

# OBTENÇÃO DE NEURÔNIOS DE REDES NEURAIS DE BASE RADIAL VIA AGRUPAMENTO DE DADOS POR FLORESTA DE CAMINHOS ÓTIMOS

PASSOS Júnior Leandro A.  
Universidade Estadual Paulista (UNESP Bauru/SP)  
leandropassosjr@fc.unesp.br

COSTA Kelton A. P.  
Universidade Estadual Paulista (UNESP Bauru/SP)  
kelton@fc.unesp.br

ROSA Gustavo Henrique  
Universidade Estadual Paulista (UNESP Bauru/SP)  
gustavo.rosa@fc.unesp.br

PAPA João Paulo  
Universidade Estadual Paulista (UNESP Bauru/SP)  
papa@fc.unesp.br

**RESUMO:** Neste artigo é apresentada uma abordagem para aumentar a eficácia das Redes Neurais Artificiais de Funções de Base Radial utilizando um algoritmo de agrupamento de dados via Floresta de Caminhos Ótimos. Algumas técnicas comumente empregadas para essa tarefa, como o conhecido  $k$ -médias, requerem um determinado número de classes/agrupamentos prévio à sua execução. Embora o número de classes seja conhecido em problemas supervisionados, o número real de agrupamentos é difícil de ser encontrado, dado que uma classe pode ser representada por mais de um agrupamento. Experimentos em nove bases de dados, em conjunto com análises estatísticas, demonstraram que o classificador por Floresta de Caminhos Ótimos possui um melhor desempenho que a técnica  $k$ -médias, bem como encontra as médias das distribuições Gaussianas em posições muito similares às encontradas por este último. Entretanto, o classificador por Floresta de Caminhos Ótimos possui um custo computacional maior, dado que a sua etapa de treinamento é mais custosa que a da técnica  $k$ -médias.

**PALAVRAS-CHAVE:** Redes Neurais Artificiais, Funções de Base Radial, Floresta de Caminhos Ótimos.

**ABSTRACT:** *In this paper, we show how to improve the Radial Basis Function Neural Networks effectiveness by using the Optimum-Path Forest clustering algorithm, since it computes the number of clusters on-the-fly, which can be very interesting for finding the Gaussians that cover the feature space. Some commonly used approaches for this task, such as the well-known  $k$ -means, require the number of classes/clusters previous its performance. Although the number of classes is known in supervised applications, the real number of clusters is hard to figure out, since one class may be represented by more than one cluster. Experiments over nine datasets together with statistical analysis have shown the suitability of Optimum-Path Forest clustering for the RBF training step.*

**KEYWORDS:** Artificial Neural Networks, Radial Basis Function, Optimum-Path Forest.

## 1. INTRODUÇÃO

Técnicas de inteligência artificial vem sendo extensivamente estudadas nas últimas décadas. Melhorias em sua formulação matemática e implementação através de programas de computadores levaram ao desenvolvimento de melhores abordagens

para lidar com o problema de encontrar funções de decisão que melhor separam o espaço de características, levando em conta o contexto de grandes bases de dados.

Existe uma grande variedade de técnicas de aprendizado, que podem partir de abordagens supervisionadas a abordagens não supervisionadas. A introdução do

conhecido *Perceptron* pelo trabalho seminal de Rosenblatt [1] foi o responsável por dar início às pesquisas guiadas pela seguinte pergunta: "Como o cérebro funciona?". Entretanto, mesmo após vários anos, essa preocupação ainda é um problema em aberto.

As Redes Neurais Artificiais (*Artificial Neural Networks - ANNs*) surgiram com a intenção de melhorar a idéia original do *Perceptron*, nas quais uma coleção de neurônios pode processar a informação e propagar sinais de entrada de uma camada para outra. Uma famosa abordagem é a Rede Neural Artificial com *Perceptrons* de Multi-Camadas (*Multi Layer Perceptrons - MLP*), onde sua arquitetura padrão é composta por três camadas principais (entrada, oculta e saída) e tem por objetivo tentar modelar o complexo e intrínseco funcionamento do cérebro [2]. A ideia é separar o espaço de características de tal forma que as amostras com propriedades similares pertençam a um mesmo agrupamento; uma ANN-MLP emprega limites de decisão lineares para cumprir tal tarefa. Funções de separação complexas podem ser obtidas utilizando camadas ocultas com mais neurônios [3].

Outros tipos de redes neurais incluem as Redes Neurais Probabilísticas (*Probabilistic Neural Networks - PNNs*) [4], as quais geram uma saída da probabilidade de cada classe para uma dada amostra de entrada. É uma escolha interessante por serem mais rápidas do que as ANN-MLP, bem como de fácil implementação. Posteriormente, Ahmadlou e Adeli [5] introduziram uma versão melhorada das PNNs para lidar com dados irregulares. Redes Neurais *Spiking* consideram o tempo de cada dado de entrada, e são vantajosas para sinais que podem variar de acordo com o tempo [6]. Mapas de Auto-Organização adotam a idéia da espacialidade do cérebro, onde neurônios próximos tendem a possuir funções similares [7]. Outros conjunto de técnicas interessante diz respeito às redes neuro-fuzzy, que empregam a lógica difusa no processo de aprendizado usualmente executado pelas redes neurais [8].

Redes Neurais de Função de Base Radial (*Radial Basis Function - RBF*) utilizam uma arquitetura de três camadas para problemas de regressão e reconhecimento

de padrões. A ideia é submeter os dados de entrada a um mapeamento não-linear executado pelas funções radiais na camada oculta e, após isso, uma combinação linear das saídas da camada oculta é utilizada na camada de saída [3]. Redes Neurais Artificiais de funções de base radial (ANN-RBF) também procuram separar o espaço de características como uma ANN-MLP, mas com diferentes limiares de decisão. Usualmente, estes tipos de redes neurais podem ser mais rápidas do que uma ANN-MLP, dado que sua fase de treinamento é mais simples e pode ser conduzida usando diferentes abordagens, sendo uma delas obtida em somente um passo (solução da Pseudo-inversa).

