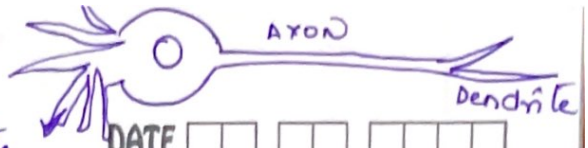
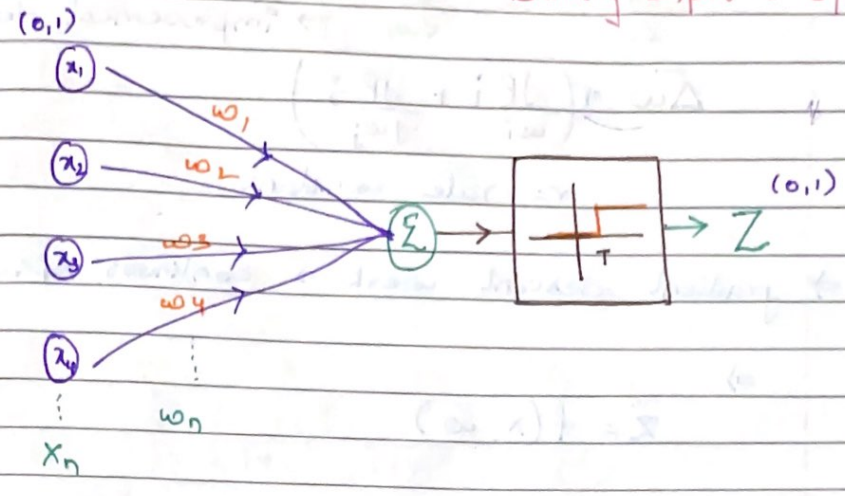


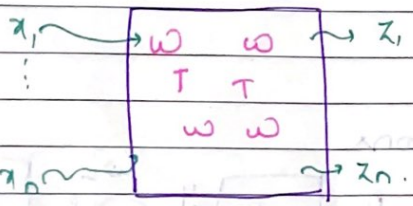
Neural Nets

DATE

Binary Input : 0/1

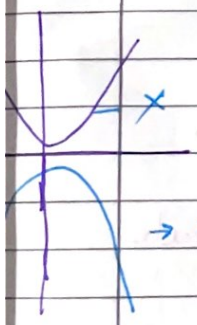


1. All or NON
2. Cumulative influence
3. SYNAPTIC weight
4. Refraction Period



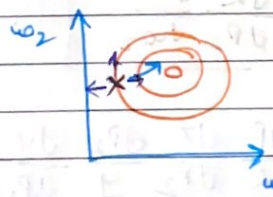
$\bar{Z} = f(x, w, T)$ → function approximator

$\bar{d} = g(x)$



$P = \| \bar{d} - \bar{Z} \|$ → Performance function

→ To get a uphill, instead of downhill



instead of taking step in every direction

↓

Take some partial derivates

DATE

$\frac{dP}{d\omega_1}$, $\frac{dP}{d\omega_2}$ $\omega_2 > \omega_1$ → improvement done

$$\Delta\omega = r\left(\frac{dP}{\omega i} i + \frac{dP}{d\omega_j} j\right)$$

r = rate constant

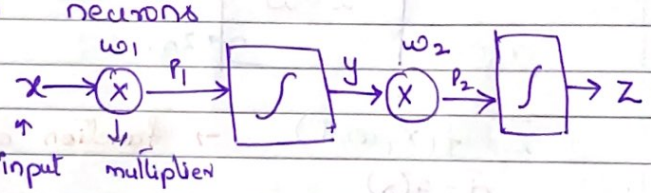
⇒ gradient descent work → continuous space

