Redes Neurais para mapeamento não linear

Neural Networks for nonlinear mapping

Nicholas Damasceno Pinto*, Thiago Antonini Alves†, Yara de Souza Tadano‡, Hugo Siqueira§

RESUMO

Este trabalho propõe a aplicação de Redes Neurais Artificiais (RNA) do tipo *Perceptron* de Múltiplas Camadas (MLP) para problemas do mundo real. São discutidas as etapas de implementação, treinamento e aplicação. O estudo de casos envolve um problema de classificação baseado nas distintas características de carros. Os resultados computacionais mostram a viabilidade da proposta.

Palavras-chave: Redes Neurais Artificiais, *Perceptron* de Múltiplas Camadas, classificação.

ABSTRACT

This work proposes the application of Multilayer Perceptron Artificial Neural Networks (MLP ANN) to real world problems. Implementation, training, and application steps are discussed. The case study involves a classification problem based on the different characteristics of cars. The computational results show the feasibility of the proposal.

Keywords: Artificial Neural Networks, Multilayer Perceptron, classification.

1 INTRODUÇÃO

As Redes Neurais Artificiais podem ser aplicadas em múltiplas áreas para facilitar processamento de dados e têm sido utilizadas com sucesso nos mais diversos problemas. Dentre as principais áreas podemos citar: sistemas de controle, reconhecimento de padrões e aproximação (FERREIRA, 2019),

A Rede Neural é um sistema de nós interconectados capaz de reconhecer padrões e correlações entre dados, assim podendo classificar e agrupar esses dados (HAYKIN, 1999). A arquitetura *Perceptron* de Múltiplas Camadas (*Multi Layer Perceptron* – MLP) é um modelo que apresenta multicamadas de neurônios, os quais são organizados para permitir processamento não linear, com pelo menos uma camada de entrada, uma intermediária e uma de saída (BROWNLEE, 2019). A primeira distribui os dados aos neurônios da camada seguinte sem modificá-los; as camadas intermediárias realizam o processamento dos dados recebidos e repassando-os adiante; por fim, a camada de saída possui apenas um único neurônio que devolve a saída da rede (BRAGA *et al*., 1999). Note que a arquitetura admite múltiplas saídas.

A MLP usada como classificador possui um parâmetro α que é otimizado através da comparação de valores diferentes do conjunto de dados. O seu aumento corrige a alta variância e encoraja pesos menores e seu decréscimo corrige o viés e encoraja pesos menores (HAYKIN, 1999).

* Ciência da Computação, Universidade Tecnológica Federal do Paraná, Ponta Grossa,PR,Brasil; nicholaspinto@alunos.utfpr.edu.br
† Universidade Tecnológica Federal do Paraná, *Campus* Ponta Grossa; antonini@utfpr.edu.br
‡ Universidade Tecnológica Federal do Paraná, *Campus* Ponta Grossa; yaratadano@utfpr.edu.br
§ Universidade Tecnológica Federal do Paraná, *Campus* Ponta Grossa; hugosiqueira@utfpr.edu.br

O mapeamento não linear obtém um mapa da distância entre as variáveis de um conjunto, representando-a em um gráfico é possível obter estimativas das relações sobre determinadas variáveis ou conjunto de variáveis. Técnicas de *encoding*, são utilizadas para tratar dados, visando transformá-los para melhor manipulação durante o trabalho. Duas técnicas serão usadas, a *one-hot* e *label*. A primeira, apesar de ser muito custosa, gera uma coluna para cada valor categórico do atributo e a segunda, apesar do problema de ranqueamento possui um ótimo custo-benefício, quantificando os atributos de forma simples e prática (LAZZERI, 2020).

Neste trabalho, a Rede Neural criada é a MLP *classifier* para mapeamento não linear, utiliza a linguagem de programação *Python* em conjunto das bibliotecas Tensorflow e Keras para realizar os cálculos, previsões e criar gráficos (FALAHATI, 2021). A base de dados que será classificada e mapeada é um *dataset* de *Marko Bohanec*, o *Car Evaluation 4*, o qual possui 3.098 entradas (completas) contendo atributos dos veículos e avaliações de 100.000 clientes.

