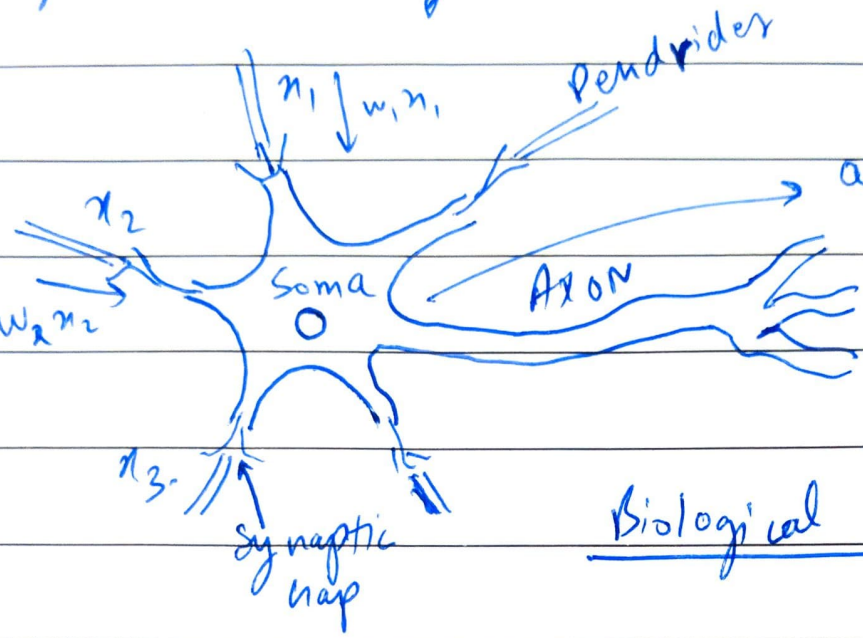
Deep Learning Introduction

Artificial Vs Biological Neurons :



Biological Neuron

Dendrites : Brings inputs (x_1, x_2, x_3, etc.) from other neurons.

Soma : Input goes here for processing.

Synaptic Gap : Modifier input (Say x_1 was passed & $w_1 x_1$ will go inside Soma)

Activation Valve :

$$z = \sum w^{(i)} x^{(i)} + b \quad \text{(bias)} \quad \text{(weighted sum)}$$

If $z >$ threshold the neuron will fire.

$$a = g(z)$$

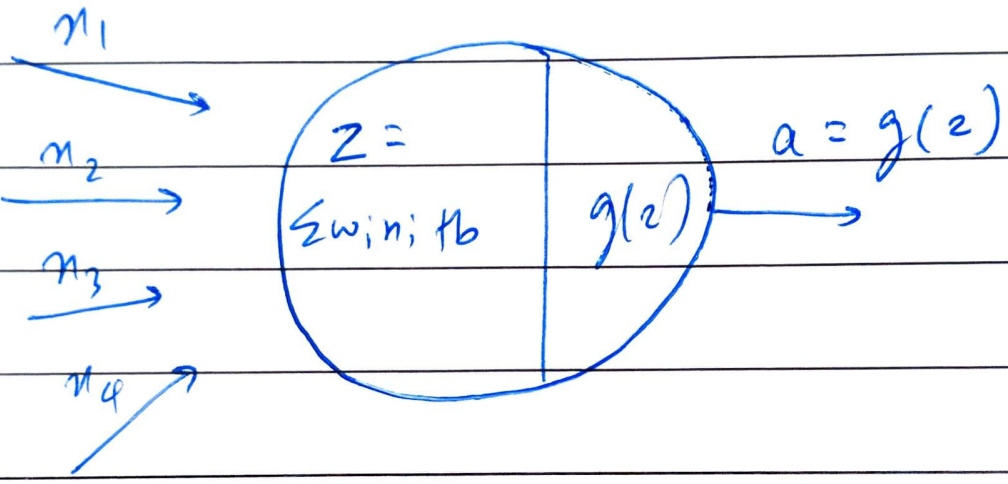
g → Threshold / Activation function / Activity

g can be a sigmoid function, but we use RELU.

