TOPIC NAME : Machine Learning DAY : TIME : DATE : / /

ANN
(Artificial Neural Network)

Single Neuron/Perception :

Basic Structure :

Inputs : $X = [x_1, x_2, ---- x_n]$

Weights : $W = [W_1, W_2, ---- W_n]$

Bias : b

Activation ~~Fuct~~ Function : φ

Single Neuron/Perception :

* Pre activation step :

$$z = \sum_{i=0}^{n} w_i x_i + b$$

$$= W_1 x_1 + W_2 x_2 + W_3 x_3 + ---- W_n x_n + b$$

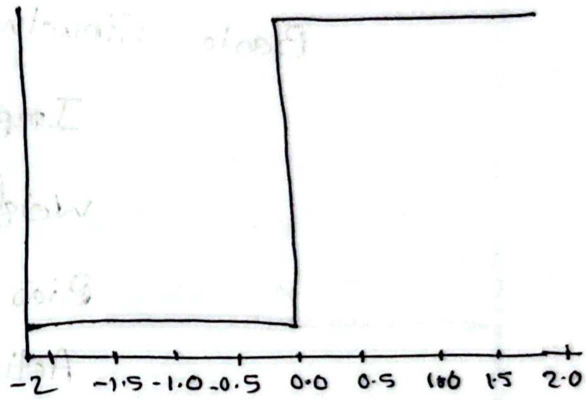
* Activation function :

$$Y = \phi(z)$$

TOPIC NAME : DAY : TIME : DATE : / /

Binary Step:

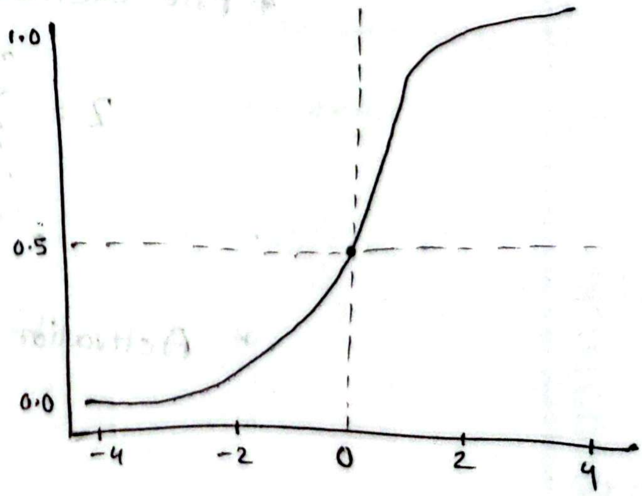
$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$



Activation Functions (Cont)