Com o potencial para diversas áreas, foi escolhido em analisar o mercado de veículos que tem tido certas dificuldades de classificar e mapear as vendas e compras de veículos e satisfação dos clientes – um problema complexo (OLIVEIRA, 2019). Dessa forma, será possível criar uma Rede Neural Artificial para mapear e classificar veículos segundo a preferência dos clientes e avaliar o seu grau de satisfação.

2 MÉTODO E ELABORAÇÃO DE UMA MLP

Como mencionado anteriormente, a entrada de dados é o *dataset Car Evaluation* 4 que apresenta os veículos com os seguintes atributos: valor, manutenção, portas, passageiros, tamanho porta-malas e segurança. As avaliações dos veículos são aceitabilidade e satisfação. Uma amostra desses dados é mostrada na Figura 1.

Figura 1 – Amostra de dados.

	Valor	Manutenção	Portas	Passageiros	Porta-Malas	Segurança	Aceitabilidade	Satisfação
24	vhigh	vhigh	2	more	big	low	unacc	unacc
237	vhigh	med	2	more	med	low	unacc	unacc
563	high	high	2	more	med	high	acc	acc
333	vhigh	low	2	4	small	low	unacc	unacc
1187	med	med	5more	more	big	high	vgood	vgood
723	high	med	4	more	med	low	unacc	unacc
1464	low	high	4	2	big	low	unacc	unacc
256	vhigh	med	3	4	med	med	unacc	unacc
0	vhigh	vhigh	2	2	small	low	unacc	unacc
138	vhigh	high	3	2	med	low	unacc	unacc

Fonte: autoria própria.

Como os dados não são todos numéricos, eles precisam ser tratados antes de serem entregues ao classificador. Para isso, são aplicadas as técnicas de *encoding one-hot* e *label*, com o código ilustrado nas Figuras 2 e 3.

Figura 2 – Tratamento de dados 1

```
#car.columns = ["Valor", "Manutenção", "Portas", "Passageiros", "Porta-malas", "Segurança", "Aceitabilidade, Satisfação"]
# Label encoding
car["Valor"] = car.apply(lambda row : transf_valorman(row["Valor"]),axis = 1).astype('float')
car["Manutenção"] = car.apply(lambda row : transf_valorman(row["Manutenção"]),axis = 1).astype('float')
car["Passageiros"] = car.apply(lambda row : transf_passageiros(row["Passageiros"]),axis = 1).astype('float')
car["Porta-malas"] = car.apply(lambda row : transf_portamala(row["Porta-malas"]),axis = 1).astype('float')
# One-hot enconding
car = pd.get_dummies(car, columns=['Portas','Segurança'])
car['Portas_2'] = car['Portas_2'] + 0.001
car['Portas_3'] = car['Portas_3'] + 0.001
car['Portas_4'] = car['Portas_4'] + 0.001
car['Portas_5more'] = car['Portas_5more'] + 0.001
car['Segurança_low'] = car['Segurança_low'] + 0.001
car['Segurança_high'] = car['Segurança_high'] + 0.001
car['Segurança_med'] = car['Segurança_med'] + 0.001
```

Fonte: autoria própria.

Figura 3 – Tratamento de dados 2.

```
#Encoding
def transf_valorman(aux): #encoding Valor e Manutenção
  valores = np.linspace(1,0.001,4) #normalização
  if aux == "vhigh":
    return valores[3]
  elif aux == "high":
    return valores[2]
  elif aux == "med":
    return valores[1]
  elif aux == "low":
    return valores[0]
  else:
    return aux

def transf_passageiros(aux): #Passageiros
  if aux =="4" or aux == "more":
    return 1.0
  elif aux == "2":
    return 0.001
  else:
    return aux

def transf_portamala(aux): #Tamanho do porta-malas
  valores = np.linspace(1,0.001,3)
  if aux == "small":
    return valores[2]
  elif aux == "med":
    return valores[1]
  elif aux == "big":
    return valores[0]
  else:
    return aux
```

Fonte: autoria própria.