A ideia básica de uma rede neural artificial de função de base radial é efetuar uma fase de treinamento seguida de classificação, sendo este procedimento composto por duas etapas: (i) uma não supervisionada, a qual é responsável por encontrar os parâmetros das funções radiais, e (ii) uma fase supervisionada, que executa um mapeamento não-linear do vetor de entrada para uma futura combinação linear de pesos. Dentre a diversas variedades de funções de base radial, a mais empregada e conhecida é a Gaussiana, que possui uma simples e efetiva formulação. Entretanto, o principal desafio de uma rede ANN-RBF é a necessidade de englobar um grande espaço pelas funções Gaussianas, que implica nesta ANN-RBF a extrema dependência da eficácia da abordagem de agrupamento de dados, a qual é responsável por encontrar os parâmetros das Gaussianas, i.e., média e variância (esta última é comumente estimada através de uma equação fechada). Embora a maior parte das implementações utilize o conhecido *k*-médias para tal tarefa, o qual é simples e fácil de ser implementado, em aplicações reais não é tão fácil de se conhecer o número de agrupamentos, como requerido pelo *k*-médias [19]. Muito embora o usuário possa ter o conhecimento do número de classes, cada uma delas pode ser representada por mais de uma função Gaussiana. Adicionalmente, o valor de *k* pode ser estimado através de um conjunto de validação, mas isto pode se tornar impraticável à medida que utilizamos bases

de dados maiores.

Atualmente, um número considerável de trabalhos tem sido responsável pelo problema de encontrar os parâmetros das Gaussianas utilizando abordagens de otimização. Esmaeili e Mozayani [9], por exemplo, utilizaram a técnica de *Particle Swarm Optimization* (PSO) para esta tarefa. Tsekouras e Tsimikas [10] propuseram uma abordagem híbrida utilizando PSO e um agrupamento de dados baseado em lógica difusa. Um algoritmo baseado em mimetismo utilizando o conceito de Evolução Diferencial também foi utilizado para o mesmo contexto por Qasema e Shamsuddina [11]. Embora estas abordagens baseadas em otimização sejam interessantes e amplamente empregadas para o treinamento de uma RBF, é importante apontar que: (i) elas podem ficar presas em um local ótimo, (ii) algumas delas tem um alto custo computacional, e (iii) muitas vezes necessitam de outros parâmetros, os quais podem ser empiricamente encontrados ou até mesmo com técnicas de meta-otimização, aumentando a complexidade de todo o sistema.

Recentemente, Rocha et al. [12] propuseram uma abordagem interessante para agrupamento de dados chamada Floresta de Caminhos Ótimos (*Optimum-Path Forest - OPF*), a qual modela a tarefa de agrupamento de dados como um problema de partição de um grafo, onde as amostras são os nós e uma relação de adjacência pré-definida é responsável por conectá-los. A partição do grafo é regulada por uma competição entre nós chaves (protótipos) utilizando uma função de custo, sendo que cada um deles é a raiz de sua floresta de caminhos ótimos (agrupamento), que contém os nós conquistados. O OPF demonstrou melhores resultados do que o tradicional algoritmo Mean-Shift [12], sendo também capaz de encontrar o número de agrupamentos em tempo de execução, i.e., ele não requer o número de agrupamentos como um dado de entrada, como o  $k$ -médias, por exemplo, sendo uma ferramenta interessante para o treinamento de redes RBF. Adicionalmente, os protótipos estimados pelo OPF, os quais serão responsáveis por conter os valores médios das distribuições Gaussianas, são posicionados no centro (ou em regiões mui-

to próximas) dos agrupamentos (regiões de alta densidade), sendo então localizados em posições similares aos computados e encontrados pelo  $k$ -médias.

Este artigo introduz o agrupamento de dados via OPF para o treinamento de redes RBF, o qual é comparado extensivamente contra o tradicional  $k$ -médias em diversas bases de dados reais e sintéticas. Os experimentos foram estatisticamente analisados e mostraram a robustez do agrupamento de dados via OPF em situações nas quais o  $k$ -médias poderia falhar. O restante deste artigo está organizado como segue. A Seção 1 apresenta um cenário teórico relacionado com as redes RBF e o agrupamento de dados via OPF. As Seções 2 e 3 descrevem a abordagem proposta para encontrar possíveis centros das Gaussianas utilizando o OPF e a seção de experimentos, respectivamente. Finalmente, as conclusões são descritas ao final deste trabalho.

## 2. CENÁRIO TEÓRICO

### 2.1. REDES NEURAIS ARTIFICIAIS DE FUNÇÃO DE BASE RADIAL

Redes Neurais Artificiais de Funções de Base Radial são algumas das técnicas de aprendizado de máquina mais utilizadas, as quais podem ser vistas como uma rede neural multi-camadas composta por três camadas: entrada, oculta e saída. A primeira camada recebe um vetor de entrada, o qual é aplicado a uma transformação não linear na camada oculta. Finalmente, uma combinação linear de saídas da camada oculta é executada na última camada.

Dado um vetor de entrada,  $x \in \mathbb{R}^n$  as saídas de uma rede neural RBF podem ser computadas como segue:

$$y_j = \sum_{i=1}^m h_i(x_i | m_i, \Sigma_i) w_{i,j}, j = 1, \dots, p, \quad (1)$$

em que  $w_{i,j} \in \mathbb{R}$  corresponde ao peso da conexão entre o neurônio oculto  $i$  e o neurônio de saída  $j$ ,  $p$  denota o número de neurônios da camada de saída, e  $h \in \mathbb{R}^m$  denota o vetor de funções base. Em relação às redes neurais RBF,  $h$  pode ser composto por qualquer tipo de função de base radial, sendo a mais comumente utilizada a Gaus-

siana de  $n$ -dimensões:

$$h_j(x|\mu_j, \Sigma_j) = \frac{1}{2\pi^{n/2}|\Sigma_j|^{1/2}} e^{\left(-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1}(x-\mu_j)\right)}, \quad (2)$$

onde  $\mu_j \in \mathbb{R}$  e  $\Sigma_j$  correspondem aos centros (vetor médio) e a matriz de covariância da Gaussiana  $j$ , respectivamente. Assim sendo, a fase de treinamento de uma ANN-RBF tem por objetivo encontrar valores plausíveis para  $\mu, \Sigma$  e a matriz de pesos  $W$ .

