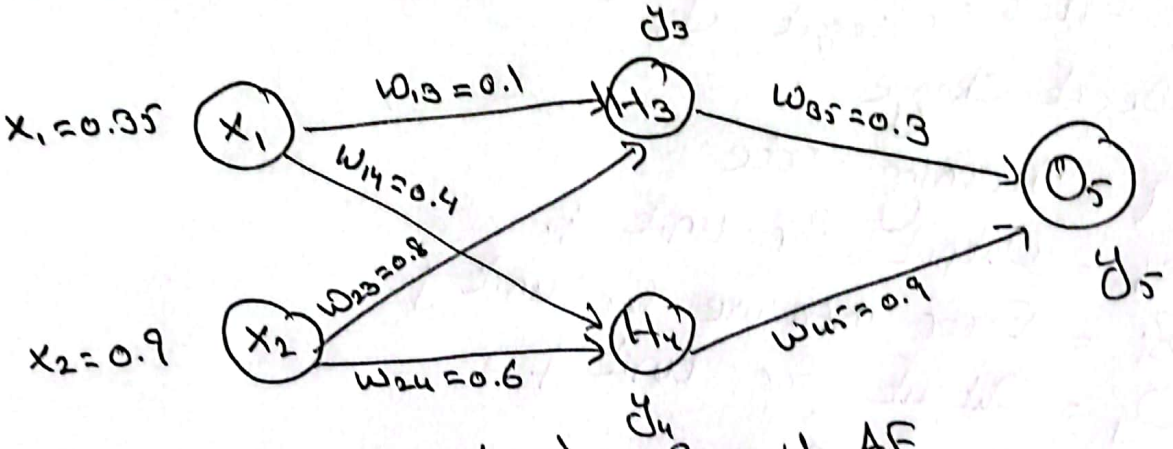
Neural Network
Lecture 9

NN Models:-

→ There are different available NN Models.

1) McCulloch-Pitts Neuron (earliest)
   ↳ M-P Neuron (1943)
2) Perception NN
   ↳ various types such as Rosenblatt (1962), Block (1962), Minsky-Papert (1969)
3) Adaptive Linear NN
4) Multiple Adaptive Linear NN

Gradient Decent:-



Activation Function (AF) = Sigmoid AF

Actual Output $y = 0.5$

Learning Rate $= 1$

Forward_Pass (propagation):

Compute y_3, y_4 & y_5

$$a_{ij} = \sum_j (w_{ij} * x_i) \qquad y_j = F(a_j) = \frac{1}{1+e^{-a_j}}$$

$$a_1 = (w_{13} * x_1) + (w_{23} * x_2)$$
$$= (0.1 \times 0.35) + (0.8 \times 0.9) = 0.755$$
$$y_3 = f(a_1) = 1/(1+e^{-0.755}) = 0.68$$

$$a_2 = (w_{14} * x_1) + (w_{24} \times x_2)$$
$$= (0.4 * 0.35) + (0.6 \times 0.9) = 0.68$$
$$y_4 = f(a_2) = 1/(1+e^{-0.68}) = 0.6337$$

$$a_3 = (w_{35} * y_3) + (w_{45} \times y_4)$$
$$= (0.3 \times 0.68) + (0.9 \times 0.6337) = 0.801$$
$$y_5 = f(a_3) = 1/(1+e^{-0.801}) = \boxed{0.69} \text{ (Network Output)}$$

So

$$\text{Error} = y_{target} - y_5 = 0.5 - 0.69 = -0.19$$

Now Weight Change

η = learning rate
t_j = Output for unit j.
δ_i = Error measure for unit j
O_j = Output for unit j.

$$\Delta w_{ij} = \eta \delta_j O_j$$
$$\delta_j = O_j(1-O_j)(t_j - O_j) \quad \therefore \text{if j is output unit}$$
$$\delta_j = O_j(1-O_j) \sum_k \delta_k w_{kj} \quad \therefore \text{if j is hidden unit}$$

Backward pass (propagation):

Compute δ_3, δ_4 & δ_5

For output unit

$$\delta_5 = y_5(1-y_5)(y_{target} - y_5)$$
$$= 0.69(1-0.69)(0.5-0.69)$$
$$= -0.0406$$

For hidden unit:

$$\delta_3 = y_3(1-y_3)(w_{35} \times \delta_5)$$
$$= 0.68(1-0.68)(0.3 \times -0.0406)$$
$$= -0.00265$$

$$\delta_4 = y_4(1-y_4)(w_{45} \times \delta_5)$$
$$= 0.6637 \times (1-0.6337)(0.9 \times -0.0406)$$
$$= -0.0082$$

