SIGMOIDAL

AULA 005

DEEP LEARNING

Alex Aparecido Ruiz - alexapruiz@hotmail.com - CPF: 199.954.008-55

Praticamente, tudo que se chama deep learning é movido por um algoritmo muito importante: Stochastic Gradient Descent (SGD).

- Goodfellow et al (2016)

MODELOS PARAMÉTRICOS

Tem a capacidade de aprender padrões a partir de dados de input durante uma etapa conhecida como treinamento.

MODELOS PARAMÉTRICOS

Esse modelo é representado não por todos os dados que o alimentam, mas por um número limitado de parâmetros.

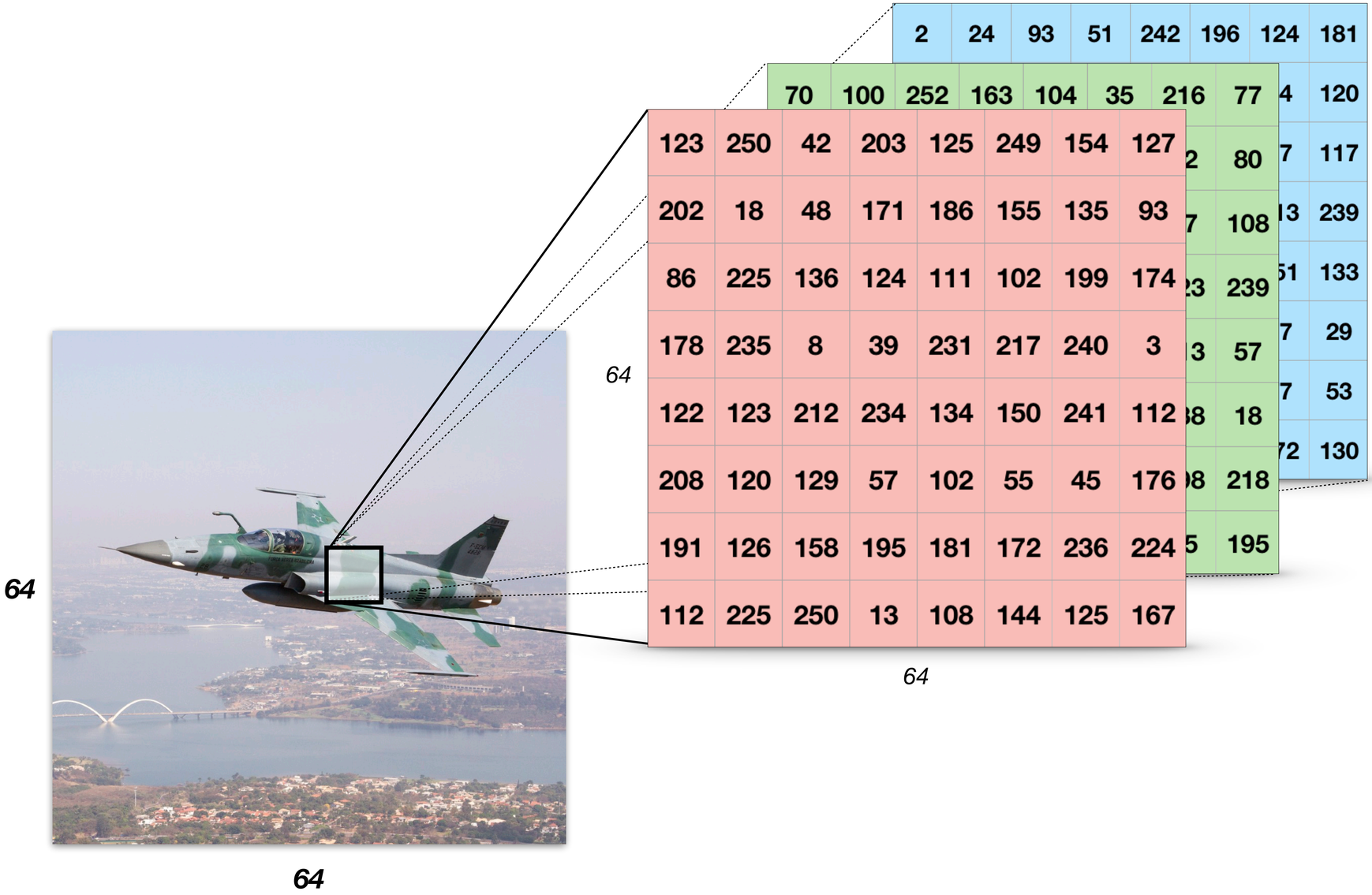
MODELOS PARAMÉTRICOS

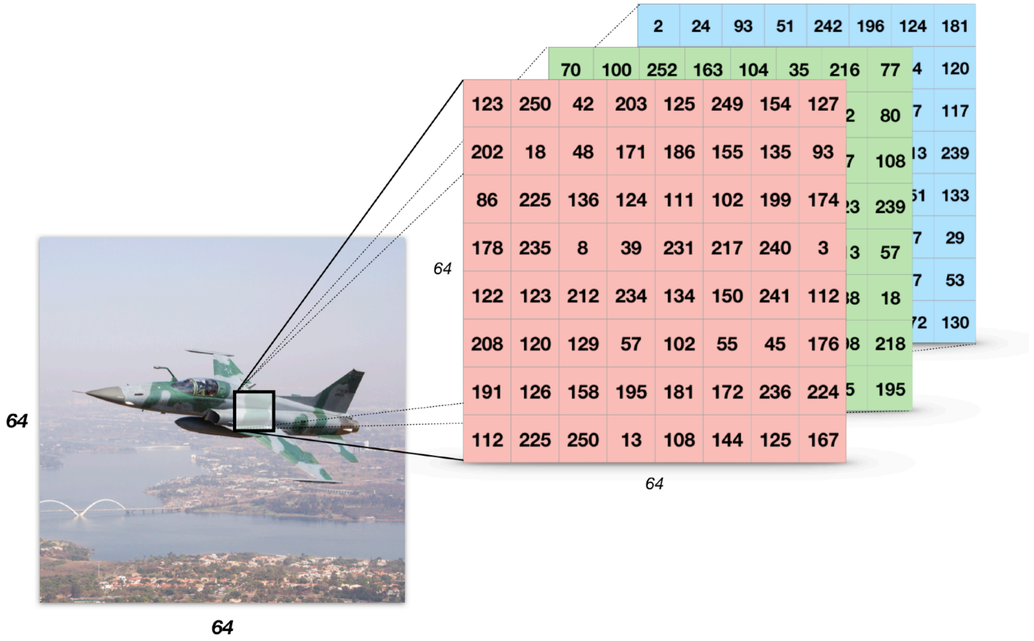
