim et al., 2024).

Além das métricas quantitativas, foram geradas diversas análises gráficas para uma avaliação qualitativa e visual do desempenho do modelo, conforme implementado no script `view_model.py`. Foram analisadas curvas de aprendizado, gráficos de dispersão (real vs previsto) e comparação da qualidade de previsão da demanda, seguindo as recomendações das bibliotecas Scikit-learn e Keras para avaliação e visualização de modelos de aprendizado de máquina. (Scikit-learn developers, 2024; Keras Ref, 2024)

- **Curvas de aprendizado**: Gráficos de evolução das métricas de perda (loss/val_loss), MAE, RMSE e R^2 ao longo das épocas de treinamento, permitindo identificar overfitting, underfitting e a estabilidade do processo de aprendizado.
- **Gráfico de dispersão (real vs previsto)**: Permite visualizar a aderência das previsões em relação aos valores reais, sendo ideal quando os pontos se alinham próximos à reta 45º.
- **Comparação de demandas**: Gráficos comparando a demanda residencial, a demanda combinada (residencial + VEs) e a demanda total prevista, facilitando a análise do impacto dos veículos elétricos no perfil de carga.

6 RESULTADOS

Este capítulo apresenta os resultados obtidos na busca por um modelo de RNA do tipo MLP eficaz para a previsão da demanda de energia elétrica residencial, com a particularidade de incorporar o impacto do carregamento de VEs. O processo experimental foi conduzido de forma iterativa, iniciando com a definição dos dados de entrada, explorando diferentes configurações de rede e culminando na seleção e avaliação detalhada do modelo mais promissor.

A etapa inicial consistiu na caracterização das variáveis de entrada (features) que alimentariam os modelos de RNA. Estas foram cuidadosamente selecionadas para abranger tanto os fatores tradicionalmente influentes no consumo residencial quanto as novas componentes de demanda introduzidas pelo carregamento de VEs. A Tabela 2 detalha o conjunto de features empregado em cada cenário de análise considerado, indicando o número total de variáveis de entrada para a rede neural em cada caso. Para os propósitos de desenvolvimento e comparação inicial de arquiteturas, este estudo concentrou-se no cenário Residencial + L1 (Re+L1), que considera a carga residencial agregada ao carregamento de VEs de nível 1.

Tabela 2 – Relação de features por cenário e total de entradas na RNA

Feature	**Residencial (Re)**	**Re+L1**	**Re+L2**	**Re+L1L2**
Dia	✓	✓	✓	✓
Mês	✓	✓	✓	✓
Hora	✓	✓	✓	✓
Dia da semana	✓	✓	✓	✓
Estação do ano	✓	✓	✓	✓
Período do dia	✓	✓	✓	✓
Fim de semana	✓	✓	✓	✓
Demanda total agregada (kWh)	✓	✓	✓	✓
VEs L1	X	✓	X	✓
VEs L2	X	X	✓	✓
Total de inputs para a RNA	8	9	9	10

Fonte: Elaborado pelo autor.

6.1 ANÁLISE PRELIMINAR E SELEÇÃO DE ARQUITETURAS MLP PROMISSORAS

Com o intuito de identificar uma arquitetura de RNA MLP robusta e eficiente para a previsão de carga no cenário Re+L1, uma investigação inicial foi conduzida avaliando-se redes de diferentes portes: uma arquitetura "grande"(128 neurônios na primeira camada oculta, 64 na segunda e 32 na terceira, denotada como 128-64-32), uma "média"(64-32-16) e uma "pequena"(32-16-8). Todas as arquiteturas nesta fase utilizaram a função de ativação ReLU nas camadas ocultas e

uma função linear na camada de saída, sendo treinadas com o algoritmo backpropagation. Foram também investigados os efeitos de técnicas de regularização como Batch Normalization (BN) e Early Stopping (ES) para mitigar o overfitting e acelerar a convergência.

O desempenho de cada uma dessas arquiteturas sob diferentes configurações de treinamento e para diversos cenários de demanda é consolidado a seguir. A Tabela 3 detalha os resultados para a arquitetura 32-16-8, enquanto as Tabelas 4 e 5 apresentam os resultados para as arquiteturas 64-32-16 e 128-64-32, respectivamente. As métricas de avaliação incluem o MAPE, MAE, RMSE e o R^2, além do número de épocas de treinamento.

Tabela 3 – Resultados da arquitetura 32-16-8

Arq.	Cenário	Parâmetros	LOSS (MSE)	MAE	RMSE	R^2	MAPE (%)	Épocas
32-16-8	RE	Sem BN/ES	0.002851	0.042233	0.053396	0.912302	10.05	100
32-16-8	RE	Sem BN, Com ES	0.003030	0.042263	0.055046	0.906797	8.96	24
32-16-8	RE	Com BN/ES	0.001897	0.031754	0.043552	0.941656	6.37	98
32-16-8	**RE + L1**	**Sem BN/ES**	**0.003293**	**0.047749**	**0.057382**	**0.921070**	**10.56**	**100**
32-16-8	**RE + L1**	**Sem BN, Com ES**	**0.003282**	**0.044516**	**0.057290**	**0.921324**	**8.39**	**34**
32-16-8	**RE + L1**	**Com BN/ES**	**0.001643**	**0.029923**	**0.040534**	**0.960616**	**5.42**	**100**
32-16-8	RE + L2	Com BN/ES	0.002466	0.037549	0.049661	0.940856	8.16	85
32-16-8	RE + L1L2	Com BN/ES	0.002891	0.041266	0.053765	0.937922	8.95	59

Fonte: Elaborado pelo autor.

Tabela 4 – Resultados da arquitetura 64-32-16

Arq.	Cenário	Parâmetros	LOSS (MSE)	MAE	RMSE	R^2	MAPE (%)	Épocas
64-32-16	RE	Com BN/ES	0.001955	0.032666	0.044213	0.939872	6.83	100
64-32-16	**RE + L1**	**Com BN/ES**	**0.001582**	**0.030093**	**0.039772**	**0.962082**	**5.55**	**95**
64-32-16	RE + L2	Com BN/ES	0.002284	0.036382	0.047791	0.945225	8.03	100
64-32-16	RE + L1L2	Com BN/ES	0.002313	0.036744	0.048089	0.950336	7.61	100

Fonte: Elaborado pelo autor.

Tabela 5 – Resultados da arquitetura 128-64-32

Arq.	Cenário	Parâmetros	LOSS (MSE)	MAE	RMSE	R^2	MAPE (%)	Épocas
128-64-32	RE	Sem BN/ES	0.003020	0.042310	0.054954	0.907111	9.75	100
128-64-32	RE	Sem BN, Com ES	0.002884	0.041197	0.053703	0.911292	8.52	19
128-64-32	RE	Com BN/ES	0.002160	0.034732	0.046471	0.933576	7.11	65
128-64-32	**RE + L1**	**Sem BN/ES**	**0.002320**	**0.037758**	**0.048166**	**0.944390**	**7.38**	**100**
128-64-32	**RE + L1**	**Sem BN, Com ES**	**0.002472**	**0.039591**	**0.049719**	**0.940745**	**7.80**	**20**
128-64-32	**RE + L1**	**Com BN/ES**	**0.002980**	**0.040772**	**0.054587**	**0.928573**	**7.11**	**100**
128-64-32	RE + L2	Com BN/ES	0.002273	0.036063	0.047677	0.945486	7.78	100
128-64-32	RE + L1L2	Com BN/ES	0.002780	0.043400	0.052728	0.940294	11.01	100

Fonte: Elaborado pelo autor.

Analisando os resultados para o cenário Re+L1, observa-se na Tabela 3 que a arquitetura 32-16-8 com BN/ES alcançou um MAPE de 5,42%, enquanto a arquitetura 64-32-16 (Tabela 4) com BN/ES obteve 5,55%. A arquitetura maior, 128-64-32, como visto na Tabela 5, apresentou um MAPE de 7,11% para este mesmo cenário, indicando que o aumento excessivo da complexidade da rede não necessariamente se traduziu em melhor desempenho, podendo inclusive levar a uma degradação ou maior dificuldade de treinamento. A utilização de Batch Normalization e Early Stopping demonstrou ser benéfica, resultando em geral em menores erros e/ou treinamento mais estável, como pode ser visto ao comparar as linhas "Com BN/ES"e "Sem BN/ES"dentro de cada tabela.

