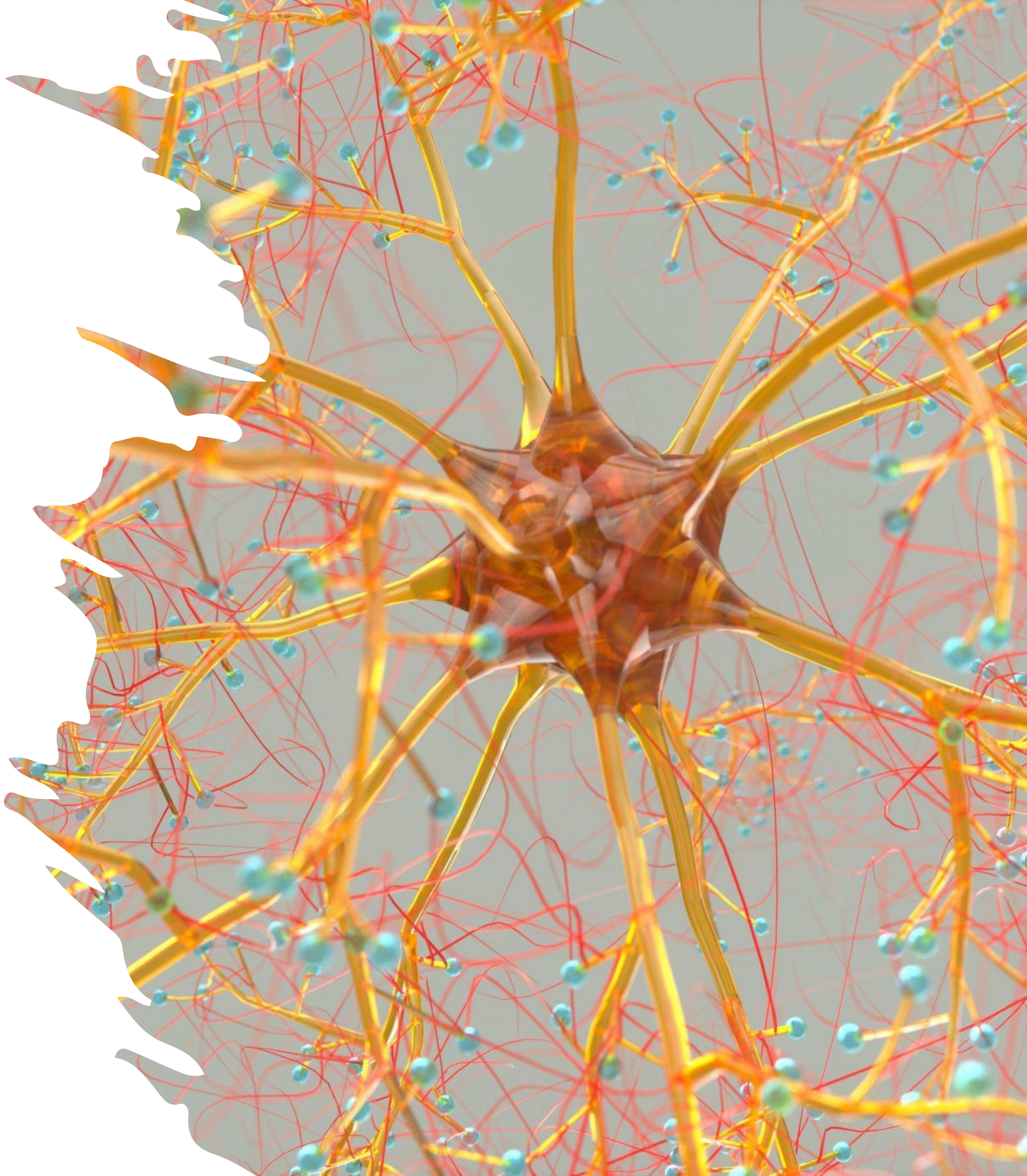
CS826

Neural Networks

Week 1 - Summary



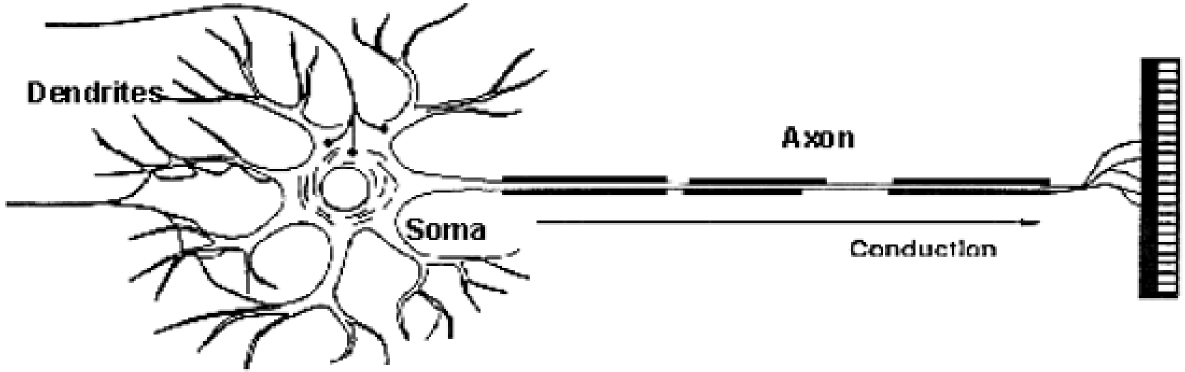
Week 1: Review and Getting Started with Neural Networks

This week you will be introduced to:

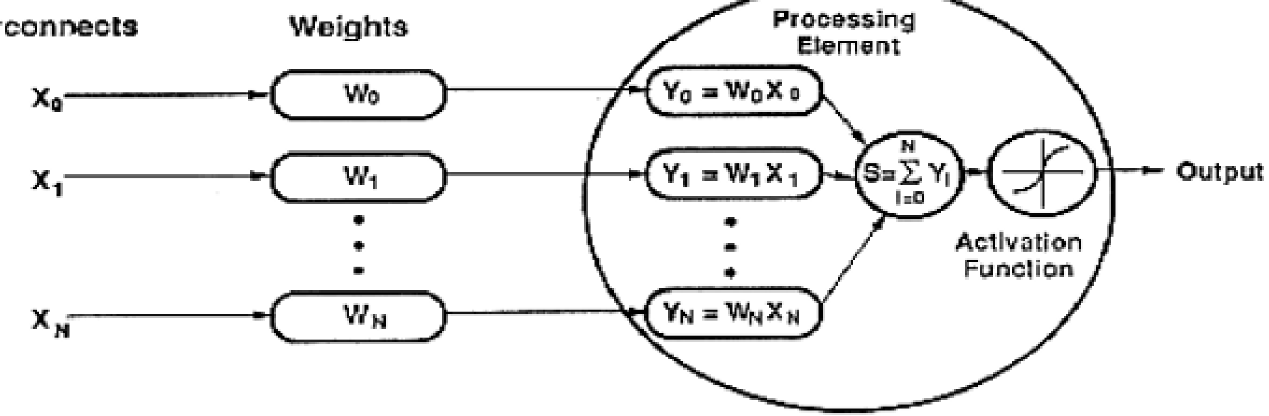
1. What is deep learning? ✅
2. Motivation, challenges, and applications ✅
3. Architecture and maths ✅
4. Simple examples
5. Google Colab for practical sessions

How do ANNs work?