Independente do tamanho do meus dados de treino, é possível generalizar, representar os mesmos por meio de parâmetros.

1 (avião)

0 (não-avião)

1 (avião)





$$x = \begin{bmatrix} 123 \\ 250 \\ \vdots \\ 70 \\ 100 \\ \vdots \\ 130 \end{bmatrix}$$

$$y = 1$$

$$n_x = 64 \times 64 \times 3 = 12288$$

$$m = 1$$

$$x^{(1)} = \begin{bmatrix} 12 \\ 122 \\ \vdots \\ 163 \end{bmatrix}$$

$$x^{(2)} = \begin{bmatrix} 231 \\ 146 \\ \vdots \\ 0 \end{bmatrix}$$

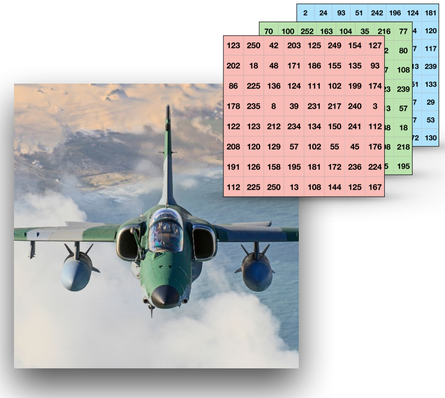
$$x^{(3)} = \begin{bmatrix} 22 \\ 13 \\ \vdots \\ 89 \end{bmatrix}$$

$$X = \begin{bmatrix} 12 & 231 & 22 \\ 122 & 146 & 13 \\ \vdots & \vdots & \vdots \\ 163 & 0 & 89 \end{bmatrix}$$

