Uso de redes neurais para classificação multirrótulos

Pedro Mariano Sousa Bezerra , João Victor Calvo Fracasso

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, 13083-970 – Campinas, SP, Brasil

Abstract – This article presents an investigation of the performance of a Multilayer Perceptron Neural Network in the task of Multi-label classification, trained with the Backpropagation algorithm and a modified version, Backpropagation Multilabel Learning algorithm proposed in [11], which seeks to explore the relationship dependency between labels whithin a data sample. The results for the Yeast Data Set were compared with the methods proposed in [8], and show that Neural Networks performed better in the task.

Keywords – multitask learning, multilabel problem, neural network

1. Introdução

Este trabalho tem como objetivo analisar o desempenho de redes neurais MLP (*Multilayer Perceptron*) em tarefas de classificação multirrótulo. Nos dias de hoje, a demanda por este tipo de tarefa tem aumentado. Exemplos de aplicação podem ser encontrados na classificação de funções proteicas, categorização de músicas, diagnósticos médicos e classificação de textos. Por exemplo, um artigo de jornal contendo uma crítica de um filme de cunho religioso pode ser rotulado nas categorias *Religião* e *Filmes*.

Em problemas de classificação multirrótulo, lida-se com conjuntos de dados onde cada amostra possui múltiplos rótulos associados. Como as amostras possuem números diferentes de rótulos, o problema se torna desafiante para muitos classificadores tradicionais. Diversos métodos foram propostos na literatura para abordar este problema, notavelmente os que utilizam conceitos de aprendizado multitarefa, para tirar proveito das informações obtidas ao se treinar um classificador para cada rótulo. Neste trabalho, iremos abordar um método de transformação do problema original proposto em [12], que consiste em treinar uma rede neural com uma camada intermediária por meio de uma versão modificada do algoritmo *Backpropagation*, denominada *Backpropagation Multilabel Learning* - BP-MLL. Este algoritmo busca obter os pesos sinápticos dos neurônios explorando as correlações entre os diferentes rótulos para cada amostra de dados. Utilizaremos um conjunto de dados conhecido na literatura, e iremos analisar os resultados da rede neural treinada com o BP-MLL e compará-los com os resultados obtidos por outros métodos.

Este trabalho está organizado da seguinte maneira: a Seção 2 apresenta o problema de classificação multirrótulo; a Seção 3 contém uma breve abordagem do conceito de aprendizado muititarefa; a metodologia *Backpropagation Multilabel Learning* utilizada no trabalho é apresentada na Seção 4; na Seção 5, é discutido um estudo de caso utilizando a metodologia proposta; por fim, a Seção 6 apresenta as conclusões acerca dos resultados obtidos no trabalho.

2. Problema de classificação multirrótulo

Como descrito em [11], em problemas tradicionais de classificação, cada amostra de um conjunto de dados é associada a um único rótulo l de um conjunto de rótulos L. Já em problemas de classificação multirrótulo, cada amostra de dados é associada a um subconjunto de rótulos $Y \subseteq L$. Amostras diferentes não possuem necessariamente o mesmo número de rótulos. O objetivo é obter um classificador a partir de um conjunto de dados conhecidos e submetê-lo a novos dados, desconhecidos até então, a fim de determinar corretamente seus subconjuntos de rótulos associados.

Como descrito em [11], existem duas principais abordagens para o problema de classificação multirrótulo, apresentadas a seguir.

Métodos de transformação do problema: Uma possível abordagem para este problema é conceber um classificador diferente para cada rótulo existente, transformando o problema multirrótulo em múltiplos problemas binários de classificação, independentes entre si. No entanto, esta técnica não explora a correlação existente entre os diferentes rótulos, podendo apresentar resultados ruins. Assim, conceitos de aprendizado multitarefa devem ser empregados com o intuito de obter um classificador de melhor desempenho. Esta é a abordagem utilizada para o treinamento de redes neurais usando o algoritmo BP-MLL.

Métodos de adaptação de algoritmos: O outro tipo de abordagem consiste em adaptar um algoritmo de aprendizado de máquina existente para a tarefa de aprendizado de vários rótulos. Como exemplo, podemos citar os algoritmos Adaboost.MH e Adaboost.MR propostos em [9], que são extensões para classificação multirrótulo do algoritmo AdaBoost [5].