Sigmoid / Logistic

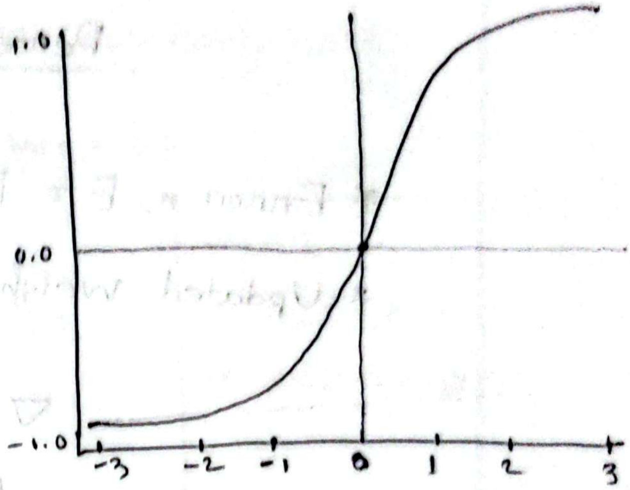
$$f(x) = \frac{1}{1+e^{-x}}$$



TOPIC NAME : DAY : TIME : DATE : / /

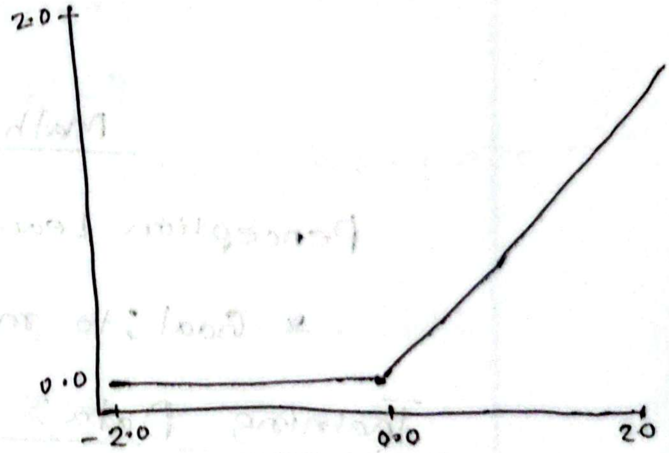
Tanh

$$f(x) = \frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})}$$

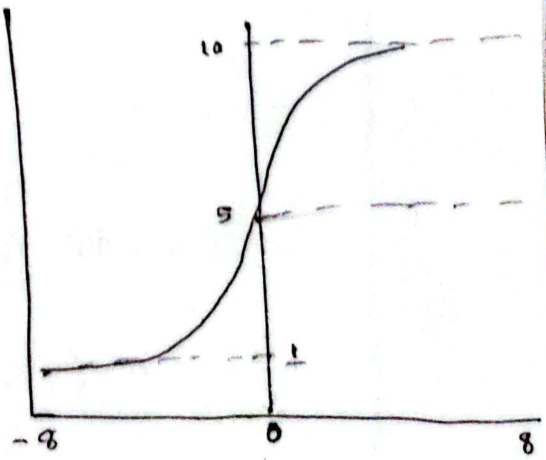


ReLU

$$f(x) = \max(0, x)$$



$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$



TOPIC NAME : DAY : TIME : DATE : / /

Perception of Learning Rule

* Error, $E = \text{Target} - \text{Output}$
* Updated weights & Bias:

$$\nabla W_i = \alpha \cdot E \cdot x_i$$

$$b_{new} = b_{old} + \alpha \cdot E$$

α is the new learning rate.

Math

Perception Learning example: AND Gate

* Goal: to mimic AND logic, $y = x_1 \cdot x_2$

Training Data:

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

GOOD LUCK

$w_1x_1 + w_2x_2 + w_3x_3 + \cdots + b$

TOPIC NAME :

DAY :

TIME : DATE : / /

Here, initialize parameters (randomly)

Weights, $w_1 = 0.5$, $w_2 = 0.5$

Bias, $b = -0.7$

Learning rate, $\alpha = 0.1$

এখানে,
if $z \geq 0 \Rightarrow y = 1$
if $z < 0 \Rightarrow y = 0$

Epoch 1 :

Sample (0, 0) :

$z = (0.5 \times 0) + (0.5 \times 0) - 0.7 = -0.7$

$y = \text{Step}(-0.7) = 0$

$\therefore$ Error $(t - y) = 0 - 0 = 0$ No update

Sample (0, 1) :

$z = (0.5 \times 0) + (0.5 \times 1) - 0.7 = -0.2$

$y = \text{step}(-0.2) = 0$

$\therefore$ Error $= (t - y) = 0$ No Update.

Sample (1, 0) :

$z = (0.5 \times 1) + (0.5 \times 0) - 0.7 = -0.2$

$y = \text{step}(-0.2) = 0$

$\therefore$ Error $= (t - y) = 0$ No Update.

Sample (1, 1) :

$z = (0.5 \times 1) + (0.5 \times 1) - 0.7 = 0.3$

$y = \text{step}(0.3) = 1$

$\therefore$ Error $= (t - y) = (1 - 1) = 0$ No Update.

TOPIC NAME : DAY : TIME : DATE : / /

∴ Epoch 2 :

* All prediction connect → Convengence achieved !