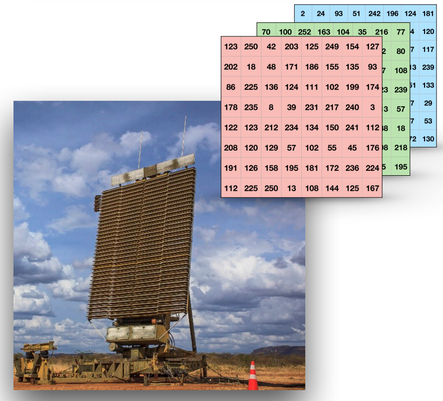
$$Y = [1,0,1]$$

$$n_x = 12288$$

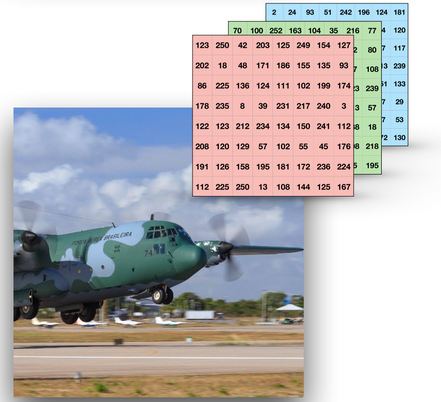
$$m = 3$$



$$x^{(1)} = \begin{bmatrix} 12 \\ 122 \\ \vdots \\ 163 \end{bmatrix}$$



$$x^{(2)} = \begin{bmatrix} 231 \\ 146 \\ \vdots \\ 0 \end{bmatrix}$$



$$x^{(3)} = \begin{bmatrix} 22 \\ 13 \\ \vdots \\ 89 \end{bmatrix}$$

$$X = \begin{bmatrix} 12 & 231 & 22 \\ 122 & 146 & 13 \\ \vdots & \vdots & \vdots \\ 163 & 0 & 89 \end{bmatrix}$$

$$Y = [1,0,1]$$

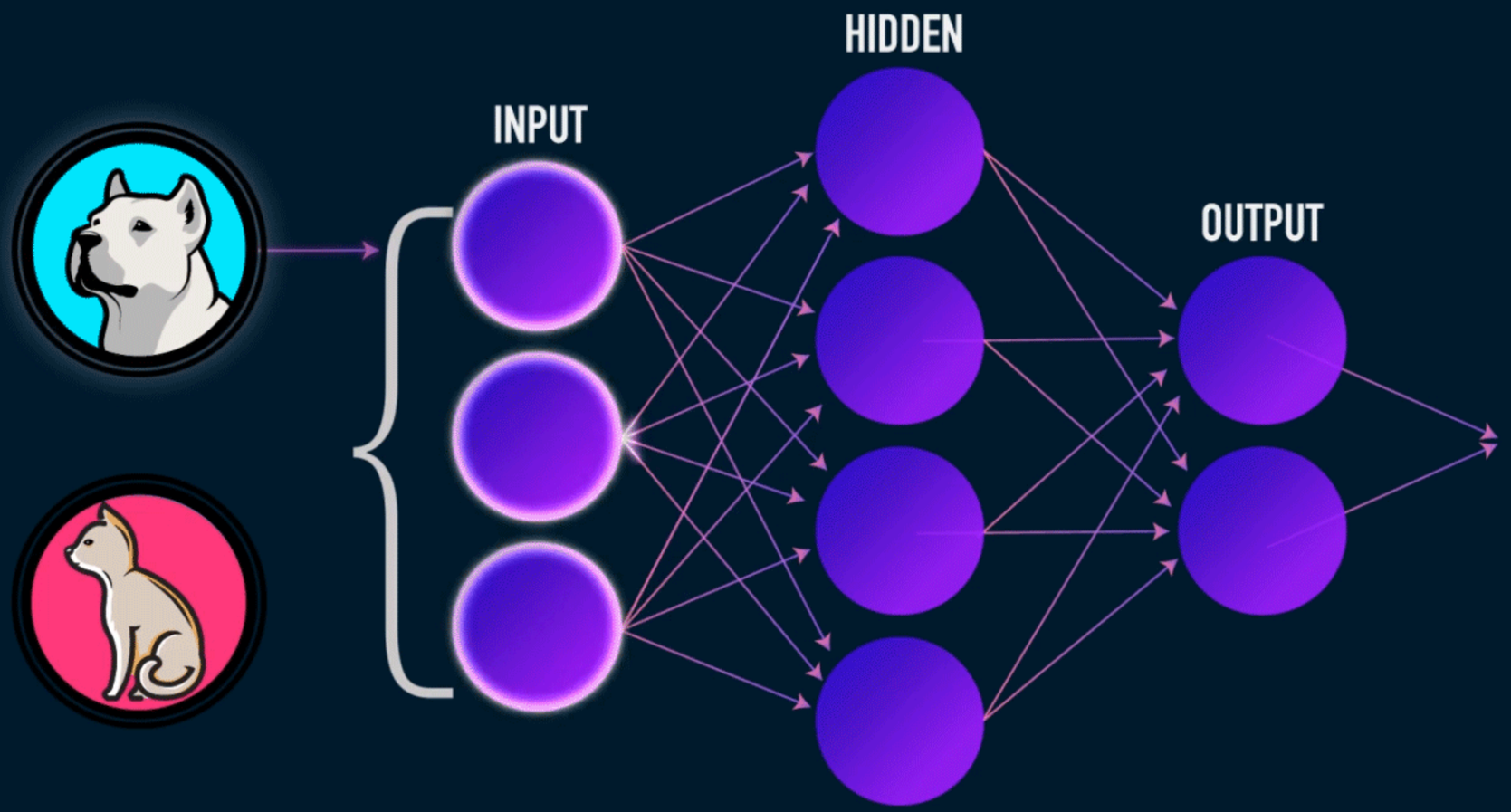
$$n_x = 12288$$

$$m = 3$$

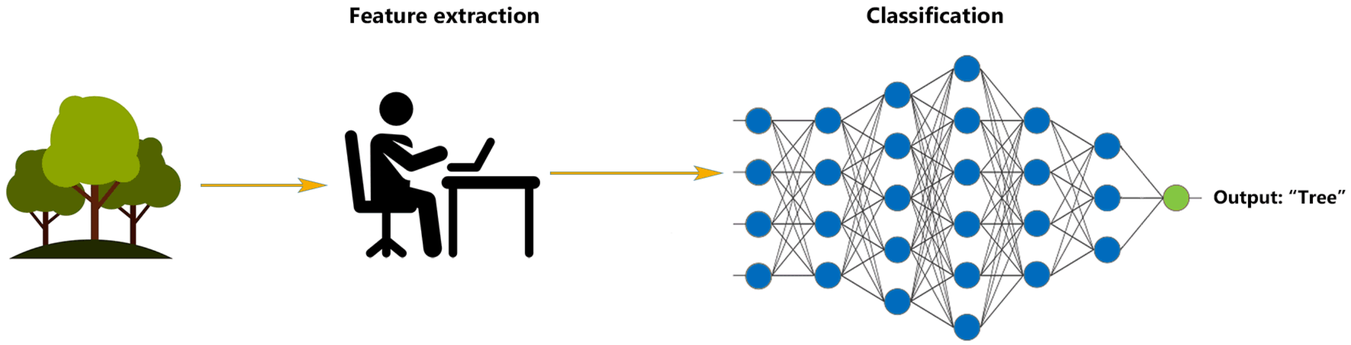
Para um conjunto de *m* amostras de treinamento

***m*-amostras:** $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$

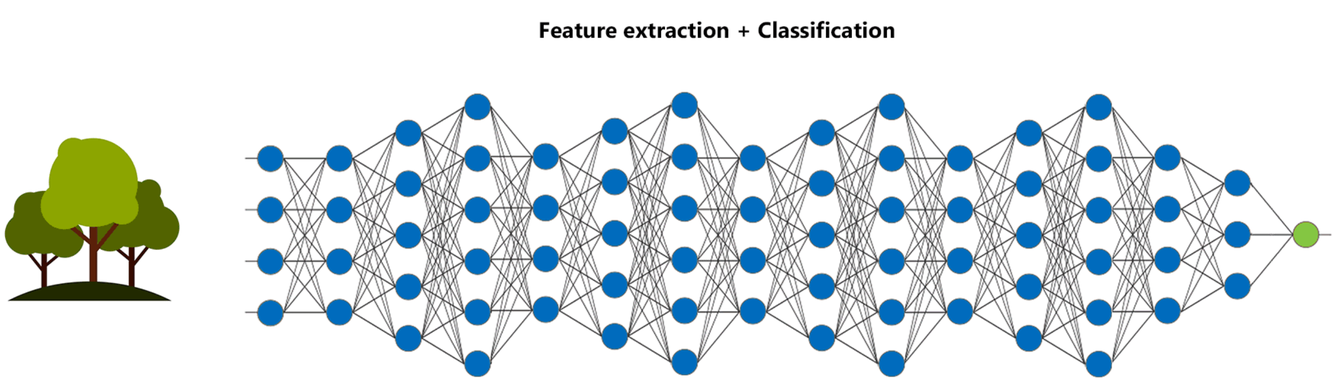
$$X = \begin{bmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \ldots & x^{(m)} \\ | & | & & | \end{bmatrix} \qquad Y = \left[y^{(1)}, y^{(1)}, \ldots, y^{(m)}\right]$$