Basicamente,  $\mu, \Sigma$  são computadas através de um aprendizado não supervisionado. Considerando  $w$ , existem algumas abordagens interessantes, como o método da pseudo-inversa e a regra delta generalizada. Neste artigo utilizaremos a primeira abordagem, a qual será descrita em mais detalhes na próxima seção.

1) *Método da Pseudo-Inversa:* O método da pseudo-inversa pode ser visto como um problema quadrático, no qual podemos computar a matriz de pesos utilizando determinadas equações. Considere  $X = \{x^1, x^2, x^3, \dots, x^r\}$  como uma base de dados de treinamento, dado que  $x^i \in \mathbb{R}^n$ . Agora, iremos encontrar utilizando  $X$  como sendo as entradas da rede neural. Assim  $\Phi$  pode ser definida como segue:

$$\Phi = \begin{bmatrix} h_1(x^1|\mu_1, \sigma_1) & h_2(x^1|\mu_2, \sigma_2) & \dots & h_m(x^1|\mu_m, \sigma_m) \\ h_1(x^2|\mu_1, \sigma_1) & h_2(x^2|\mu_2, \sigma_2) & \dots & h_m(x^2|\mu_m, \sigma_m) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(x^r|\mu_1, \sigma_1) & h_2(x^r|\mu_2, \sigma_2) & \dots & h_m(x^r|\mu_m, \sigma_m) \end{bmatrix}, \quad (3)$$

onde  $\Phi_{r \times m}$  é a matriz responsável pelos neurônios da camada oculta  $m$  para a entrada das  $r$  amostras de treinamento. Nós temos também:

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,p} \\ w_{2,1} & w_{2,2} & \dots & w_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,1} & w_{m,2} & \dots & w_{m,p} \end{bmatrix}, \quad (4)$$

sendo essa a já conhecida matriz de pesos, em que  $w_{ij}$  corresponde ao peso de cada conexão entre o neurônio oculto  $i$  para o neurônio de saída  $j$ . Finalmente, considere  $y_{r \times p}$  como a matriz de saída:

$$Y = \begin{bmatrix} y_{1,1} & y_{1,2} & \dots & y_{1,p} \\ y_{2,1} & y_{2,2} & \dots & y_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ y_{r,1} & y_{r,2} & \dots & y_{r,p} \end{bmatrix}, \quad (5)$$

em que  $y_{ij}$  corresponde ao valor desejado para a amostra de treinamento  $i$  no neurônio de saída  $j$ . Note que  $y_i$  foi definido na Equação 1, mas para apenas uma amostra  $x \in \mathbb{R}^n$ . Agora, existe um conjunto  $x$  com  $r$  amostras, e cada  $y_{ij}$  é agora redefinido como parte da matriz  $y$ , mas que também pode ser computada como anteriormente, i.e., utilizando a Equação 1, a qual pode ser reescrita para lidar com todo o conjunto de treinamento:

$$Y = \Phi W \quad (6)$$

Por agora, temos que encontrar  $W$ , que pode ser calculado como a seguir:

$$W = \Phi^{-1}Y, \quad (7)$$

em que  $\Phi^{-1}$  é a inversa de  $\Phi$ . Como  $\Phi$  pode não ser quadrada, temos que computar sua pseudo-inversa  $\Phi^+$ , que é dada pela fórmula a seguir:

$$\Phi^+ = VS^+U^T, \quad (8)$$

a qual também é chamada de pseudo-inversa de Moore-Penrose. Neste caso, a Equação 7 pode ser reescrita como:

$$W = \Phi^+Y. \quad (9)$$

Portanto,  $W$  pode ser computada e seus pesos estimados. Após isso, é necessário calcular a variância de cada distribuição Gaussiana. Normalmente, é assumida a sua distribuição isotrópica, o que significa que podemos utilizar a mesma variância para todas Gaussianas. Dentre várias equações para este propósito, utilizemos a seguinte:

$$\sigma = 2D, \quad (10)$$

onde  $D$  denota a distância média entre os centros. Como estamos trabalhando com distribuições Gaussianas de  $n$ -dimensões (Equação 2), é possível criar matrizes diagonais de covariâncias com tamanho  $n \times n$ , como segue:

$$\Sigma_i = \begin{bmatrix} \sigma & 0 & \dots & 0 \\ 0 & \sigma & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma \end{bmatrix}, \quad (11)$$

em que  $\Sigma_i$  denota a matriz de covariância para a distribuição Gaussiana  $i$ .



## 2.2. AGRUPAMENTO DE DADOS VIA FLORESTA DE CAMINHOS ÓTIMOS

A estrutura de classificadores de padrões baseados em Florestas de Caminhos Ótimos é proposta por uma metodologia baseada em grafos para ajustar as relações de conectividade entre amostras de dados em um determinado espaço de características. A metodologia interpreta a etapa de treinamento como um grafo, no qual seus nós são as amostras e os arcos que conectam pares de amostras satisfazem uma determinada relação de adjacência. Para uma possível função de valor (conectividade), o algoritmo OPF [13] particiona o grafo em árvores de caminhos ótimos com raízes em amostras chaves, nomeadas protótipos. Os protótipos competem entre si pelas amostras mais fortemente conexas na etapa de treinamento, em que cada amostra é assinalada para uma árvore cujo protótipo oferece a ela um caminho de custo ótimo. A classificação de uma nova amostra é feita encontrando a raiz mais conexa em um método incremental através da avaliação dos valores do caminho ótimo nas amostras de treinamento.