3. Aprendizado multitarefa

Aprendizado multitarefa (MTL - *Multitask Learning*), segundo [2], é um mecanismo de transferência indutivz cujo objetivo principal é melhorar a capacidade de generalização. Isto é atingido quando se aproveitam informações específicas do problema contidas no processo de treinamento de tarefas relacionadas, ao realizar em paralelo o treinamento destas tarefas usando uma representação compartilhada.

Para o caso específico de redes neurais, dentro da abordagem de métodos de transformação do problema, podemos resolver um problema de Classificação Multirrótulo de diversas formas. Em uma delas, denominada abordagem *Single Task Learning* (STL), cada rede neural possui os mesmos dados de entrada e uma saída diferente, correspondente a cada rótulo, e seu processo de treinamento é feito de forma independente. Como as redes não são conectadas, não é possível que o que foi aprendido por uma rede seja aproveitado pelas outras. Já na abordagem MTL, uma única rede neural é treinada, com o número de saídas igual ao número de rótulos. Todas as saídas se conectam às mesmas camadas intermediárias da rede, que elas compartilham. O treinamento é feito de forma paralela para todas as saídas, e assim é possível que representações internas que surgem nas camadas intermediárias para uma dada tarefa sejam aproveitadas por outras.

No âmbito de aprendizado de máquina, existem outras áreas com alta correlação com aprendizado multitarefa, que incluem regressão de múltiplas e transferência de aprendizado, além de classificação multirrótulo abordada neste trabalho.

4. Backpropagation Multilabel Learning

O algoritmo de Retropropagação do Erro para Aprendizado Multirrótulo (*Backpropagation Multilabel Learning* - BP-MLL) foi proposto em [12]. A ideia principal consiste em redefinir o funcional de erro E a ser minimizado durante o processo de treinamento da rede neural. A arquitetura da rede neural proposta é composta de duas camadas de neurônios com função de ativação não-linear (neste caso, tangente hiperbólica): uma camada intermediária e uma camada de saída. Assim, é proposto um novo funcional de erro E que busque explorar as correlações existentes entre os diferentes rótulos associados a uma amostra de dados:

$$E = \sum_{i=1}^{m} E_i = \sum_{i=1}^{m} \frac{1}{|Y_i||\overline{Y_i}|} \sum_{(k,l)\in Y_i \times \overline{Y_i}} exp(-(c_k^i - c_l^i)) \quad (1)$$

onde m é o número de amostras de dados, E_i é o erro da i-ésima amostra, $Y_i \subseteq L$ é o seu conjunto de rótulos associados, $\overline{Y_i}$ é o conjunto de rótulos de L ausentes na amostra e c_k^i é a k-ésima saída da rede neural em resposta ao padrão de entrada i. O termo $c_k^i - c_l^i$ mede a diferença entre as saídas da rede neural para um rótulo k associado à amostra i e para um rótulo l ausente nesta amostra. Quanto maior a diferença, melhor o desempenho. A exponencial negativa penaliza severamente a situação em que $c_k^i \ll c_l^i$. Para a i-ésima amostra, a soma desses termos leva em conta a diferença acumulada entre as saídas de todos os pares de rótulos com um deles pertencentes a Y_i e o outro a $\overline{Y_i}$, normalizada pelo número total de combinações $|Y_i||\overline{Y_i}|$. Desta forma, as correlações entre os diferentes rótulos dessa amostra são devidamente exploradas, ou seja, a rede deve apresentar maiores valores na saída para rótulos pertencentes a Y_i do que para rótulos de $\overline{Y_i}$.

Para o treinamento, é proposto o método do gradiente com passo de ajuste α fixo. Seja ν_{hs} o peso sináptico da conexão entre o s-ésimo neurônio da camada intermediária e a h-ésima entrada, e ω_{sj} o peso sináptico da conexão entre o s-ésimo neurônio da camada intermediária e o j-ésimo neurônio da camada de saída. É adicionado um termo de regularização ao funcional de erro E da equação 1, para controlar a norma do vetor de pesos. Assim, o funcional $J(\boldsymbol{\theta})$ ser minimizado durante o treinamento da rede neural é dado por:

$$\min J(\boldsymbol{\theta}) = E + \frac{C}{2}\|\boldsymbol{\theta}\|_2^2 \quad (2)$$

onde $\boldsymbol{\theta}$ é o vetor contendo todos os pesos da rede e C é uma constante a definir. As direções de ajuste