* Decision Boundary : $0.5x_1 + 0.5x_2 - 0.7 = 0$

Example 2

Weights, $w_1 = 0.2.5$
$w_2 = 0.25$

Bias, $b = -0.8$

Learning rate, $\alpha = 0.5$

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

Epoch 1 :

Sample (0, 0) :

$z = (0.25 \times 0) + (0.25 \times 0) - 0.8 = -0.8$

$y = \text{step}(-0.8) = 0$

Ennon $(t - y) = (0 - 0) = 0$ ⊞ No Update

Sample (0, 1) :

$z = (0.25 \times 0) + (0.25 \times 1) - 0.8 = -0.55$

$y = \text{step}(-0.55) = 0$

∴ Ennon $(t - y) = (0 - 0) = 0$ ⊞ No Update

TOPIC NAME : DAY :
TIME : DATE : / /

Sample (1, 0) :

$\therefore z = (0.25 \times 1) + (0.25 \times 0) - 0.8 = -0.55$

$y = \text{step}(-0.55) = 0$

$\therefore$ Error $(t - y) = (0 - 0) = 0 \Rightarrow$ No Update

Sample (1, 1) :

$z = (0.25 \times 1) + (0.25 \times 1) - 0.8 = -0.3$

$y = \text{Step}(-0.3) = 0$

$\therefore$ Error $(t - y) = (1 - 0) = 1$ ~~No Update~~

$\therefore$ Updated weights and Bias :

$\Delta w_1 = \alpha \cdot (t - y) \cdot x_1 = 0.5 \times 1 \times 1 = 0.5$ $\}$ $w_1 = 0.25 + 0.5 = 0.75$

$\Delta w_2 = \alpha \cdot (t - y) \cdot x_2 = 0.5 \times 1 \times 1 = 0.5$ $\}$ $w_2 = 0.25 + 0.5 = 0.75$

$\therefore \Delta b = \cancel{\alpha}\, b_{old} + \alpha \cdot E$

$= -0.8 + 0.5 \times 1 = -0.3$

After epoch 1 :

$w_1 = 0.75$

$w_2 = 0.75$

$b = -0.3$

TOPIC NAME : DAY : TIME : DATE : / /

Epoch 2 :

☐ ~~Inpu~~ Sample (0, 0), $t = 0$

$$z = (0.75 \times 0) + (0.75 \times 0) - 0.3 = -0.3$$

$$y = \text{step}(-0.3) = 0$$

Error $= (t - y) = 0$ ☐ No Update

☐ Sample (0, 1), $t = 0$

$$z = (0.75 \times 0) + (0.75 \times 1) - 0.3 = 0.45$$

$$y = \text{step}(0.45) = 1$$

$\therefore$ Error $= (t - y) = -1$

$$\therefore \nabla w_1 = \alpha \cdot E \cdot x_i = 0.5 \cdot (-1) \cdot 0 = 0 \quad w_1 = 0.75 + 0 = 0.75$$

$$\therefore \nabla W_2 = \alpha \cdot E\, x_i = 0.5 \times (-1) \cdot 1 = -0.5 \quad W_2 = 0.75 - 0.5 = 0.25$$

$$\therefore b = b_{old} + \alpha \cdot E = -0.3 + (0.5 \cdot (-1)) = -0.8$$

$\therefore w_1 = 0.75$

$w_2 = 0.25$

$b = -0.8$

TOPIC NAME :

DAY :

TIME : DATE : / /

Sample (1,0), $t=0$

$z = (0.75\times1) + (0.25\times0) - 0.8$

$= -0.05 \geq 0$ ~~No Update~~

$\therefore y = step(-0.05) = 1$ $\therefore$ Error $= (t-y) = (1-1) = 0$

No Update.

Sample (1,1), $t=1$

$z = (0.75\times1) + (0.25\times1) - 0.8$

$= 1 - 0.8 = 0.2 \geq 1$.

$\therefore y = step(1) = 1$

$\therefore$ Error $= (t-y) = (1-1) = 0$ $\therefore$ No Update.

$\therefore w_1 = 0.75$

$w_2 = 0.25$

$b = -0.8$

$\therefore$ Decision Function:

$z = w_1x_1 + w_2x_2 + b$

$= 0.75x_1 + 0.25x_2 - 0.8$

GOOD LUCK

TOPIC NAME: DAY: TIME: DATE: / /

Funtamental Question

Q1: What is Antificial Neural Network (ANN)? How is it inspired by the human brain?

