

Trabalho Final: Redes Neurais Artificiais

Rúbia Reis Guerra

Abstract—Aplicação de técnicas redes neurais de múltiplas camadas a problemas de classificação utilizando variações de funções de treinamento.

I. INTRODUÇÃO

Em Estatística e Aprendizado de Máquina, a classificação é o problema da identificação de um conjunto de categorias (subpopulações) que uma nova observação pertence, com base em um conjunto de treinamento contendo informações (ou instâncias) cujo pertencimento às categorias base seja conhecido. Um exemplo seria atribuir um determinado email às classes "spam" ou "não-spam" ou atribuir um diagnóstico a um determinado paciente conforme descrito pelas características observadas do paciente (gênero, pressão arterial, presença ou ausência de certos sintomas, etc.). Neste trabalho, serão usadas técnicas de Redes Neurais Artificiais e Aprendizado de Máquina com intuito de resolver problemas de classificação em três bases de dados distintas.

II. RESUMO TEÓRICO

A. Multilayer Perceptron

Um perceptron multicamada (MLP) é um modelo de rede neural artificial *feedforward* que mapeia conjuntos de dados de entrada para um conjunto de saídas apropriadas. Um MLP consiste em várias camadas de nós em um grafo direcionado, com cada camada totalmente conectada à próxima. Exceto os nós de entrada, cada nó é um neurônio (ou elemento de processamento) com uma função de ativação não-linear. O MLP é uma modificação do perceptron linear e pode distinguir dados que não são linearmente separáveis.

B. Backpropagation with Weight Decay

O algoritmo de backpropagation faz ajustes calculando a derivada ou a inclinação do erro da rede em relação a saída de cada neurônio. Ele tenta minimizar o erro geral ao diminuir essa inclinação para o valor mínimo para cada peso, avançando um passo abaixo da inclinação de cada época. Se a rede tomar medidas que muito grandes, pode passar o mínimo global. Se tomar medidas muito pequenas, pode estabelecer-se em mínimos locais, ou gastar uma quantidade excessiva de tempo para chegar ao mnimo global. O Weight Decay foi introduzido por P. Werbos, em 1988. A ideia do algoritmo é simples: em cada iteração do ciclo de treinamento, após a atualização dos valores de pesos e bias da rede neural, os pesos e os biases são diminuídos por uma pequena quantidade. Isso tende a manter as amplitudes do peso e dos valores de bias pequenos, o que, por sua vez, ajuda a impedir o overfitting.

C. Resilient Backpropagation (Rprop)

Rprop, ou resilient backpropagation, é uma heurística para aprendizagem supervisionada em redes neurais artificiais *feedforward*, criada por Martin Riedmiller e Heinrich Braun em 1992. A heurística leva em consideração apenas o sinal da derivada parcial sobre todos os padrões (e não a magnitude), e age de forma independente em cada peso. Para cada peso, se houver uma mudança de sinal da derivada parcial da função de erro total em comparação com a última iteração, o valor de atualização para esse peso é multiplicado por um fator η^- , onde $\eta^- < 1$. Se a última iteração produzir mesmo sinal, o valor de atualização é multiplicado por η^+ , onde $\eta^+ > 1$. Assim, são calculados os valores de atualização para cada peso, e, por fim, cada peso é alterado pelo seu próprio valor de atualização, na direção contrária à derivada parcial do peso, de modo a minimizar a função de erro total. η^+ é configurado empiricamente para 1.2, e, η^- , para 0.5. O algoritmo Rprop possui três parâmetros: o valor de atualização inicial, um limite para o tamanho máximo do passo e o expoente de decaimento de peso.

D. Scaled Conjugate Gradient

O algoritmo básico de backpropagation ajusta os pesos na direção de descida mais íngreme (contrária ao gradiente). Essa é a direção em que a função de custo está diminuindo com mais rapidez. Embora a função diminua mais rapidamente ao longo do negativo do gradiente, isso não produz necessariamente a convergência mais rápida. Nos algoritmos de gradiente conjugado, uma busca é realizada ao longo das direções conjugadas, o que produz convergência geralmente mais rápida do que as direções de descida mais íngremes. Na maioria dos algoritmos de gradiente conjugado, o tamanho do passo é ajustado em cada iteração. Uma busca é feita ao longo da direção do gradiente conjugado para determinar o tamanho da etapa, que minimiza a função de desempenho ao longo dessa linha. Porém, a busca por essa linha é computacionalmente dispendiosa, uma vez que exige que a resposta da rede a todas as entradas de treinamento seja calculada várias vezes para cada pesquisa. O algoritmo de *Scaled Conjugate Gradient* (SCG), desenvolvido por Moller [Mol93], foi projetado para contornar o problema, e tem como ideia básica combinar a abordagem da região modelo-confiança (usada no algoritmo Levenberg-Marquardt), com a abordagem do gradiente conjugado.

E. Receiver Operating Characteristic

Uma forma de comparar o desempenho de classificadores é por meio da área abaixo da curva ROC (AUC). De forma resumida, a área sob a curva ROC especifica a probabilidade

de que, quando é escolhido um exemplo positivo e um negativo ao acaso, a função de decisão atribui um valor maior ao positivo do que ao exemplo negativo. Assim, tem-se um "melhor" resultado quando a área aproxima-se de 1.

III. COMPARAÇÃO EMPÍRICA

A solução proposta consiste em aleatorizar os índices das observações, dividir o dataset em conjuntos de treino (70% dos dados) e teste (30% dos dados) e realizar o treinamento utilizando métodos do pacote RSNNs, descritos nas seções anteriores. Serão realizados 10 treinamentos para cada método e, ao final, será feita uma comparação dos resultados de classificação para cada técnica utilizada.

TABLE I
BASES DE DADOS UTILIZADAS

Nome	Origem	Atributos	Exemplos	Classes
Titanic	Kaggle	12	892	2
DNA	mlbench	181	3186	3
Spambase	UCI	57	4601	2

A. DNA

Para esta base, é resolvido um problema de classificação **multiclasse**, cujo objetivo é encontrar a classe a qual um novo exemplo pertence, dentre as categorias já existentes (ei, ie, n).

TABLE II
RESULTADOS: DNA

Treinamento	Média AUC (%)	Std. AUC (%)
Rprop	96.91	0.47
Backprop. w/ W.D.	97.72	0.62
SCG	97.18	0.75

B. Spambase

Para esta base, é resolvido um problema de classificação simples, cujo objetivo é encontrar a classe a qual um novo exemplo pertence, dentre as categorias (spam e não-spam).

TABLE III
RESULTADOS: SPAMBASE

Treinamento	Média AUC (%)	Std. AUC (%)
Rprop	92.44	0.34
Backprop. w/ W.D.	92.46	0.31
SCG	92.49	0.43

C. Titanic

Para esta base, é resolvido um problema de classificação simples, cujo objetivo é encontrar a classe a qual um novo exemplo pertence, dentre as categorias (survived ou not-survived). Porém, a falta de estrutura de alguns dos atributos de entrada e os baixos resultados (Rprop: 62.46%, 0.38% std.) obtidos se utilizada a base em seu estado bruto, demonstra a necessidade de se realizar manipulações nos dados previamente ao treinamento.

1) *Análise Exploratória*: A fim de entender melhor as possíveis correlações entre os atributos de entrada, plotou-se as seguintes grupos de variáveis:

- Sex x Survived (fig. 1): mulheres têm maiores chances de sobreviver;
- Age x Survived (fig. 2): crianças têm maiores chances de sobreviver;
- Fare x Survived (fig. 3): dentro das pessoas que sobreviveram, a maior porcentagem pagou uma tarifa elevada para o bilhete, o que indica que a tarifa alta pode corresponder melhores cabines, que são fáceis de escapar.

