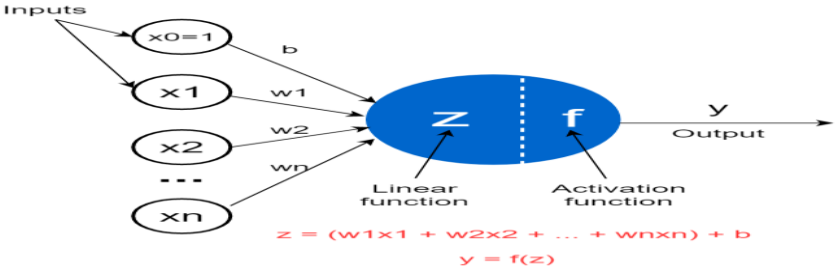
SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)
Approved by AICTE, New Delhi, Affiliated to Anna University, Chennai
Accredited by NAAC-UGC with 'A++' Grade (Cycle III) &
Accredited by NBA (B.E - CSE, EEE, ECE, Mech&B.Tech.IT)
COIMBATORE-641 035, TAMIL NADU

Reg. No: | | | | | | | | | | | | |

B

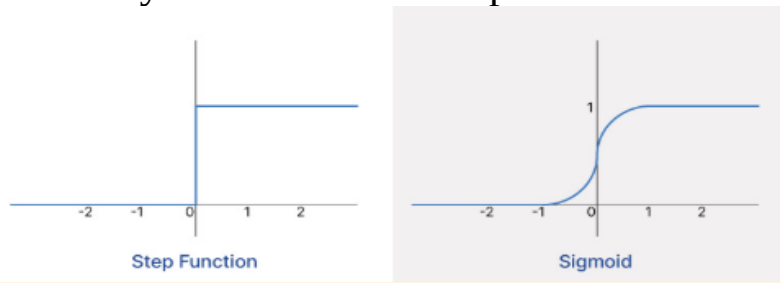
B.E/B.Tech- Internal Assessment – I
Academic Year 2024-2025 (Even Semester)
Sixth Semester
19CST302-NEURAL NETWORKS AND DEEP LEARNING
Answer Key
Answer All Questions

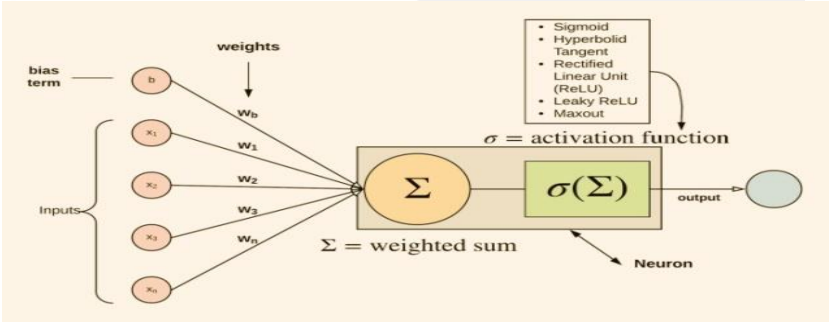
		CO	Blooms
1.	Draw the structure of artificial Neuron. 	CO1	REM
2.	What are the three basic elements of the neuronal model? 1. Input Layer 2. Hidden Layer(Weighted Sum, Calculate Weighted Input) 3. Activation Function 4. Output Layer	CO1	REM
3.	Write the mathematical notation of setting Threshold activation function. It is a commonly used activation function. As depicted in the diagram, it gives **1 as output** of the input is either 0 or positive. If the input is negative, it gives **0 as output**. Expressing it mathematically, $y_{out} = f(y_{sum}) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}$	CO1	UND
4.	Is Gradient descent optimization algorithm used for finding the local minimum of a function? Justify. Yes. Gradient descent is an iterative optimization algorithm used for finding the local minimum of a differentiable function	CO2	ANA
5.	How can you prevent overfitting in neural network?	CO2	ANA

Use a larger training set, Use a smaller network, Weight-sharing (as in convolutional neural networks, Early stopping, Transfer Learning

PART – B (2*13=26 Marks) & (1*14=14 Marks)

