

#### 4o LISTA DE EXERCÍCIOS - CAP 10

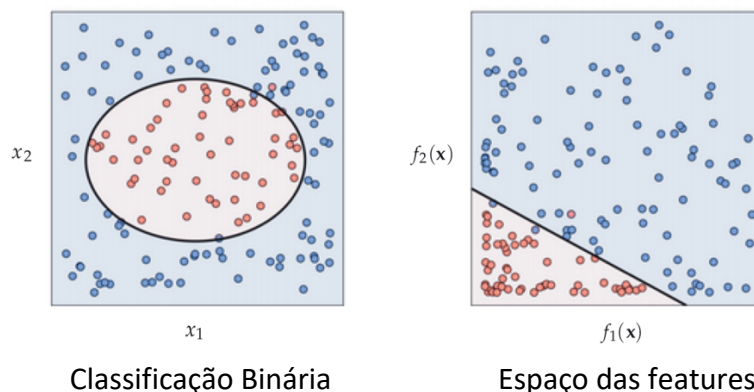
1) No exemplo em sala de aula da regressão não-linear, implemente a normalização dos dados ( $x_{\text{new}} = (x_{\text{old}} - \text{media})/\text{desvio\_padrao}$ ) e compare a convergência, exibindo a comparação dos dois gráficos da função custo: normalizado e não-normalizado.

2) Implementar a regressão não-linear para o caso bi-dimensional, usando o conjunto gerado abaixo:

```
import numpy as np
def pontos_2d(P):
    np.random.seed(1)
    X = [5*np.random.random_sample(P), 5*np.random.random_sample(P)]
    X = np.array(X)
    Y = np.sin(X[0,:]+X[1,:]) #+ 0.2 * np.random.random_sample(P)
    return X,Y

import matplotlib.pyplot as plt
P = 300
[X,Y] = pontos_2d(P)
ax = plt.axes(projection='3d')
ax.view_init(10, 115)
for i in range(P):
    ax.scatter(X[0,:], X[1,:], Y, c=Y, cmap='viridis', linewidth=0.5)
```

3) No exemplo de sala de aula da classificação binária, implemente a visualização do espaço das features, como na figura 10.11 (direita) reproduzida a seguir.



4) Para o modelo do Autoencoder, implemente a função custo (10.53) da pag 296:

$$g(\Theta) = \frac{1}{P} \sum_{p=1}^P \left\| \mathbf{f}_d(\mathbf{f}_e(\mathbf{x}_p)) - \mathbf{x}_p \right\|_2^2.$$

com as funções não-lineares descritas na página 297:

$$\mathbf{f}_e(\mathbf{x}_p) = \theta_p = \arctan\left(\frac{x_{p,2} - w_2}{x_{p,1} - w_1}\right) \quad \mathbf{f}_d(\theta_p) = \begin{bmatrix} r \cos(\theta_p) + w_1 \\ r \sin(\theta_p) + w_2 \end{bmatrix}$$

E com os dados gerados pelo código abaixo:

```
import numpy as np
def pontos_2d(P):
    np.random.seed(1)
    X = [np.zeros(P), np.zeros(P)]
```

```
X = np.array(X)
theta = -1.5
dt = 3.14/P
for i in range(P):
    r = 5 + 2 * np.random.random_sample(1)
    X[0][i] = r * np.cos(theta) + 2
    X[1][i] = r * np.sin(theta) - 1.2
    theta += dt
return X
```

Exercícios para serem entregues: 2) e 4)

Entrega da Lista: 03/Dezembro