

# Implementação do Perceptron

Murillo Freitas Bouzon

**Resumo**—Redes Neurais Artificiais é um conceito que surgiu já há algum tempo, porém é uma área que atualmente está em constante crescimento graças a popularização do *Deep Learning*. A base desta área são os neurônios artificiais, modelados com base nos neurônios biológicos para simular o comportamento cerebral. Um modelo tradicional desses neurônios é o Perceptron, um classificador binário que recebe uma lista de entradas e retorna como saída apenas dois valores. O Perceptron serviu como base para diversos algoritmos que surgiram no futuro do estudo de Redes Neurais. Dito isso, neste trabalho foi implementado um modelo de Perceptron tradicional para realizar a classificação binária de uma base de dados. Os resultados mostraram que utilizando o Perceptron, é possível classificar dados corretamente que são linearmente separáveis. Porém, para dados que não são linearmente separáveis, o Perceptron mostrou-se ineficaz, necessitando de outras abordagens para realizar esses tipos de classificações.

**Index Terms**—K-means, Aprendizado não-supervisionado, Machine Learning, Cluster

## I. INTRODUÇÃO

O conceito de Redes Neurais Artificiais surgiu há mais de meio século, sendo proposto por [1], com o intuito de se assimilar ao modelo cerebral para a realização de tarefas computacionais, acreditando que seria possível simular comportamentos com mais inteligência.

Para isso, foram propostos modelos de neurônios artificiais baseados nos neurônios biológicos, sendo conectados entre si para formar uma rede neural, onde cada neurônio recebe uma entrada multiplicada por um peso e retorna uma saída. Desta forma, os neurônios comunicam entre si para encontrar uma resposta final para um problema qualquer.

Um dos primeiros modelos de neurônios artificiais é o Perceptron, sendo um neurônio com uma função de classificação binária, atribuindo apenas o valor 1 ou 0 dependendo do valor das entradas e dos pesos.

Sendo o modelo mais simples de neurônio, esse trabalho tem como objetivo implementar o Perceptron para classificação binária de dados, sendo testada na base de dados Iris e para calcular as funções AND, OR e XOR.

## II. CONCEITOS FUNDAMENTAIS

Nesta seção serão apresentadas as teorias relacionadas ao método desenvolvido neste trabalho.

### A. Perceptron

O Perceptron foi proposto em [2], sendo um modelo computacional para classificação binária baseado na estrutura de um neurônio biológico que dada uma entrada  $x$ , retorna um valor de saída  $f(x)$  dada pela Equação (1):

$$f(x) = \begin{cases} 1 & \text{se } w \cdot x + b \geq 0 \\ 0 & \text{caso contrário} \end{cases}, \quad (1)$$

sendo  $w$  o vetor de pesos,  $x$  as entradas e  $b$  o *bias* da função.

A Figura 1 mostra um exemplo de um modelo de Perceptron, possuindo 3 entradas e retornando 1 saída.

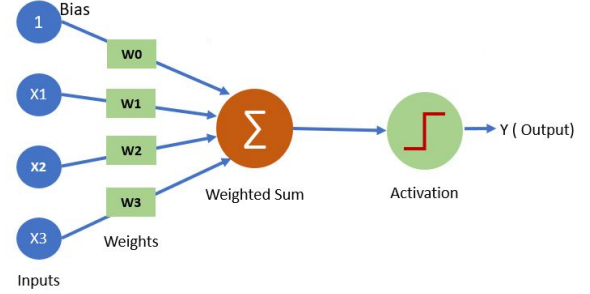


Figura 1: Ilustração de um modelo de Perceptron.

O Perceptron atribui o valor dos seus pesos utilizando um modelo de aprendizado supervisionado, dado pela Equação (2):

$$w_i = w_i + \eta(t - o)x_i, \quad (2)$$

onde  $w_i$  é o peso que está sendo atualizado,  $t$  é o valor desejado,  $o$  é o valor estimado pelo perceptron,  $x_i$  é a entrada do perceptron e  $\eta$  é a taxa de aprendizado.

## III. METODOLOGIA

A metodologia deste trabalho consiste na implementação da classe Perceptron, possuindo os métodos *train* e *predict*. O método *train* realiza o treinamento do neurônio de acordo com a Equação (2) descrita na Seção II-A e o método *predict* prediz uma saída dada uma entrada de acordo com a função da Equação (1) apresentada na Seção II-A.