⇒

$$\hat{z} = f(x, \omega)$$

Sigmoid :-

⇒ Two neurons



$$P = \frac{1}{2}(d - z)^2$$

d = desired o/p
Z = actual o/p

Partial derivative using chain rule

$$\frac{dP}{d\omega_2} = \frac{dP}{dz} \frac{dz}{d\omega_2} \quad \frac{dz}{dP_2} \frac{dP_2}{d\omega_2}$$

$$\frac{dP_1}{d\omega_1} = \frac{dP}{dz} \frac{dz}{dP_2} \frac{dP_2}{y} \frac{dy}{dP_1} \frac{dP_1}{d\omega_1}$$

$$\frac{dP_2}{d\omega_2} = \frac{dP_2^{(y)}}{d\omega_2} \frac{dz}{dP_2} \frac{dP}{dz}\,(d-z)$$

DATE

$B = \frac{1}{1+e^{-\alpha}}$; derivative with α.

$\frac{dB}{d\alpha} = \frac{d(1-e^{-\alpha})^{-1}}{d\alpha}$

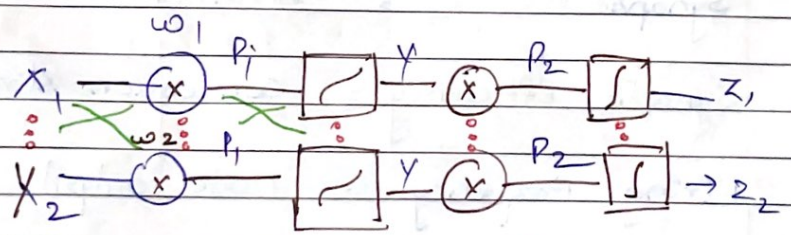
$= (1+e^{-\alpha})^{-2} e^{-\alpha}$

$= \frac{1+e^{-\alpha}-1}{1+e^{-\alpha}} \times \frac{1}{1+e^{-\alpha}}$

$= \left(\frac{1+e^{-\alpha}}{1+e^{-\alpha}} - \frac{1}{1+e^{-\alpha}}\right)\left(\frac{1}{1+e^{-\alpha}}\right)$

$= B(1-B)$

$\Rightarrow z(1-z) \rightarrow$ o/p $\rightarrow$ Sigmoid.



$P = -\frac{1}{2}(0-z)^2$

$\Rightarrow$ A repetition occurs

influences of fast fourier transform

1. Linear in depth
2. w.r. width $= \omega^2$

$\Rightarrow$ reuse principle :-

vision

PAGE

Presynaptic → neuron that Sends Signal

Postsynaptic → neuron that receives

DATE

Artifical Neuron

Synaptic contacts

1. Excitatory - asymmetrical, membrane thickening > Postsynaptic
2. Inhibitory - symmetrical

Strength of connections = Efficiency of Synaptic Transmission

Threshold of Neuron = net excitation exceeds its inhibition by critical amount.

Refractory Period = Neuron is inactive

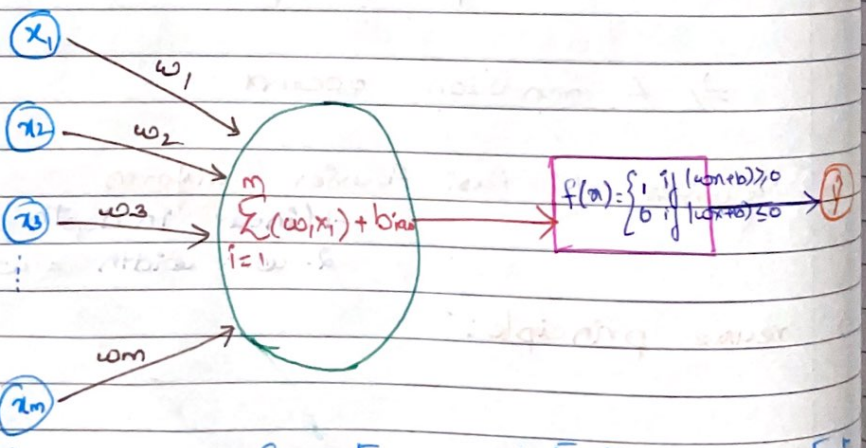
output of Neuron = 1. strong input
2. Strong frequency

Neuron. = Node / cell / unit

Synapse = Connection

Synaptic Efficiency = connection strength.