A Tabela 6 e a Figura 6 fornecem uma visão consolidada e comparativa do desempenho das arquiteturas nos diferentes cenários, utilizando o MAPE como métrica principal de referência.

Tabela 6 – Desempenho consolidado inicial: Melhor cenário por arquitetura e melhor arquitetura por cenário (MAPE)

Resultados por arquitetura (foco no melhor MAPE)		
Arquitetura	**Melhor cenário (MAPE)**	**MAPE médio**
32-16-8	RE + L1 (5,42%)	7,22%
64-32-16	RE + L1 (5,55%)	7,00%
128-64-32	RE + L1 (7,11%)	8,25%
Resultados por cenário (foco no melhor MAPE)		
Cenário	**Melhor arquitetura (MAPE)**	**MAPE médio**
RE	32-16-8 (6,37%)	6,77%
RE + L1	32-16-8 (5,42%)	6,02%
RE + L2	128-64-32 (7,78%)	7,99%
RE + L1L2	64-32-16 (7,61%)	9,19%

Fonte: Elaborado pelo autor.

O heatmap na Figura 6 ilustra visualmente que, para o cenário Re+L1, as arquiteturas 32-16-8 e 64-32-16 (ambas com BN/ES) apresentaram os menores MAPEs. Considerando o princípio da parcimônia (preferência por modelos mais simples com desempenho similar), a arquitetura 32-16-8 surge como uma candidata particularmente interessante. Seu desempenho foi competitivo com a rede 64-32-16 no cenário Re+L1, mas com menor complexidade.

Com base nesta análise preliminar, o cenário Re+L1 foi selecionado como o caso de referência para a fase de otimização à seguir. Para este cenário, a matriz de entrada final para a RNA, detalhada na Tabela 7, possui 9 entradas e um total de 8.760 amostras horárias, correspondentes a um ano completo.

Este conjunto de dados foi então dividido em subconjuntos para o treinamento, validação e teste do modelo. A divisão resultou em uma matriz de treinamento com 5.605 amostras ($\approx 64\%$), uma de validação com 1.402 amostras ($\approx 16\%$), e uma de teste com 1.752 amostras ($\approx 20\%$).

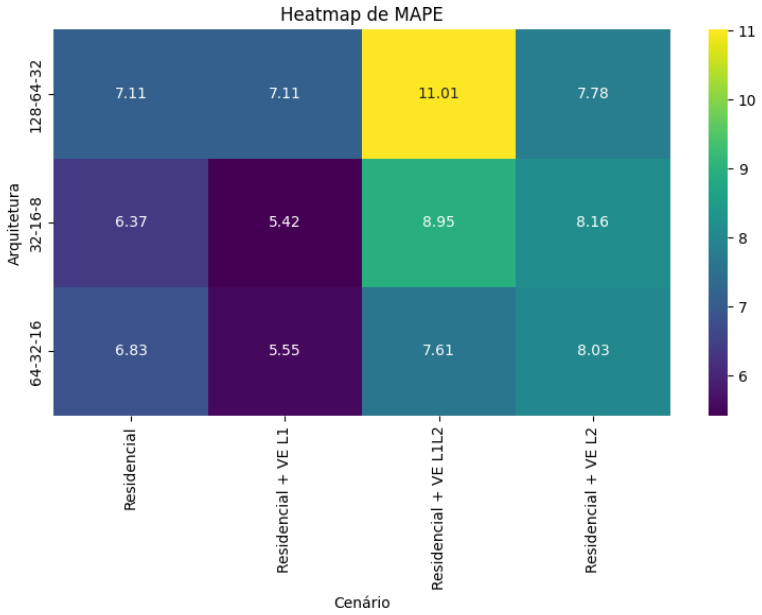
Tabela 7 – Matriz de Entrada final para treinamento da RNA

Dia	Mês	Hora	Dia Sem.	Demanda L1	Estação	Período	Fim Sem.	VEs L1
1	1	0	2	1436.41	4	0	0	1
1	1	1	2	1334.25	4	0	0	0
1	1	2	2	1269.48	4	0	0	0
				⋮				
31	12	21	2	3004.91	4	3	0	3
31	12	22	2	2793.84	4	0	0	3
31	12	23	2	2448.99	4	0	0	2

Fonte: Elaborado pelo autor.

A segunda fase de experimentação, focada em refinar o modelo e explorar o impacto de diferentes funções de ativação e possíveis simplificações na arquitetura, foi seguindo a partir dessa configuração.

Figura 6 – Heatmap do MAPE para as arquiteturas e cenários avaliados com BN/ES



Fonte: Elaborado pelo autor.

6.2 OTIMIZAÇÃO DA ARQUITETURA, AVALIAÇÃO DE FUNÇÕES DE ATIVAÇÃO E IMPACTO DO BATCH NORMALIZATION NO CENÁRIO RE+L1

Com base na análise preliminar, que apontou a arquitetura 32-16-8 com função de ativação ReLU e o uso de *Batch Normalization* (BN) e *Early Stopping* (ES) como uma configuração promissora, esta seção detalha uma fase de experimentação mais aprofundada. O objetivo foi refinar o modelo de RNA para a previsão de carga no cenário Re+L1, investigando:

1. O impacto da técnica de *Batch Normalization*, comparando sistematicamente modelos treinados com e sem sua aplicação.

2. A influência de diferentes funções de ativação nas camadas ocultas, especificamente ReLU versus Sigmoid.

3. O desempenho de arquiteturas mais compactas em relação à 32-16-8, buscando um equilíbrio ótimo entre capacidade preditiva e eficiência computacional. As topologias avaliadas foram: 32-16-8, 24-12-6, 16-10-4 e 9-6-2.

Todos os testes subsequentes foram conduzidos no cenário Re+L1, mantendo-se o uso de *Early Stopping* (ES) para prevenir o *overfitting* e otimizar o tempo de treinamento. Uma função de ativação linear foi utilizada na camada de saída para a tarefa de regressão.

Tabela 8 – Resultados da Otimização: Desempenho das Arquiteturas com Variação de Função de Ativação e Uso de Batch Normalization no Cenário Re+L1 (com ES)

Arquitetura	Função Ativação	Batch Norm.	LOSS (MSE)	MAE	RMSE	R^2	MAPE (%)	Épocas
32-16-8	ReLU	Com BN	0.002171	0.03569	0.046594	0.947960	6.662316	100
32-16-8	ReLU	Sem BN	0.003282	0.044516	0.05729	0.921324	8.388743	34
32-16-8	Sigmoid	Com BN	0.001994	0.03405	0.044651	0.952208	6.307805	54
32-16-8	Sigmoid	Sem BN	0.002001	0.034404	0.044738	0.952023	6.531992	100
24-12-6	ReLU	Com BN	0.001812	0.032319	0.042567	0.956566	5.913319	91
24-12-6	ReLU	Sem BN	0.001666	0.030998	0.040817	0.960064	5.760098	67
24-12-6	Sigmoid	Com BN	0.001942	0.033271	0.044072	0.953440	6.066131	93
24-12-6	Sigmoid	Sem BN	0.002145	0.035335	0.046314	0.948583	6.588826	79
16-10-4	ReLU	Com BN	0.002574	0.037509	0.050735	0.938299	6.367132	100
16-10-4	ReLU	Sem BN	0.001896	0.032604	0.043549	0.954540	5.914040	100
16-10-4	Sigmoid	Com BN	0.002174	0.036479	0.046627	0.947886	7.032152	47
16-10-4	Sigmoid	Sem BN	0.002160	0.036063	0.046477	0.948220	6.960916	100
9-6-2	ReLU	Com BN	0.001715	0.031535	0.041407	0.958901	5.763818	100
9-6-2	ReLU	Sem BN	0.001785	0.031692	0.042253	0.957205	5.747468	100
9-6-2	Sigmoid	Com BN	0.001994	0.034050	0.044651	0.952208	6.307805	55
9-6-2	Sigmoid	Sem BN	0.002012	0.034315	0.044857	0.951767	6.361369	100

Fonte: Elaborado pelo autor.

Para apresentar os resultados desta fase de otimização de forma consolidada, a Tabela 8 contém as métricas de desempenho para cada combinação de arquitetura, função de ativação (ReLU e Sigmoid) e uso ou não de *Batch Normalization*.

