Classificação Multirrótulo com Mapas de *Kohonen* e Vizinhanças Vencedoras

J. G. M. Barbirato[1], R. Cerri[2]

Universidade Federal de São Carlos, Brazil
joaobarbirato@gmail.com, cerri@ufscar.br

Abstract. O problema convencional de classificação no contexto do aprendizado de máquina consiste em classificar exemplos de conjuntos de dados em categorias pré-definidas, de acordo com uma ou mais características semelhantes. Contudo, alguns conjuntos de dados possuem classes com intersecções, ou seja, exemplos podem pertencer a mais de uma classe simultaneamente. Exemplos desses problemas podem ser encontrados, por exemplo, na identificação de gêneros de livros e na classificação de imagens. Esses tipos de problemas são denominados multirrótulo. O objetivo deste artigo é propor um novo método de classificação multirrótulo com Mapas de *Kohonen*. A ideia é utilizar o neurônio vencedor do processo competitivo do mapa auto-organizável, juntamente com a vizinhança ao redor desse neurônio, para a classificação de dados. Assim, um novo exemplo é classificado nas classes pertencentes aos exemplos de treino mapeados para o neurônio vencedor e sua vizinhança. A linguagem *Python* e a biblioteca de aprendizado de máquina *Scikit-Learn* foram utilizadas para implementação do modelo da rede neural, para implementação das medidas de avaliação, e para a geração de conjuntos de dados sintéticos. A utilização de uma vizinhança de neurônios foi comparada com uma proposta anterior utilizando apenas um neurônio vencedor. Os resultados mostraram que a utilização de uma vizinhança ao redor do neurônio vencedor é promissora, obtendo melhores reesultados.

Categories and Subject Descriptors: I.2.6 [**Artificial Intelligence**]: Learning

Keywords: aprendizado de máquina, classificação, classification, machine learning, mapas de kohonen, multilabel, multirrótulo, neural networks, redes neurais, self-organizing maps

1. INTRODUÇÃO

Em aprendizado de máquina, a classificação convencional consiste em separar conjuntos de dados em classes, de forma a se rotular um exemplo em uma determinada classe. Para isso, pode-se utilizar, por exemplo, a semelhança dos atributos de um exemplo a ser classificado, com os atributos de outros exemplos de treinamento. Contudo, há situações nas quais apenas um rótulo não modela adequadamente o problema. É o caso da categorização de documentos, do diagnóstico médico e da classificação de imagens; um documento geralmente envolve mais de uma área de conhecimento, assim como um paciente pode sofrer de duas doenças simultaneamente. Para essas situações, utiliza-se a classificação multirrótulo, em que um novo exemplo é associado a um conjunto de classes simultaneamente.

Diversos modelos de aprendizado de máquina, na literatura, se adaptam a problemas desse tipo. Entre esses, estão as redes neurais artificiais, modelos que tentam simular o comportamento do cérebro humano para tomada de decisões. Trata-se de um modelo fortemente inspirado no processo de aprendizado humano, em que as informações passam por uma estrutura de neurônios: as sinapses transformam e processam as entradas.

Um desses modelos é o Mapa de *Kohonen* ou Mapa Auto-Organizável. Em seu uso convencional, Mapas de *Kohonen* são redes neurais estruturadas em uma superfície bidimensional de neurônios, de modo que os neurônios se auto-organizam formando regiões no mapa – determinando grupos ou

classes de exemplos. Assim, pode-se utilizar a regra dos vizinhos mais próximos para classificar um novo exemplo. O neurônio geometricamente mais próximo da entrada é denominado como "neurônio vencedor" e, evidentemente, categoriza a entrada.

Diante do exposto, o presente artigo tem por objetivo, dentro do contexto de redes neurais artificiais, utilizar neurônios de uma região do Mapa de *Kohonen* para a classificação multirrótulo de exemplos. Para isso, será estendido o trabalho proposto por [Colombini et al. 2017], a fim de que as classes atribuídas a uma entrada de teste sejam escolhidas não apenas com base em um único neurônio vencedor, mas também com base na vizinhança ao redor desse neurônio. Utilizando conjuntos de dados artificiais e reais, esse trabalho visa investigar se a utilização de uma vizinhança de neurônios ao redor do neurônio vencedor leva a melhores resultados se comparado com a utilização de apenas um neurônio vencedor.