Ans: ANN is a computer model that works like human brain to solve problems by learning from data. It uses antificial neurons that mimic how brain neurons send signals.

Math & Learning

Q2: Explain the penception learning rule. How are weights & Bias Updated?

Ans: It is a method used to adjust the weights and bias of a neuron based on the error (diffenence between expected and actual output) using a formula.

Updated w and b:

$$\nabla W_i = \alpha \cdot E \cdot x_i$$

$$b_{new} = b_{old} + \alpha \cdot E$$

learning rate.

TOPIC NAME :

DAY :

TIME : DATE : / /

Q3: Write the mathematical expression for the output for a single neuron?

Ans: pre-activation:

$$z = \sum_{i=0}^{n} w_i x_i + b$$

$$= w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$$

Activation Function:

$$y = \phi(z)$$

Q4: Why is the bias term important & what role does the learning rate (α) play in training a perception?

Ans: It helps shift the output so the neuron can learn better. The learning rate controls how big the updates to weights and bias are.

Q5: What is activation function?

Ans: It decides whether a neuron should be "activated" or not by applying a non-linear transformation to the input.

TOPIC NAME : DAY : TIME : DATE : / /

Q6: Diffenence between step, sigmoid, and ReLU?

Ans: Step: Outputs 0 on 1, used in basic models.

Sigmoid: Smooth S-shaped curve, output between 0 on 1.

ReLU: Outputs 0 if input is negative, otherwise outputs ~~the~~ the input itself

Q7: Why is ReLU betten than Sigmoid?

Ans: ReLU is faster and helps avoid the vanishing gradient problem, making it better for deep networks.

GOOD LUCK™

TOPIC NAME : $z = w_1x_1 + w_2x_2 + \cdots + b$

DAY :

TIME : DATE : / /

ML Math

☐ Given,

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

weights, $w_1 = 0.25$

$w_2 = 0.25$

Bias, $b = -0.8$

learning rate, $\alpha = 0.5$

∴ AND Gate training Steps:

Epoch 1:

☐ Sample (0,0), $t = 0$:

$z = (0.25 \times 0) + (0.25 \times 0) - 0.8 = -0.8$

$y = step(-0.8) = 0$

Error $= (t - y) = (0 - 0) = 0$ ☐ No Update.

☐ Sample (0,1), $t = 0$:

$z = (0.25 \times 0) + (0.25 \times 1) - 0.8 = -0.55$

$y = step(-0.55) = 0$

Error $= (t - y) = (0 - 0) = 0$ ☐ No Update

☐ Sample (1,0), $t = 0$:

$z = (0.25 \times 1) + (0.25 \times 0) - 0.8 = -0.55$

$y = step(-0.55) = 0$

Error $= (t - y) = (0 - 0) = 0$ ☐ No Update.

☐ Sample (1,1), $t = 1$:

$z = (0.25 \times 1) + (0.25 \times 1) - 0.8 = -0.3$

$y = step(-0.3) = 0$

∴ Error $= (t - y) = (1 - 0) = 1$ ☐ Error

∴ Updated Weight and Bias:

Given,
$\alpha = 0.5$

$$\Delta W_1 = \alpha \cdot E \cdot x_1$$
$$= 0.5 \times 1 \times 1$$
$$= 0.5$$
$$\therefore W_1 = 0.5 + 0.25 = 0.75$$

∴ ~~Updated~~ $\Delta W_2 = \alpha \cdot E \cdot x_2$
$$= 0.5 \times 1 \times 1$$
$$= 0.5$$
$$\therefore W_2 = 0.5 + 0.25 = 0.75.$$

$$\therefore b = b_{old} + \alpha \cdot E = -0.8 + 0.5 \times 1 = -0.3$$

Epoch 2:

Sample: $(0,0), t=0$:

$$z = (0.75 \times 0) + (0.75 \times 0) - 0.3 = -0.3$$
$y = step(-0.3) = 0$ No Update.
$Error = t - y = 0$

