IMRAZ ZIYA
241058045
BIG DATA ANALYTICS

ASSIGNMENT-3:

BASICS OF ANN

(1) Write the purposes of weights in the artificial neural network models

Ans:- Weights in artificial neural network (ANN) models play a critical role in determining the network's functionality and learning ability. Their purpose include

(i) Capturing Relationships between Inputs and Outputs
Weights represent the strength and direction of the connection between neurons. They determine how much influence an input feature or previous layer's output has on the subsequent layer.

(ii) Facilitating Learning:
During training, weights are adjusted through backpropagation and optimization algorithms (e.g. gradient descent). This adjustment enables the network to minimize the loss function and improve its predictive accuracy

(iii) Storing knowledge:
The learned weights encode patterns and information about the training data. These weights collectively define the model's ability to generalize to unseen data.

(iv) Enabling Feature Scaling:
Weights scale the input data, enabling the network to combine or modify input features in ways that are optimal for a given task, such as classification or regression.

(v) Providing Flexibility:
By adjusting weights, ANNs can adapt to different problems and datasets, making them versatile for various applications, such as image recognition, natural language processing, and time series prediction.

(vi) Supporting Non-Linearity:
Combined with activation functions, weights help the

network model complex, non-linear relationships in data, which is crucial for solving real-world problems.

(vii) Contributing to Decision Boundaries:-

Weights help define the decision boundaries in classification tasks, enabling the network to distinguish between different classes in the input data.

(2.) Illustrate the uses of different activation functions for artificial neural networks

Ans:- Activation functions are mathematical functions applied to the output of a neuron in an artificial neural network (ANN) to introduce non-linearity and enable the network to learn complex patterns.

Different activation functions have specific characteristics that make them suitable for various tasks.

(i) Sigmoid Activation Function:

Formula: $\sigma(x) = \frac{1}{1+e^{-x}}$

Range: $(0,1)$

Uses:-

- Ideal for binary classification tasks, as it maps outputs to a probability range (0 to 1).
- Suitable for the output layer of binary logistic regression models.

Limitations:-

- Vanishing gradient problem for large positive or negative inputs.
- Outputs are not zero-centered.

(ii) Hyperbolic Tangent (Tanh) Activation Function:-

Formula: $\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$

Range: $(-1,1)$

Uses:

- Often used in hidden layers for networks requiring zero-centered outputs.
- Suitable for modeling data where negative outputs are meaningful.

Limitations:

- Suffers from the vanishing gradient problem, similar to the sigmoid function.

(iii) Rectified Linear Unit (ReLU) Activation Function:

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

Range: $[0, \infty]$

Uses:

- Widely used in hidden layers of deep neural networks due to its simplicity and computational efficiency.
- Mitigates the vanishing gradient problem by maintaining gradients for positive inputs.

Limitations:

- Can suffer from the "dying ReLU" problem, where neurons become inactive for all inputs.

(iv) Leaky ReLU Activation Function:

Formula: $$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{otherwise} \end{cases}$$

where α is a small positive value (e.g., 0.01).

Range: $(-\infty, \infty)$

Uses:

- Addresses the "dying ReLU" problem by allowing small gradients for negative inputs.
- Suitable for deep networks with sparse data.

(v) Softmax Activation Function:

Formula: $$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

Range: $(0, 1)$, summing to 1 across all outputs

Uses:

- Commonly used in the output layer for multi-class classification tasks.
- Converts raw scores into class probabilities

(vi) **Linear Activation Function**

Formula: $f(x) = x$

Range: $(-\infty, \infty)$

Uses:

- Used in the output layer for regression tasks.
- Suitable when the output is continuous.

Limitations:

- Cannot introduce non-linearity, making it unsuitable for hidden layers

(vii) **Exponential Linear Unit (ELU)**

Formula: $f(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha(e^{x} - 1) & \text{otherwise} \end{cases}$

where $\alpha > 0$

Range: $(-\alpha, \infty)$

Uses:

- Addresses the dying neuron problem by smoothing the negative part of the output.
- Retains benefits of ReLU while allowing negative values.