2. CLASSIFICAÇÃO MULTIRRÓTULO

Na literatura de aprendizado de máquina, problemas de classificação convencionais são chamados de problemas simples-rótulo ou monorrótulo. Nesses problemas, um classificador é treinado em um conjunto de exemplos que estão associados com uma única classe l de um conjunto de classes disjuntas L, onde $|L| > 1$. Se $|L| = 2$, então o problema é chamado de problema de classificação binária, e se $|L| > 2$, o problema é chamado de problema de classificação multiclasse. Em uma classificação multirrótulo, os exemplos de treino estão associados a um conjunto de classes $Y \subseteq L$, sendo $|Y| > 1$ [Tsoumakas et al. 2009].

Muitos trabalhos têm sido propostos na literatura de classificação multirrótulo, mas poucos utilizam redes neurais artificiais, e especificamente mapas de *Kohonen*. [Zhang and Zhou 2006] propõe a utilização de redes neurais multicamadas para classificação multirrótulo. Os autores propuseram uma medida de erro multirrótulo para ser incorporada no algoritmo back-propagation, associando um neurônio de saída a cada classe do problema. A ideia central dessa medida de erro é considerar os erros em todos os neurônios no momento do cálculo do erro de um neurônio específico.

Em [Borges and Nievola 2012], foi desenvolvido um método utilizando uma rede neural competitiva para classificação multirrótulo em cenários onde as classes pertencem a uma hierarquia. O método segue a mesma ideia de treinamento utilizada nos mapas de *Kohonen*. A rede neural consiste de uma camada de entrada, formada pelos atributos dos exemplos, e de uma camada de saída, na qual cada neurônio corresponde a um nó da hierarquia. Assim como nos mapas de *Kohonen*, o processo de treinamento da rede consiste de três etapas: competição, cooperação e adaptação.

No trabalho de [Cerri et al. 2016], redes neurais para classificação multirrótulo foram utilizadas para classificação de funções de proteínas também em cenários hierárquicos. Bons resultados foram obtidos por meio da utilização de redes perceptron multicamadas (*Multi-layer Perceptron*), associando uma rede neural a cada nível da hierarquia [Cerri et al. 2016].

Seguindo a ideia de [Cerri et al. 2016], recentes trabalhos [Wehrmann et al. 2017; Wehrmann et al. 2018] também utilizaram redes neurais artificiais para problemas multirrótulo hierárquicos. Porém, ao invés de dividir o problema como feito por [Cerri et al. 2016] otimizando funções locais, os autores propuseram arquiteturas para otimização de funções tanto locais quanto globais (considerando todas as classes simultaneamente).

O trabalho proposto por [Colombini et al. 2017] inspira grande parte do presente artigo. Naquele trabalho, os autores demonstraram que a tarefa de classificação multirrótulo utilizando Mapas de *Kohonen* é promissora, mesmo trazendo resultados de classificação utilizando apenas o neurônio vencedor. Assim, neste trabalho propomos uma extensão do método de [Colombini et al. 2017], utilizando, além do neurônio vencedor, uma vizinhança ao redor desse neurônio.

2.1 Medidas de avaliação

Medidas de avaliação convencionais não são adequadas para problemas multirrótulo, pois uma classificação pode ser parcialmente correta. Assim, seguiu-se o trabalho proposto por [Godbole and Sarawagi 2004].

Seja C um classificador multirrótulo, com $L_i = H(x_i)$ o conjunto de rótulos previstos por C para um exemplo x_i; Y_i o conjunto de classes verdadeiras, e X o conjunto de exemplos. Assim, [Godbole and Sarawagi 2004] propõem as medidas de precisão (*precision*) e revocação (*recall*), conforme as Equações 1 e 2; medidas estas utilizadas no presente artigo.

$$\text{precisão}\,(C, X) = \frac{1}{|X|}\sum_{i=1}^{|X|}\frac{|Y_i \cap L_i|}{|L_i|} \tag{1}$$

$$\text{revocação}\,(C, X) = \frac{1}{|X|}\sum_{i=1}^{|X|}\frac{|Y_i \cap L_i|}{|Y_i|} \tag{2}$$