				CO	Blooms
6.	(a)	i) Analyze the importance of activation functions in neural networks and provide examples of commonly used functions. 1. An activation function is a mathematical function applied to the output of a neuron. 2. It introduces non-linearity into the model, allowing the network to learn and represent complex patterns in the data. 3. Without this non-linearity feature, a neural network would behave like a linear regression model, no matter how many layers it has. **Sigmoid function:** The sigmoid function is a widely used tool for binary classification problems, where it maps any input value to a value between 0 and 1. This allows us to interpret the output as representing the probability of a positive class.	7	CO1	**ANA**





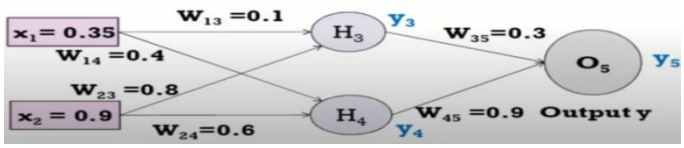
ii) Compare Single Layer Neural Network and Multilayer Neural Network with real time example 6

Comparison Table

Aspect	Single-Layer Perceptron (SLP)	Multi-Layer Perceptron (MLP)
Architecture	One input layer, one output layer (no hidden layers)	Input, hidden, and output layers
Problem Solvability	Solves only linearly separable problems (e.g., AND)	Solves both linear and non-linear problems (e.g., XOR)
Activation Function	Step function (binary output)	Non-linear functions (e.g., sigmoid, ReLU, tanh)
Learning Algorithm	Perceptron learning rule (no backpropagation)	Trained using backpropagation and gradient descent
Output	Binary (0 or 1)	Continuous (for regression) or multi-class (for classification)
Applications	Simple binary classification	Complex tasks (classification, regression, image recognition)
Complexity	Simple and limited	Complex and powerful

(OR)

(b) Examine the error rate using Feed Forward Neural Network. Learning rate= 0.5 , Target output = 0.6



Input layer: i/p – [x1 x2] = [0.35, 0.9]
Hidden layer: z1 – [v11 v21 v01]
Output layer: yin – [w11 w21 w01] 13 CO1 **ANA**
Activation function = Binary sigmoid function
Target value, t = 0.6
Learning rate, α = 0.5
in is calculated as:
yin = w01 + z1*w11 + z2*w21 (for all)
Calculating the error between output and hidden layer
(Target – Output) i.e., (t – y) is the error in the output not in the layer.

7. (a) How the Gradient descent concept helps us to making predictions in deep Neural Networks? Justify with your answer.
Serves as a fundamental optimization technique to minimize the cost function of a model by iteratively adjusting the model parameters to reduce the difference between predicted and actual values, improving the model's performance. 13 CO2 **ANA**
Training Machine Learning Models

Neural networks are trained using Gradient Descent (or its variants) in combination with backpropagation. Backpropagation computes the gradients of the loss function with respect to each parameter (weights and biases) in the network by applying the chain rule. The process involves:

- Forward Propagation: Computes the output for a given input by passing data through the layers.
- Backward Propagation: Uses the chain rule to calculate gradients of the loss with respect to each parameter (weights and biases) across all layers.

Minimizing the Cost Function

- The algorithm minimizes a cost function, which quantifies the error or loss of the model's predictions compared to the true labels
- Gradient descent minimizes the Mean Squared Error (MSE) which serves as the loss function to find the best-fit line.
- Gradient Descent is used to iteratively update the weights (coefficients) and bias by computing the gradient of the MSE with respect to these parameters.
- MSE is a convex function gradient descent guarantees convergence to the global minimum if the learning rate is appropriately chosen.

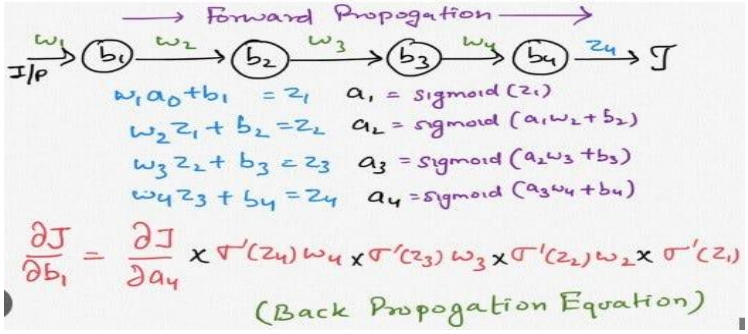
(OR)

(b) Analyze the impact of the Vanishing Gradient Problem and how to improve initialization and active function in Deep Learning Models?