Sample: $(0,1), t=0$:

$$z = (0.75 \times 0) + (0.75 \times 1) - 0.3 = 0.45$$

~~Sample: (0,1), t=0~~

$$y = step(0.45) = 1$$
$$Error = t - y = 0 - 1 = -1$$

~~Sample: (1,0), t=0~~

$z = (0.75 \times 1) + \cancel{(0.75 \times 0) - 0.3}$

∴ Updated w & b :

$$\Delta W_1 = \alpha \cdot E \cdot x_1$$
$$= 0.5 \times (-1) \times 0 = 0$$
$$\therefore w_1 = 0.75 + 0 = 0.75$$

$$\Delta W_2 = \alpha \cdot E \cdot x_2$$
$$= 0.5 \times (-1) \times 1$$
$$= -0.5$$
$$\therefore w_2 = 0.75 - 0.5 = 0.25$$

$$\therefore b = b_{old} + \alpha \cdot E$$
$$= -0.3 + 0.5 \times (-1) = -0.8$$

∴ Sample (1, 0), t = 0 :

$$z = (0.75 \times 1) + (0.25 \times 0) - 0.8$$
$$= -0.05$$
$$\therefore y = step(-0.05) = 0$$
∴ Error = t − y = 0 No Update

∴ Sample (1, 1), t = 1 :

$$z = (0.75 \times 1) + (0.25 \times 1) - 0.8 = 0.2$$
$$\therefore y = Step(0.2) = 1$$
∴ Error = t − y = 1 − 1 = 0 No Update

TOPIC NAME :

DAY :

TIME : DATE : / /

Epoch 3:

Sample (0,0), t=0

$z = (0.75 \times 0) + (0.25 \times 0) - 0.8 = -0.8$

$y = \text{step}(-0.8) = 0$

$\therefore$ Error $= t - y = 0$ $\boxed{\text{No Update}}$

Sample (0,1), t=0

$z = (0.75 \times 0) + (0.25 \times 1) - 0.8 = -0.55$

$y = \text{step}(-0.55) = 0$

Error $= t - y = 0$ $\boxed{\text{No Update}}$

Sample (1,0), t=0

$z = (0.75 \times 1) + (0.25 \times 0) - 0.8 = -0.05$

$y = \text{step}(-0.05) = 0$

$\therefore$ Error $= 0$ $\boxed{\text{No Update}}$

Sample (1,1), t=1

$z = (0.75 \times 1) + (0.25 \times 1) - 0.8 = 0.2$

$y = \text{step}(0.2) = 1$

$\therefore$ Error $= 1 - 1 = 0$ $\boxed{\text{No Update}}$

$\therefore$ All prediction correct $\rightarrow$ Convergence Achieved

$\therefore$ Decision Boundary:

$$0.75x_1 + 0.25x_2 - 0.8 = 0$$

Ques1: What is an Artifical Neural Network (ANN)?

Ans: An ANN is a computer model inspired by the human brain. It has neurons (perceptions) that take inputs, multiply them by weights, add a bias, and pass them through an activation function to produce an output.

Ques:2: Explain the role of activation functions in ANN.

Ans: Activation functions decide if a neuron should 'fire' (activate) or not.

* Step function: Simple (0 or 1). $\boxed{f(x) = \frac{1}{1+e^{-x}}}$

~~Sigm~~

* Sigmoid/Tanh: Smooth curves, good for probabilites $\boxed{f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}}$

* ReLU: Fast learning, avoids negative outputs $\boxed{f(x) = max(0,x)}$

* Softmax: Used for clasification

TOPIC NAME :

DAY :

TIME : DATE : / /

Q3: How does a perceptron learn? Explain the weight update rule.

Ans:

1. It calculates Error = Target − Output
2. It Updates weights:

$$w_i = w_i + \alpha \cdot E \cdot x_i$$

3. It Updates bias: $b = b_{old} + \alpha \cdot E$ (α: learning rate)

Q4: difference between sigmoid and ReLU?

Sigmoid	ReLU
Output: 0 to 1	Output: 0 or linear
Slow for deep networks	Faster, avoids vanishing gradient
Used in old models	popular in deep learning