6.2.1 Avaliação do Impacto do Batch Normalization (BN)

Para entender a contribuição do Batch Normalization, foi comparado o desempenho dos modelos com e sem BN para cada par de arquitetura e função de ativação. Ou seja, as colunas de diferença (Δ) na Tabela 9 foram calculadas subtraindo o valor da métrica do modelo sem BN do valor obtido pelo modelo com BN ($M_{\text{BN}} - M_{\text{NoBN}}$). Dessa forma, um resultado negativo para as métricas de erro (LOSS, MAE, RMSE, MAPE) significa que o BN reduziu o erro, representando

uma melhora. Para o R^2, um resultado positivo indica que o BN melhorou o ajuste do modelo. A Tabela 9 resume as diferenças observadas nas métricas.

A análise da Tabela 9 revela que o BN demonstrou ser majoritariamente benéfico. Em 5 das 8 configurações testadas, o BN resultou em melhorias (ou seja, redução) nas métricas de erro LOSS, MAE e RMSE, e no aumento do R^2. Para o MAPE, o BN mostrou melhoria em 4 dos 8 casos. Destaca-se a combinação da arquitetura 32-16-8 com ReLU, onde o BN proporcionou uma melhora expressiva em todas as métricas, notadamente uma redução de 1,73 pontos percentuais (p.p.) no MAPE, apesar de ter demandado um número maior de épocas para convergência (100 com BN vs. 34 sem BN). Isso sugere que o BN pode ter auxiliado o modelo a atingir um ponto ótimo de melhor qualidade. Comportamento similar de melhoria com BN foi observado para 24-12-6 Sigmoid, 32-16-8 Sigmoid e 9-6-2 Sigmoid. Por outro lado, para as arquiteturas 16-10-4 ReLU e 24-12-6 ReLU, a ausência de BN levou a resultados superiores. No caso da 24-12-6 ReLU, o modelo sem BN não apenas performou melhor, mas também treinou em menos épocas (67 vs 91).

Tabela 9 – Comparativo do Efeito do Batch Normalization por Arquitetura e Função de Ativação (Cenário Re+L1 com ES)

Arquitetura	**Ativação**	**Δ LOSS**	**Δ MAE**	**Δ RMSE**	**Δ R^2**	**Δ MAPE (%)**	**Épocas BN**	**Épocas NoBN**
16-10-4	ReLU	0.000678	0.004905	0.007186	-0.016241	0.453092	100	100
16-10-4	Sigmoid	1.4e-05	0.000416	0.000150	-0.000334	0.071236	47	100
24-12-6	ReLU	0.000146	0.001321	0.001750	-0.003498	0.153221	91	67
24-12-6	Sigmoid	-0.000203	-0.002064	-0.002242	0.004857	-0.522695	93	79
32-16-8	ReLU	-0.001111	-0.008826	-0.010696	0.026636	-1.726430	100	34
32-16-8	Sigmoid	-7.0e-06	-0.000354	-8.7e-05	0.000185	-0.224187	54	100
9-6-2	ReLU	-7.0e-05	-0.000157	-0.000846	0.001696	0.016350	100	100
9-6-2	Sigmoid	-1.8e-05	-0.000265	-0.000206	0.000441	-0.053564	55	100

Fonte: Elaborado pelo autor.

Quanto ao número de épocas, em média, os modelos com BN treinaram por 80 épocas, enquanto os sem BN por 85 épocas. Contudo, essa média é influenciada por casos específicos. Para a função `Sigmoid`, o BN consistentemente levou a uma convergência mais rápida (menos épocas).

6.2.2 Seleção do Modelo de Melhor Desempenho

Para identificar o modelo com o melhor desempenho entre todas as configurações testadas nesta fase de otimização, foi calculado um score combinado. Este score foi obtido em duas etapas: primeiro, as cinco métricas de avaliação (LOSS, MAE, RMSE, MAPE e R^2) foram normalizadas para uma escala comum de 0 a 1, onde 1 representa o melhor desempenho relativo e 0, o pior. Em seguida, o score combinado de cada modelo foi calculado como a soma dessas cinco métricas normalizadas. Um score mais alto indica, assim, um melhor desempenho geral

entre as arquiteturas testadas, considerando todas as métricas de forma equilibrada. A Tabela 10 apresenta os cinco modelos com as maiores pontuações.

Tabela 10 – Top 5 Modelos por Score Combinado (Cenário Re+L1 com ES)

Arquitetura	Ativação	Batch Norm.	LOSS	MAE	RMSE	R^2	MAPE (%)	Épocas	Score Norm.
24-12-6	ReLU	Sem BN	0.001666	0.030998	0.040817	0.960064	5.76010	67	4.99522
9-6-2	ReLU	Com BN	0.001715	0.031535	0.041407	0.958901	5.76382	100	4.85793
9-6-2	ReLU	Sem BN	0.001785	0.031692	0.042253	0.957205	5.74747	100	4.71405
24-12-6	ReLU	Com BN	0.001812	0.032319	0.042567	0.956566	5.91332	91	4.55261
16-10-4	ReLU	Sem BN	0.001896	0.032604	0.043549	0.954540	5.91404	100	4.36736

Fonte: Elaborado pelo autor.

Conforme a Tabela 10, a arquitetura `24-12-6` (ReLU, Sem BN) alcançou o maior score normalizado (4.99522), apresentando boas métricas individuais entre seus pares, como um MAPE de 5,76010%. Logo em seguida, a arquitetura `9-6-2` (ReLU, Com BN) obteve um score de 4.85793, com um MAPE muito similar de 5,76382%.

Apesar da arquitetura `24-12-6` (ReLU, Sem BN) ter apresentado um score relativamente superior, seguindo **princípio da parcimônia**, optou-se pela seleção da arquitetura **`9-6-2` utilizando a função de ativação `ReLU` e treinada `Com Batch Normalization`** como o modelo final para este estudo. A arquitetura `9-6-2` é significativamente menos complexa (menor número de neurônios e, consequentemente, de parâmetros treináveis) que a `24-12-6`. Dado que a diferença no MAPE entre os dois modelos é de apenas aproximadamente 0,0037 p.p. e o score combinado é muito próximo, a simplicidade da `9-6-2` a torna uma escolha razoável e de menor custo computacional para implantação, sem um sacrifício relevante na performance.

É interessante notar que quatro dos cinco modelos com melhor pontuação utilizaram a função de ativação `ReLU`, e três deles (incluindo o primeiro classificado) não aplicaram `Batch Normalization`. No entanto, o segundo melhor modelo, e o escolhido, `9-6-2 ReLU`, beneficiou-se do uso de `Batch Normalization`.

6.2.3 Análise de Tendências Gerais

6.2.3.1 Desempenho por Função de Ativação

Avaliando o desempenho médio agrupado por função de ativação, conforme a Tabela 11, observa-se que `ReLU` e `Sigmoid` apresentaram performances médias competitivas no contexto dos experimentos realizados.

Em média, `Sigmoid` resultou em LOSS e RMSE ligeiramente inferiores e um R^2 um pouco superior. Contudo, `ReLU` alcançou um MAPE médio inferior. As diferenças médias gerais são pequenas, mas o modelo individual de melhor desempenho empregou `ReLU`. `Sigmoid` tendeu a convergir em menos épocas, em média, especialmente quando combinada com BN.

Tabela 11 – Desempenho Médio Agregado por Função de Ativação (Cenário Re+L1 com ES)

Ativação	LOSS Méd.	MAE Méd.	RMSE Méd.	R^2 Méd.	MAPE Méd. (%)	Épocas Méd.
ReLU	0.002113	0.034608	0.045651	0.949357	6.31462	86.5
Sigmoid	0.002053	0.034746	0.045298	0.950792	6.51962	78.5

Fonte: Elaborado pelo autor.

6.2.3.2 Desempenho por Complexidade da Arquitetura

Na tabela 12 é possível observar que há uma tendência onde as arquiteturas mais compactas (`9-6-2` e `24-12-6`) apresentaram, em média, os melhores resultados em termos de MAPE, LOSS, MAE, RMSE e R^2. A arquitetura `32-16-8`, a mais complexa entre as testadas nesta fase, obteve o desempenho médio mais fraco na maioria das métricas de erro, embora tenha convergido, em média, em menos épocas (resultado influenciado pelo caso `32-16-8 ReLU Sem BN` que interrompeu o treinamento rapidamente com desempenho inferior aos demais da mesma arquitetura). Isso sugere que, para o presente cenário de estudo e volume de dados, arquiteturas excessivamente complexas podem não oferecer vantagens e serem mais suscetíveis a dificuldades de treinamento ou *overfitting*, a menos que uma regularização e ajuste fino muito precisos sejam empregados.