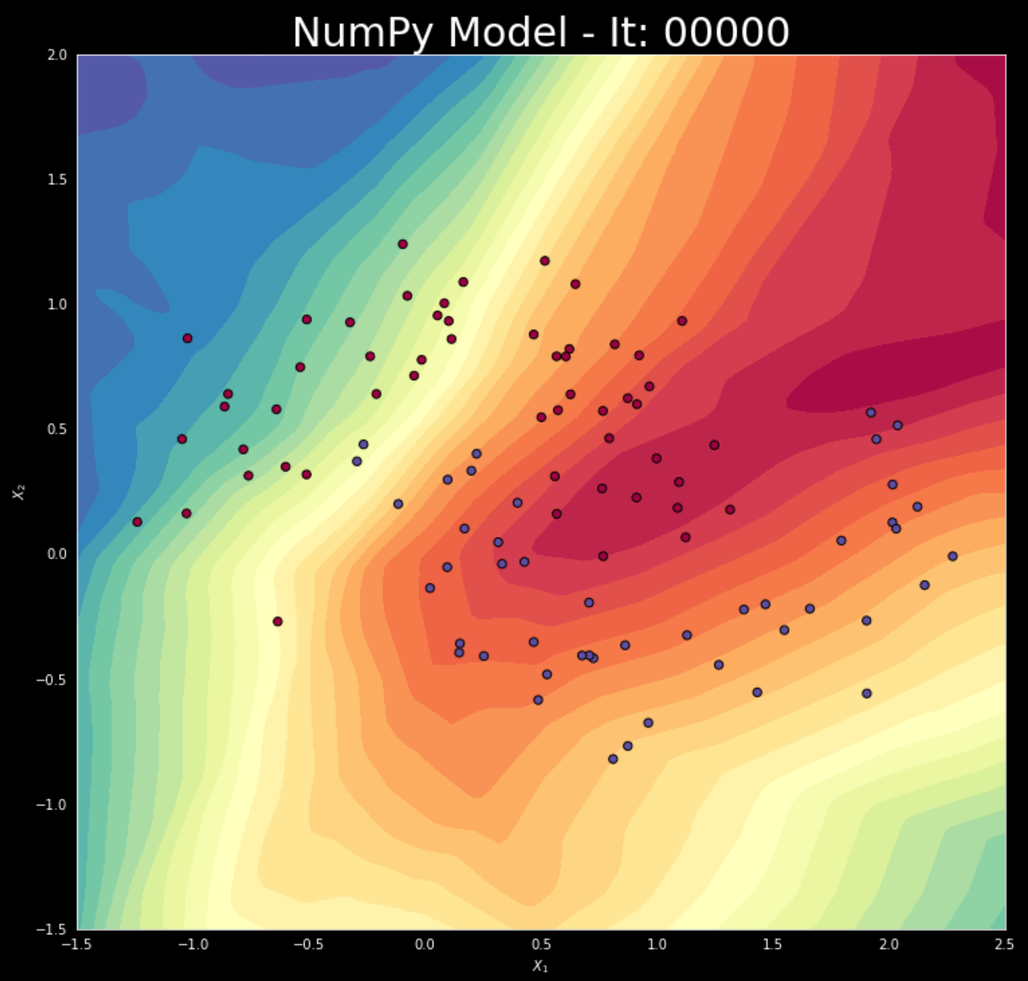


Machine Learning

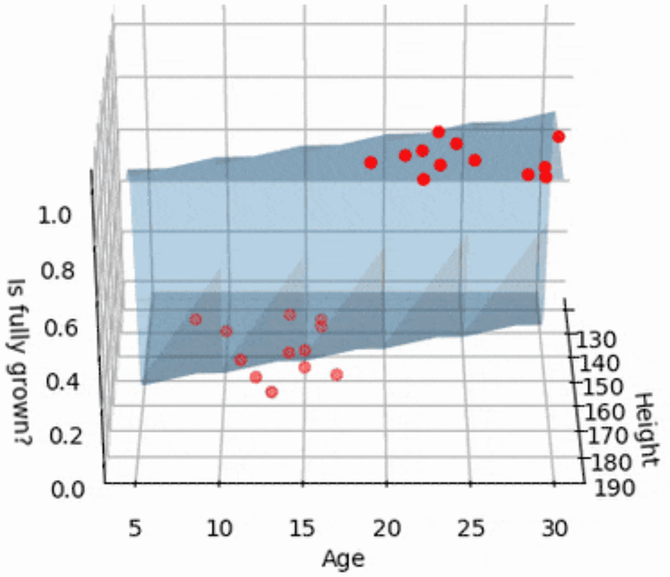


Deep Learning

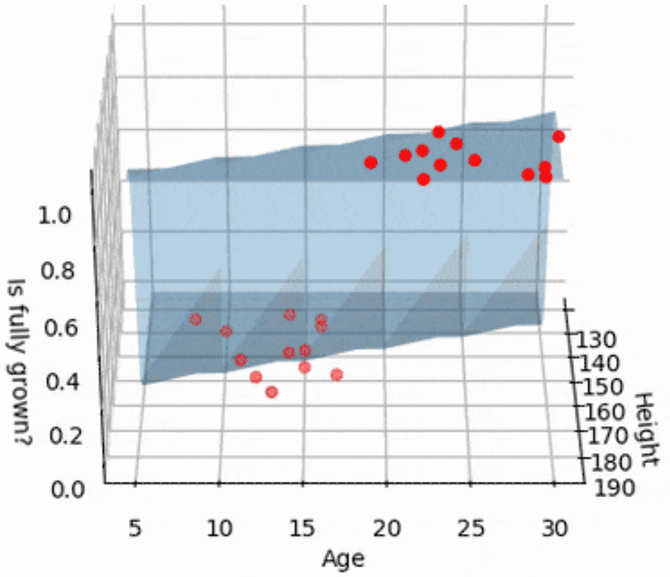



NumPy Model - It: 00000
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
x_2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
x_1

REGRESSÃO LOGÍSTICA



REGRESSÃO LOGÍSTICA



$$\hat{y} = P(y = 1 \mid x)$$

1. SCORE FUNCTION

A score function é uma função que vai receber o nosso input (no exemplo usado aqui, uma imagem) e mapear os dados para as classes de labels.

$$f(x, \theta, b) = \hat{y} = \theta^T x + b$$

Lembra que estamos falando de classificação de imagens, e da probabilidade dessa imagem ser um avião ou não?