Comput New Weights

$$\Delta w_{ji} = \eta \delta_j O_j$$

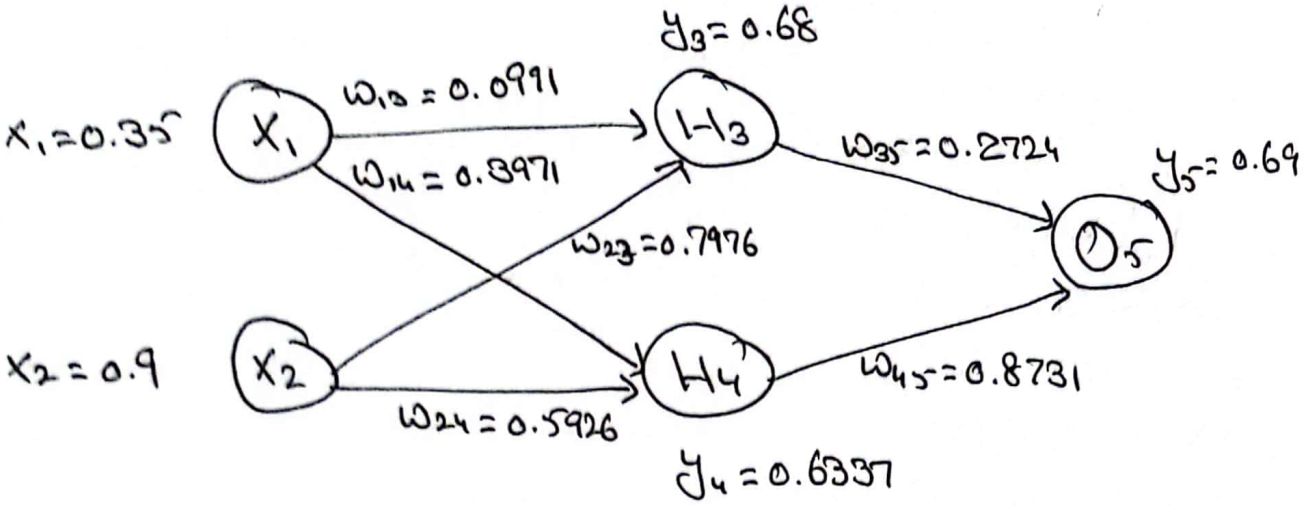
$$\Delta w_{45} = \eta \delta_5 y_4$$
$$= 1 \times (-0.0406) \times 0.6637 = -0.0269$$
$$w_{45}(\text{New}) = \Delta w_{45} + w_{45}(\text{Old}) = -0.0269 + (0.9)$$
$$= 0.8731$$

$$w_{14} = \eta \delta_4 x_1$$
$$= 1 \times (-0.0082) \times 0.35 = -0.00287$$
$$\Delta w_{14}(\text{New}) = \Delta w_{14} + w_{14}(\text{Old})$$
$$= -0.00287 + 0.4 = 0.3971$$

Similarly

i	j	w_{ij}	δ_j	x_i	η	Updated w_{ij}
1	3	0.1	-0.00265	0.35	1	0.6991
2	3	0.8	-0.00265	0.9	1	0.7976
1	4	0.4	-0.0082	0.35	1	0.3971
2	4	0.6	-0.0082	0.9	1	0.5926
3	5	0.3	-0.0406	0.68	1	0.2724
4	5	0.9	-0.0406	0.6337	1	0.8731

The Update Neural Network is



Forward pass (propagation):

$$a_j = \sum_j (w_{ij} \times x_i) \qquad y_{ij} = F(a_j) = \frac{1}{1+e^{a_j}}$$

$$a_1 = (w_{13} * x_1) + (w_{23} * x_2)$$
$$= (0.0991 \times 0.35) + (0.7976 \times 0.9) = 0.7525$$
$$y_3 = f(a_1) = 1/(1+e^{-0.7525}) = 0.6797$$

$$a_2 = (w_{14} * x_1) + (w_{24} * x_2)$$
$$= (0.3971 \times 0.35) + (0.5926 \times 0.9) = 0.6723$$
$$y_4 = f(a_2) = 1/(1+e^{-0.6723}) = 0.6620$$