Tabela 12 – Desempenho Médio Agregado por Arquitetura (Cenário Re+L1 com ES, ordenado por MAPE crescente)

Arquitetura	LOSS Méd.	MAE Méd.	RMSE Méd.	R^2 Méd.	MAPE Méd. (%)	Épocas Méd.
9-6-2	0.001877	0.032898	0.043292	0.955020	6.04512	88.75
24-12-6	0.001891	0.032981	0.043443	0.954663	6.08209	82.50
16-10-4	0.002201	0.035664	0.046847	0.947236	6.56856	86.75
32-16-8	0.002362	0.037165	0.048318	0.943379	6.97271	72.00

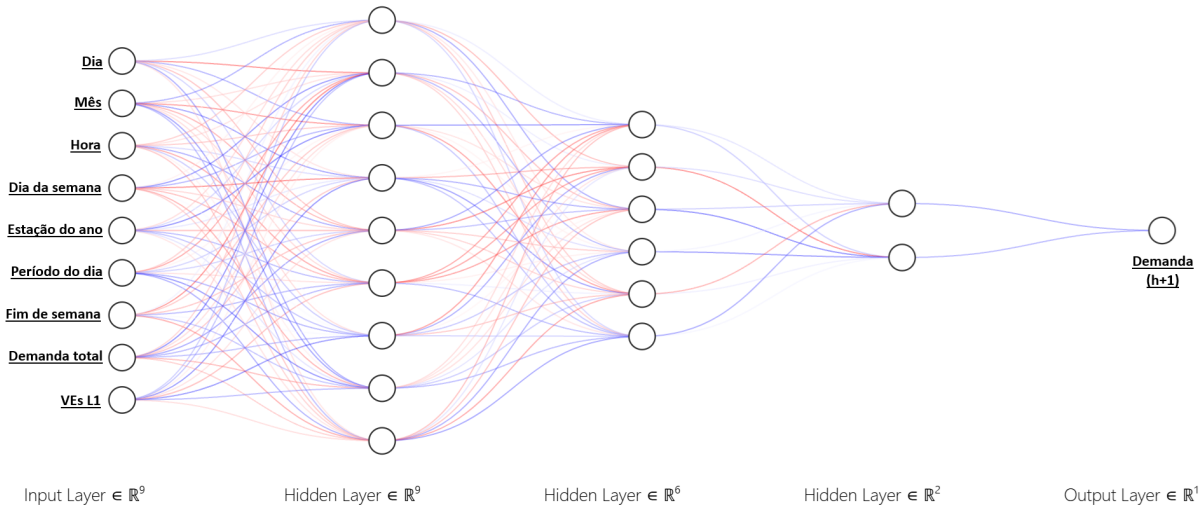
Fonte: Elaborado pelo autor.

Com base nesta extensa fase de otimização e aplicando o princípio da parcimônia, o modelo **MLP com arquitetura 9-6-2, função de ativação ReLU nas camadas ocultas, e treinado Com Batch Normalization (e com Early Stopping)** foi selecionado como o modelo final para análise detalhada subsequente.

6.3 ANÁLISE DETALHADA DO MODELO MLP SELECIONADO (9-6-2 RELU COM BN)

O modelo final selecionado, com arquitetura **9-6-2**, função de ativação **ReLU** nas camadas ocultas e treinado **Com *Batch Normalization*** (além de *Early Stopping*), é representado esquematicamente na Figura 7.

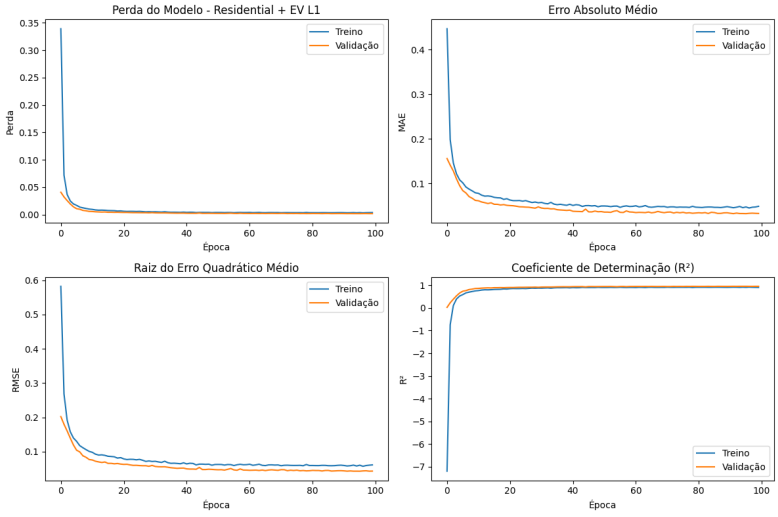
Figura 7 – Diagrama da arquitetura da Rede Neural Artificial MLP 9-6-2 selecionada para o estudo



Fonte: Elaborado pelo autor.

A avaliação do processo de treinamento do modelo `9-6-2 ReLU Com BN` é apresentada através das curvas de aprendizado na Figura 8, que ilustram a evolução das métricas de Loss, MAE, RMSE e R^2 ao longo das épocas para os conjuntos de treinamento e validação.

Figura 8 – Curvas de aprendizado (Loss, MAE, RMSE, R^2) para o modelo 9-6-2 ReLU Com BN, cenário Re+L1, com early stopping.



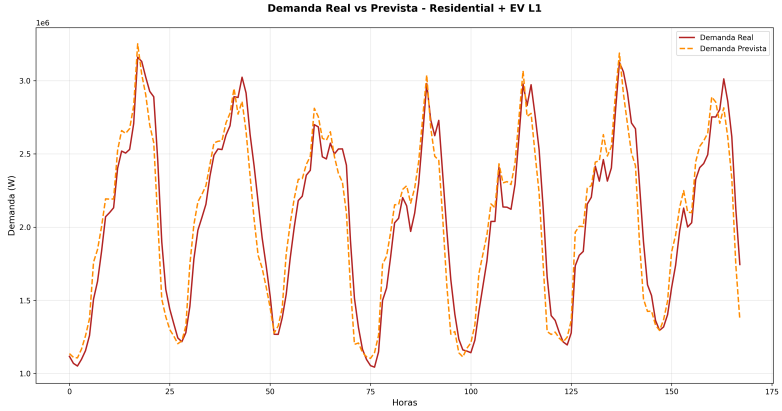
Fonte: Elaborado pelo autor.

As curvas de aprendizado do modelo `9-6-2 ReLU Com BN` (Figura 8) demonstram um treinamento eficaz. As perdas de treinamento e validação convergem suavemente para valores baixos, com suas curvas permanecendo visualmente próximas. O *gap* de generalização entre o *Loss* de treinamento (MSE final 0.001631) e o de validação (MSE final 0.001715) é mínimo

(0.000084), indicando ausência de *overfitting* expressivo e boa capacidade de generalização. As demais métricas (MAE, RMSE, R^2) apresentam comportamento similar, com as curvas de validação acompanhando de perto as de treinamento, o que reforça a robustez do modelo. A estabilidade do aprendizado é confirmada pelo baixo desvio padrão da `val_loss` nas últimas 10 épocas (0.00000911), sugerindo a atingimento de um platô de desempenho estável. As métricas finais (MAPE 5,7638%, R^2 0.9589) corroboram o bom resultado.

A comparação com o modelo `9-6-2 ReLU Sem BN` destaca o benefício do *Batch Normalization*. Embora ambos tenham treinado por 100 épocas, o modelo sem BN apresentou um *gap* de generalização para o *Loss* visivelmente maior (0.000193), perceptível pela maior separação entre suas curvas de treino e validação (analisadas a partir de seu histórico, não exibidas aqui). A estabilidade da `val_loss` também foi ligeiramente inferior no modelo sem BN (desvio padrão de 0.00001182). Apesar de um MAPE final similar (5,7475% para o modelo sem BN), o menor *gap* e a maior estabilidade gráfica do `9-6-2 ReLU Com BN` sustentam sua escolha.

Figura 9 – Comparação entre a demanda real e a demanda prevista pelo modelo 9-6-2 ReLU Com BN para uma semana no cenário Re+L1.

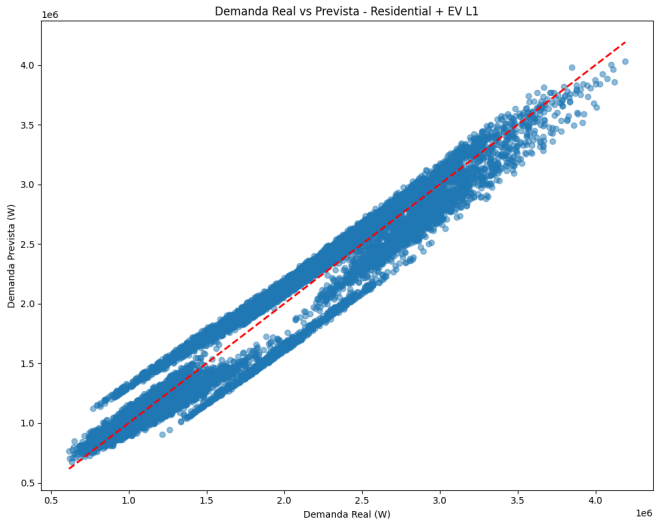


Fonte: Elaborado pelo autor.

O modelo `24-12-6 ReLU Sem BN`, que obteve o maior score (Tabela 10), foi interrompido pelo *Early Stopping* em 67 épocas (MAPE 5,7601%, R^2 0.9600). Seu *gap* de generalização para o *Loss* (0.000158) foi intermediário. As curvas de aprendizado (Apêndice A) deste modelo mais complexo revelam na Figura 12 leve instabilidade nas métricas de MAE, RMSE e R^2, onde as curvas de validação ocasionalmente cruzam as de treinamento. Essa interação sugere uma variabilidade maior no processo de validação para esta arquitetura sem BN, contrastando com o paralelismo mais consistente das curvas do `9-6-2 ReLU Com BN`. O desvio padrão da `val_loss` do `24-12-6 Sem BN` (0.00002516) também foi superior, indicando menor estabilidade ao final do treinamento.

Para visualizar o desempenho preditivo do modelo ao longo de um período contínuo, a Figura 9 compara a demanda real de energia com os valores previstos pelo modelo para uma semana (168 horas) do conjunto de teste. A figura demonstra a capacidade do modelo em acompanhar as variações diárias e semanais da demanda de carga. As previsões seguem de perto

Figura 10 – Dispersão dos valores reais vs. previstos para o modelo 9-6-2 ReLU Com BN, cenário Re+L1.

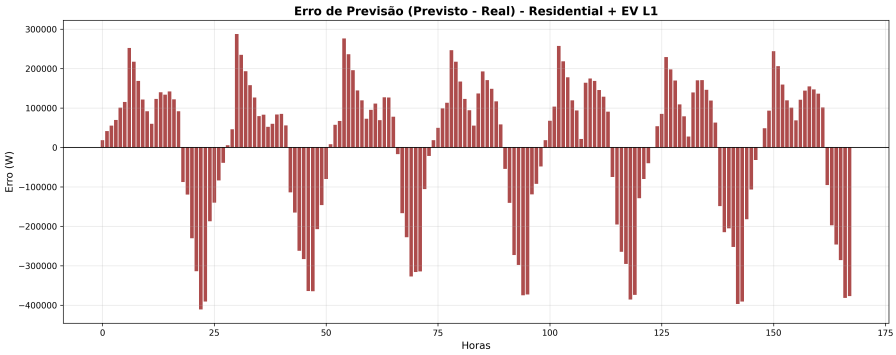


Fonte: Elaborado pelo autor.

os picos e vales da demanda real, capturando a dinâmica do consumo residencial acrescido do carregamento de veículos elétricos. Pequenas discrepâncias são observadas, especialmente em transições rápidas de carga, mas o perfil geral é bem representado. A distribuição e magnitude dos erros de previsão podem ser analisadas na Figura 11, que plota o erro (Real - Previsto) para o mesmo período de uma semana.

A qualidade da previsão do modelo selecionado pode ser visualizada através do gráfico de dispersão na Figura 10, que demonstra uma forte correlação linear positiva entre os valores reais e os previstos, com a maioria dos pontos agrupados próximos à linha diagonal (y=x), que representa a previsão perfeita. A dispersão relativamente baixa dos pontos indica uma boa acurácia do modelo em estimar a demanda de carga no conjunto de teste.

Figura 11 – Erro de previsão (Real - Previsto) do modelo 9-6-2 ReLU Com BN para uma semana no cenário Re+L1.



Fonte: Elaborado pelo autor.

Estes resultados demonstram que a arquitetura `9-6-2` com função de ativação `ReLU` e

uso de *Batch Normalization*, aliada à técnica de *Early Stopping*, proporcionou um modelo consistente, robusto e preciso para a previsão de carga residencial incluindo o carregamento de VEs no cenário Re+L1, alcançando um MAPE de 5,76382%.

7 DISCUSSÃO DOS RESULTADOS

Este capítulo dedica-se à análise e interpretação dos resultados obtidos no desenvolvimento e avaliação dos modelos de Redes Neurais Artificiais (RNA) do tipo Perceptron Multicamadas (MLP) para a previsão da demanda de energia elétrica residencial, com a inclusão do impacto do carregamento de veículos elétricos (VEs). A discussão se concentrará nos principais achados referentes à seleção de arquiteturas, ao impacto das técnicas de regularização e funções de ativação, e ao desempenho do modelo final selecionado.

A primeira constatação relevante vem da análise preliminar de arquiteturas (Seção 6.1), onde se observou que um aumento indiscriminado na complexidade da rede (número de neurônios e camadas) não necessariamente se traduz em um ganho proporcional de desempenho. Arquiteturas mais robustas, como a `128-64-32`, embora capazes de aprender os padrões dos dados, não superaram consistentemente as arquiteturas mais enxutas, como a `32-16-8`, especialmente quando se considera o cenário Re+L1 e a aplicação de técnicas como *Batch Normalization* (BN) e *Early Stopping* (ES). Este achado sugere que, para o conjunto de dados e o problema em questão neste trabalho, modelos excessivamente complexos podem ser mais propensos a dificuldades de treinamento ou a um ajuste excessivo aos dados de treino, mesmo com mecanismos de regularização. A eficácia do BN e do ES, evidenciada pela melhoria nas métricas e/ou estabilidade do treinamento (Tabela 9), reforça a importância dessas técnicas na modelagem com RNAs.

A fase seguinte de otimização (Seção 6.2), focada no cenário Re+L1, permitiu uma investigação mais granular do impacto do *Batch Normalization* e das funções de ativação (ReLU vs. Sigmoid) em arquiteturas ainda mais compactas. A análise comparativa (Tabela 9) revelou que o BN, embora majoritariamente benéfico, não é uma panaceia universal[1]. Sua contribuição positiva foi notável em diversas configurações, como na arquitetura `32-16-8` com ReLU, onde proporcionou uma redução expressiva no MAPE, e na família de modelos com Sigmoid, onde também tendeu a acelerar a convergência. No entanto, em casos específicos, como na arquitetura `24-12-6` com ReLU, o modelo sem BN apresentou um desempenho ligeiramente superior. Isso traz a tona a natureza empírica da otimização de hiperparâmetros em aprendizado profundo, onde a interação entre arquitetura, técnicas de regularização e características dos dados determina a configuração ótima do modelo.

No que tange às funções de ativação, tanto ReLU quanto Sigmoid se mostraram competitivas (Tabela 11). Embora Sigmoid tenha, em média, apresentado LOSS e RMSE ligeiramente melhores e convergido mais rapidamente em alguns casos (especialmente com BN), a função ReLU esteve presente nos modelos com os menores MAPEs individuais, incluindo o modelo final selecionado. A preferência por ReLU em muitas aplicações de aprendizado profundo, devido à sua

[1] Panaceia universal: remédio ou solução que se acredita curar todos os males ou resolver todos os problemas. Fonte: https://pt.wikipedia.org/wiki/Panaceia

simplicidade e capacidade de mitigar o problema do desaparecimento do gradiente (Goodfellow; Bengio; Courville, 2016), parece encontrar respaldo também neste estudo.

A análise criteriosa dos resultados deve ser destacada como norteador crucial na seleção do modelo final. A arquitetura **`9-6-2` com ReLU e *Batch Normalization*** foi escolhida (Seção 6.2) não por deter o maior score absoluto por uma margem significativa, mas por oferecer um excelente equilíbrio entre simplicidade computacional (menor número de parâmetros treináveis) e um desempenho preditivo robusto (MAPE de 5,76382% no cenário Re+L1) em comparação aos modelos pares deste estudo.

A análise detalhada do modelo `9-6-2 ReLU Com BN` (Seção 6.3) confirmou sua eficácia. As curvas de aprendizado (Figura 8) indicaram um bom ajuste, com um *gap* de generalização estável e mínimo entre os conjuntos de treino e validação, sugerindo que o modelo não sofreu de *overfitting* expressivo. A capacidade do modelo em acompanhar as flutuações da demanda real, incluindo os picos e vales característicos (Figura 9), e a forte correlação entre valores reais e previstos (Figura 10) atestam sua qualidade preditiva. A comparação com sua contraparte sem BN e com o modelo `24-12-6 ReLU Sem BN` (que obteve o maior score) reforçou a validade da escolha, destacando a contribuição do BN para a estabilidade e generalização no modelo selecionado.

O desempenho alcançado, com um MAPE na faixa de 5-6% para o cenário Re+L1, pode ser considerado muito promissor no contexto de previsão de carga horária residencial, especialmente ao se levar em conta a complexidade adicional introduzida pela componente estocástica do carregamento de veículos elétricos. A capacidade do modelo em aprender padrões a partir das *features* (como período do dia, estação do ano, a quantidade de VEs plugados e etc) foram relevante para este resultado.

É importante contextualizar que os dados utilizados, embora validados e empregados na literatura internacional, são sintéticos e representam o cenário norte-americano. Portanto, enquanto a metodologia de desenvolvimento e otimização da RNA possui validade geral, a transposição direta dos valores de acurácia para outras realidades, como a brasileira, requereria estudos adicionais com dados locais. Contudo, o rigor metodológico empregado e os insights obtidos sobre a modelagem da demanda com VEs constituem uma contribuição relevante.

8 CONCLUSÃO

O presente Trabalho de Conclusão de Curso dedicou-se ao desenvolvimento e avaliação de um modelo de Rede Neural Artificial (RNA) do tipo Perceptron Multicamadas (MLP) para a previsão de carga em unidades residenciais, com a particularidade de incorporar o impacto decorrente do carregamento de veículos elétricos (VEs). Diante da crescente eletrificação da mobilidade e dos desafios impostos às redes de distribuição de energia, o objetivo central foi investigar a eficácia da RNA MLP, treinada com o algoritmo *backpropagation* e implementada por meio de bibliotecas da linguagem Python, como ferramenta útil para esta complexa tarefa de previsão.

Ao longo deste estudo, diversas arquiteturas de rede, funções de ativação e técnicas de regularização como o *Batch Normalization* foram sistematicamente avaliadas, focando-se no cenário de demanda residencial acrescida do carregamento de VEs de Nível 1 (Re+L1). O processo experimental, conduzido em fases exploratórias e de otimização, culminou na seleção da arquitetura MLP `9-9-6-2-1` (nove entradas, três camadas ocultas com nove, seis e dois neurônios e uma saída) com função de ativação ReLU e aplicação de *Batch Normalization*. A escolha deste modelo, considerando o princípio da parcimônia, demonstrou que uma rede relativamente simples pode alcançar um desempenho notável, obtendo um Erro Percentual Absoluto Médio (MAPE) de 5,76% e um Coeficiente de Determinação (R^2) de 0.96 para o conjunto de teste. Estes valores indicam uma boa precisão e capacidade de generalização do modelo, especialmente ao considerar a natureza estocástica do consumo residencial e do carregamento de VEs, que torna a tarefa mais complexa.

A análise detalhada das curvas de aprendizado e das métricas de desempenho confirmou a confiabilidade do modelo selecionado, com um *gap* de generalização mínimo entre os dados de treinamento e validação e um comportamento de treinamento estável ao olharmos para as épocas de aprendizado. Constatou-se também a relevância do *Batch Normalization* para aprimorar a estabilidade e a capacidade de generalização da arquitetura escolhida, bem como a eficácia da função de ativação ReLU. A investigação sobre a complexidade da arquitetura revelou que modelos excessivamente complexos não necessariamente resultam em melhor desempenho para o conjunto de dados utilizado, reforçando a importância de um dimensionamento adequado da rede e um bom olhar crítico a todo o contexto do problema.

As contribuições deste trabalho estão na demonstração prática da aplicação bem-sucedida de uma RNA MLP para a previsão de carga com inclusão de VEs, fornecendo um modelo com bom desempenho e complexidade reduzida. A metodologia empregada, desde a seleção de *features* até a otimização e escolha do modelo, oferece um caminho replicável para estudos futuros.

Reconhece-se, contudo, as limitações inerentes ao estudo, como a utilização de dados sintéticos representativos do cenário norte-americano, o que demanda cautela na transposição direta dos resultados de acurácia para outras realidades, como a brasileira, sem estudos adicionais

com dados locais. Adicionalmente, o foco no carregamento de Nível 1 e a não exploração exaustiva de todas as arquiteturas neurais avançadas ou combinações de hiperparâmetros abrem caminhos para investigações futuras.

Como norteador para trabalhos futuros, é possível realizar a validação do modelo com dados reais de consumidores brasileiros, a incorporação de diferentes níveis e estratégias de carregamento de VEs (Nível 2, carregamento inteligente etc.), aprofundar na análise de sensibilidade do modelo `9-9-6-2-1 ReLU Com BN` em relação às suas entradas a fim de explorar simplificações e/ou incrementações na arquitetura. Destaca-se também a possibilidade de exploração de arquiteturas neurais alternativas, como Redes Neurais Recorrentes (RNNs) e Long Short-Term Memory (LSTMs), que podem oferecer vantagens na modelagem de séries temporais complexas.

Portanto, este trabalho atingiu seus objetivos ao desenvolver e validar um modelo de RNA MLP que demonstrou uma boa performance na previsão da demanda energética residencial, incluindo o carregamento de veículos elétricos, utilizando uma abordagem metodológica sistemática e ferramentas computacionais amplamente acessíveis. Os resultados e *insights* obtidos reforçam o potencial das técnicas de inteligência artificial como ferramentas valiosas para enfrentar os desafios contemporâneos e dinâmicos no setor elétrico, contribuindo para a operação mais eficiente e confiável das redes de energia do futuro.

REFERÊNCIAS

ABADI, M. et al. **TensorFlow: A system for large-scale machine learning**. 2016. Disponível em: https://arxiv.org/abs/1605.08695.

AGENCY, I. E. **Global EV Outlook 2022**. 2022. Acesso em: abr. 2024. Disponível em: https://www.iea.org/reports/global-ev-outlook-2022.

AGENCY, I. E. **Global EV Outlook 2024**. 2024. Acesso em: abr. 2024. Disponível em: https://www.iea.org/reports/global-ev-outlook-2024.

Associação Brasileira do Veículo Elétrico. **Relatório Anual 2023: Mercado Brasileiro de Veículos Elétricos**. São Paulo: [s.n.], 2023. Acesso em: 22 mar. 2025. Disponível em: https://abve.org.br/abve-data/bi-geral/.

Bloomberg NEF. **Electric Vehicle Outlook 2023**. 2023. Acesso em: 20 mar. 2025. Disponível em: https://about.bnef.com/electric-vehicle-outlook.

BROWNLEE, J. **Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python**. [S.l.]: Machine Learning Mastery, 2018.

China Electric Vehicle Association. **EV Charging Infrastructure Report 2023**. 2023. Acesso em: 11 mar. 2025. Disponível em: https://theicct.org/wp-content/uploads/2024/01/ID-93-%E2%80%93-China-charging-Report-A4-70131-v6.pdf.

EHSANI, M.; GAO, Y.; LONGO, S. **Modern Electric, Hybrid Electric, and Fuel Cell Vehicles**. 3. ed. Boca Raton: CRC Press, 2018. ISBN 978-1498761567.

ELéTRICO, A. B. do V. **Infraestrutura de recarga no Brasil**. 2024. Acesso em: abr. 2024. Disponível em: https://abve.org.br/infraestrutura-de-recarga-acelera-no-pais-e-apresenta-crescimento-de-179/.

European Commission. **Regulation (EU) 2023/857 on CO2 emission performance standards**. Brussels, 2023. Acesso em: 18 mar. 2025. Disponível em: https://eur-lex.europa.eu/eli/reg/2023/857/oj/eng.