Biological Neuron

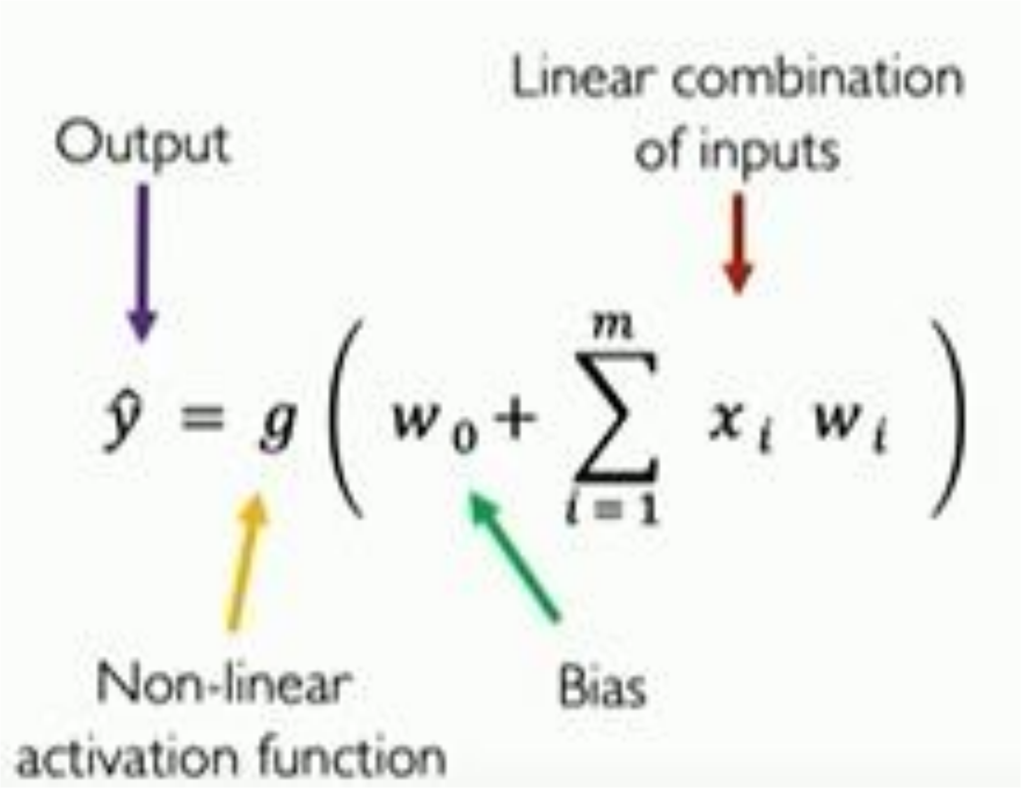


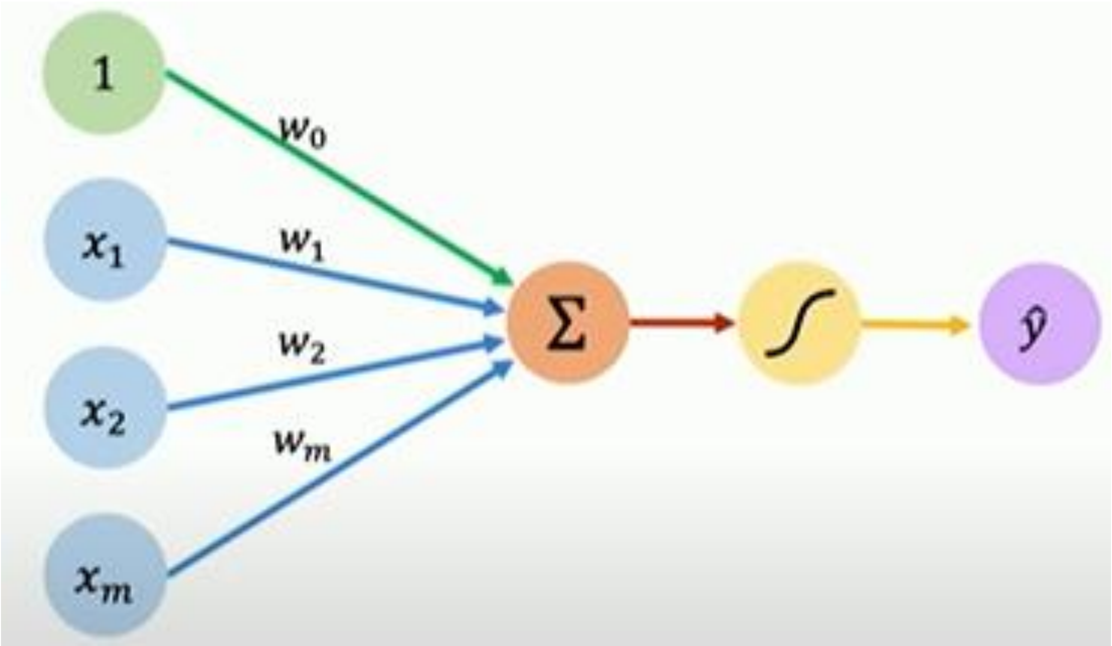
Artificial Neuron



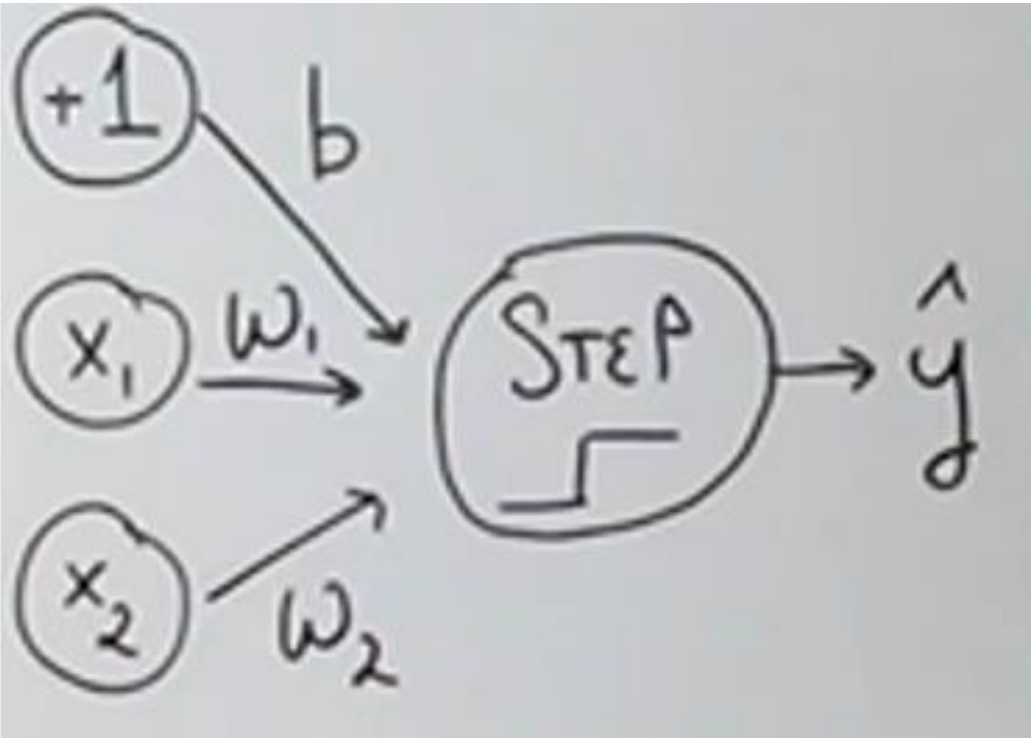
An artificial neuron is an imitation of a human neuron

Forward Propagation - Perceptron





Perceptron



Let the input feature vector be:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad y \in \{1, 0\}$$

where y is the binary response.

The perceptron model computes a weighted sum of inputs:

$$z = w_1 x_1 + w_2 x_2 + b$$

The perceptron then applies a step function to determine the predicted label $\hat{y}$:

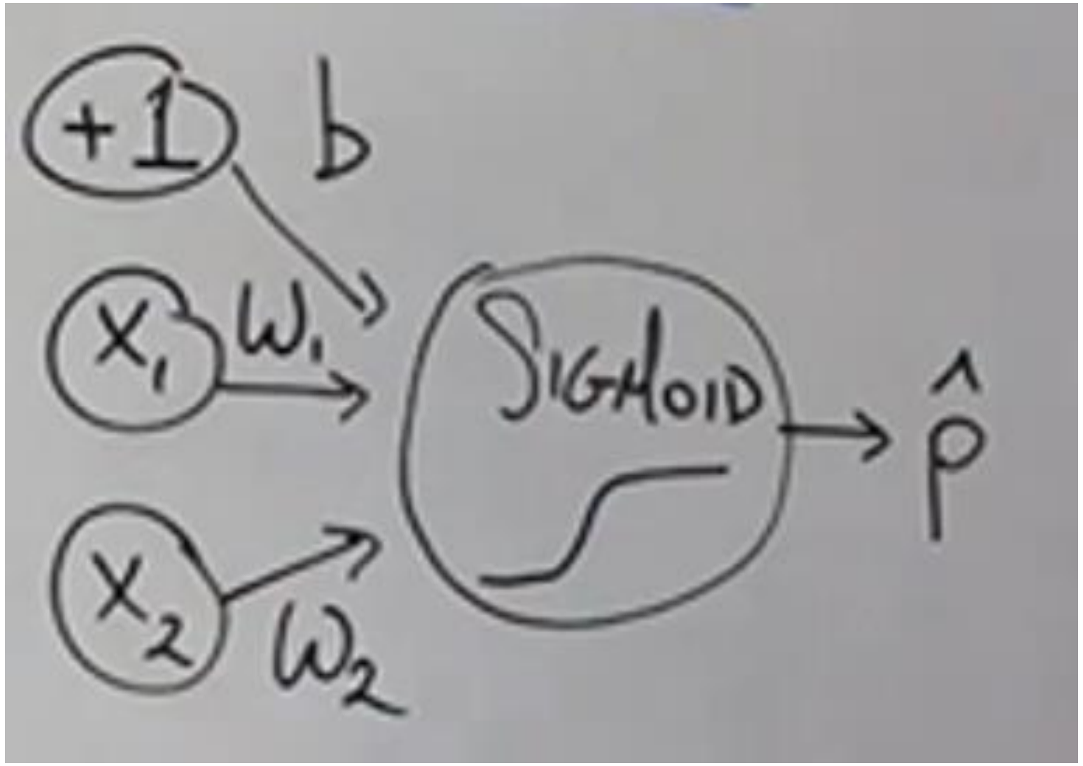
$$\hat{y} = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$$

Logistic Regression

Let the input feature vector be:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad y \in \{1, 0\}$$

where y is the binary response.



In logistic regression, we compute the same linear combination of inputs:

$$z = w_1 x_1 + w_2 x_2 + b$$

but instead of a step function, we apply the sigmoid function $\sigma(z)$ to get the predicted probability $\hat{p}$:

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

Neural Network

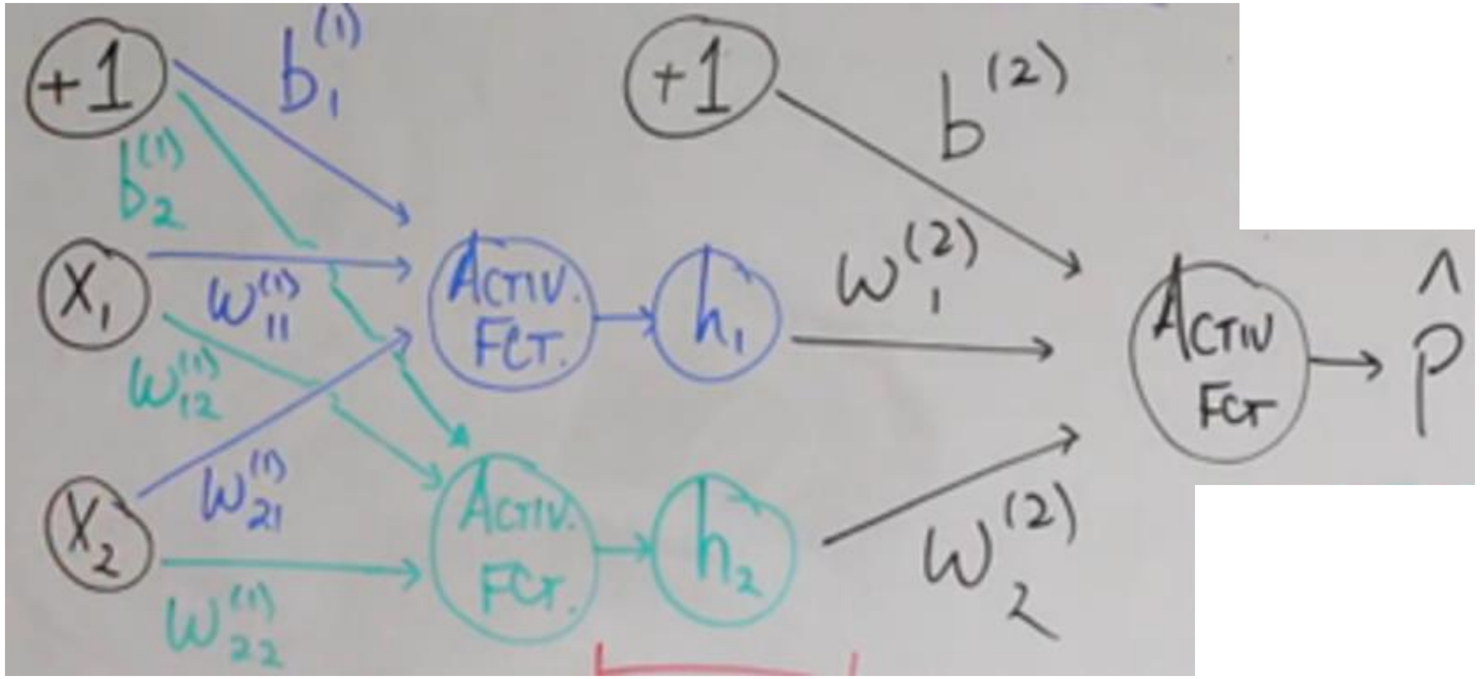
$$z_1^{(1)} = W_{11} X_1 + W_{21} X_2 + b_1^{(1)}$$

