

CÁLCULO DO RAIÃO EM REDES RBF BASEADO NA DISTÂNCIA MÉDIA ENTRE OS CENTROS

PAULO V. C. BATISTA*, ANTÔNIO P. BRAGA*, LUIZ C. B. TORRES*

*Programa de Pós-Graduação em Engenharia Elétrica - Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brasil

Emails: paulovitorsi@gmail.com, apbraga@ufmg.br, luizliltc@gmail.com

Abstract— The technique of database classification as well as pattern recognition consists of applying a well-trained algorithm with the purpose of separating or classifying classes of objects based on their information. The result of this classification depends directly on the normalization of the data, the model used as several other parameters well adjusted. This work presents a method for calculating the Gaussian radius in radial base function networks, as well as its results applied to the benchmark databases. The results obtained are relevant, compared to other methods in the literature.

Keywords— Neural Networks, Radial Basis, Lightning, Classification.

Resumo— A técnica de classificação de base de dados bem como reconhecimento de padrões consiste em aplicar determinado algoritmo bem treinado com o objetivo de separar ou classificar classes de objetos baseando-se em suas informações. O resultado desta classificação depende diretamente da normalização dos dados, do modelo utilizado, assim como de vários outros parâmetros bem ajustados. Neste trabalho é apresentado um método para cálculo do raio das gaussianas em redes de bases radiais, bem como seus resultados aplicados à bases de dados de *benchmark*. Os resultados obtidos são relevantes, comparados a outros métodos da literatura.

Palavras-chave— Redes Neurais, Bases Radiais, Raio, Classificação.

1 Introdução

De acordo com Haykin (2007) uma rede neural tem sua arquitetura definida pelo número de camadas, pelo número de nós em cada camada, pelo tipo de conexão entre os nós e por sua topologia. Ao contrário de uma rede MLP (*Multi layer Perceptron*) em que a função de ativação é baseada no produto do vetor de entradas e o vetor de pesos, uma rede RBF (*Radial Basis Function*) é fundamentada no cálculo da distância entre o vetor de entradas e o centro da unidade. A rede estará bem construída quando esse vetor de entradas tem ativação significativa feita por apenas algumas unidades (funções) ocultas.

Uma rede RBF possui três tipos de parâmetros de caráter distinto: centros, variâncias (larguras) e pesos da camada de saída (Haykin (1994)).

Diversos são os métodos encontrados na literatura para determinação dos centros. Pode-se defini-los de maneira aleatória ou através de *clustering*, utilizando métodos como o *k-means* proposto por Hartigan and Wong (1979), SOM (*Self Organizing Maps*) proposto por Kohonen (1998), OLS (*Orthogonal Least Squares*) proposto por Chen et al. (1991), FCM (*Fuzzy C-Means*) proposto por Bezdek et al. (1984) e algoritmos genéticos.

Para cálculo do raio, existem vários métodos, como o que utiliza a distância máxima entre os centros proposto por Haykin (1994); o que utiliza o desvio padrão de cada agrupamento, sugerido por Benoudjit et al. (2002); segundo um ajuste iterativo, proposto por Verleysen and Hlavackova (1994); e ainda a distância entre o centro do neurô-

nio e ponto médio mais próximo, proposto por Torres (2013).

O treinamento de uma rede RBF consiste em ajustar os parâmetros a partir dos dados de entrada e consequentemente encontrar os pesos da camada de saída, dado um resultado conhecido.

Após o treinamento da rede, outra fase existente é a generalização que consiste na capacidade de aproximar a função representada pelo conjunto de dados. Uma dessas formas é a interpolação exata, onde é gerada uma rede RBF com um neurônio para cada vetor de entrada, passando exatamente por todos os pontos do conjunto de dados, gerando consequentemente *overfitting* (Figura 1).

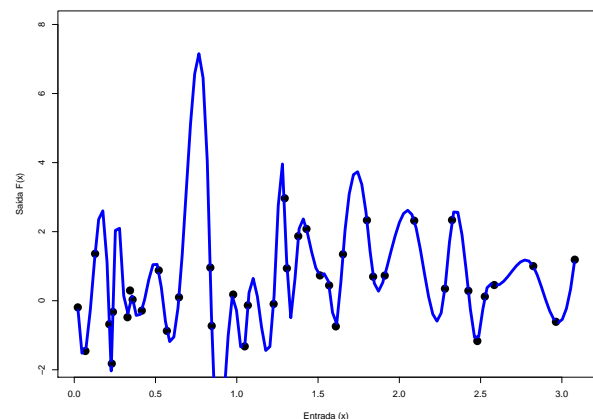


Figura 1: Overfitting

O objetivo deste trabalho é, utilizando uma rede neural baseado em Função de base radial,

apresentar um novo método para cálculo do raio da função e aplicá-lo à bases de dados muito referenciadas na literatura. Para o desenvolvimento deste trabalho utilizou-se 14 bases de dados que foram retiradas a partir dos repositórios públicos *UCI benchmark repository* (Bache and Lichman (2013)) e *Keel Data Set Repository*. Os resultados obtidos são promissores, mostrando-se equivalente a outros métodos da literatura e com menor tempo de execução.