O problema é que a nossa score function, do jeito que está, pode retornar qualquer valor.

$$0 \leq \hat{y} \leq 1$$

FUNÇÕES DE ATIVAÇÃO

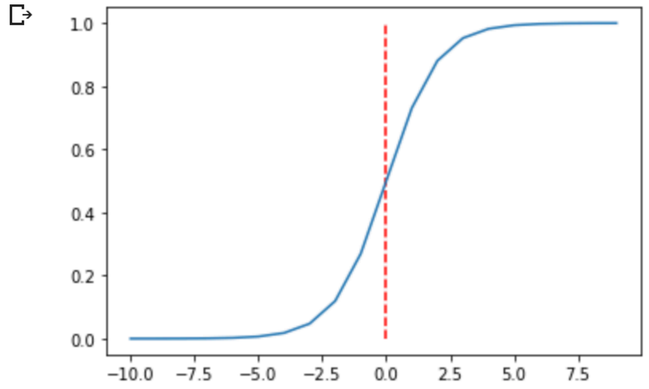
FUNÇÃO SIGMÓIDE

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

```
[ ]  # importar pacotes necessários
     import numpy as np
     import matplotlib.pyplot as plt

     # criar um set entre -10 e 10 e aplicar a função sigmóide
     x = np.arange(-10, 10)
     y = 1 / (1 + np.exp(-x))

     # plotar a curva sigmoidal
     plt.plot(x, y)
     plt.vlines(0, 0, 1, colors='r', linestyles='dashed')
     plt.show()
```