FORMAGGIO, G. P. **Previsão de demanda de carregamento para veículos elétricos a curto prazo utilizando rede neural artificial**. 2021. Dissertação (Dissertação (Mestrado em Engenharia Elétrica)) — Faculdade de Engenharia de Ilha Solteira, Universidade Estadual Paulista "Júlio de Mesquita Filho", Ilha Solteira, 2021. 92 f. Orientadora: Anna Diva Plasencia Lotufo.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. http://www.deeplearningbook.org.

HARRIS, C. R. et al. Array programming with numpy. **Nature**, Springer Science and Business Media LLC, v. 585, n. 7825, p. 357–362, set. 2020. ISSN 1476-4687. Disponível em: http://dx.doi.org/10.1038/s41586-020-2649-2.

HAYKIN, S. **Redes Neurais: Princípios e Prática**. Bookman Editora, 2001. ISBN 9788577800865. Disponível em: https://books.google.com.br/books?id=bhMwDwAAQBAJ.

HUNTER, J. D. Matplotlib: A 2d graphics environment. **Computing in Science Engineering**, v. 9, n. 3, p. 90–95, 2007.

HYNDMAN, R. J.; ATHANASOPOULOS, G. **Forecasting: Principles and Practice**. 3. ed. OTexts, 2021. Acesso em: 22 mar. 2025. Disponível em: https://otexts.com/fpp3/.

INSTITUTE, T. E. **EV Consumer Behavior**. 2021. Acesso em: abr. 2024. Disponível em: https://www.transportationenergy.org/wp-content/uploads/2022/11/21FI_-EVC_ConsumerBehaviorReport_V07-FINAL.pdf.

Intergovernmental Panel on Climate Change. **Climate Change 2023: Mitigation of Climate Change**. 2023. Acesso em: 18 mar. 2025. Disponível em: https://www.ipcc.ch/report/ar6/wg3.

International Energy Agency. **Global EV Outlook 2023**. 2023. Acesso em: 12 mar. 2025. Disponível em: https://www.iea.org/reports/global-ev-outlook-2023.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. **CoRR**, abs/1502.03167, 2015. Disponível em: http://arxiv.org/abs/1502.03167.

JAIN, P. **Breakdown & Simplify: AI, ML, NLP, Deep Learning, Computer Vision**. 2023. Acesso em: 29 mar. 2025. Disponível em: https://medium.com/@jainpalak9509/breakdown-simplify-ai-ml-nlp-deep-learning-computer-vision-c76cd982f1e4.

Keras Ref. **Training and evaluation with built-in methods**. 2024. Acesso em: 21 maio 2025. Disponível em: https://keras.io/guides/training_with_built_in_methods/.

KIM, J. et al. **A Comprehensive Survey of Deep Learning for Time Series Forecasting: Architectural Diversity and Open Challenges**. 2024. Disponível em: https://arxiv.org/abs/2411.05793.

LEARNLIB scikit. **scikit-learn: Machine Learning in Python**. 2024. https://scikit-learn.org/stable/index.html.

LOPES, J. A. P.; SOARES, F. J.; ALMEIDA, P. M. R. Integration of electric vehicles in the electric power system. **Proceedings of the IEEE**, v. 99, n. 1, p. 168–183, 2011. Acesso em: 14 mar. 2025.

MCKINNEY, W. Data structures for statistical computing in python. **Proceedings of the 9th Python in Science Conference**, v. 445, p. 56–61, 01 2010.

MURATORI, M. **Impact of uncoordinated plug-in electric vehicle charging on residential power demand - supplementary data**. 2017. NREL Data Catalog. Golden, CO: National Renewable Energy Laboratory. Last updated: December 18, 2024. Disponível em: https://data.nrel.gov/submissions/69.

MURATORI, M. Impact of uncoordinated plug-in electric vehicle charging on residential power demand. **Nature Energy**, Nature Publishing Group, v. 3, n. 3, p. 193–201, 2018.

MURATORI, M. et al. A highly resolved modeling technique to simulate residential power demand. **Applied Energy**, v. 107, p. 465–473, 2013. ISSN 0306-2619. Disponível em: https://www.sciencedirect.com/science/article/pii/S030626191300175X.

OBSERVATORY, E. A. F. **EAFO analysis: Trends in EV charging infrastructure across Europe**. 2024. Acesso em: abr. 2024. Disponível em: https://alternative-fuels-observatory.ec.europa.eu/general-information/news/eafo-analysis-trends-ev-charging-infrastructure-across-europe.

PANDASLIB. **pandas: powerful Python data analysis toolkit**. 2024. https://pandas.pydata.org/.

PEDREGOSA, F. et al. **Scikit-learn: Machine Learning in Python**. 2018. Disponível em: https://arxiv.org/abs/1201.0490.

PROCEFFA. **Mape: Guide To Mean Absolute Percentage Error**. 2023. Acesso em: 8 maio 2025. Disponível em: https://proceffa.org/mape-guide-to-mean-absolute-percentage-error/.

QUIRóS-TORTóS, J.; OCHOA, L. F.; LEES, B. A statistical analysis of ev charging behavior in the uk. In: **2015 IEEE PES Innovative Smart Grid Technologies Latin America (ISGT LATAM)**. [S.l.: s.n.], 2015. p. 445–449.

RASHID, M.; ELFOULY, T.; CHEN, N. Short-term load forecasting using artificial neural networks: A review. In: **2023 20th International Conference on the European Energy Market (EEM)**. IEEE, 2024. p. 1–6. Acesso em: 29 mar. 2025. Disponível em: https://ieeexplore.ieee.org/document/10670452.

SAGAN, C. **O Mundo Assombrado pelos Demônios: A Ciência Vista Como Uma Vela no Escuro**. São Paulo: Companhia das Bolso, 1996. Tradução de The Demon-Haunted World: Science as a Candle in the Dark, 1995.

SANDERSON, G. **Neural Networks**. 2017. 3Blue1Brown. Acesso em: 29 jan. 2025. Disponível em: https://www.3blue1brown.com/lessons/neural-networks#title.

Scikit-learn developers. **Model evaluation: quantifying the quality of predictions**. [S.l.], 2024. Acesso em: 8 maio 2025. Disponível em: https://scikit-learn.org/stable/modules/model_evaluation.html.

SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. **The journal of machine learning research**, v. 15, n. 1, p. 1929–1958, 2014.

TENSORFLOWLIB. **TensorFlow**. 2024. https://www.tensorflow.org/.

Tesla Motors. **Technical Specifications of Tesla Model 3**. 2023. Acesso em: 8 mar. 2025. Disponível em: https://www.tesla.com/model3.

U.S. Department of Energy. **Electric Vehicles Tax Credits**. 2023. Acesso em: 10 mar. 2025. Disponível em: https://www.energy.gov/eere/electricvehicles/electric-vehicles-tax-credits.

U.S. Energy Information Administration. **2009 Residential Energy Consumption Survey: Overview of U.S. Residential Energy Consumption in 2009**. 2013. Acesso em: 29 abr. 2025. Disponível em: https://www.eia.gov/consumption/residential/reports/2009/overview.php.

VERMA, H.; TRIPATHI, P.; NAQVI, A. M. A comprehensive review on impacts of electric vehicle charging on the distribution network. p. 1–6, 2021. Acesso em: 28 jul. 2024.

WASKOM, M. L. seaborn: statistical data visualization. **Journal of Open Source Software**, The Open Journal, v. 6, n. 60, p. 3021, 2021. Disponível em: https://doi.org/10.21105/joss.03021.