O modelo de Rede Neural Artificial MLP *classifier* foi construído em linguagem *Python* com bibliotecas auxiliares (BROWNLEE, 2019). Essa RNA recebe os dados já tratados pelo *encoding* e então separa esses dados em 3 grupos: treinamento que recebe 70% dos dados; validação que recebe 15%; e teste que recebe os demais 15%. O código é ilustrado na Figura 4. Após essa etapa, a Rede Neural Artificial MLP pode ser executada. O código da MLP *classifier* está ilustrado na Figura 5.

Figura 4 – Separação de dados.

```
#Separando os dados
xdata = car[["Valor", "Manutenção", "Passageiros", "Porta-malas", "Portas_2", "Portas_3", "Portas_4", "Portas_5more", "Segurança_high", "Segurança_low", "Segurança_low"]]
ydata = car["Aceitabilidade, Satisfação"]
# Separação de dados para a MLP 70,15,15
xMLP_train, A, yMLP_train, B = train_test_split(xdata, ydata, train_size = 0.70, shuffle = True)
xMLP_val, xMLP_test, yMLP_val, yMLP_test = train_test_split(xdata, ydata, train_size = 0.70, shuffle = True)
```

Fonte: autoria própria.

Figura 5 – MLP *classifier*

```
#MLP classifier
yMLP_train = pd.get_dummies(yMLP_train) #one hot encoder pra mlp
yMLP_test = pd.get_dummies(yMLP_test) #one hot encoder pra mlp
yMLP_val = pd.get_dummies(yMLP_val) #one hot encoder pra mlp

from keras.models import Sequential
from keras.layers import Dense
#Mlp
modelo = Sequential()
modelo.add(Dense(300, activation='relu', input_dim=len(xMLP_train.columns)))
modelo.add(Dense(4))

modelo.compile(optimizer='adam', loss='mse', metrics=['accuracy'])

history = modelo.fit(
    xMLP_train,
    yMLP_train,
    epochs=100,
    verbose=0,
    validation_data=(xMLP_val, yMLP_val))

acc2 = modelo.evaluate(xMLP_test, yMLP_test, verbose=0)
```

Fonte: autoria própria.

3 RESULTADOS

Após a elaboração da Rede Neural e do pré-processamento dos dados, a etapa de execução foi realizada. A taxa de acerto na melhor execução foi de quase 99%, ilustrada na Figura 6, como mostra a saída do código. Feito isso, foram gerados os gráficos da Figura 7, que mostram a aceitabilidade das dimensões dos dados. É possível deduzir que carros com capacidade para apenas 2 passageiros e baixa segurança são pouco atraentes aos clientes, os quais dão preferências para os mais baratos e com baixo custo de manutenção, além de preferirem carros com porta-malas mais espaçosos e alta segurança.

Ao analisar a Figura 8 pode ser observado que os clientes possuem grau de aceitação conciso. Uma vez que as variáveis “Satisfação” e “Classificação de Veículos” ficaram bem entrelaçadas, mostrando que a classificação foi a adequada. Dessa forma, a Rede Neural Artificial MLP foi capaz de classificar veículos que satisfizeram a maior parte dos clientes de forma positiva. Isto mostra que a relação da satisfação com classificação, mesmo quando o número de clientes aumentou.