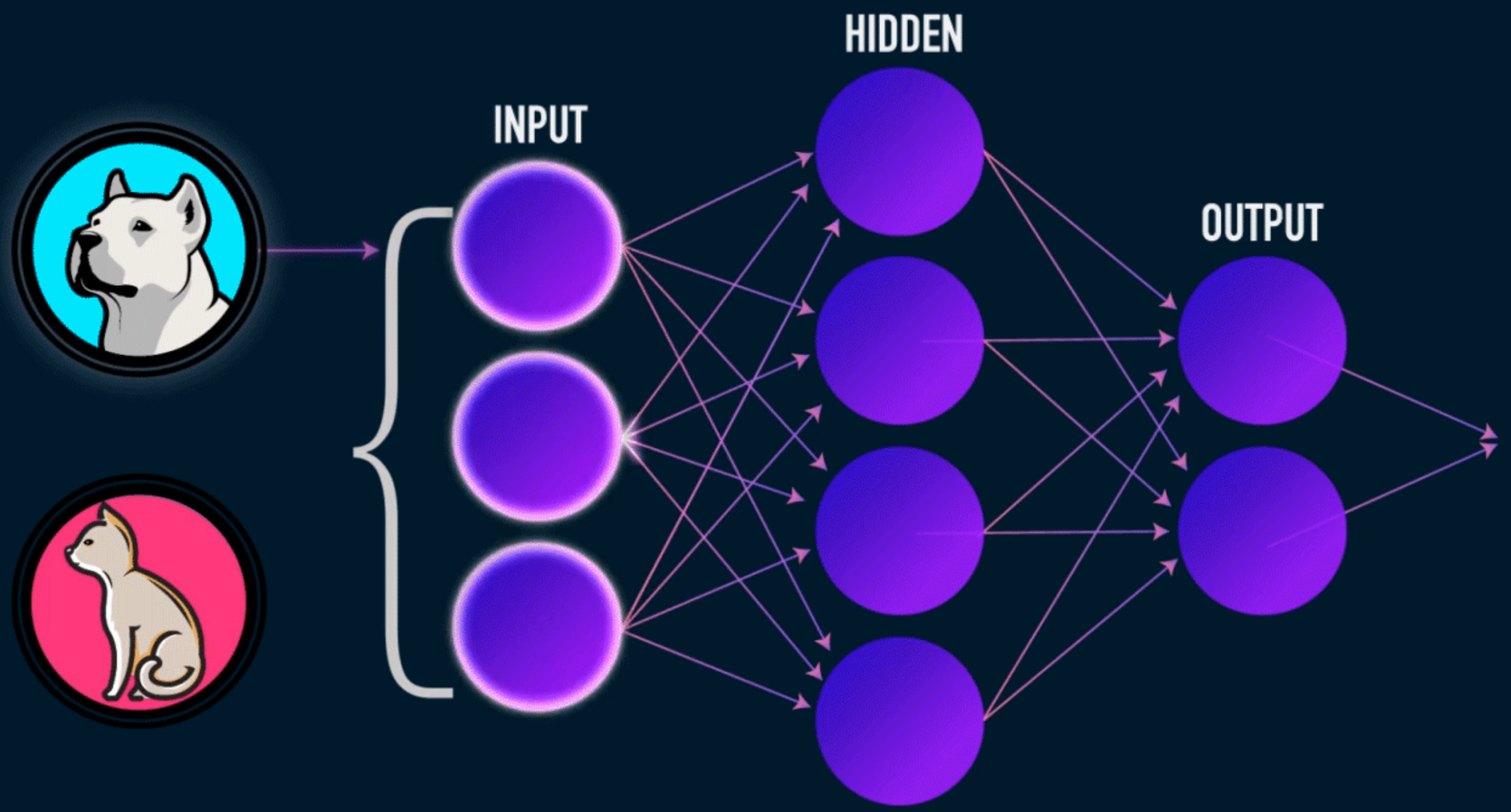


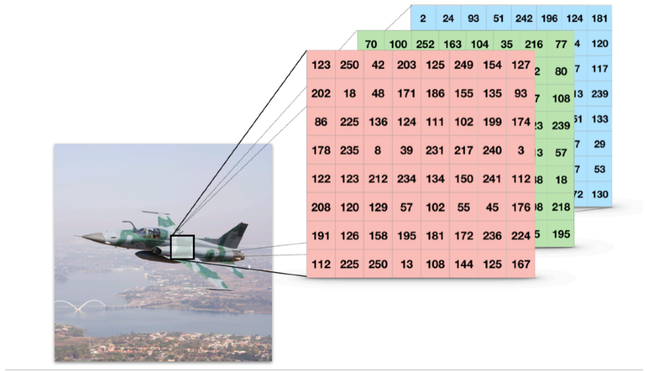
1. SCORE FUNCTION

A score function é uma função que vai receber o nosso input (no exemplo usado aqui, uma imagem) e mapear os dados para as classes de labels.