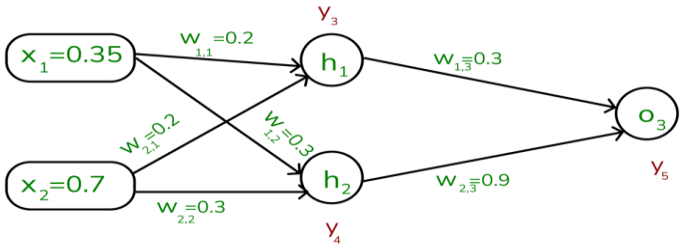
The vanishing gradient problem is caused by the derivative of the activation function used to create the neural network. The simplest solution to the problem is to replace the activation

13 CO2 **ANA**

function of the network. Instead of sigmoid, use an activation function such as ReLU.



8. (a) Find the total error at the output for a given neural network using back propagation.y=0.4



Initial Calculation

The weighted sum at each node is calculated using:

aj=∑(wi,j∗ xi)aj=∑(wi,j∗ xi)

Sigmoid Function

The sigmoid function returns a value between 0 and 1, introducing non-linearity into the model. 14 CO1 **APP**

yj=11+e−ajyj=1+e−aj1

Computing Outputs

At h1 node,

a1=(w1,1x1)+(w2,1x2)=(0.2∗ 0.35)+(0.2∗ 0.7)= 0.21a1=(w1,1x1)+(w2,1x2)=(0.2∗ 0.35)+(0.2∗ 0.7)=0.21

Once, we calculated the a1 value, we can now proceed to find the y3 value:

yj=F(aj)=11+e−a1yj=F(aj)=1+e−a11

y3=F(0.21)=11+e−0.21y3=F(0.21)=1+e−0.211

y3=0.56y3=0.56

Similarly find the values of y4 at h2 and y5 at O3

a2=(w1,2∗x1)+(w2,2∗x2)=(0.3∗0.35

y5=F(0.702)=11+e−0.702=0.67y5 =F(0.702)=1+e−0.7021=0.67

Error Calculation

Note that, our actual output is 0.5 but we obtained 0.67.

To calculate the error, we can use the below formula:

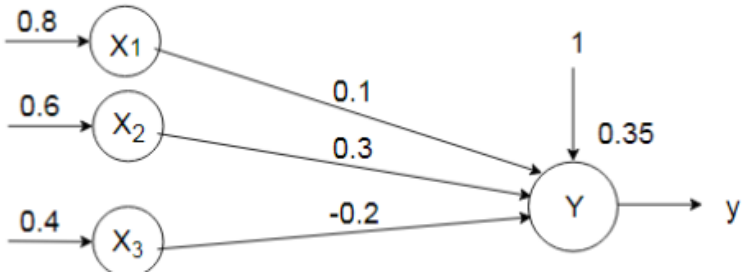
Errorj=ytarget−y5Errorj=ytarget−y5

Error=0.5−0.67=−0.17Error=0.5−0.67=−0.17

Using this error value, we will be backpropagating.

(OR)

(b) Simplify a perceptron to solve simple OR problem with two inputs and obtain the output of the neuron Y. y=0.7



14 CO2 **APP**

The weighted sum is computed as:

$$y_{out} = f(y_{sum}) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}$$

z=w1x1+w2x2+…+wnxn=XTWz=$w_1x_1+w_2x_2+\ldots+w_nx_n=X_TW$

The weight update formula is:

wi,j=wi,j+η(yj−y^j)xi$w^{i,j}=w^{i,j}+\eta(y^j-y\hat{}^j)x^i$

Bloom's Taxonomy:

REM – Remember **UND** – Understand **APP**– Apply **ANA**–Analyse

Faculty Incharge **Teaching Coordinator** **HoD/Dean**