O artigo está estruturado a partir desta introdução, e conta na segunda seção com a explicação das redes RBF's; na terceira seção são apresentados trabalhos relacionados ao tema do artigo, onde são discutidas pesquisas envolvendo o cálculo do raio das funções gaussianas; na quarta seção são detalhadas as base de dados utilizadas bem como o método proposto; na quinta seção são apresentados os resultados obtidos, bem como uma discussão sobre eles; e por fim, na sexta seção, as bibliografias consultadas.

2 Radial Basis Function

Como discutido por Braga et al. (2000), funções de base radiais representam uma classe especial de funções não-lineares cujo valor aumenta ou diminui de acordo com a distância a um ponto central.

De acordo com Rumelhart (1986), as redes de funções de base radial ou RBF's são modelos multi-camadas semelhantes aos modelos MLP que são assim distribuídas: camada de entrada, na qual os padrões são apresentados à rede; a camada intermediária (única escondida), onde são aplicadas as funções de ativação (alta dimensionalidade) e a camada de saída que utilizando uma função linear, apresenta a resposta da rede ao padrão informado.

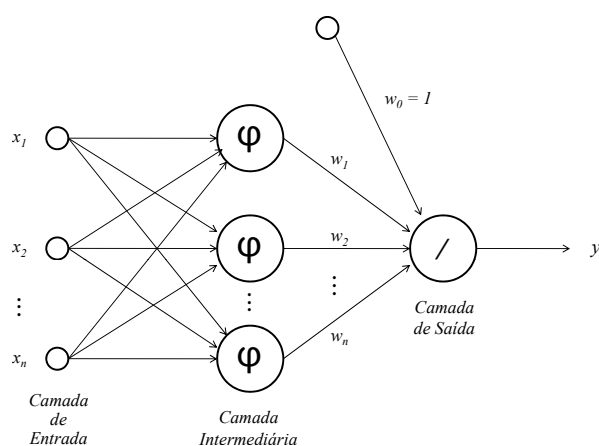


Figura 2: Rede RBF

Como na Figura 2 e na Equação (1), cada um dos n componentes do vetor de entrada x são aplicados às funções de ativação φ , cujas saídas são linearmente combinadas com pesos w_j , com j

variando de 1 a n . Para o caso de uma única saída $y = f(x)$, o mapeamento entrada-saída é dado por:

$$y = \sum_{j=1}^m w_j h_j(\vec{X}) \quad (1)$$

onde h_j representa a saída da camada escondida tendo como entrada na rede o vetor \vec{X} .

As funções de base radiais mais comuns são:

- Função de Base Logística

$$z_j(x) = \left(1 + \exp \left(\frac{\|x - \mu_j\|}{\sigma_j^2} - \theta_j \right) \right) \quad (2)$$

- Função de Base Gaussiana

$$z_j(x) = \exp \left(\frac{-\|x - \mu_j\|^2}{2\sigma_j^2} \right) \quad (3)$$

- Função Multiquadrática Inversa

$$z_j(x) = \frac{1}{\left(\sqrt[2]{\|x - \mu_j\|^2 + \sigma_j^2} \right)} \quad (4)$$

- Função *Thin-Plate-Spline*

$$z_j(x) = \|x - \mu_j\|^2 \ln(\|x - \mu_j\|) \quad (5)$$

Devido sua função de ativação na camada oculta, as redes RBF são aproximadores universais dado um número suficiente de neurônios, pois de acordo com Cover (1965), um problema de classificação de padrões que, quando é transformado para um espaço de maior dimensão é mais provável ser linearmente separável do que em espaço de menor dimensão, tornando consequentemente o problema mais simples de ser resolvido.