$$f(x, \theta, b) = \hat{y} = \theta^T x + b$$

$$\hat{y} = \sigma(\theta^T x + b) = \frac{1}{1 + e^{-(\theta^T x + b)}}$$





Transformar em vetor

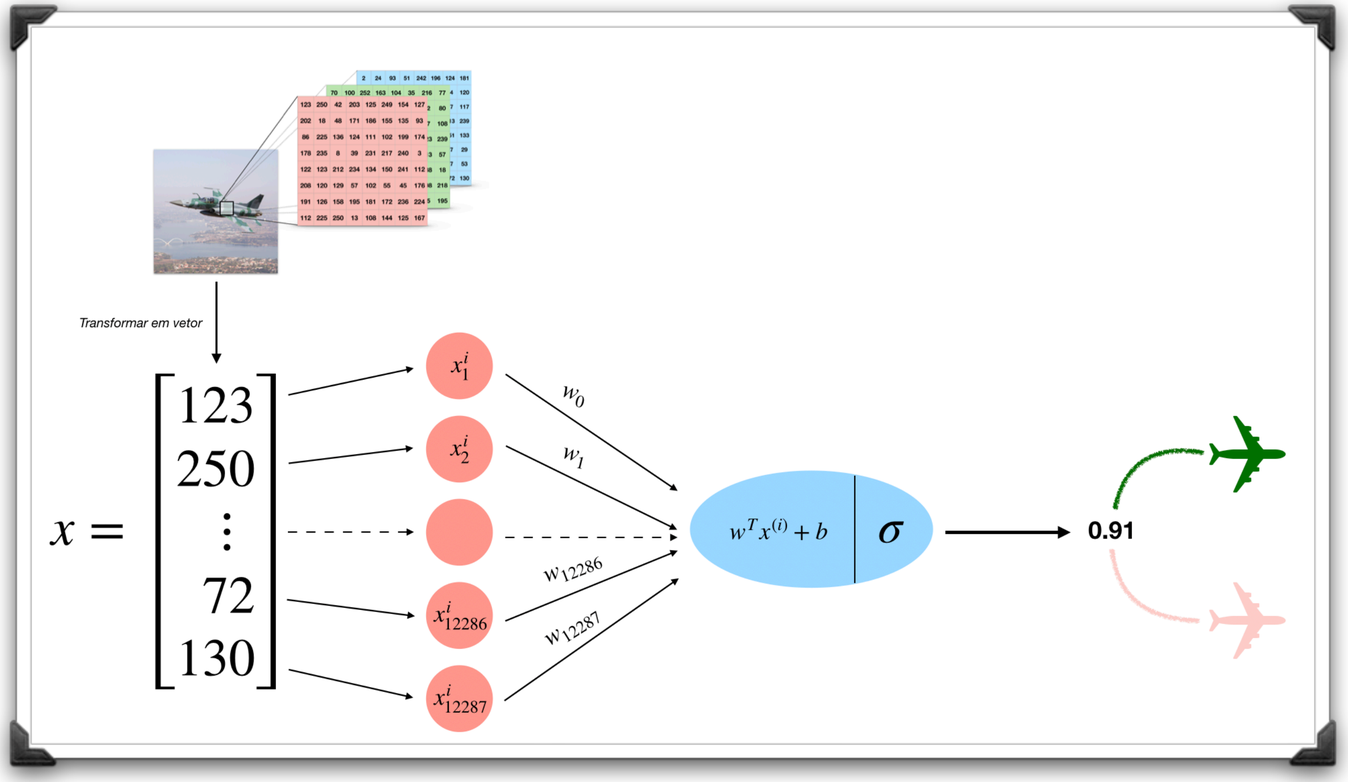
$$x = \begin{bmatrix} 123 \\ 250 \\ \vdots \\ 72 \\ 130 \end{bmatrix}$$

x_1^i x_2^i x_{12286}^i x_{12287}^i

w_0 w_1 w_{12286} w_{12287}

$w^T x^{(i)} + b \mid \sigma$

0.91



1. Transformar a imagem em um vetor.
2. Multiplicar o valor de cada pixel de x pelo seu peso w.
3. Obter o valor de z.
4. Obter a probabilidade de ser um avião. Ou seja, obter um valor entre 0 e 1.
5. Classificar a imagem com o label “*avião*”.