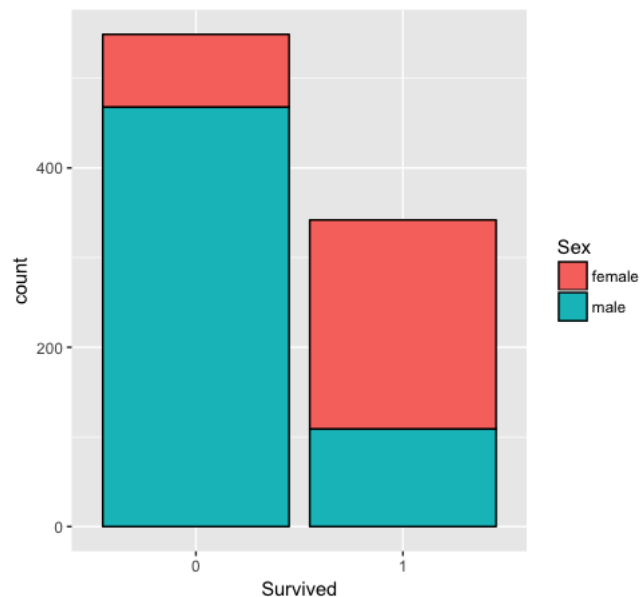


Fig. 1. Sex x Survived

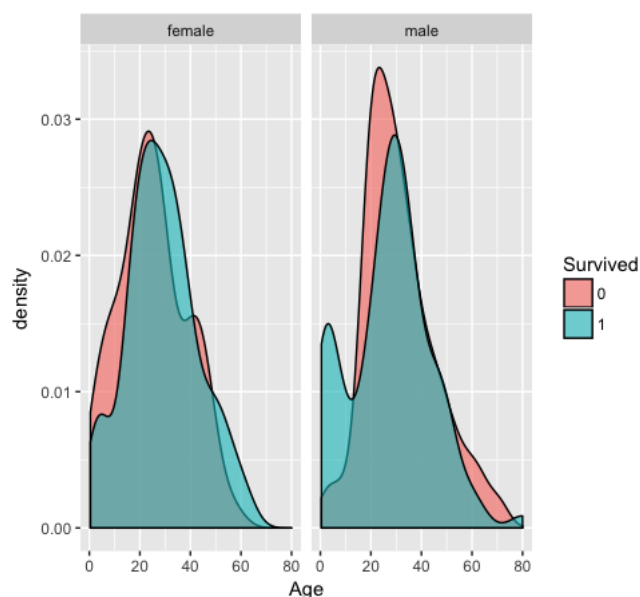


Fig. 2. Age x Survived

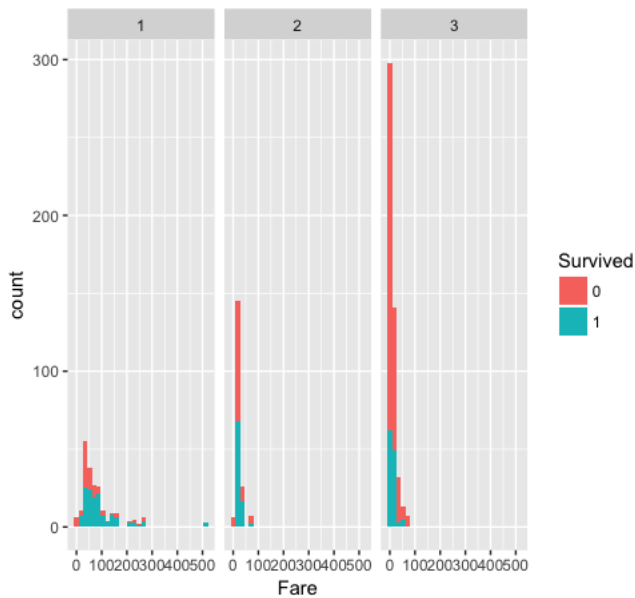


Fig. 3. Fare x Survived

2) *Extração de Características*: Para tentar melhorar os resultados, foi realizada extrações de características com base nos atributos existentes:

- Name: a partir de *Name*, pode-se extrair informações sobre famílias (passageiros que compartilham do mesmo sobrenome) e classe social (a partir do tratamento, por exemplo, Mr. e Master.)
- Ticket: pode-se extrair informações sobre pessoas que viajam juntas (possuem o mesmo número de bilhete);
- Fare: pode-se extrair o preço pago por pessoa para um grupo que viaja junto;

3) *Tratamento de Valores Faltantes*: Os valores faltantes foram tratados da seguinte forma:

- NA em Age: substituir pela média de idade do grupo de mesmo título (Mr., Lady, Master);
- NA em Embarked: substituir pelo valor mais frequente;
- NA em Cabin: substituir por Unknown ou substituir pelo valor do grupo (se o passageiro viaja com um grupo);
- NA em Fare: substituir pela média dos valores de Fare.

A partir das modificações acima, foram obtidos os seguintes resultados:

TABLE IV
RESULTADOS: TITANIC

Treinamento	Média AUC (%)	Std. AUC (%)
Rprop	78.21	2.04
Backprop. w/ W.D.	78.18	2.86
SCG	77.34	2.48

IV. CONCLUSÃO

Neste trabalho, foi realizada a aplicação de técnicas de redes neurais com diferentes funções de treinamento para resolver problemas de classificação em aprendizado de máquina.

Ainda, observou-se os diferentes subproblemas existentes no domínio de classificação, como problemas multiclasse e a necessidade de extração de características. Futuramente, para melhorar os resultados encontrados, outras ferramentas como seleção de características e validação cruzada podem ser implementadas.

REFERENCES

- [1] A. Braga, A. Carvalho and T. Ludermir, *Redes neurais artificiais*. Rio de Janeiro: LTC Editora, 2007.
- [2] R. Duda, D. Stork and P. Hart, *Pattern classification and scene analysis*. New York: Wiley, 2000.
- [3] "Encyclopedia Titanica", *Encyclopedia Titanica*, 2017. [Online]. Available: <https://www.encyclopedia-titanica.org/>. [Accessed: 30- Jun- 2017].
- [4] "SNNS- Stuttgart Neural Network Simulator", *Ra.cs.uni-tuebingen.de*, 2017. [Online]. Available: <http://www.ra.cs.uni-tuebingen.de/SNNS/>. [Accessed: 30- Jun- 2017].