Seja  $Z$  uma base de dados onde, para cada amostra  $s \in Z$ , exista um vetor de características  $\vec{v}(s)$ . Seja  $d(s,t)$  a distância entre  $s$  e  $t$  no espaço de características. Um grafo  $(Z, A_k)$  pode ser definido como seus arcos  $(s,t) \in A$  conectando  $k$ -próximos vizinhos no espaço de características. Os arcos possuem seus pesos definidos por  $d(s,t)$  e seus nós  $s \in Z$  tem seus pesos computados por um valor de densidade de probabilidade  $\rho(s)$ :

$$\rho(s) = \frac{1}{\sqrt{\pi\sigma^2}|A_k(s)|} \sum_{t \in A_k(s)} \exp\left(\frac{-d^2(s,t)}{2\sigma^2}\right), \quad (12)$$

onde  $|A_k(s)| = k$ ,  $\sigma = \frac{d_f}{3}$ , e  $d_f$  é o peso máximo do arco em  $(Z, A_k)$ . Esta escolha de parâmetro considera todos os nós adjacentes para o cálculo da densidade, dado que uma função Gaussiana engloba quase todas as amostras entre  $d(s,t) \in [0, 3\sigma]$ . Adicionalmente, como  $A_k$  é assimétrica, arcos de simetria devem ser adicionados aos platôs de uma função de densidade de pro-

babilidade (fdp) com o objetivo de garantir uma única raiz por máximo.

Tradicionalmente, a estimativa de uma fdp é realizada pelo método de Janela de Parzen. A Equação 12 pode prover uma estimativa da Janela de Parzen [20] baseada no núcleo isotrópico de uma Gaussiana quando os arcos são definidos por  $(s,t) \in A$  se  $d(s,t) \leq d_f$ . Essa escolha, entretanto, apresenta problemas nas diferenças de escala e concentração de amostras. Soluções para esse problema levaram a escolhas adaptativas de  $df$  dependendo da região do espaço de características [14]. Levando em consideração os  $k$ -vizinhos próximos, o método lida com diferentes concentrações e reduz o problema de escala encontrando o melhor valor de  $k$ , digamos  $k^*$ , com  $[K_{min}, K_{max}]$ , para  $1 \leq k_{min} < k_{max} \leq |Z|$ .

A solução proposta por Rocha et al. [12] para encontrar  $k^*$  considera o corte mínimo no grafo entre todos os resultados do agrupamento para  $k \in [1, k_{max}]$ , de acordo com a medida normalizada  $GC(A_k, L, d)$  sugerida por Shi e Malik [15]:

$$GC(A_k, L, d) = \sum_{i=1}^c \frac{T'_i}{T_i + T'_i}, \quad (13)$$

$$T_i = \sum_{(s,t) \in A_k \vee L(s)=L(t)=i} \frac{1}{d(s,t)}, \quad (14)$$

$$T'_i = \sum_{(s,t) \in A_k \vee L(s)=i, L(t) \neq i} \frac{1}{d(s,t)}, \quad (15)$$

onde  $L(t)$  é o rótulo da amostra  $t$ ,  $T'_i$  utiliza todos os pesos dos arcos entre o agrupamento  $i$  e todos os outros agrupamentos, e  $T_i$  usa todos os pesos dos arcos com o agrupamento  $i = 1, 2, \dots, c$ .

O método define um caminho  $\pi_t$  como uma sequência de amostras adjacentes começando pela raiz  $R(t)$  e terminando na amostra  $t$ , sendo  $\pi_t$  um caminho trivial e  $\pi_s \cdot (s,t)$  a concatenação de  $\pi_s$  e o arco  $(s,t)$ . Esta ideia associa para cada caminho  $\pi_t$  um valor  $f(\pi_t)$  dado pela função de conectividade  $f$ . Um caminho  $\pi_t$  é considerado ótimo se  $f(\pi_t) \geq f(\tau_t)$  para qualquer outro caminho  $\tau_t$ .

Dentre todos os possíveis caminhos  $\pi_t$  a partir do máximo da fdp, este método assume para  $t$  um caminho o qual o valor mínimo de densidade é seu máximo. Isto é, o método encontra

$V(t) = \max_{\pi_t \in (Z, A_k)} \{f(\pi_t)\}$  para  $f(\pi_t)$  definida por:

$$f((t)) = \begin{cases} \rho(t) & \text{se } t \in R \\ \rho(t) - \delta & \text{caso contrário} \end{cases}$$

$$f((\pi_s \cdot (s, t))) = \min\{f(\pi_s, \rho(t))\}, \quad (16)$$

para  $\delta = \min_{(s,t) \in A_k, \rho(t) \neq \rho(s)} |\rho(t) - \rho(s)|$  e  $R$  sendo um conjunto de raízes computadas em tempo de execução, com um elemento por cada máximo na fdp. Note que valores maiores de  $\delta$  reduzem o número de máximos. No artigo é considerado  $\delta = 1.0$  e redimensionando números reais  $\rho(t) \in [1,100]$ . O algoritmo do OPF maximiza o mapa de conectividade  $V(t)$  computando uma floresta de caminhos ótimos — um mapa de predecessores  $P$  sem conter ciclos e associa para cada amostra  $t \in R$  seu predecessor  $P(t)$  no caminho ótimo a partir  $R$  de ou um marcador nil quando  $t \in R$ .

### 3. TREINAMENTO VIA FLORESTA DE CAMINHOS ÓTIMOS

Nesta seção, apresentamos a abordagem proposta que emprega o agrupamento de dados via OPF para encontrar os centros das distribuições Gaussianas. Dada uma base de dados supervisionada, a mesma é particionada em conjuntos de treinamento e teste, sendo o primeiro utilizado pelo OPF para agrupar as amostras, e então encontrar os seus protótipos. Após isso, suas posições são armazenadas e utilizadas como os valores médios para as distribuições Gaussianas (as matrizes de covariância são computadas utilizando a Equação 11).