$$z_2^{(1)} = W_{12} X_1 + W_{22} X_2 + b_2^{(1)}$$

$$h_1 = \sigma(z_1^{(1)}) \quad \text{and} \quad h_2 = \sigma(z_2^{(1)})$$

$$z^{(2)} = w_1^{(2)} h_1 + w_2^{(2)} h_2 + b^{(2)}$$

$$\hat{p} = \sigma(z^{(2)})$$



Matrix Form:

$$\mathbf{z}^{(1)} = \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}$$

$$\mathbf{W}^{(1)} = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} \end{bmatrix}, \quad \mathbf{b}^{(1)} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(1)} \end{bmatrix}$$

$$\mathbf{h} = \sigma(\mathbf{z}^{(1)})$$

$$z^{(2)} = \mathbf{w}^{(2)}\mathbf{h} + b^{(2)}$$

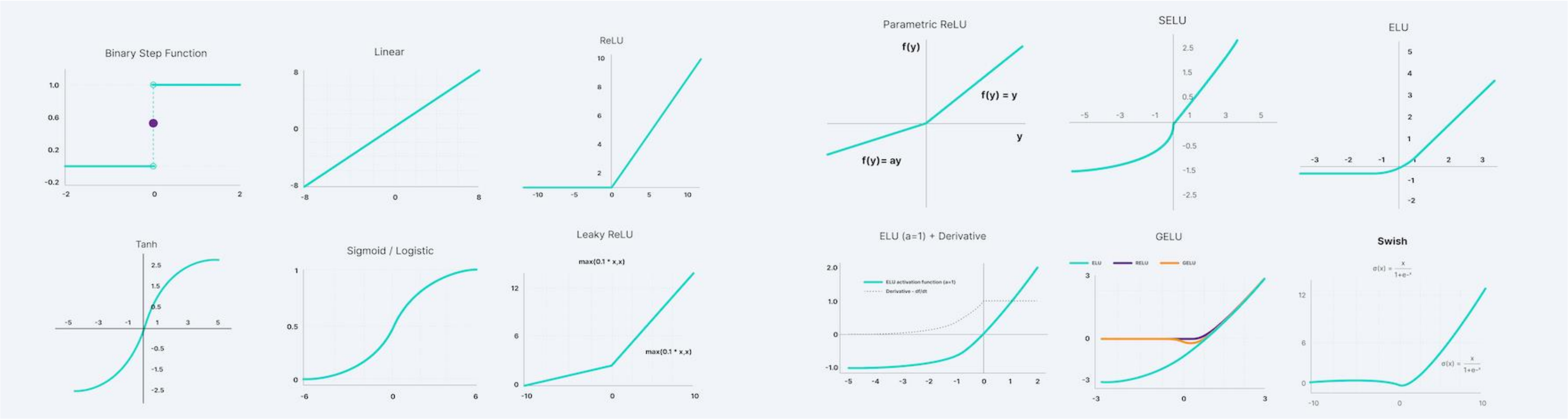
$$\hat{p} = \sigma(z^{(2)})$$

Vital and Lingering Questions

Some questions regarding neural network architecture and training include:

- How to get weights and biases (w and b)?
- How many hidden layers should be used?
- What activation function to choose?