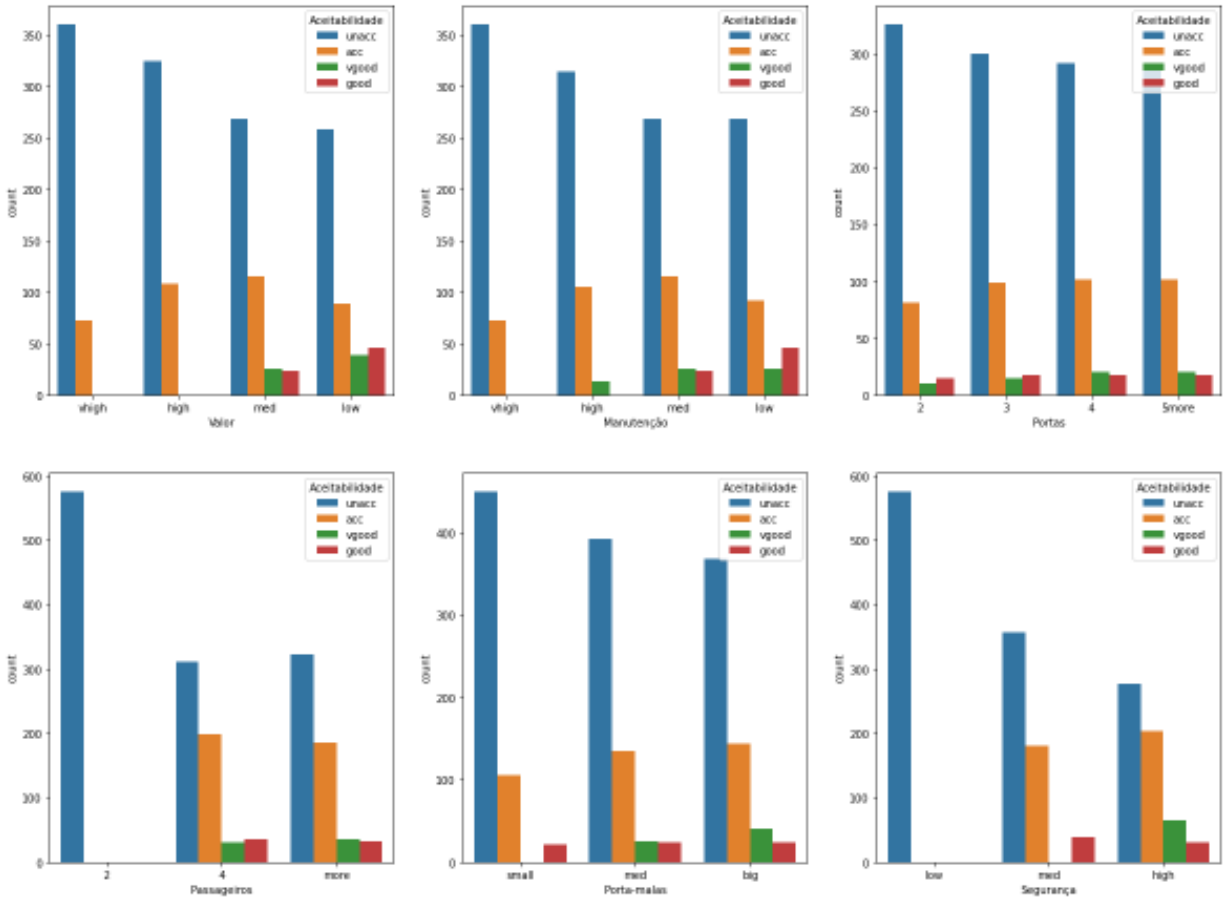
Figura 6 – Taxa de acerto.

```
print("MLP taxa de acerto: " + format((acc2[1] * 100.00),'.2f') + "%")

MLP taxa de acerto: 98.84%
```