A implementação foi feita em *Python*, utilizando a biblioteca *numpy* e a biblioteca *matplotlib* para plotagem de gráficos.

## IV. BASE DE DADOS

Para validar se a implementação foi feita corretamente, foram realizados experimentos em três bases de dados, sendo descritas a seguir:

### A. Iris Data Set

Uma das bases mais conhecidas na literatura utilizada para reconhecimento de padrões, sendo apresentada em [3]. Esta base possui 3 classes com 50 amostras cada classe e 5 características, sendo elas:

- Comprimento da sépala em cm
- Largura da sépala em cm

- Comprimento da pétala em cm
- Largura da pétala em cm
- Classes: *Iris Setosa*, *Iris Versicolor*, *Iris Virginica*

## V. EXPERIMENTOS E RESULTADOS

Para avaliar o método implementado, realizou-se dois experimentos, utilizando 10 iterações para treinamento e uma taxa de aprendizado  $\eta = 0.1$ . No primeiro experimento, foi utilizado o modelo de Perceptron implementado para resolver as funções AND, OR e XOR com duas entradas.

A Figura 2 apresenta os resultados do primeiro experimento. Os resultados foram demonstrados por gráficos onde o eixo  $X$  é a primeira entrada e o eixo  $Y$  é a segunda entrada. Os dados classificados como *False* são representados por um  $x$  vermelho e os dados classificados como *True* são representados por um círculo azul.

Os resultados obtidos mostram que o Perceptron implementado classificou corretamente as funções AND e OR, porém para a função XOR não foi possível classificar de maneira correta, já que a função XOR não pode ser corretamente classificada com uma única reta.

Para o segundo experimento, foi utilizada a base de dados Iris, descrita na Seção IV-A, como dados de entrada para o Perceptron. Utilizou-se o modelo de Perceptron para classificar as classes *Iris Setosa* e *Iris Versicolor*.

A Figura 3 apresenta o gráfico com o resultado do segundo experimento, onde o eixo  $x$  representa o tamanho das sépalas em cm e o eixo  $y$  representa a largura das sépalas em cm. A classe *Iris Setosa* é representada pelo  $x$  vermelho e a classe *Iris Versicolor* é representada pelo círculo azul.

Os resultados apontam que é possível classificar as duas classes utilizando o modelo de neurônio Perceptron sem nenhum problema, já que ele consegue discriminar os dados por uma reta.

## VI. CONCLUSÃO

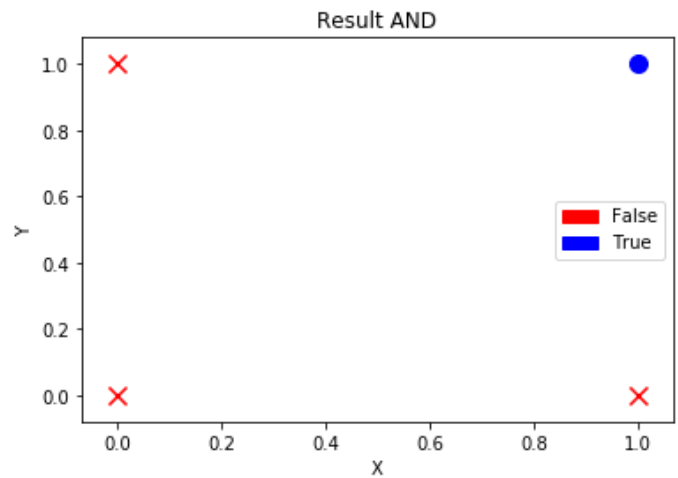
Neste trabalho foi implementado o modelo de neurônio artificial Perceptron para classificação binária de dados supervisionados, utilizando a base da Iris como exemplo de teste.

A avaliação do método foi feita através de dois experimentos diferentes. No primeiro experimento, o Perceptron foi testado para calcular as funções lógicas AND, OR e XOR. Para o segundo experimento, utilizou-se a base da Iris para ser classificada em duas classes, *Iris Setosa* e *Iris Versicolor*.