$\Delta\nu_{hs}$ e $\Delta\omega_{sj}$ para os pesos são proporcionais ao oposto do gradiente do funcional:

$$\Delta\omega_{sj} = \alpha d_j b_s - C\omega_{sj} \tag{3}$$

$$\Delta\nu_{hs} = \alpha e_s a_h - C\nu_{hs} \tag{4}$$

onde b_s é a saída do s-ésimo neurônio da camada intermediária, a_h é o h-ésimo atributo de entrada ($b_s = a_h = 1$ para os pesos correspondentes aos *bias*), e d_j e e_s são dados pelas expressões a seguir:
$d_j =$

$$\begin{cases} \left(\frac{1}{|Y_i||\overline{Y_i}|}\sum_{l\in\overline{Y_i}} exp(-(c_j^i - c_l^i))\right)(1-(c_j^i)^2), \\ \text{se } j \in Y_i \\ \left(-\frac{1}{|Y_i||\overline{Y_i}|}\sum_{k\in Y_i} exp(-(c_k^i - c_j^i))\right)(1-(c_j^i)^2), \\ \text{se } j \in \overline{Y_i} \end{cases} \tag{5}$$

$$e_s = \left(\sum_{j=1}^{|L|} d_j w_{sj}\right)(1+b_s)(1-b_s) \tag{6}$$

A dedução das expressões 3 a 6 pode ser encontrada em [12].

Finalizado o treinamento, é necessário obter as métricas de avaliação para cada classificador obtido. Assim, é necessário determinar, para cada amostra de dados $\mathbf{x_i}$, o limiar $t(\mathbf{x_i})$ dos valores das saídas da rede que servirá de referência para determinar o conjunto de rótulos associados a essa amostra, ou seja, $Y_i = \{j | c_j^i > t(\mathbf{x_i}), j \in L\}$. Adotamos o mecanismo de aprendizado de limiares (*Threshold Learning Mechanism*), proposto em [3], para determinar seus valores. A técnica consiste em aproximar $t(\mathbf{x})$ por um modelo linear $t(\mathbf{x}) = \mathbf{w}^T \cdot c(\mathbf{x}) + b$, onde $c(\mathbf{x})$ é o vetor contendo as saídas da rede para a amostra $\mathbf{x}$. Primeiramente, determina-se para cada amostra $\mathbf{x_i}$ de dados de treinamento o valor $t(\mathbf{x_i})$ dado por:

$$t(\mathbf{x_i}) = \underset{t}{\text{argmin}}(|\{k | k \in Y_i, c_k^i \leq t\}| + |\{l | l \in \overline{Y_i}, c_l^i \geq t\}|) \tag{7}$$

que corresponde ao limiar que minimiza as classificações incorretas. Quando o valor mínimo desta expressão não é único, escolhe-se o valor médio dos valores ótimos. Assim, os parâmetros do modelo de $t(\mathbf{x})$ podem ser obtidos resolvendo a equação matricial $\Phi \cdot \mathbf{w'} = \mathbf{t}$, onde Φ é uma matriz cuja i-ésima linha é $(c_1^i, ..., c_{|L|}^i, 1)$, $\mathbf{w'}$ é um vetor de dimensão $|L|+1$ tal que $\mathbf{w'} = (\mathbf{w}, b)$ e $\mathbf{t}$ é o vetor contendo os valores dos limiares $t(\mathbf{x_i})$ para cada amostra $\mathbf{x_i}$. Neste trabalho, aplicamos o método dos mínimos quadrados para resolver a equação acima. Para determinar o limiar $t(\mathbf{x})$ de uma amostra de dados de teste $\mathbf{x}$, calcula-se a saída da rede $c(\mathbf{x})$ em resposta a esta amostra, e então $t(\mathbf{x})$ é obtido a partir do modelo encontrado $t(\mathbf{x}) = \mathbf{w}^T \cdot c(\mathbf{x}) + b$.

5. Estudo de Caso

Vamos analisar o desempenho de redes neurais em tarefas de classificação multirrótulo trabalhando com o banco de dados *Yeast*, disponível em [1], o qual é bem conhecido na literatura ([3] [7]) de classificação multirrótulo. O conjunto de dados *Yeast* é formado por dados de fungos da espécie *Saccharomyces cerevisiae*, popularmente conhecidos como levedura de cerveja. Cada amostra possui 103 atributos contendo informações sobre expressões gênicas de microarranjos e perfis filogenéticos, e o conjunto possui 1500 amostras de genes no conjunto de treinamento e 917 no conjunto de teste. Cada gene está associado a um conjunto de 14 rótulos hierárquicos que identifica as suas classes funcionais. Este conjunto de dados é reconhecidamente difícil de ser tratado, como apontado em [3]. Neste trabalho, não iremos explorar diretamente as relações de hierarquia entre os rótulos: tais relações serão descobertas indiretamente pelas técnicas de *Multitask Learning* aqui empregadas.