As medidas de precisão e revocação podem ser combinadas da medida-F (*F-measure*), apresentada na Equação 3. Essa medida também é utilizada nos experimentos desse trabalho.

$$\text{medidaf}\,(C, X) = 2 \times \frac{\text{precisão}\,(C, X) \times \text{revocação}\,(C, X)}{\text{precisão}\,(C, X) + \text{revocação}\,(C, X)} \tag{3}$$

3. MAPAS DE *KOHONEN*

Mapas auto-organizáveis, SOM (*Self-Organizing Maps*) ou Mapas de *Kohonen* são modelos de redes neurais cujos neurônios artificiais se dispõem em uma grade usualmente bidimensional e são interligados. O caráter auto-organizável desse modelo torna seu aprendizado não supervisionado, por padrão. Seu desenvolvimento como um modelo de redes neurais é motivado biologicamente porque sinais de entradas semelhantes estimulam neurônios de uma mesma região e de maneira ordenada. Além disso, ocorre o denominado aprendizado competitivo, visto que os neurônios competem entre si para serem ativados. Os neurônios sintonizados - denominados de neurônios vencedores - tornam-se ordenados com relação uns aos outros. Assim, a localização espacial dos neurônios torna-se indicativo de características contidas no padrão de entrada, formando um mapa topográfico dos padrões de entrada, agrupando neurônios que mapeiam entradas com características semelhantes. Cada neurônio do mapa possui um vetor de pesos que está relacionado às características dos dados de entrada. A Figura 1 ilustra um Mapa de *Kohonen* recebendo um novo exemplo para mapeamento.

Para os procedimentos de treinamento e de teste, o neurônio vencedor é geralmente dado pela distância Euclidiana entre o exemplo de entrada e os pesos dos neurônios. O neurônio de menor distância Euclidiana é denominado como vencedor. A Equação 4 representa o cálculo da distância Euclidiana entre o neurônio k com o exemplo x_i, onde N representa o número de características (atributos) do exemplo entrada e w_k representa o vetor de pesos do neurônio k.

$$d_k(x) = \sqrt{\sum_{i=1}^{N}(x_i - w_{ki})^2} \tag{4}$$

Quando um exemplo é mapeado para um neurônio, os pesos do mesmo são ajustados de maneira que exemplos semelhantes sejam mapeados para a mesma região do mapa. A vizinhança do neurônio

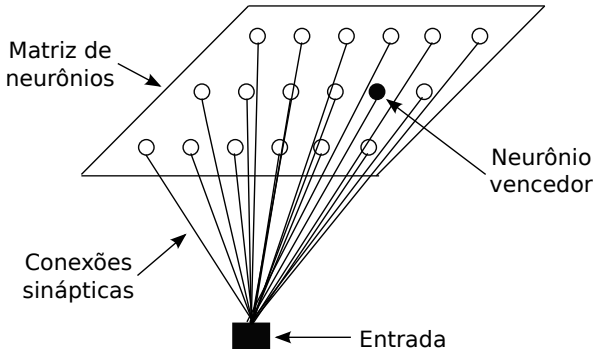

Fig. 1. Mapa de *Kohonen* com grade bidimensional de neurônios recebendo um exemplo de entrada. Percebe-se que todos os neurônios da grade se interligam com o exemplo.

vencedor também tem os pesos ajustados de modo que se crie uma vizinhança de neurônios semelhantes. Após a determinação do neurônio vencedor, seus pesos precisam ser aproximados à entrada, e isso se dá utilizando uma função de vizinhança. É razoável escolher a função Gaussiana, representada na Equação 5, onde h_{ji} representa a vizinhança do neurônio vencedor i formado pelos neurônios excitados j; d representa a distância Euclidiana calculada do neurônio vencedor; σ é um parâmetro que define a largura da função, utilizado no contexto do Mapa de *Kohonen* comumente por um valor da função de aprendizado.