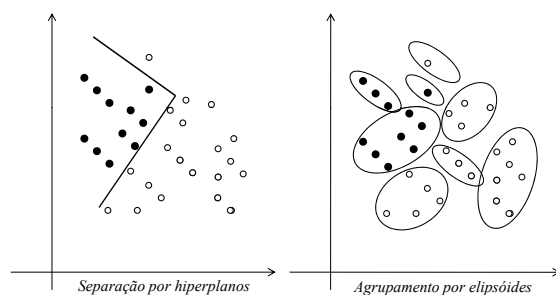


Figura 3: Classificação utilizando MLP e RBF

Como mostrado na Figura 3, ao contrário de uma rede neural MLP que utiliza hiperplanos para separar o espaço de entradas, uma rede com função de ativação RBF, por estar em um espaço de alta dimensionalidade, utiliza elipsóides para particionar esse espaço.

3 Trabalhos Relacionados

Provenientes de pesquisa bibliográfica nesta seção são apresentados 4 métodos para o cálculo do raio de cada gaussiana.

Assim como na determinação dos centros, o cálculo dos raios pode ser realizado de diversas formas, como proposto por Haykin (1994) em (6):

$$r = \frac{dmax}{\sqrt[2]{2M}} \quad (6)$$

onde M é o número de centros e $dmax$ é a distância máxima entre todos os centros.

Outro método proposto, agora por Benoudjit et al. (2002), consiste em determinar um fator escalar q heurísticamente baseado no menor erro de validação multiplicado pelo desvio padrão de cada agrupamento, como em (7):

$$r'_j = qr_j \quad (7)$$

Já Verleysen and Hlavackova (1994) propõe calcular o valor do raio segundo um ajuste iterativo, como em (8):

$$r(t+1) = (1 - \beta(t))r(t) + \beta(t)2||x_i - c(t)|| \quad (8)$$

onde β é um fator de adaptação entre 0 e 1.

Em Torres (2013), o raio do neurônio da rede é encontrado através da distância entre o centro do neurônio e o ponto médio mais próximo a ele, como mostrado em (9).

$$r = 2(\min(\delta(c_i, P_M(j))), \forall j \in P_M \quad (9)$$

4 Método Proposto

A rede foi construída utilizando-se na camada oculta a função gaussiana, representada pela Equação (3), para ativação dos neurônios.

No desenvolvimento do método proposto foi utilizado para definição dos centros da rede o método *k-means* proposto por MacQueen et al. (1967) que consiste em dado um número k de agrupamentos, as posições de seus centros são ajustadas iterativamente.

Primeiramente são iniciados os pontos de cada agrupamento e em seguida, tais pontos são atribuídos a um conjunto de dados, de forma que todos os dados pertençam ao agrupamento mais próximo.

A quantidade de centros utilizada variou para cada base, visto que atingiu-se melhores resultados ao utilizar-se quantidade de centros diferentes.

Em um processo de aprendizagem de máquina em uma rede neural, os dados são divididos em dois subconjuntos: treinamento e teste. O subconjunto de treinamento é usado para obter uma rede treinada para aquele conjunto de dados. Obtido um modelo de rede treinada, é utilizado assim

o subconjunto de teste com o objetivo de verificar a performance da rede, visto que esses dados não foram utilizados na fase anterior.

Definidos os centros, foi calculada a distância entre todos eles, obtendo-se posteriormente a média das distâncias. Com o conhecimento desta média, foi realizada a média entre este resultado e a menor distância obtida pelo respectivo centro, com o objetivo de aumentar a suavidade da gaussiana. O método proposto neste artigo é descrito matematicamente como

$$\sigma = \frac{1}{2} \left[\left(\frac{1}{k} \sum_{i=1}^k d_i \right) + \min(d_i) \right] \quad (10)$$

onde d_i representa a distância entre os centros das gaussianas.

A partir desse cálculo, aplicou-se o resultado alcançado para cada centro, à função da gaussiana, obtendo-se a saída de cada neurônio da camada escondida para os dados de treinamento. Obteve-se então o vetor de pesos, que consiste na multiplicação entre a inversa da matriz de saída da camada escondida, e a saída dos dados de treinamento y_n , conforme a Equação (11).

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} h(||x_1 - x_1||) & \dots & h(||x_1 - x_n||) \\ h(||x_2 - x_1||) & \dots & h(||x_2 - x_n||) \\ \vdots & \ddots & \vdots \\ h(||x_n - x_1||) & \dots & h(||x_n - x_n||) \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (11)$$

Foi definido como critério de parada o número máximo de iterações e nomeado o método como MM-RBF.

5 Resultados e Discussão

Para definição dos conjuntos de treinamento e teste foi utilizado o método *cross-validation* (Kohavi et al. (1995)) que divide a base de dados em k partes (*folds*). A partir dessa divisão, $k - 1$ partes são utilizadas para o treinamento e 1 parte é usada para teste. Essa operação é repetida k vezes, garantindo que todas as partes serão utilizadas como dados de teste.