Sendo assim, executamos o algoritmo BP-MLL para treinar uma rede neural com o conjunto de dados aqui descrito. Os pesos sinápticos das conexões de todos os neurônios são inicializados de forma aleatória. Os parâmetros utilizados são os sugeridos em [12]: 20 neurônios na camada intermediária, número máximo de épocas igual a 100, parâmetro de regularização $C = 0,1$ e taxa de aprendizagem $\alpha = 0,05$. Utilizamos validação cruzada com 10 pastas, formadas pela divisão em 10 da junção dos conjuntos originalmente destinados a treinamento e teste. Em cada execução, uma pasta é separada para teste e as demais são utilizadas no treinamento. Não trabalhamos com conjunto de validação pois a rede é regularizada.

Para fim de comparação, treinamos uma rede neural com a mesma arquitetura e parâmetros com o algoritmo *Backpropagation* padrão, utilizando o *toolbox*

disponibilizado em [4]. Também apresentamos os resultados de outras quatro metodologias presentes na literatura [10]:
Binary Relevance (BR), transforma o problema de classificação multirrótulo com L rótulos em um problema de classificação binário, com L classificadores independentes e tendo a saída sendo a união dos L classificadores;
Classifier chains (CC), utiliza a regra de *Bayesian chain*, onde, dado um conjunto com L rótulos, são treinados L classificadores, sendo que o primeiro classificador é treinado somente com os padrões de entrada, e para os demais são utilizados os padrões de entrada e os classificadores já treinados;
Label Powerset (LP), transforma o problema multirrótulo em um problema multiclasse, onde é atribuída uma classe para cada combinação de rótulos existentes no conjunto de rótulos, tendo como saída uma classe com a combinação dos rótulos.

5.1. Métricas de avaliação

Diferentemente de um classificador tradicional com um único rótulo, um classificador multirrótulo faz uso de diferentes métricas propostas na literatura. Dado que D é um conjunto de dados de testes, consistindo em |D| amostras rotuladas $(\mathbf{x_i}, Y_i)$, $i = 1 \ldots |D|$, e conjunto de rótulos $Y_i \subseteq L$. Seja H um classificador multirrótulo e $Z_i = H(\mathbf{x_i})$ o conjunto de rótulos preditos em H para a amostra $\mathbf{x_i}$.

Em [9], apresenta-se a métrica *Hamming Loss*, que avalia a frequência média com a qual uma amostra é classificada incorretamente:

$$HammingLoss(H, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \Delta Z_i|}{|L|} \quad (8)$$

Quanto menor o valor de *Hamming Loss*, melhor o desempenho de H. Outras métricas aplicadas neste trabalho são apresentadas em [6] para a avaliação de H em D:

$$Accuracy(H, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \quad (9)$$

$$Precision(H, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \cap Z_i|}{|Z_i|} \quad (10)$$

$$Recall(H, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \cap Z_i|}{|Y_i|} \quad (11)$$

Estas métricas medem o quão próximo os valores preditos Z_i estão próximos de Y_i, em relação a conjuntos distintos. Quanto maior os seus valores, melhor o desempenho de H.

5.2. Resultados

Os resultados das métricas descritas na Seção 5.1 para as metodologias aplicadas estão presentes na Tabela 1, onde NN BP-MLL refere-se à rede neural treinada com o algoritmo BP-MLL, e NN BP refere-se à rede neural treinada com o algoritmo padrão. Para esses dois métodos, são apresentados os valores médios das métricas para os 10 conjuntos de teste avaliados. Os valores das demais metodologias foram extraídas de [8]. Os melhores valores para cada métrica estão destacados.
Analisando a Tabela 1, nota-se que os resultados para a metodologia BP-MLL apresentada são semelhantes aos obtidos pelas outras metodologias presentes na literatura: BR, CC e LP. Destaca-se ainda que, dentre esses métodos, a rede BP-MLL obteve os melhores valores para *Accuracy* e *Recall*. Assim, o método proposto mostra-se competitivo para resolver um problema em um banco de dados reconhecidamente difícil de se tratar.
No entanto, a rede neural treinada com o algoritmo padrão atingiu os melhores resultados nas três primeiras métricas, só ficando abaixo em *Recall*. Destaca-se ainda que o tempo de treinamento desta rede é muito menor do que o da rede treinada com BP-MLL: em um computador com processador Intel Core i7-4720HQ e 16GB de memória RAM, o tempo de treinamento médio para 100 épocas da rede neural com o algoritmo BP padrão foi de aproximadamente 1 segundo, enquanto que para o algoritmo BP-MLL foi de aproximadamente 58 minutos - cerca de 3500 vezes mais.
Para este conjunto de dados, a rede neural NN BP se mostra a mais adequada para tratar o problema de classificação multirrótulo. No entanto, não é possível afirmar que uma rede neural com um algoritmo de treinamento padrão se sobressairá sempre neste tipo de problema. Ainda assim, não podemos ignorar o fato de que a arquitetura de rede utilizada, compartilhando uma camada intermediária conectada a todas as saídas de cada rótulo, já pressupõe a utilização de um algoritmo de treinamento com base em aprendizado multitarefa, como apontado