$$h_{ij} = e^{\frac{-d^2}{2\sigma^2}} \tag{5}$$

Assim, o ajuste dos pesos Δw do neurônio j dado um exemplo x e uma taxa de aprendizado η é calculado de acordo com a Equação 6. Dessa forma, o processo de treinamento se repete com todos os exemplos. Evidentemente, é esperado que quanto mais dados de treinamento a rede receba como entrada, melhor ajustados serão os pesos e mais precisos serão os resultados.

$$\Delta w_j = \eta h_{ji}(x - w_j) \tag{6}$$

Para a tarefa de classificação, as classes de todos os exemplos são representadas por um vetor binário. Neste, a posição j corresponde à j-ésima classe. Se um exemplo é classificado na j-ésima classe, a posição j do vetor recebe o valor 1, e 0 caso contrário. Cada neurônio n possui S_n vetores binários de classe, significando que S_n exemplos de treino foram mapeados para esse neurônio, ou seja, ele é o neurônio vencedor para esses S_n exemplos.

Para a classificação de um exemplo de teste, o mesmo é mapeado para o mapa de neurônios. Após identificado o neurônio vencedor n, a classificação desse exemplo se dá por um vetor binário $\overline{\mathbf{v}}_n$ cujos valores são as médias aritméticas das respectivas classes dos S_n vetores binários de treino associados ao neurônio vencedor, como apresentado na Equação 7. Nessa equação, S_{nj} representa o conjunto de exemplos de treino mapeados para o neurônio n que pertencem à j-ésima classe do problema.

$$\overline{v}_{nj} = \frac{|S_{nj}|}{S_n} \tag{7}$$

Após a obtenção do vetor $\overline{\mathbf{v}}_n$, um limiar precisa ser utilizado para que seja obtida a classificação final binária. Assim, utilizando por exemplo um limiar de valor 0, todas as posições cujos valores são maiores ou iguais a 0 recebem o valor 1, e 0 caso contrário.

4. MÉTODO PROPOSTO UTILIZANDO UMA VIZINHANÇA DE NEURÔNIOS

Neste artigo, pretende-se explorar os vetores binários não só do neurônio vencedor, mas também da sua vizinhança. Sendo H_n o conjunto formado pelo neurônio vencedor n e seus neurônios vizinhos, a Equação 7 pode ser facilmente reescrita de acordo com a Equação 8, agora considerando uma vizinhança de neurônios ao redor do neurônio vencedor.

$$\overline{v}_{nj} = \frac{\left[\sum_{k \in H_n}^{H_n} |S_{kj}|\right]}{|H_n|} \tag{8}$$

A Equação 8 calcula a média entre todos os vetores binários do neurônio vencedor e seus vizinhos para classificar um novo exemplo, estabelecendo, assim, uma "vizinhança vencedora". A Figura 2 ilustra o processo de classificação utilizando o método proposto. É importante ressaltar que os vetores binários usados no cálculo são os vetores de classes dos exemplos de treino que foram mapeados para a rede durante o processo de treinamento. Para uma classificação binária final, um limiar deve ser utilizado.

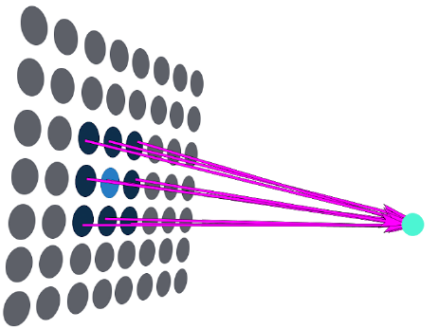

Fig. 2. Ilustração do processo de classificação utilizando o neurônio vencedor e seus 1-vizinhos mais próximos (Fonte: autoral).

5. MODELO *BASELINE*

Alguns trabalhos relacionados a aprendizado de máquina e redes neurais [Mao et al. 2016] mostram que o uso de um modelo *baseline* para comparar resultados traz conclusões mais evidentes. Por isso, adotou-se o modelo de classificação proposto por [Colombini et al. 2017] como *baseline*, já que o presente artigo propõe uma extensão para o trabalho citado.