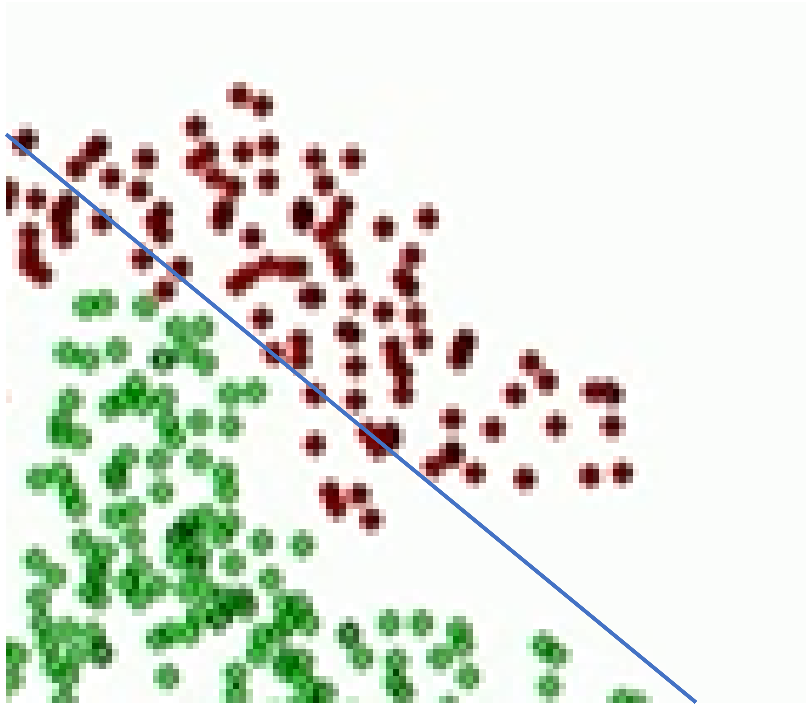
Activation Functions



Source: v7labs.com

Importance of Activation Functions

Distinguish Red vs Green points



Introduce
non-linearities
to the network

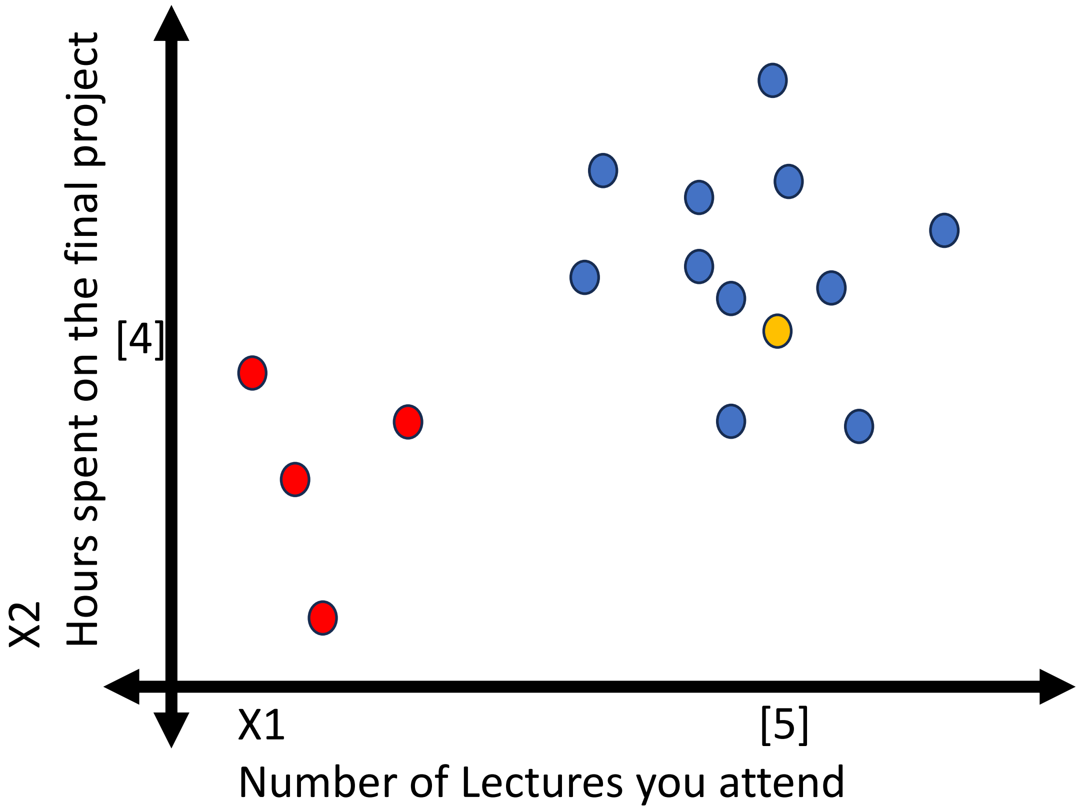
What is bias, how is used
and why it is important?

Toy Problem – Motivating Example

Two Feature Model:

X1 = Number of lectures you attend
X2 = Hours spent on the final group Project

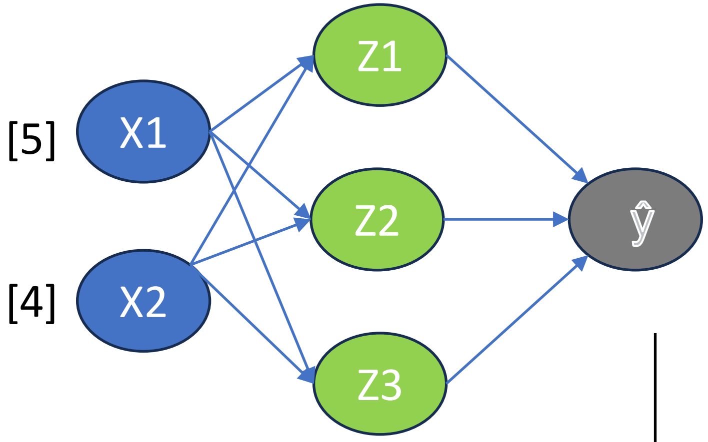
WILL I PASS THIS CLASS?



PASS

FAIL

X = [5,4]



Predicted: $\hat{y} = 0.2$
Actual: $y = 1$

Why NN is so wrong?

Not trained properly!

Loss function

$$\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}\Big(\underbrace{f\big(x^{(i)};\boldsymbol{W}\big)}_{\text{Predicted}},\underbrace{y^{(i)}}_{\text{Actual}}\Big)$$

$$\begin{array}{ccc} f(x) & & y \\ \begin{bmatrix}0.1\\0.8\\0.6\\\vdots\end{bmatrix} & \begin{matrix}\times\\\times\\\checkmark\\ \end{matrix} & \begin{bmatrix}1\\0\\1\\\vdots\end{bmatrix}\end{array}$$

Loss Optimization – Gradient Descent

1. Initialize weights randomly: $W \sim \mathcal{N}(0, \sigma^2)$
2. Loop until convergence:
   (a) Compute gradient, $\frac{\partial J(W)}{\partial W}$
   (b) Update weights, $W \leftarrow W - \eta \frac{\partial J(W)}{\partial W}$
3. Return weights

Backpropagation – Next Lecture

Example in Python from scratch:
NEURAL NETWORK MODEL