5.1 Base de Dados

Foram utilizadas 14 bases de dados para realização dos testes. Tais bases, disponibilizadas por Bache and Lichman (2013), foram selecionadas em sua maioria a partir do repositório público *UCI benchmark repository*. Apenas a base de dados *Appendicitis data set* foi retirada do sítio *Keel Data Set Repository*. Estas bases de dados são referenciadas em diversas pesquisas trabalhos na literatura, o que as torna um ótimo *benchmark* para este trabalho. Foram utilizadas:

- *Appendicitis data set (append)*;
- *Banknote Authentication (bank)*;

- *Blood Transfusion (blood)*;
- *Bupa liver disorders (bupa)*
- *Connectionist Bench Sonar (sonar)*;
- *Fertility (fertil)*;
- *Haberman's Survival (haberm)*
- *Indian Liver Patient (ilpd)*;
- *Ionosphere (ionosp)*;
- *Parkinsons (parkins)*;
- *Pima Indians Diabetes (diabet)*
- *Statlog Australian Credit (austr)*;
- *Statlog Heart Disease (heart) e*
- *Wisconsin Breast Cancer (cancer)*.

Na Tabela 1, são apresentadas para cada base de dados o número de características bem como o número de amostras. Nota-se que há uma grande diversidade entre as bases utilizadas; visto que as bases variam entre 3 e 60 características, e entre 100 e 1372 amostras.

Tabela 1: Características das bases de dados

Bases	Número de características	Número de amostras
append	7	106
austr	14	690
bank	5	1372
blood	5	748
bupa	6	345
cancer	9	683
diabet	8	768
fertil	10	100
haberm	3	306
heart	13	270
ilpd	10	583
ionosp	34	351
parkins	22	195
sonar	60	208

Os experimentos foram realizados utilizando-se a estratégia de validação cruzada *k-fold cross-validation*, adotando-se $k = 10$. Foi calculado o erro de teste para cada uma das 10 execuções e utilizada a média dos erros percentuais de classificação. Este procedimento foi repetido 5 vezes para cada base, com o objetivo de reduzir ainda mais a variância.

O método desenvolvido é comparado na Tabela 2 com o RBF utilizando o menor raio, com o classificador *Support Vector Machine* (SVM) implementado utilizando o método LIBSVM proposto por Chang and Lin (2011) com *kernel* RBF, $\sigma = 1$, e método para encontrar o hiperplano de separação SMO (*Sequential Minimal*

Optimization). O três métodos comparados foram desenvolvidos e executados na mesma máquina, evitando assim diferenças computacionais.

Tabela 2: - Resultados: Médias das acurácias

Bases	RBF	SVM	MMRBF
append	0.8855	0.8691	0.8882
austr	0.5580	0.8522	0.8696
bank	1	1	1
blood	0.7741	0.7567	0.7954
bupa	0.6402	0.6870	0.7424
cancer	0.9209	0.9707	0.9341
diabet	0.7342	0.7511	0.7800
fertil	0.8800	0.8800	0.8800
haberm	0.7576	0.7548	0.7555
heart	0.8407	0.8407	0.8519
ilpd	0.7185	0.6252	0.7236
ionosp	0.6410	0.9544	0.9459
parkins	0.8466	0.9282	0.8611
sonar	0.8026	0.8512	0.8271

Em negrito estão destacados os melhores resultados para as bases testadas. Percebe-se que o algoritmo proposto obteve bons resultados em comparação com outros métodos relevantes da literatura.

Entre os resultados, destaca-se a precisão para a base *bupa* onde o método proposto neste artigo obteve em torno de 75% de acurácia, enquanto os métodos RBF e SVM obtiveram em torno de 64% e 68% respectivamente; destaca-se ainda como relevantes as acurácias nas bases *diabetes* e *ilpd* onde o método apresentou resultados consideravelmente superiores ao método SVM, assim como nas bases *austr* e *ionosp* que o método desenvolvido apresentou resultados consideravelmente superiores ao método RBF.

Tabela 3: Tempo computacional (ms)

Bases	RBF	MMRBF
append	15.52	15.30
austr	262.00	262.11
bank	361.09	401.01
blood	179.57	169.26
bupa	171.85	169.85
cancer	198.94	196.24
diabet	324.15	327.77
fertil	14.21	13.15
haberm	109.84	108.73
heart	93.89	90.54
ilpd	248.86	253.73
ionosp	343.93	342.92
parkins	73.67	73.55
sonar	125.98	125.45

A Tabela 3, apresenta uma comparação entre o tempo computacional dos métodos RBF e RBFMM. Como pode-se observar, o método proposto neste artigo obteve tempo de execução com-

putacional inferior ao método RBF em 10 das 14 bases utilizadas.