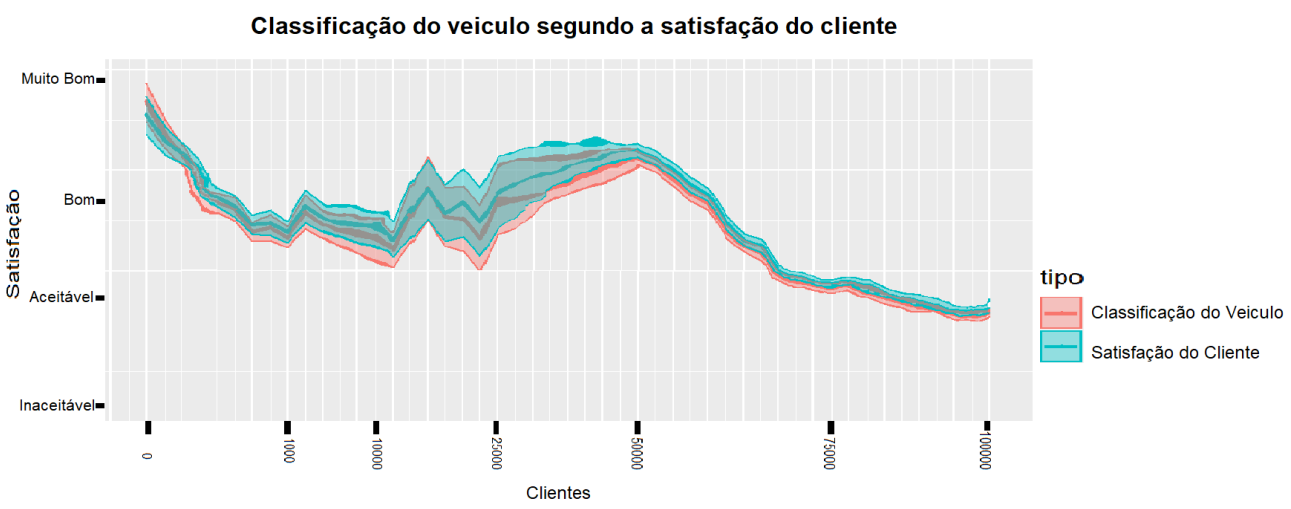
Fonte: autoria própria.

Figura 7 – Gráfico da aceitabilidade.



Fonte: autoria própria.

Figura 8 – Mapeamento da satisfação.



Fonte: autoria própria.

4 CONCLUSÃO

Este trabalho apresentou as etapas de elaboração e aplicação de uma Rede Neural Artificial do tipo *Perceptron* de Múltiplas Camadas. Foram discutidos aspectos construtivos e de parametrização da Rede Neural. Como estudo de caso, foi utilizada uma base de classificação de veículos. A taxa de acerto da Rede Neural foi de aproximadamente 99%, um valor satisfatório se considerar o número de atributos. O tratamento que foi capaz de melhor aproveitar as amostras facilitando o trabalho do classificador. Com a elaboração de um gráfico de aceitabilidade foi possível notar que tipo de veículos os clientes preferem. Dessa forma, definidos os veículos com maior probabilidade de venda, pode ser inferido quais são os que levam a um bom grau de satisfação dos clientes. Finalmente, pode ser notado que a criação de uma Rede Neural Artificial MLP para tratar de mundo real é possível e relevante. Dessa maneira, trabalhos futuros devem ser desenvolvidos no sentido de aplicar outras arquiteturas de Redes Neurais Artificiais em tarefas como previsão de séries temporais e impacto da poluição atmosférica na saúde humana

AGRADECIMENTOS

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pela bolsa de Iniciação Tecnológica concedida ao acadêmico.

REFERÊNCIAS

BROWNLEE, J. **Introduction to time series forecasting with Python: How to prepare data and develop models to predict the future**. Canberra: Books AU, 2019.

HAYKIN, S. **Neural networks: A comprehensive foundation**, Prentice Hall, New York, NY, USA, 1999.

BRAGA, A.; FERREIRA, A.; LUDERMIR, T. **Redes neurais artificiais: Teoria e aplicações**. LTC Editora, Rio de Janeiro, RJ, Brasil, 2007.

LAZZERI, F. **Machine Learning for time series forecasting with Python**. Wiley & Sons, New Jersey, NY, 2020.

FALAHATI, A. **Titanic comprehensive notebook (EDA+ML)**. Hoda Katebi, Tehran, Iran, 2021.

FERREIRA, C.A. **MLP Classifier**. Saraiva, Guaxupé, MG, Brasil, 2019.

OLIVEIRA, C. **Atividade da força de venda em relação ao comportamento de compra de um automóvel.** Saraiva, Santos, SP, Brasil, 2019.