Sendo  $c$  o número de classes de uma determinada base de dados, após a criação dos modelos Gaussianos com parâmetros estimados pelo OPF, empregamos uma arquitetura neural com  $n$  neurônios de entrada,  $m$  neurônios para a camada escondida e  $|c|$  neurônios para a camada de saída. A Figura 1 ilustra o procedimento proposto.

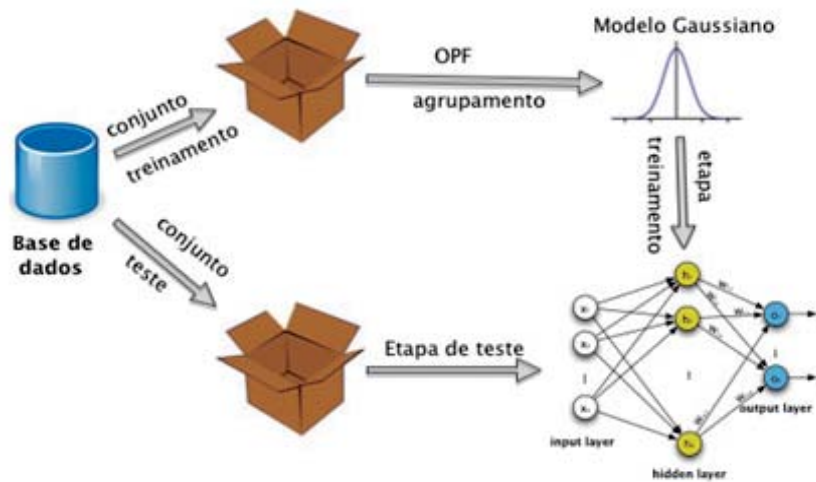


Figura 1 - Arquitetura proposta para o treinamento de uma rede neural utilizando OPF.

### 4. RESULTADOS EXPERIMENTAIS

Os experimentos foram executados em duas diferentes fases: (i) na primeira fase, apresentamos a similaridade (localização) entre os protótipos encontrados pelo OPF e as médias computadas pelo  $k$ -médias em quatro bases de dados sintéticas de duas dimensões, e também foi discutida a robustez do OPF em relação ao uso do parâmetro  $K_{max}$ ; (ii) no último experimen-

to, empregamos o classificador OPF na etapa de estimação dos parâmetros das distribuições Gaussianas visando o treinamento das redes neurais para as mesmas bases de dados sintéticas utilizadas na etapa anterior, bem como para cinco novas bases de dados que correspondem a problemas reais. É importante salientar que, para todos os experimentos, foi utilizado um computador com processador Intel I5®, 8Gb RAM e sistema operacional OS X 10.8.5.

#### 4.1. BASE DE DADOS

A Tabela 1 apresenta as bases de dados empregadas neste trabalho. Note que as primeiras quatro bases de dados são as mesmas utilizadas no primeiro experimento (Seção 3.2.1.), e as restantes são bases de dados de domínio público.

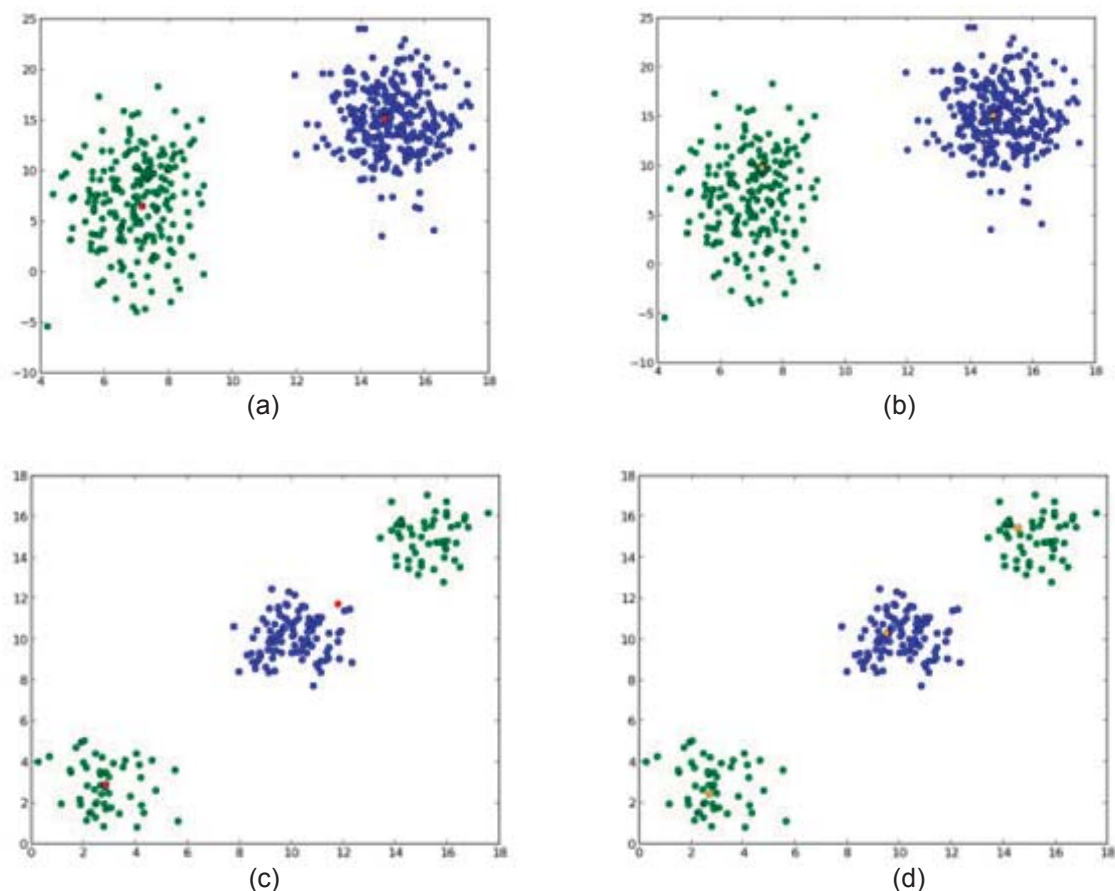
**Tabela 1** - Descrição das bases de dados utilizadas neste trabalho