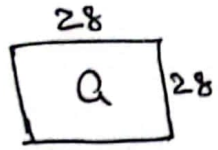
$$a_3 = (w_{35} \times y_3) + (w_{45} \times y_4)$$
$$= (0.2724 \times 0.6797) + (0.8731 \times 0.6620)$$
$$= 0.7631$$
$$y_5 = f(a_3) = 1/(1+e^{-0.7631})$$
$$= \boxed{0.6820} \text{ (Network Output)}$$

$$\text{Error} = y_{target} - y_5$$
$$= 0.5 - 0.6820$$
$$= -0.182$$

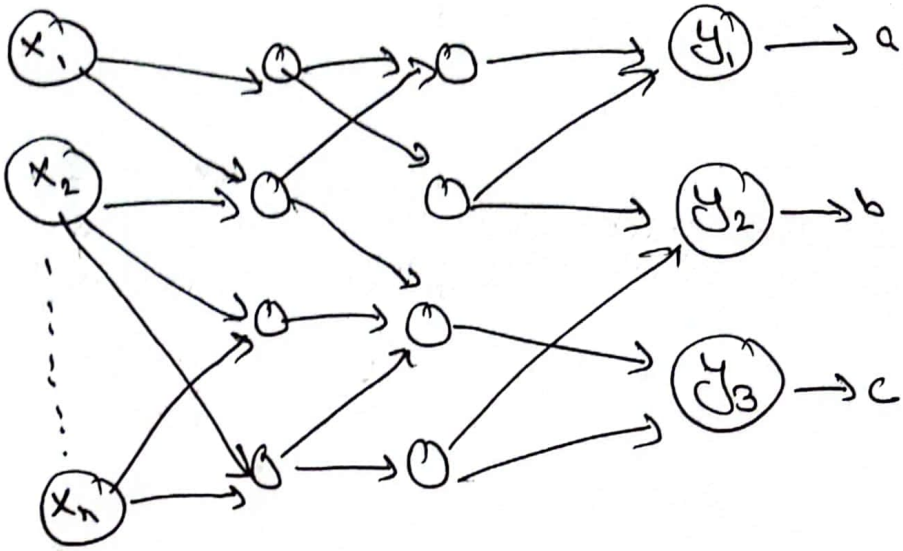
Gradient Decent:

→ Three types (Batch, Stocastic, mini-Batch)

→ Choice is between Accuracy & Time

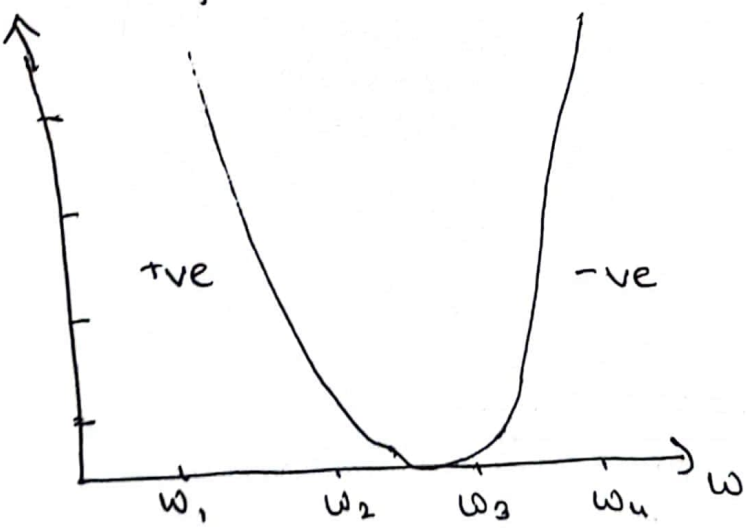


$28 \times 28 = 784$



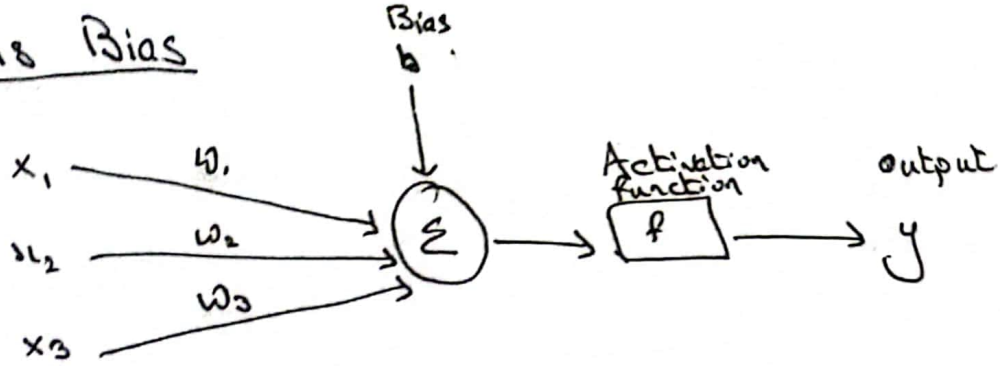
$n = 784$

Loss = [(actual Output) − (predicted Output)]



Every Learning Step = 1 iteration.

What is Bias



→ Bias can shift the AF. $y = mx + c$
→ Used to control AF (triggering, delay, steepness).
= sum (weight × inputs) + bias

$y = m x + c$
weight, Bias