

Perceptron

November 16, 2020

1 Exercício - Perceptron

1. Implemente o Perceptron para classificar o dataset Iris, separando as classes “setosa” e “versicolor”.
2. Implemente o Perceptron Multiclasses e use no dataset Iris, separando as 3 classes (é possível?).

Não é possível. Com este modelo de rede neural artificial estamos limitados a classificações binárias

```
[6]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

def load_data():
    """
    Carrega os dados do dataset iris
    :return: dados carregados em uma matriz
    """
    data = pd.read_csv("https://archive.ics.uci.edu/ml/
↪machine-learning-databases/iris/iris.data", header=None)

    # utiliza somente as duas primeiras classes
    data = data[:100]
    # transforma as classes em 0 e 1
    data[4] = np.where(data.iloc[:, -1] == 'Iris-setosa', 0, 1)
    data = np.asmatrix(data, dtype='float64')
    return data

def plot_data(data):
    """
    Exibe os dados
    :param data: dados do dataset iris
    """
```

```

plt.scatter(np.array(data[:50, 0]), np.array(data[:50, 2]), marker='o',
→label='setosa')
plt.scatter(np.array(data[50:, 0]), np.array(data[50:, 2]), marker='x',
→label='versicolor')
plt.xlabel('petal length')
plt.ylabel('sepal length')
plt.legend()
plt.show()

def perceptron(data, num_iter):
    """
    Rede neural artificial: Perceptron
    :param data: dados do dataset iris
    :param num_iter: número de iterações
    """
    input = data[:, :-1]
    labels = data[:, -1]

    # inicia o vetor de pesos com 0
    w = np.zeros(shape=(1, input.shape[1] + 1))

    erro_classificacao_epoch = []

    for epoch in range(num_iter):
        erro_classificacao = 0
        for x, label in zip(input, labels):
            x = np.insert(x, 0, 1)
            y = np.dot(w, x.transpose())
            if y > 0:
                target = 1.0
            else:
                target = 0.0

            delta = (label.item(0, 0) - target)

            if delta:
                erro_classificacao += 1
                w += (delta * x)

        erro_classificacao_epoch.append(erro_classificacao)

    return w, erro_classificacao_epoch

def plot_error(class_incorreto):
    """

```

```

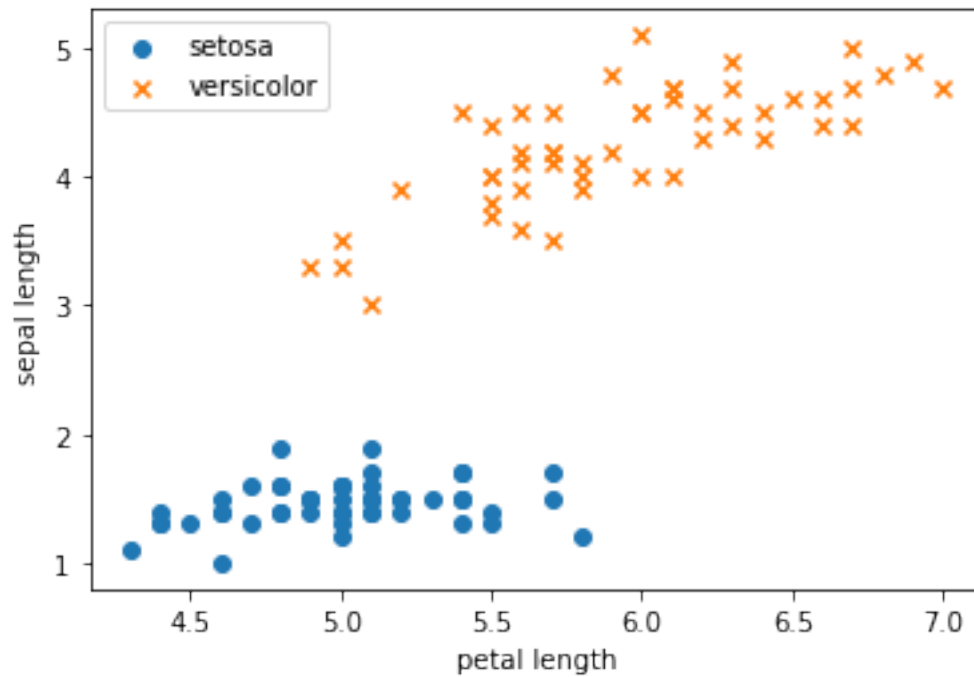
Exibe o erro ao longo das epochs
:param class_incorreto: dados classificados incorretamente
"""
epochs = np.arange(1, num_iter + 1)
plt.plot(epochs, class_incorreto)
plt.xlabel('Iterações')
plt.ylabel('Classificados incorretamente')
plt.show()

```

```

[7]: data = load_data()
plot_data(data)

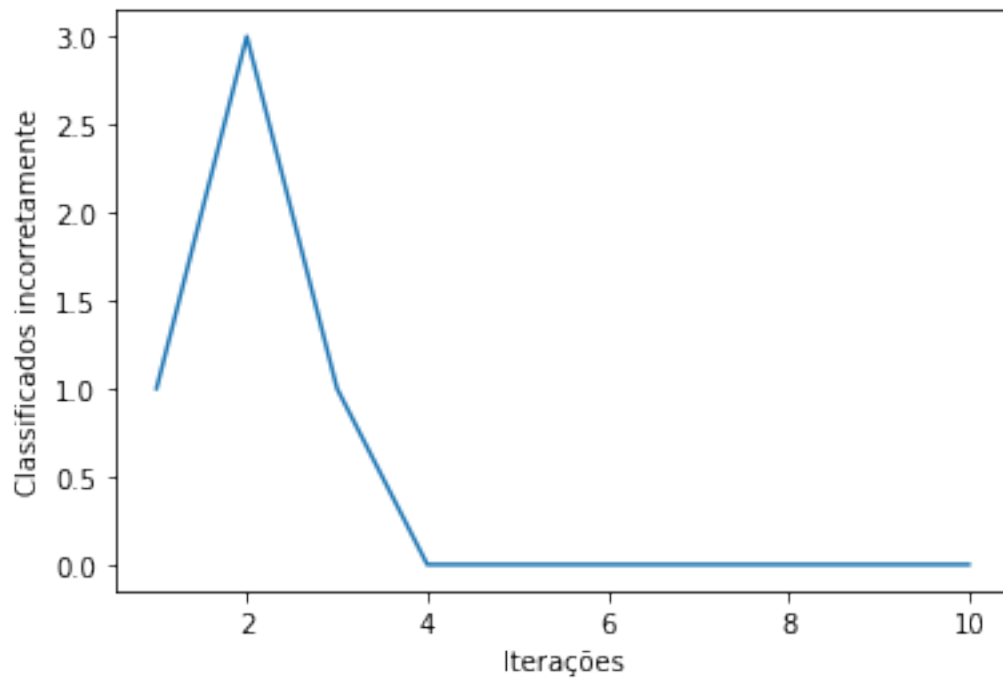
```



```

[8]: num_iter = 10
w, erro_classificacao = perceptron(data, num_iter)
plot_error(erro_classificacao)

```



[]: