

Atividade: Neurônio Artificial

INSTITUTO FEDERAL DE MINAS GERAIS

Departamento de Engenharia e Computação

Professor: Ciniro Nametala Leite

Aluno: Luan Carlos dos Santos

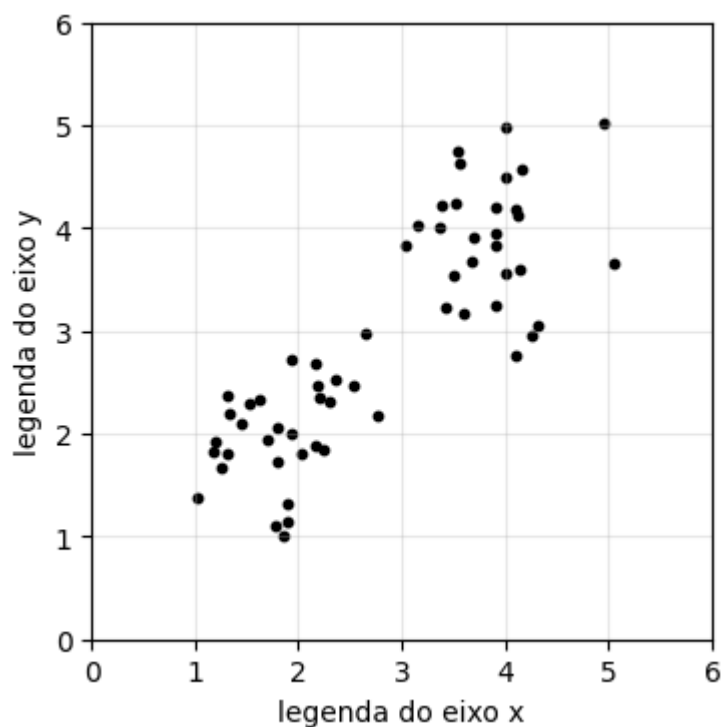
```
In [6]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# somatorio do produto de x pelo seu erro
def neuronio(x1, x2, bias, w0, w1, w2):
    u = (x1 * w1) + (x2 * w2) + (bias * w0)
    return 1 if u > 0 else 0

# leitura dos dados
dataset = pd.read_csv('amostravariada.csv', sep=';', decimal=',')

# convertendo os dados
dataset['x'] = pd.to_numeric(dataset['x'], errors='coerce')
dataset['y'] = pd.to_numeric(dataset['y'], errors='coerce')

plt.figure(figsize=(4, 4))
plt.scatter(dataset['x'], dataset['y'], c='black', s=10) # configurações do gráfico de dispersão
plt.xlim(0,6) # tamanho do eixo x
plt.ylim(0,6) # tamanho do eixo y
plt.xlabel("legenda do eixo x")
plt.ylabel("legenda do eixo y")
plt.grid(True, alpha=0.3) # opacidade das linhas do grid
plt.show() # plotando o gráfico
```



Classificação

```
In [7]: # Erros
w0 = 6 # termo do viés
w1 = 1 # erro 1
w2 = 1 # erro 2
bias = -1 # viés

classificacoes = []
for i, linha in dataset.iterrows():
    classe = neuronio(linha['x'], linha['y'], bias, w0, w1, w2)
    classificacoes.append(classe)
dataset['classe'] = classificacoes
dataset.head()
```

```
Out[7]:
```

	x	y	classe
0	1.183988	1.832880	0
1	1.523565	2.293337	0
2	2.199241	2.342880	0
3	2.768052	2.179136	0
4	2.165374	1.888445	0

Gráfico

```
In [8]: plt.figure(figsize=(4,4))
cores = ['blue' if c == 0 else 'green' for c in dataset['classe']]

# gerando a reta de separação
eixox = np.linspace(0, 6, 100)
eixoy = w0 - eixox

plt.scatter(dataset['x'], dataset['y'], c = cores, s=10)
plt.plot(eixox, eixoy, 'r-', linewidth=1)
plt.xlim(0,6)
plt.ylim(0,6)
plt.xlabel("Eixo x")
plt.ylabel("Eixo y")
plt.grid(True, alpha=0.5)
plt.show()
```