Bases de Dados	# amostras	# características	# classes
Db1	500	2	2
Db2	200	2	2
Db3	150	2	3
Db4	445	2	3
Wine [16]	178	13	4
Iris [16]	150	4	3
Breast-cancer [16]	683	10	2
Saturn [17]	200	2	2
Cone-Torus [17]	200	2	3

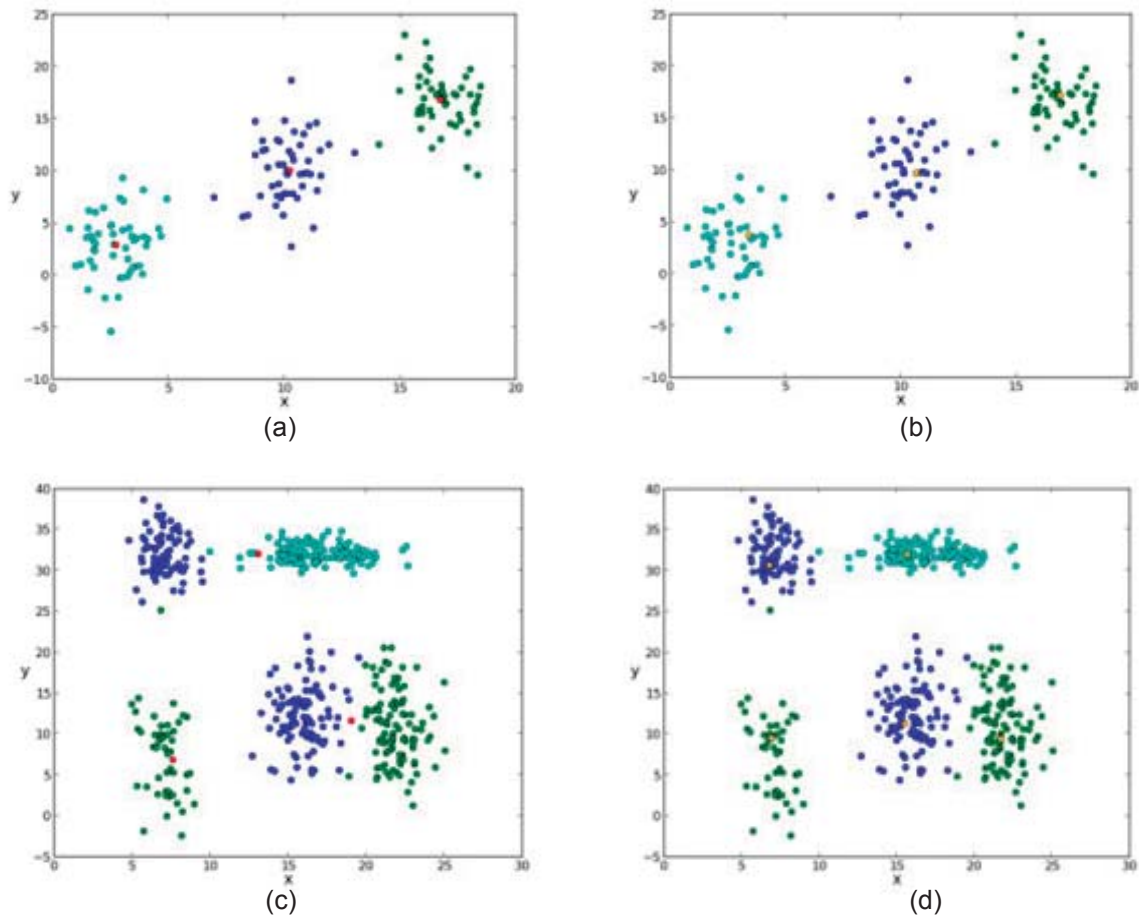
#### 4.2. EXPERIMENTOS

##### 4.2.1 OPF VERSUS K-MÉDIAS

Como mencionado anteriormente, a primeira rodada de experimentos teve por objetivo comparar a posição dos protótipos do OPF com os centros obtidos pelo  $k$ -médias. As Figuras 2 e 3 apresentam tais resultados, sendo que  $k_{max} = 50$  foi utilizado para todas as bases de dados. Pode-se observar que ambas técnicas alcançaram resultados similares para as bases Db1 (Figuras 2a e 2b) e Db3 (Figuras 3a e 3b). Entretanto, em situações em que se tem mais de um agrupamento por classe,  $k$ -médias pode não conseguir alcançar bons resultados (número incorreto de agrupamentos), como apontado pelas bases de dados Db2 (Figuras 2c e 2d) e Db4 (Figuras 3c e 3d). Portanto, tal comportamento pode diminuir a eficácia da ANN-RBF, dado que o espaço de características não foi englobado corretamente pelas distribuições Gaussianas. Adicionalmente, pode-se observar que OPF encontrou os centros para todos os agrupamentos de todas as bases de dados.



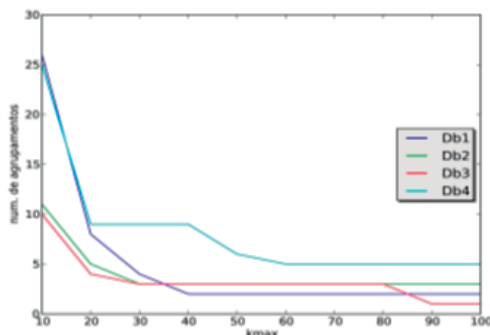
**Figura 2** - Bases de Dados sintéticas: protótipos do  $k$ -médias em vermelho para as bases de dados (a) Db1 e (c) Db2, e protótipos da técnica OPF em laranja para as bases de dados (b) Db1 e (d) Db2.