Firing frequency = Node output



VISION

PAGE

Activation function :- o/p of neuron, given set of inputs.

DATE

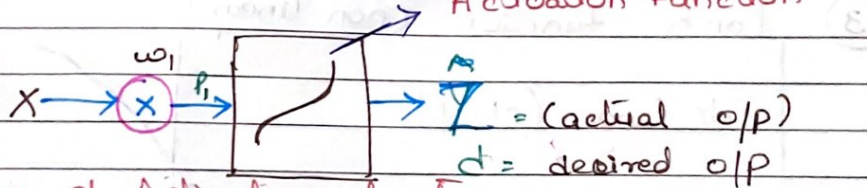
Back Propogation : Repeatedly adjust the weights to minimize difference between d and Z

Hidden layers : Neuron nodes stacked in between inputs

$$W \cdot X = w_1 x_1 + w_2 x_2 \dots w_m x_m$$

$$= \sum_{i=1}^{m} w_i x_i$$

$$Z = \sum_{i=1}^{m} w_i x_i + \text{bias}$$



Types of Activation function

↝ Sigmoid function =

Exclude 2

$$Z = \frac{1}{2}(d - z)$$

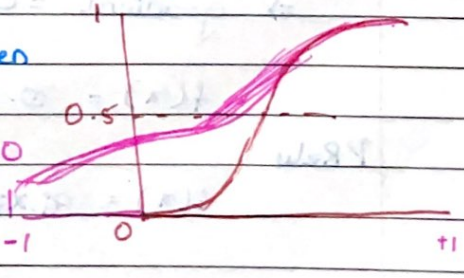
$$Z = \sum_{i=1}^{m} w_i x_i + b_i$$

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

⇒ Sigmoid function

⇒ Transform value between 0 to 1

⇒ if less than 0.5 - o/p = 0
more than 0.5 - o/p = 1



⇒ classification Problem.

if i/p values are increased, o/p becomes stagnant (inactive)
↳ Vanishing Gradient

DATE

(2) Relu AF (Rectified linear unit)

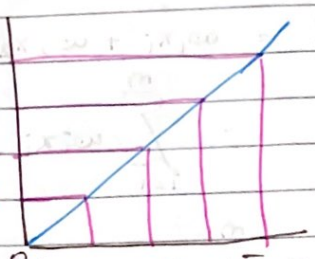
$$f(x) = \begin{cases} x_i, & \text{if } x_i > 0 \\ 0, & \text{if } x_i < 0 \end{cases}$$

max (Z, 0)

if Z is -ve → max(-ve, 0)
↳ o/p = 0

if Z is +ve → max(+ve, 0)
↳ o/p = +ve number

=> Regression
=> Problem use Relu, less computation

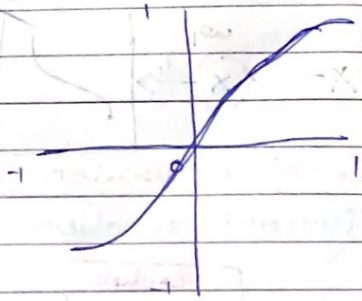


disadvantage :- if -ve, zero value, gets stuck

(3) tanh function (non-linear)

=> Range (-1, 1)

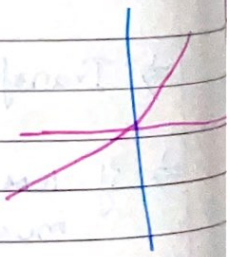
=> zero-centered



(4) Leaky ReLU
↳ Dead Neurons (due to -ve value)

=> gradient = 0 if x < 0

$f(x) = 0.01x, \quad x < 0$



PRelu

$f(x) = a_i x_i$

Leaky Rectified Linear Unit, or Leaky ReLU, is a type of activation function based on a ReLU, but it has a small slope for negative values instead of a flat slope. ... This type of activation function is popular in tasks where we we may suffer from sparse gradients, for example training generative adversarial networks.

VISION

PAGE