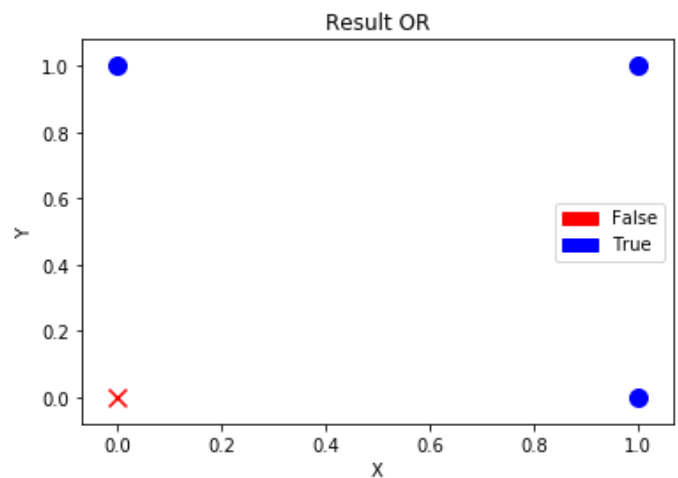
Os resultados do primeiro experimento mostraram que foi possível resolver corretamente as funções AND e OR. No entanto, para resolução da função XOR, não foi possível alcançar o resultado correto devido a limitação do Perceptron em separar dados com mais de uma reta.

No segundo experimento, foi possível classificar com sucesso duas das três classes presentes na base de dados da Iris. O modelo de Perceptron conseguiu discriminar corretamente as classes *Iris Setosa* e *Iris Versicolor*.

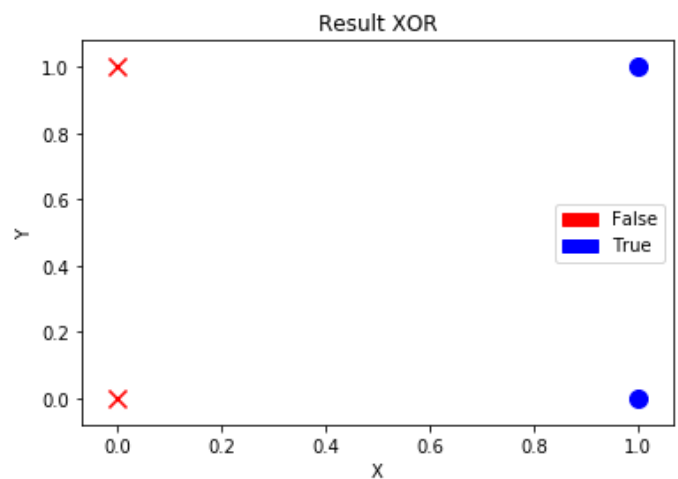
De forma geral, o Perceptron mostrou-se ser eficaz em classificar dados que conseguem ser divididos linearmente. No entanto, para dados não-linearmente separáveis o Perceptron não é uma boa solução. Para resolver esse tipo de problema,



(a) Resultado da classificação da função AND.



(b) Resultado da classificação da função OR.



(c) Resultado da classificação da função XOR.

Figura 2: Resultados do primeiro experimento.

o Perceptron será utilizado como base para a implementação de Multi-Layer Perceptron em trabalhos futuros que serão feitos para obter um melhor entendimento a respeito das Redes Neurais Artificiais.

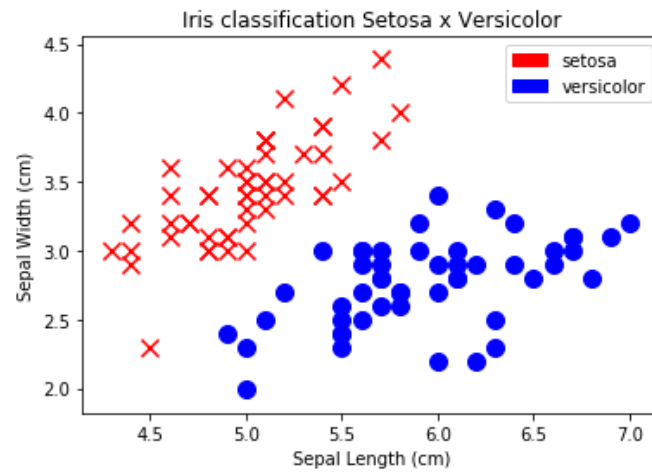


Figura 3: Resultado da classificação da base Iris.

#### REFERÊNCIAS

- [1] Warren Sturgis McCulloch and Walter F. Pitts. A logical calculus of the ideas immanent in nervous activity. 1990.
- [2] Frank F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958.
- [3] R. A. FISHER. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.