

Leonardo Santos Paulucio

Tarefa Computacional 3

Vitória - ES

03 de Dezembro de 2019

1 Introdução

Técnicas de *feature selection* têm o propósito de identificar um conjunto menor de *features*, que sejam relevantes, de forma que seja possível realizar uma simplificação do modelo, e, assim, melhorar sua performance.

Apesar das inúmeras vantagens, a escolha das melhores *features* não é uma tarefa simples de ser realizada. Diversos estudos na literatura apresentam alguns problemas como: alto custo de tempo computacional, falta de consistência na seleção de *features* relevantes, e pequeno aumento na acurácia do modelo. Sendo assim, essa é uma área que ainda está em grande relevância na literatura.

Nessa trabalho será implementado a ideia proposta por [Deraeve e Alexander \(2018\)](#), onde o autor propõe uma nova abordagem para *feature selection* utilizando Redes Neurais.

2 Arquitetura

A arquitetura proposta por Deraeve e Alexander (2018) consiste em uma arquitetura simples formada por uma *single-layer neural network* onde as *features* são dadas como entradas e a camada de saída são formadas por nós que representam cada uma das classes. Uma ilustração dessa arquitetura está ilustrada na Figura 1. O pseudocódigo do algoritmo proposto está ilustrado na Figura 2.

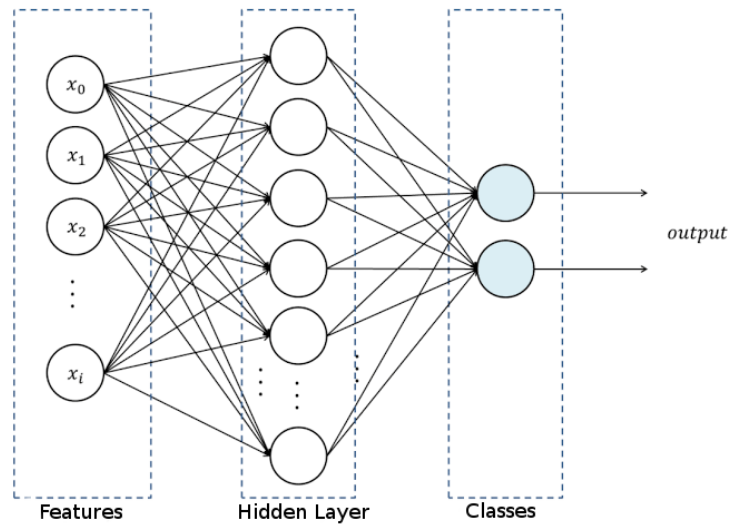


Figura 1 – Arquitetura proposta

Algorithm Feature selection: Neural network

```

1: for 20 iterations do
2:   train_samples = 90 % of training set
3:   for 50 epochs do
4:     forward propogation using train_samples
5:     update weights with delta rule
6:     weights = updated weights
7:     val_acc = acc on validation set
8:   end for
9:   best_weights = weights for epoch with max(val_acc)
10: end for
11: summed_weights = abs(sum(best_weights))
12: sorted_importance = sort(summed_weights)
13: return sorted_importance

```

Figura 2 – Pseudocódigo do algoritmo proposto

3 Resultados e Discussão

Com o objetivo de comparar os resultados do método com os obtidos no trabalho computacional 1, foram utilizados os mesmos valores dos parâmetros e no número de execuções e épocas. A Tabela 1 apresenta os valores utilizados.

Parâmetro	Valor	Significado
NumRuns	6	Número de execuções
NumEpochs	100	Num. épocas de treino do classificador
HidSize	$\sqrt{NumFeatures * NumClasses}$	Tam. camada oculta do classificador

Tabela 1 – Parâmetros utilizados

Para analisar o desempenho do método foram utilizados *datasets* com diferentes quantidades de *features*. Os *datasets* selecionados foram: Wine, Ionosphere e Arrhythmia. A Tabela 2 apresenta com mais detalhes cada um.

Dataset	Features	Amostras	Classes	Balanceado
Wine	13	178	3	Sim
Ionosphere	34	351	2	Sim
Arrhythmia	279	452	16	Não

Tabela 2 – Detalhes dos datasets utilizados

3.1 Baseline

Para o *baseline* todas as *features* foram utilizadas como entrada em um classificador simples. Os resultados obtidos estão ilustrados na Tabela 3.

Dataset	Acurácia
Wine	0.9722
Ionosphere	0.8873
Arrhythmia	0.6593

Tabela 3 – Resultados da acurácia média obtida no baseline

3.2 Experimento

As tabelas de 4-6 apresentam os resultados obtidos nos experimentos juntamente com os do trabalho computacional 1 para que seja possível realizar uma comparação dos mesmos. Como o modelo retorna um vetor ordenado das importâncias de cada *feature*, para fins de comparação com o trabalho 1, o número de *features* selecionadas escolhido para cada *dataset* foi o mesmo número obtido para a melhor acurácia do trabalho 1. Assim o número de *features* selecionadas para os *datasets* Wine, Ionosphere e Arrhythmia foram 7, 19 e 132 respectivamente.

A Tabela 4 apresenta as acurácias obtidas. Já a Tabela 5 apresenta a média dos tempos de execuções obtidos. E, por último, a Tabela 6 apresenta os índices de Jaccard obtidos. As matrizes de confusão obtidas estão ilustradas na Figura 3.

Dataset	Baseline	TC1	TC3
Wine	0.97	0.99	0.93
Ionosphere	0.89	0.93	0.89
Arrhythmia	0.66	0.75	0.65

Tabela 4 – Média da acurácia obtida

Dataset	TC1	TC3
Wine	0.62	3.83
Ionosphere	0.65	4.76
Arrhythmia	9.81	23.19

Tabela 5 – Média do tempo de execução

Dataset	Média	Desvio Padrão
Wine	0.40	0.10
Ionosphere	0.15	0.12
Arrhythmia	0.21	0.06