**Figura 3** - Bases de Dados sintéticas: protótipos do k-médias em vermelho para as bases de dados (a) Db3 e (c) Db4, e protótipos da técnica OPF em laranja para as bases de dados (b) Db3 e (d) Db4.

Sobre o agrupamento de dados via OPF, pode-se perguntar: “E o parâmetro  $K_{max}$ ?”. Este parâmetro controla o espaço de uma amostra durante o processo de competição, e seu valor deve, previamente, definir o número de agrupamentos (Seção 1.2). Entretanto, o parâmetro  $K_{max}$  é muito menos sensível ao erro para estimativas de

agrupamentos do que o valor de  $k$  para o k-médias. A Figura 4 apresenta o resultado do experimento sobre as bases de dados Db1, Db2, Db3 e Db4 utilizando diferentes valores de  $K_{max}$ . Pode-se observar que existem vários platôs nas curvas, i. e., existem diversos valores de  $K_{max}$  que não alteram o número de agrupamentos. Portanto, um usuário com uma pequena experiência na utilização do OPF pode aprender possíveis valores de entrada para  $K_{max}$ .



**Figura 4** - Robustez do OPF considerando diferentes valores de  $k_{max}$ .

#### 4.2.2 AVALIANDO A EFICÁCIA DO OPF PARA ESTIMATIVAS DE NEURÔNIOS DE UMA ANN-RBF

Neste experimento, a qualidade dos neurônios encontrados pelo OPF é avaliada contra aqueles computados pelo k-médias. Para tal propósito, empregamos as bases de dados apresentadas na Tabela 1 em um procedimento de validação cruza-



da com 10 execuções, sendo 50% da base de dados utilizada para compor a etapa de treinamento, e o restante das amostras utilizadas para o teste. Após essas execuções, avaliamos a robustez dos experimentos por meio de um *t*-teste pareado com 95% de confiança. A Tabela 2 apresenta a acurácia média da ANN-RBF com os neurônios obtidos pelo OPF e *k*-médias. Note que empregamos uma medida de acurácia média proposta por Papa et al. [18], que considera classes não balanceadas. Os valores dos colchetes na coluna do OPF correspondem aos valores de *k<sub>max</sub>*, os quais foram empiricamente escolhidos.

**Tabela 2** - Acurácia média para a classificação ANN-RBF utilizando OPF e *k*-médias: os desvios padrões são pequenos demais para serem apresentados.

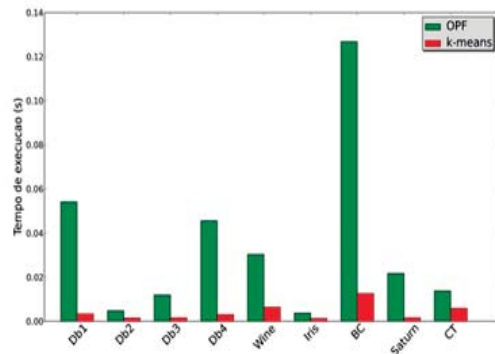
Base de Dados	OPF	<i>k</i> -médias
Db1	98.33%[50]	97.16%
Db2	100.00%[50]	74.00%
Db3	99.00%[50]	99.00%
Db4	97.49%[50]	65.81%
Wine	81.99%[5]	74.38%
Iris	99.00%[10]	94.00%
Breast-cancer	96.34%[50]	80.24%
Saturn	53.99%[50]	44.00%
Cone-Torus	84.33%[10]	73.17%

Os resultados dos neurônios obtidos pelo OPF são claramente melhores do que os computados pelo *k*-médias. Em alguns casos, a taxa de reconhecimento de uma ANN-RBF utilizando neurônios obtidos por OPF foi 48.13% melhor do que usando *k*-médias (base de dados Db4, por exemplo). Tal diferença é justificada por situações em que o número de Gaussianas que compõem o espaço de características é diferente do número de classes. Na prática, como pode-se ver em aplicações reais, a diferença entre o OPF e o *k*-médias é menor, mas ainda sim o OPF possui uma estimativa melhor de neurônios do que o *k*-médias.

Em relação à avaliação estatística, o *t*-teste pareado com 95% de confiança rejeitou a hipótese nula, que declara que o OPF e o *k*-médias são similares entre si. Portanto, pode-se concluir que o OPF providencia melhores neurônios para o treinamento de uma ANN-RBF do que o *k*-médias,

considerando as bases de dados empregadas neste trabalho.

Adicionalmente, computamos o tempo de execução (segundos) para o treinamento da ANN-RBF por OPF e *k*-médias. A Figura 5 ilustra esta informação para cada base de dados. O tempo médio de execução para o *k*-médias (considerando todas as bases de dados) foi, aproximadamente, 0.0041s, enquanto que o treinamento via OPF teve seu tempo estimado em 0.0348s. Portanto, a etapa de treinamento utilizando *k*-médias é 8.48 vezes mais rápida do que utilizando OPF. Novamente, aplicamos um *t*-teste pareado com 95% de confiança nos dados de tempos de execução, os quais evidenciam que *k*-médias é estatisticamente mais rápido que OPF. Entretanto, OPF pode ser uma possível alternativa ao *k*-médias, dado que suas taxas de reconhecimento são consideravelmente melhores.



**Figura 5** - Tempos de execução para OPF e *k*-médias baseado no treinamento de uma ANN-RBF

## 5. CONCLUSÕES

O procedimento do treinamento de uma rede RBF é composto por duas etapas: (i) uma não supervisionada, a qual é responsável por encontrar os neurônios da camada oculta; e (ii) uma supervisionada, que computa os pesos que serão utilizados para combinar a saída dos neurônios escondidos. A abordagem mais utilizada para esse propósito emprega o *k*-médias, dado que os neurônios compreendem o parâmetro da média de uma distribuição Gaussiana. Portanto, tal abordagem é uma boa candidata para esta tarefa. Entretanto, o usuário é

responsável por inserir o número de médias ( $k$ ), o qual pode não corresponder ao número real de agrupamentos existentes. Neste artigo, propusemos a utilização do algoritmo de agrupamento de dados via OPF para tal abordagem, pois ele consegue computar o número de agrupamentos (neurônios) em tempo de execução, sendo sua utilização mais intuitiva do que a do  $k$ -médias.