O Algoritmo 5.1 apresenta o método de classificação proposto por [Colombini et al. 2017]. Trata-se do procedimento executado após o treinamento da rede. Nota-se que há apenas um neurônio vencedor escolhido para determinar a classe de um exemplo de teste. Assim, nossa proposta consiste na modificação do cálculo do vetor protótipo $\overline{\mathbf{v}}_j$ utilizando a Equação 8.

6. EXPERIMENTOS

Os experimentos consistiram em adaptar a implementação de [Colombini et al. 2017] para utilizar os neurônios vizinhos do neurônio vencedor - e os exemplos por eles rotulados - para classificar um novo

Algorithm 5.1: Procedimento de Classificação da rede SOM multirrótulo

Função: classificação-SOM-Multirrótulo(X,e)
Entrada $X^{treino} = [q, (a + l)]$*: conjunto de dados com q exemplos, a atributos e l rótulos*
$X^{test} = [m, a]$*: conjunto de dados com m exemplos e a atributos*
$W = [n, a]$*: matriz de pesos com n neurônios e a pesos*
Saida $P = [m, q]$*: matriz de predição com m linhas e l colunas*

para $j \leftarrow 1$ **ate** m `// Seleciona o neurônio vencedor da grade de neurônios` Ω
$o(\mathbf{x}_j^{teste}) = argmin_k ||\mathbf{x}_j^{test} - \mathbf{w}_k || k \in \Omega$;
`// Recupera os exemplos de treino mapeados no neurônio vencedor`
$T \leftarrow$exemplo mapeados em $o(\mathbf{x}_j^{teste})$;
`// Recupera o vetor protótipo`
$\overline{\mathbf{v}}_j \leftarrow$ média dos vetores de classes de T;
`// Associa o vetor protótipo ao exemplo`
$\mathbf{x}_j^{teste*} \leftarrow \mathbf{x}_j^{teste} + \overline{\mathbf{v}}_j$;
$\mathbf{p}_j \leftarrow \overline{\mathbf{v}}_j$;
retorne $\{P\}$;

exemplo de teste. Para isso, implementou-se o modelo da rede neural, as medidas de avaliação, e os método de treino e de teste na linguagem *Python* utilizando a biblioteca de aprendizado de máquina *Scikit-Learn* [Pedregosa et al. 2011]. Essa biblioteca também foi utilizada para a geração de conjuntos de dados multirrótulo sintéticos.

A biblioteca *Scikit-learn* fornece um módulo para criação de conjuntos de dados para classificação multirrótulo. Podem ser fornecidos parâmetros como o número de exemplos da base de dados, o número de características de cada exemplo, e o número médio de rótulos de cada exemplo. Assim, foram gerados conjuntos de dados artificiais com diferentes características. Além disso, foram utilizadas as bases de dados reais *cal500*, *emotions* e *flags*, disponíveis em `http://mulan.sourceforge.net/datasets-mlc.html`. A Tabela I exibe as características de cada base de dados. Nota-se que foram geradas nove variações para uma base de dados sintética, combinando número de atributos e número de rótulos (classes).

Base de dados	Número de exemplos	Número de atributos	Número de rótulos
Sintética	150	5, 15, 30	3, 5, 10
cal500	502	64	174
emotions	593	72	6
flags	194	14	12

Table I. Características de cada base de dados utilizada.

As classes de cada exemplo são representadas por um vetor binário de n posições, sendo n o número total de classes do conjunto de dados. Para os experimentos, foi utilizada a estratégia de validação cruzada *10-fold cross-validation* para separar a base de dados em conjuntos de treino e teste.

O Mapa de *Kohonen* implementado é o de uma *Self-Organizing Map* padrão, como descrito na Seção 3. Tanto o método de classificação *baseline* quanto a nova proposta foram implementados utilizando a linguagem *Python* e a biblioteca *Scikit-Learn* .

Como parâmetros da rede, utilizou-se uma taxa de aprendizado inicial de 0.1, com decaimento entre épocas de treinamento seguindo a Equação 9, e coeficiente de decaimento igual a 1000; como função de vizinhança, utilizou-se a gaussiana apresentada na Equação 4. Além disso, utilizou-se $\sigma = 1000$ como coeficiente de decaimento da função de vizinhança.