Tabela 6 – *Jaccard index*

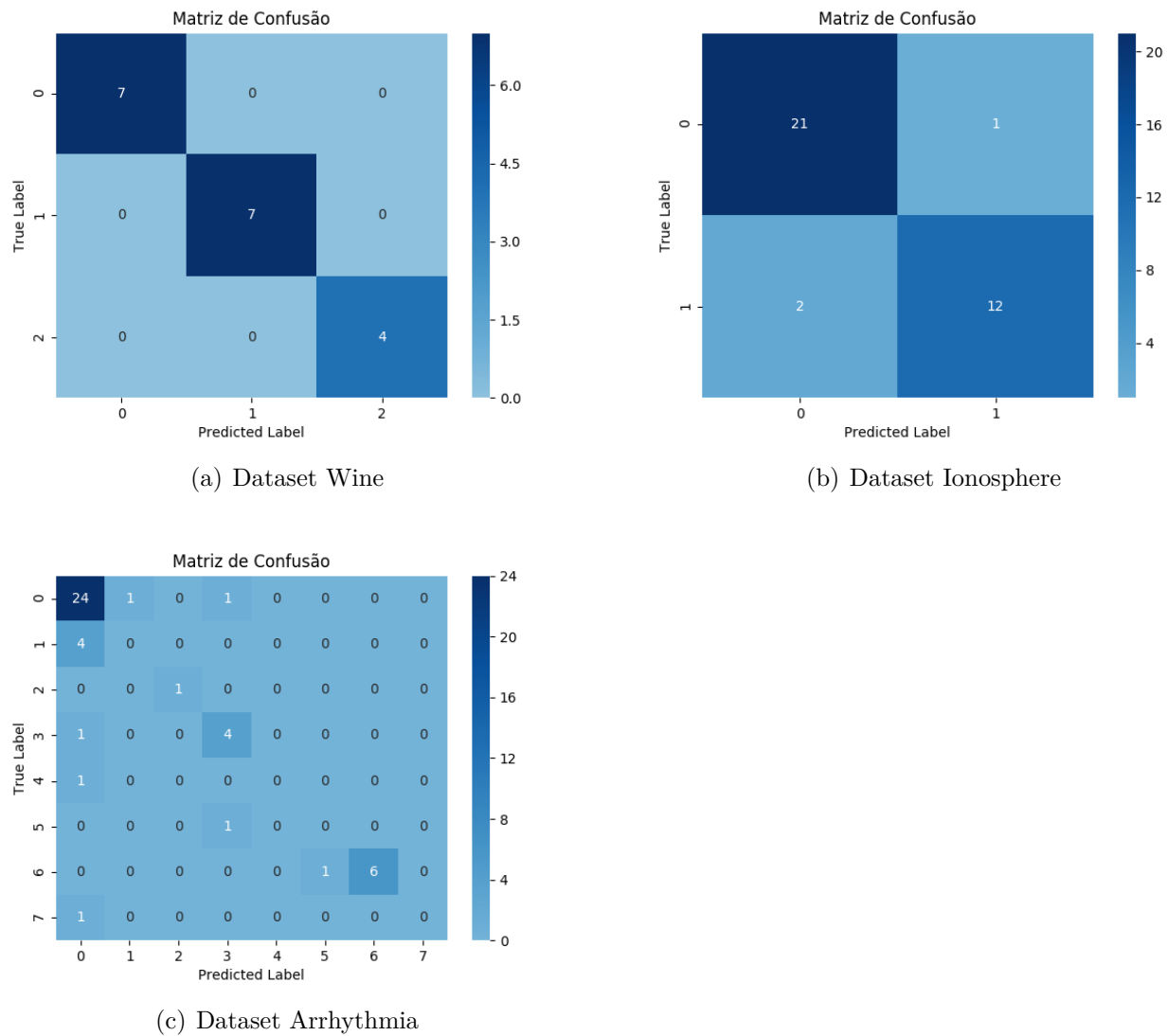


Figura 3 – Matrizes de confusão obtidas para as melhores acurácias

4 Discussões

Analisando-se os resultados apresentados nas tabelas, percebe-se que a acurácia obtida com a nova proposta para os *datasets*: Wine, Ionosphere e Arrhythmia não foram melhores do que os obtidos no trabalho 1 e foram muito próximos aos obtidos com o *baseline*.

Com relação ao tempo de execução, o novo método foi mais demorado do que os obtidos no trabalho 1. Isso se deve ao fato de serem realizadas várias etapas de treinamento durante a seleção de *features*.

O índice de *Jaccard* obtido para os *datasets* não foram muito bons, o que indica que as *features* selecionadas a cada execução não são totalmente iguais. Porém como os resultados são diferentes de zero existem *features* selecionadas que se repetem nas diferentes execuções, algo que é desejável.

Os resultados apresentados no artigo original mostram que o modelo se saiu muito bem em *datasets* simulados, quando comparados com outros métodos existentes. Esses *datasets* simulados não foram possíveis de serem repetidos para tentar replicar os resultados do artigo original.

5 Conclusão

Nesse trabalho foi desenvolvida a forma de se realizar *feature selection* utilizando redes neurais proposta por [Deraeve e Alexander \(2018\)](#). O método foi analisado nos *datasets* Wine, Ionosphere e Arrhythmia.

O modelo obteve bons resultados, porém não foi melhor do que o baseline e o modelo utilizando NN+GA desenvolvido no trabalho computacional 1. Dessa forma, talvez seja necessário realizar estudos mais aprofundados para se tentar melhorar a eficiência do modelo proposto.

Referências

DERAEVE, J.; ALEXANDER, W. H. Fast, accurate, and stable feature selection using neural networks. *Neuroinformatics*, Springer, v. 16, n. 2, p. 253–268, 2018. Citado 3 vezes nas páginas [1](#), [2](#) e [7](#).