Os experimentos mostraram que as posições dos neurônios OPF são similares àquelas encontradas pelo  $k$ -médias, e também que o OPF supera a performance do  $k$ -médias em certas situações, onde se requerem mais Gaussianas do que classes para cobrir o espaço amostrado. Resultados experimentais sobre 9 bases de dados mostraram a robustez da fase de treinamento da rede RBF com parâmetros estimados pelo OPF. A principal desvantagem na utilização do OPF consiste no seu custo computacional, que é maior que o do  $k$ -médias.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.
- [2] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.
- [3] S. Haykin, *Neural Networks: A comprehensive foundation*. Prentice-Hall, 1998.
- [4] D. F. Specht, "Probabilistic neural networks," *Neural Networks*, vol. 3, no. 1, pp. 109–118, 1990.
- [5] M. Ahmadlou and H. Adeli, "Enhanced probabilistic neural network with local decision circles: A robust classifier," *Integrated Computer-Aided Engineering*, vol. 17, no. 3, pp. 197–210, 2010.
- [6] E. M. Izhikevich, "Polychronization: Computation with spikes," *Neural Computation*, vol. 18, no. 2, pp. 245–282, 2006.
- [7] T. Kohonen, *The Self-Organizing Maps*, 3rd ed. Springer, 2001.
- [8] J.-S. R. Jang and C.-T. Sun, *Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1997.
- [9] A. Esmaeili and N. Mozayani, "Adjusting the parameters of radial basis function networks using particle swarm optimization," in *Proceedings of the IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, 2009, pp. 179–181.
- [10] G. E. Tsekouras and J. Tsimikas, "On training RBF neural networks using input-output fuzzy clustering and particle swarm optimization," *Fuzzy Sets and Systems*, vol. 221, pp. 65–89, 2013.
- [11] S. N. Qasem and S. M. Shamsuddin, "Memetic elitist pareto differential evolution algorithm based radial basis function networks for classification problems," *Applied Soft Computing*, vol. 11, no. 8, pp. 5565–5581, 2011.
- [12] L. M. Rocha, F. A. M. Cappabianco, and A. X. Falcão, "Data clustering as an optimum-path forest problem with applications in image analysis," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 50–68, 2009.
- [13] A. Falcão, J. Stolfi, and R. Lotufo, "The image foresting transform theory, algorithms, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 19–29, 2004.
- [14] D. Comaniciu, "An algorithm for data-driven bandwidth selection," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 281–288, 2003.
- [15] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug 2000.
- [16] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [17] J. P. Papa, C. T. N. S., and A. X. Falcão, *LibOPF: A library for the design of optimum-path forest classifiers*, 2009, software version 2.0 available at <http://www.ic.unicamp.br/~afalcao/LibOPF>.
- [18] J. P. Papa, A. X. Falcão, and C. T. N. Suzuki, "Supervised pattern classification based on optimum-path forest," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 120–131, 2009.
- [19] R. O. Duda, P. E. Hart e D. G. Stork, "Pattern Classification", John Wiley & Sons, 2001.
- [20] E. Parzen, "On the estimation of a probability density function and the mode", *Annals of Mathematical Statistics*. Vol. 33, no. 3, 1065–1076, 1962.

**Leandro Aparecido Passos Júnior:** Possui graduação em Informática com Gestão da Produção Industrial pela FATEC – Faculdade de Tecnologia de Jahu, SP, Brasil (2009). É aluno de mestrado em Ciência da Computação na UNESP – Universidade Estadual Paulista Júlio de Mesquita Filho. Atualmente, o interesse de suas pesquisas incluem aprendizado de máquinas e reconhecimento de padrões.

**Kelton Pontara Augusto da Costa:** Possui graduação em Análise de Sistemas pela Universidade do Sagrado Coreção, mestrado em Ciência da Computação pelo Centro Universitário Eurípedes (2004), doutorado em Engenharia Elétrica pela Universidade de São Paulo (2009) e pós-doutorado pela Universidade Estadual de Campinas (2011). Atualmente é pesquisador no departamento de computação da Universidade Estadual Paulista Júlio de Mesquita Filho, Unesp-Bauru e professor na Faculdade de Tecnologia de São Paulo desde 2008. Atua nos seguintes temas: Aprendizagem de máquina, segurança em redes de computadores e detecção de anomalias em redes de computadores.

**Gustavo H. de Rosa:** É aluno de graduação em Ciência da Computação na UNESP – Universidade Estadual Paulista Júlio de Mesquita Filho, SP, Brasil desde 2012. Além disso faz parte do grupo de pesquisas Recogna - Grupo de pesquisas em Reconhecimento de Padrões e Biometria e tem interesse nas seguintes áreas: Aprendizagem de máquinas, reconhecimento de padrões e processamento de imagens.

**João Paulo Papa:** Possui graduação em Bacharelado em Sistemas de Informação pela Universidade Estadual Paulista Júlio de Mesquita Filho (2002), mestrado em Ciência da Computação pela Universidade Federal de São Carlos (2005), doutorado em Ciência da Computação pela Universidade Estadual de Campinas (2008) e pós-doutorado em Ciência da Computação pela Universidade Estadual de Campinas (2009). Atualmente, é Professor Assistente Doutor no Departamento de Computação da Universidade Estadual Paulista Júlio de Mesquita Filho, Unesp-Bauru e pesquisador colaborador junto ao Instituto de Computação da Universidade Estadual de Campinas. Tem experiência na área de Ciência da Computação, com ênfase em Sistemas de Computação, atuando principalmente nos seguintes temas: processamento de imagens, aprendizado de máquina e reconhecimento de padrões.