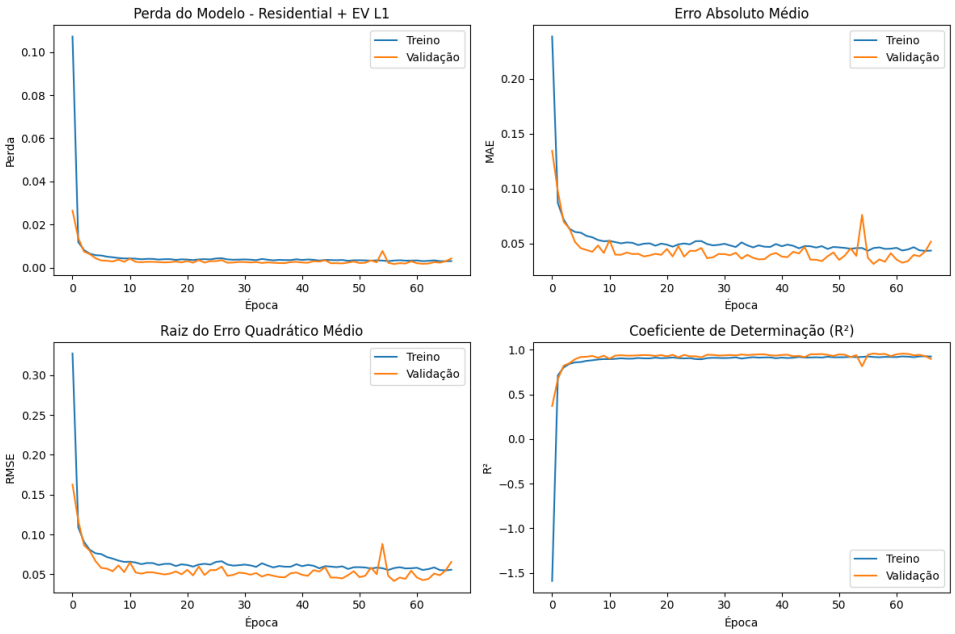
WERBOS, P. J. **Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences**. 1974. Tese (Ph.D. Dissertation) — Harvard University, Cambridge, MA, 1974.

ZHANG, H.-T.; XU, F.-Y.; ZHOU, L. Artificial neural network for load forecasting in smart grid. In: . [S.l.: s.n.], 2010. p. 3200–3205.

APÊNDICE A – CURVAS DE APRENDIZADO ADICIONAIS

As figuras a seguir apresentam as curvas de aprendizado para as demais configurações analisadas:

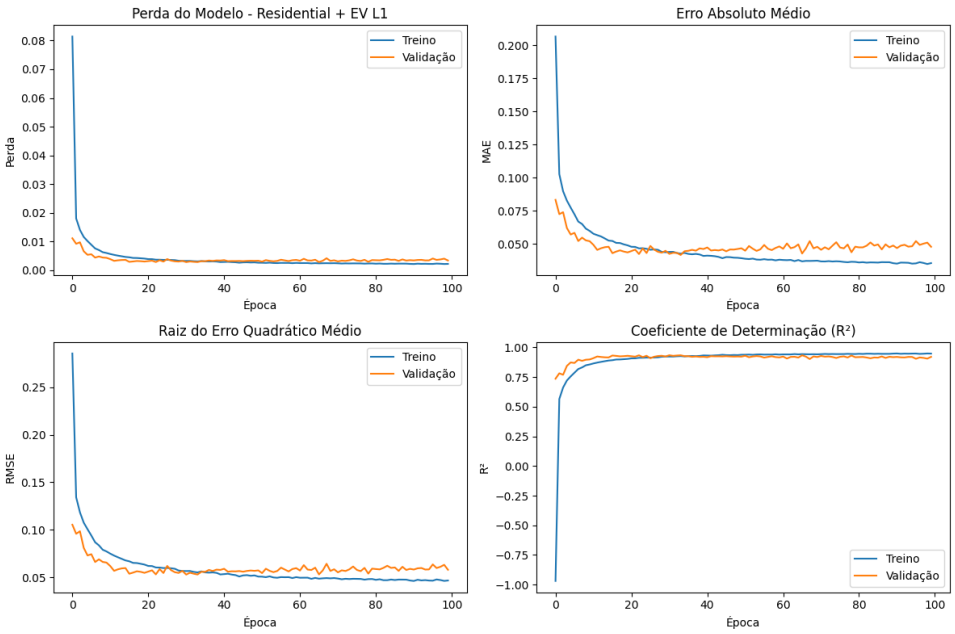
Figura 12 – Curvas de aprendizado (Loss, MAE, RMSE, R^2) para o modelo 24-12-6 ReLU Sem BN, cenário Re+L1.



Fonte: Elaborado pelo autor.

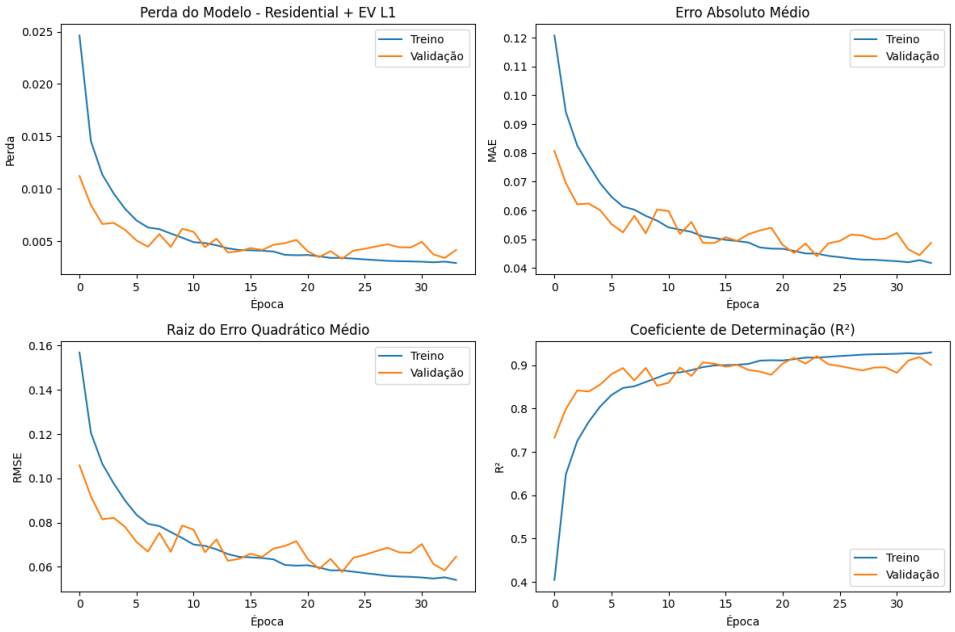
Observar a interação entre as curvas de treino e validação para MAE, RMSE e R^2.

Figura 13 – Curvas de aprendizado para o modelo 32-16-8, cenário Re+L1, sem batch normalization e sem early stopping.



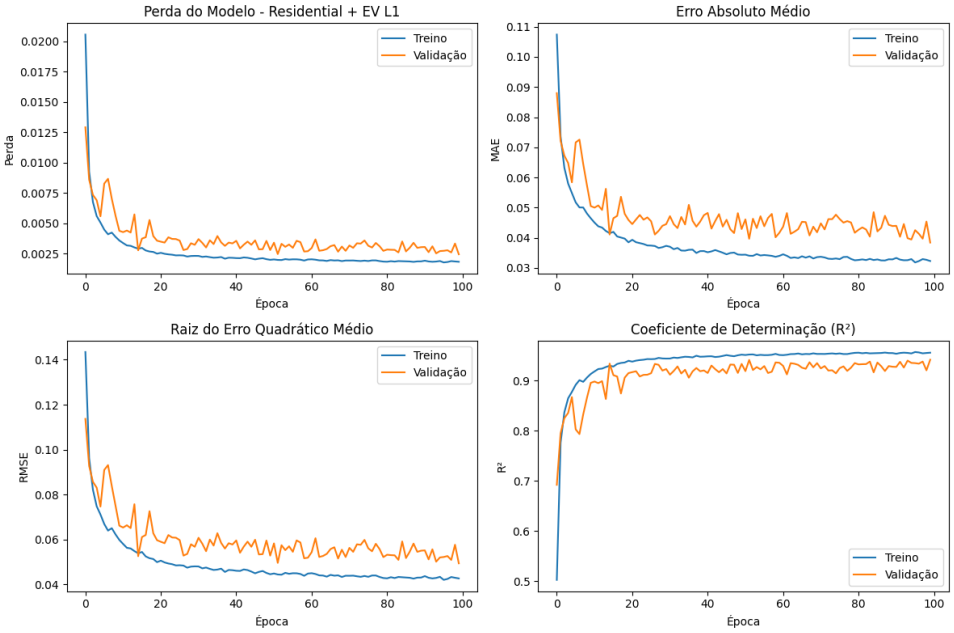
Fonte: Elaborado pelo autor.

Figura 14 – Curvas de aprendizado para o modelo 32-16-8, cenário Re+L1, sem batch normalization e sem early stopping.



Fonte: Elaborado pelo autor.

Figura 15 – Curvas de aprendizado para o modelo 128-64-32, cenário Re+L1, sem batch normalization e sem early stopping.



Fonte: Elaborado pelo autor.