por Caruana em [2]: "No algoritmo *Backpropagtion*, MTL permite o surgimento de atributos na camada intermediária que sejam aproveitados por diferentes tarefas, atributos estes que não apareceriam se o treinamento fosse realizado de forma independente. Acima de tudo, MTL também permite que alguns neurônios da camada intermediária se especializem para algumas tarefas e sejam ignorados por outras que não os julguem úteis". Isto pode explicar o fato de que um algoritmo mais simples também foi capaz de explorar as correlações existentes entre as classificações dos múltiplos rótulos.

Tabela 1. Resultados das métricas para cada método

	Hamming loss	Accuracy	Precision	Recall
NN BP-MLL	0.2077	0.5244	0.6672	**0.6516**
NN BP	**0.1887**	**0.5325**	**0.7175**	0.6222
BR	0.2027	0.5027	0.6962	0.5966
CC	0.2053	0.5139	0.6826	0.6129
LP	0.2164	0.5119	0.6507	0.6092

6. Conclusões

Neste trabalho, realizamos o treinamento de redes neurais para avaliar o seu desempenho em tarefas de classificação multirrótulo. Este tipo de tarefa consiste em determinar o conjunto de rótulos para uma instância de dados não utilizada no treinamento do classificador.

Utilizamos uma rede neural com uma camada intermediária e número de saídas igual ao número total de rótulos. Para o treinamento da rede, dois algoritmos foram utilizados: o algoritmo *Backpropagtion* padrão, e o algoritmo *Backpropagtion Multilabel Learning - BP-MLL*, que modifica o funcional de erro a fim de explorar as correlações existentes entre os diferentes rótulos para uma dada amostra.

Os resultados obtidos para as redes neurais foram comparados com outros métodos da literatura para o conjunto de dados *Yeast*, reconhecidamente difícil de se tratar. A rede treinada com o algoritmo BP-MLL obteve resultados semelhantes aos métodos da literatura. Já a rede treinada com o algoritmo padrão, de menor custo computacional, apresentou o melhor desempenho dentre todos os métodos.

Referências

[1] Mulan: A Java library for multi-label learning. `http://mulan.sourceforge.net/datasets-mlc.html`. Accessado em: 27/06/2017.

[2] Rich Caruana. Multitask learning. *Mach. Learn.*, 28(1):41–75, July 1997.

[3] André Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pages 681–687, Cambridge, MA, USA, 2001. MIT Press.

[4] FEEC/Unicamp. IA353 - Redes Neurais - toolbox de apoio à questão 10 do EFC3, 1 camada intermediária. `http://www.dca.fee.unicamp.br/~lboccato/toolbox_Q10_EFC3_IA353_nn1h.zip`. Accessado em: 29/06/2017.

[5] Y. Freund and Schapire R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[6] Shantanu Godbole and Sunita Sarawagi. *Discriminative Methods for Multi-labeled Classification*, pages 22–30. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[7] Andrew Kachites McCallum. Multi-label text classification with a mixture model trained by em. In *AAAI 99 Workshop on Text Learning*, 1999.

[8] Marcos M. Raimundo and Fernando J. Von Zuben. Many-objective ensemble-based multilabel classification. Working paper, 2017.

[9] Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2):135–168, 2000.

[10] P. Szymański and T. Kajdanowicz. A scikit-based Python environment for performing multi-label classification. *ArXiv e-prints*, February 2017.

[11] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *Int J Data Warehousing and Mining*, 2007:1–13, 2007.

[12] Min-Ling Zhang and Zhi-Hua Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Trans. on Knowl. and Data Eng.*, 18(10):1338–1351, October 2006.