Para o cálculo da matriz p_j na Equação 8, a matriz de saída do Algoritmo 5.1 foi normalizada para conter valores no intervalo $[-1, 1]$. Assim, foi possível a obtenção de uma melhor separação entre as

classes, permitindo a aplicação de um limiar de valor 0 para coverter o vetor de médias v_j em um vetor binário com valores 1 e 0. Todas as posições que continham valores iguais ou maiores que 0 recebiam o valor 1, e 0 caso contrário.

$$\phi(epoca+1) = \phi(epoca) \times e^{\left(\frac{epoca \times \log \phi_0}{1000}\right)} \tag{9}$$

Para resultados relacionados ao classificador proposto, variou-se o tamanho da vizinhança entre 1, 2 e 3 vizinhos mais próximos do neurônio vencedor. Tanto para o método baseline, como para o método proposto, foram aplicadas as variações apresentadas na Tabela II, com exceção da largura da vizinhança, que para o baseline é sempre 0. Assim, foram comparadas duas variações do baseline e seis variações do método proposto.

Parâmetros	Variações
Dimensão da grade	5x5 e 10x10
Largura da vizinhança	0 (*baseline*) 1, 2 e 3 vizinhos mais próximos

Table II. Variações de parâmetros utilizados nos experimentos. Todas as permutações foram utilizadas para a apuração dos resultados.

7. RESULTADOS

O resultado da implementação de todos os módulos e das tabelas completas geradas como resultados encontram-se em `https://github.com/joaobarbirato/python-SOM-MLL/`. A Tabela III mostra as medidas-f em cada base de dados - cal500, emotions, flags e os nove conjuntos de dados gerados sinteticamente.

	Baseline				Proposto			
Base de dados	5x5	10x10	k=1 5x5	k=1 10x10	k=2 5x5	k=2 10x10	k=3 5x5	k=3 10x10
Sintet.: 3 classes, 5 atr.	0,538 ± 0,069	0,447 ± 0,220	0,499 ± 0,160	0,516 ± 0,088	0,466 ± 0,130	0,557 ± 0,048	0,469 ± 0,095	0,591 ± 0,100
Sintet.: 5 classes, 5 atr.	0,354 ± 0,120	0,429 ± 0,076	0,500 ± 0,120	0,422 ± 0,120	0,541 ± 0,050	0,453 ± 0,140	0,529 ± 0,058	0,493 ± 0,064
Sintet.: 10 classes, 5 atr.	0,272 ± 0,071	0,262 ± 0,073	0,321 ± 0,078	0,292 ± 0,060	0,331 ± 0,064	0,321 ± 0,074	0,318 ± 0,091	0,329 ± 0,051
Sintet.: 3 classes, 15 atr.	0,444 ± 0,150	0,371 ± 0,110	0,433 ± 0,100	0,456 ± 0,080	0,363 ± 0,150	0,516 ± 0,110	0,470 ± 0,087	0,534 ± 0,088
Sintet.: 5 classes, 15 atr.	0,387 ± 0,092	0,416 ± 0,086	0,344 ± 0,120	0,461 ± 0,150	0,418 ± 0,073	0,439 ± 0,100	0,559 ± 0,100	0,477 ± 0,083
Sintet.: 10 classes, 15 atr.	0,264 ± 0,077	0,238 ± 0,060	0,290 ± 0,057	0,282 ± 0,069	0,277 ± 0,054	0,259 ± 0,073	0,283 ± 0,040	0,318 ± 0,090
Sintet.: 3 classes, 30 atr.	0,491 ± 0,098	0,444 ± 0,097	0,395 ± 0,160	0,492 ± 0,130	0,401 ± 0,140	0,501 ± 0,090	0,499 ± 0,140	0,526 ± 0,110
Sintet.: 5 classes, 30 atr.	0,394 ± 0,130	0,337 ± 0,140	0,456 ± 0,036	0,381 ± 0,077	0,478 ± 0,079	0,421 ± 0,098	0,491 ± 0,120	0,466 ± 0,100
Sintet.: 10 classes, 30 atr.	0,254 ± 0,055	0,299 ± 0,077	0,312 ± 0,038	0,285 ± 0,061	0,313 ± 0,060	0,329 ± 0,051	0,320 ± 0,076	0,319 ± 0,071
cal500	0.257 ± 0,020	0,242 ± 0,020	0,257 ± 0,040	0,260 ± 0,020	0,234 ± 0,010	0,246 ± 0,020	0,245 ± 0,020	0,254 ± 0,020
emotions	0,424 ± 0.070	0,374 ± 0,080	0,433 ± 0,030	0,441 ± 0,060	0,414 ± 0,070	0,353 ± 0,050	0,386 ± 0,040	0,387 ± 0,050
flags	0,458 ± 0,060	0,440 ± 0,050	0,452 ± 0,050	0,464 ± 0,070	0,426 ± 0,080	0,422 ± 0,110	0,436 ± 0,060	0,378 ± 0,070