RELU : Rectified Linear Unit.

$$g(z) = \begin{cases} z, & z \geq 0 \quad // \text{Neuron fires} \\ 0, & z < 0 \quad // \text{doesn't fire} \end{cases}$$

Artificial neuron :

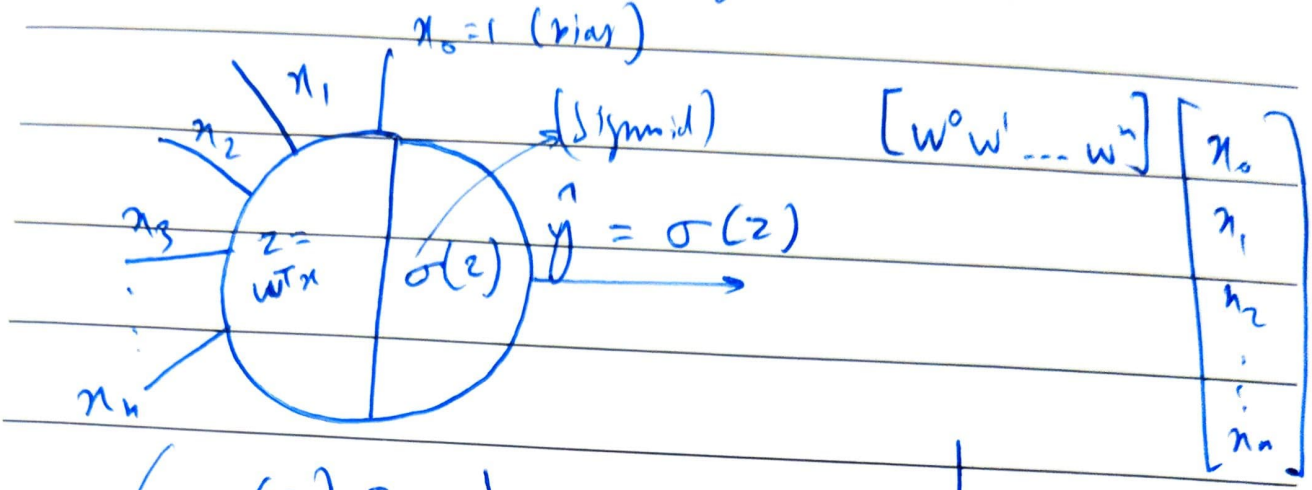


w_i are weights.

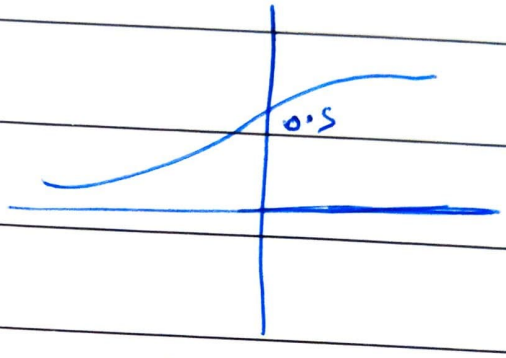
* If $g(z)$ is sigmoid, then this is logistic regression!

Biological	Artificial
Dendrides	Inputs
Synaptic cap	weights
Axon	Output
Soma	Activation

Perceptron : Single layer NN.



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



→ Acts as linear Classifier (Binary Classifier).

* This simple perception acts as logistic Reg.
* Goal is to learn parameter (w^T).

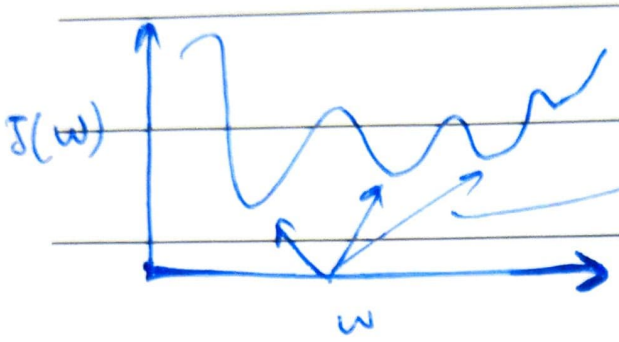
How to train a perception?

→ model here is single node of perception. (more nodes make it a NN calling it multi-layer perception).

→ loss function define :

Ex: $J(w) = MSE = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)})^2$

→ But this is non-convex



* Another kind of loss is log loss / Binary Cross entropy :

$$J(w) = -\sum_{i=1}^{m} \left(y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i) \right)$$

$$\hat{y}^{(i)} = \sigma(z^{(i)}) = \sigma(w^T x^{(i)})$$

$$\frac{\partial J(w)}{\partial w} = \frac{\partial J}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial w}$$

$\hat{y} = \sigma(z)$

$z = w^T x$

$$\frac{\partial J(w)}{\partial w} = \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)} \right) x_j^{(i)}$$

(j^{th} feature)

* $$w_j = w_j - \eta \frac{\partial J(w)}{\partial w}$$