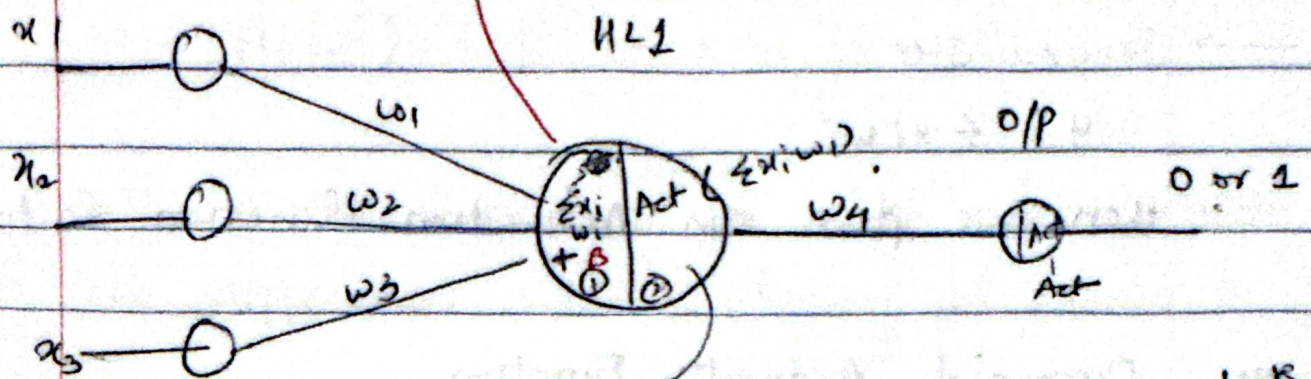


Imp Forward Propagation



we add this if weights are zero
 $w_1 = 0$ then $x_1 w_1 = 0$ so for
that we add Bias

$$\sum x_i w_i = x_1 w_1 + x_2 w_2 + x_3 w_3 + B$$
$$= \underline{w^T x}$$

L.R

$$y = \beta_0 + \beta_1 x$$
$$y = \beta^T x$$

We pass it to an Activation function.

→ Importance of Weights

So if you are placing hot object on your hand you will move your hand right as the neurons there are getting activated.

Weights will help your neurons to act (what level it should be activated)

While training also we would be assigning weights so as to what level it should be activated.

(Linear Activatⁿ) for L.R

→ Second Step

$$y = \sum x_i w_i$$

then we pass an Activation function on top of

eg. Sigmoid Activatⁿ Function

→ Binary classificⁿ

$$\text{Sigmoid} = \frac{1}{1 + e^{-y}}$$

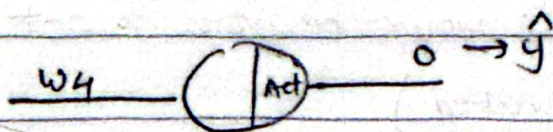
$$= \frac{1}{1 + e^{-(\sum x_i w_i + b)}}$$

↙ Bias is also getting added.
→ (0 to 1) op.

Activation functⁿ is a functⁿ that decides whether a neuron should be activated or not.

look at
dataset
value

So you doing the things listed in diagram on last page if we get output of 0 for forward propagation



If \hat{y} (predicted value is 0) then $y = 1$

$$(y - \hat{y}) = 1$$

So now what you've to make sure is that the difference $(y - \hat{y})$ should be near to 0

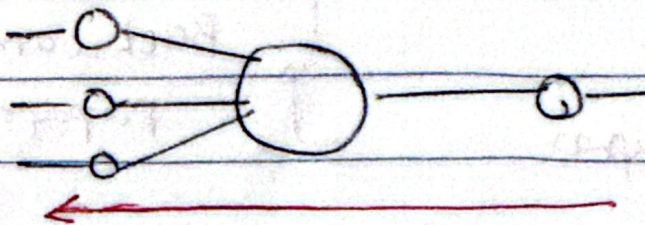
→ And what is $(y - \hat{y})$
its the Loss Function

Aim should be minimizing the difference

So now you know the difference is huge now what you do is

→ * Back propagation

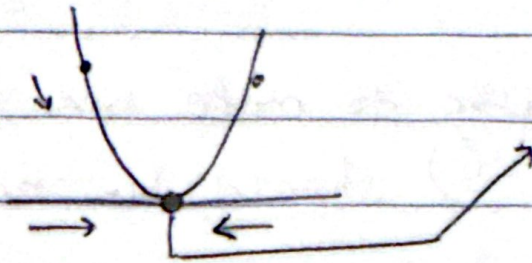
Aim of back propagatⁿ is to update weights



Optimizers → ensures each weights get updated while we are doing back propagatⁿ

[One example of optimizer is Gradient Descent]

Gradient Descent \rightarrow Kind of optimizer



Here $\hat{y} - y = \underline{\text{minimal}}$

it will make sure to update the coefficient to basically come to the global minima

Steps

- ① I/P layers
 - ② Weights
 - ③ Bias
 - ④ Activation function
 - ⑤ Loss Function $(y - \hat{y})$ ↓↓↓
 - ⑥ Optimizers
 - ⑦ Update the weights
- Forward propagation
- Backward propagation

Now if we run this 1000 times the neural network will get train just as a baby gets trained