Para comparar os resultados obtidos, foi realizado o teste estatístico *Wilcoxon signed rank test* Wilcoxon (1945). O teste de *Wilcoxon* é um método não-paramétrico utilizado para comparar duas amostras não-pareadas. As hipóteses para realização do teste foram assim definidas:

- H_0 (Hipótese nula): Os classificadores são iguais;
- H_1 (Hipótese alternativa): Os classificadores são diferentes.

Como apresentado na Tabela 4, a comparação realizada pelo teste de *Wilcoxon* entre os métodos RBF e RBFMM retornou *p-value* igual a 0.01827. Sendo assim, com um nível de significância de 5% rejeita-se a hipótese de igualdade entre os dois métodos.

Ao realizar a comparação entre os método RBFMM e o método SVM percebe-se que o *p-value* é maior que 0.05. Conclui-se então, com um nível de significância de 5%, que a hipótese nula não deve ser rejeitada, ou seja, o método desenvolvido RBFMM mostrou-se equivalente ao SVM neste domínio experimental.

Tabela 4: Resultado teste Wilcoxon

Método 1	Método 2	P-value
RBFMM	RBF	0.01827926
SVM	RBF	0.54994380
SVM	RBFMM	0.21752083

6 Considerações Finais

Este trabalho apresentou um novo método para cálculo do raio de cada gaussiana, baseado na média e na menor distância entre os centros. Utilizou-se o *k-means* para definição dos centros e o *cross-validation* para treinamento e teste dos dados. Foram utilizadas bases multivariadas conhecidas da literatura, o que as torna um ótimo *benchmark* para o trabalho.

Com os resultados preliminares apresentados, comprova-se que este método apresentou-se equivalente a outros métodos da literatura. Como neste trabalho adotou-se a quantidade de *clusters* variados, em trabalhos futuros pretende-se otimizar a escolha desse número baseado na melhor acurácia, aplicá-lo a mais bases de dados, compará-lo a mais métodos existentes na literatura e aplicar diferentes métricas de desempenho, como a AUC (*Area under curve*).

7 Agradecimentos

Meus agradecimentos ao Instituto Federal de Educação, Ciência e Tecnologia do Norte de Minas Ge-

rais (IFNMG-PBQS) pelo suporte financeiro oferecido.

Referências

- Bache, K. and Lichman, M. (2013). Uci machine learning repository.
- Benoudjit, N., Archambeau, C., Lendasse, A., Lee, J. A., Verleysen, M. et al. (2002). Width optimization of the gaussian kernels in radial basis function networks., *ESANN*, Vol. 2, pp. 425–432.
- Bezdek, J. C., Ehrlich, R. and Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm, *Computers & Geosciences* **10**(2-3): 191–203.
- Braga, A. d. P., Carvalho, A. and Ludermir, T. B. (2000). *Redes neurais artificiais: teoria e aplicações*, Livros Técnicos e Científicos.
- Chang, C.-C. and Lin, C.-J. (2011). Libsvm: a library for support vector machines, *ACM transactions on intelligent systems and technology (TIST)* **2**(3): 27.
- Chen, S., Cowan, C. F. and Grant, P. M. (1991). Orthogonal least squares learning algorithm for radial basis function networks, *IEEE Transactions on neural networks* **2**(2): 302–309.
- Cover, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition, *IEEE transactions on electronic computers* (3): 326–334.
- Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm, *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **28**(1): 100–108.
- Haykin, S. (1994). Neural networks: A comprehensive foundation: Macmillan college publishing company, New York .
- Haykin, S. (2007). *Redes neurais: princípios e prática*, Bookman Editora.
- Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection, *Ijcai*, Vol. 14, Montreal, Canada, pp. 1137–1145.
- Kohonen, T. (1998). The self-organizing map, *Neurocomputing* **21**(1-3): 1–6.
- MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations, *Proceedings of the fifth Berkeley symposium on mathematical statistics*

and probability, Vol. 1, Oakland, CA, USA, pp. 281–297.

Rumelhart, D. E. (1986). Learning internal representations by error propagation, *Parallel distributed processing* **1**(8).

Torres, L.C.B. ; Lemos, A. . B. A. P. (2013). projeto de redes rbf baseado na estrutura dos dados e em informações de margem.

Verleysen, M. and Hlavackova, K. (1994). An optimized rbf network for approximation of functions.

Wilcoxon, F. (1945). Individual comparisons by ranking methods, *Biometrics bulletin* **1**(6): 80–83.