Table III. Medidas-F entre os exemplos de cada base de dados utilizando o método *baseline* e o método proposto de classificação para três diferentes camadas de neurônios vizinhos.

Como pode ser observado na Tabela III, a utilização de uma vizinhança de neurônios levou a melhores resultados se comparado ao modelo baseline. Fica claro também que a escolha correta do tamanho da vizinhança deve ser combinada com a escolha correta dos valores dos outros parâmetros.

8. CONSIDERAÇÕES FINAIS

Pelos experimentos pode-se verificar que, na maioria dos casos, a utilização de uma vizinhança de neurônios levou a um melhor desempenho se comparado a utilização de apenas um neurônio vencedor. Assim, a proposta da arquitetura de rede neural se faz adequada para utilização em problemas de classificação multirrótulo.

Sugere-se, para trabalhos futuros, testar os impactos de alguns parâmetros no resultado final. Por limitações do presente estudo, não foram exploradas diversas possibilidades de funções de vizinhança ou funções de aprendizado; não foram medidos os tempos de processamento de cada método, limitando

análises de desempenho; não foi testado o uso de camadas mais extensas que 3 vizinhos mais próximos para a determinação da classe. Além disso, os resultados mostram que é necessário explorar a relação da largura da vizinhança vencedora com outros parâmetros, como o número de exemplos na base de dados.

Essas análises adicionais têm potencial de serem exploradas em trabalhos futuros, para que se possa inferir mais profundamente no modelo proposto.

REFERENCES

Borges, H. and Nievola, J. Multi-label hierarchical classification using a competitive neural network for protein function prediction. In *International Joint Conference on Neural Networks*. pp. 1–8, 2012.

Cerri, R., Barros, R. C., P. L. F. de Carvalho, A. C., and Jin, Y. Reduction strategies for hierarchical multi-label classification in protein function prediction. *BMC Bioinformatics* 17 (1): 373, Sep, 2016.

Colombini, G. G., de Abreu, I. B. M., and Cerri, R. A self-organizing map-based method for multi-label classification. In *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, pp. 4291–4298, 2017.

Godbole, S. and Sarawagi, S. Discriminative methods for multi-labeled classification. In *Pacific-Asia conference on knowledge discovery and data mining*. Springer, pp. 22–30, 2004.

Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A. L., and Murphy, K. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 11–20, 2016.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* vol. 12, pp. 2825–2830, 2011.

Tsoumakas, G., Katakis, I., and Vlahavas, I. Mining multi-label data. In *Data mining and knowledge discovery handbook*. Springer, pp. 667–685, 2009.

Wehrmann, J., Barros, R. C., Dôres, S. N. d., and Cerri, R. Hierarchical multi-label classification with chained neural networks. In *Proceedings of the Symposium on Applied Computing*. SAC '17. ACM, New York, NY, USA, pp. 790–795, 2017.

Wehrmann, J., Cerri, R., and Barros, R. Hierarchical multi-label classification networks. In *Proceedings of the 35th International Conference on Machine Learning*, J. Dy and A. Krause (Eds.). Proceedings of Machine Learning Research, vol. 80. PMLR, Stockholmsmässan, Stockholm Sweden, pp. 5225–5234, 2018.

Zhang, M.-L. and Zhou, Z.-H. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE transactions on Knowledge and Data Engineering* 18 (10): 1338–1351, 2006.