(viii) **Swish Activation Function**

Formula: $f(x) = x \cdot \sigma(x)$,

where $\sigma(x)$ is the sigmoid function

Range: $(-\infty, \infty)$

Uses:
- Works well in deep networks and provides smooth gradients
- Often outperforms ReLU in specific architecture

③ Write the algorithm used in a single layer perceptron model for learning

Ans: The Single Layer Perceptron (SLP) is a type of neural network that uses a single layer of weights to map input features to outputs. Its learning process involves adjusting weights to minimize errors in classification or regression. Below is the step-by-step algorithm used for training a single-layer perceptron.

Single Layer Perceptron Learning Algorithm:

(i) Initialize Parameters:
- Randomly initialize the weight vector 'w' and bias 'b' with small random values (e.g., close to zero).
- Set the learning rate η (a small positive constant)

(ii) Input and Output:
- Prepare the training dataset with 'N' samples, where each sample consists of:
  - Input feature vector $x_i = [x_{i1}, x_{i2}, \dots x_{id}]$ (of size d).
  - Target output y_i (e.g., 0 or 1 for binary classification)

(iii) Activation Function:
- Use a step function (or other suitable activation functions) to compute the perceptron's output:

$$\hat{y}_i = f(w \cdot x_i + b)$$

where $f(z)$ is typically

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

(iv) Training Loop:
- For each training epoch (iteration over the dataset)

- For each training sample (x_i, y_i):

(i) Calculate the predicted output

$$\hat{y}_i = f(w \cdot x_i + b)$$

(ii) Compute the error:

$$e_i = y_i - \hat{y}_i$$

(iii) Update the weights and bias:

- If $e_i \neq 0$ (i.e there is an error):

$$w = w + \eta e_i x_i$$

$$b = b + \eta e_i$$

(v) Stopping Criteria:

- Continue the training loop until:
  - Errors are minimized (e.g. no misclassified samples)
  - A maximum number of epochs is reached.

(vi) Output the Final Model:

- The perceptron learns the final weight vector 'w' and bias 'b', which defines the decision boundary.

Key Points:

- The learning process adjusts weights iteratively to reduce the classification error.
- A single-layer perceptron can only solve linearly separable problems. If the data is not linearly separable, the perceptron will fail to converge
- The learning rate η determines the step size for weight updates and must be chosen carefully to ensure convergence.

④ Distinguish the following
(a) Threshold
(b) Learning Rate
(c) Activation

Ans (a) Threshold:

Definition:
- The threshold is a parameter that determines whether the output of a neuron is activated (fires) or not, based on the weighted sum of inputs.
- It is used in conjunction with a step function or similar activation functions.

Purpose:
- To define a boundary or limit for the neuron to produce an output.
- In simple perceptrons, if the weighted sum of inputs exceeds the threshold, the output is 1; otherwise, it is 0.

Role in a Model:
- Threshold often appears as a bias term 'b', which is learned during training to shift the decision boundary.
- For instance, the perceptron's output is determined by:

$$output = f(w \cdot x + b)$$

(b) Learning Rate (η)

Definition:
- The learning rate is a hyperparameter that controls the step size during weight updates in the training process.

Purpose:
- To regulate how much the weights are adjusted in response to the error for each iteration.
- Ensures smooth convergence to an optimal solution during training.

Role in a model:
- Used in weight and bias update rules

$$w = w + \eta e x$$

$$b = b + \eta e$$

- Here, η is the learning rate, and e is the error.

Key Considerations:
- A small learning rate ensures slow but stable convergence, reducing the risk of overshooting the optimal solution.
- A large learning rate may speed up convergence but can lead to oscillations or divergence

(C) Activation:

Definition:
- Activation refers to the function applied to the weighted sum of inputs (including bias) to determine the output of a neuron.
- Common activation functions include step, sigmoid, ReLU and softmax

Purpose:
- To introduce non-linearity into the model, enabling it to learn complex patterns and relationships.
- Determines the output of a neuron based on the weighted sum and bias.

Role in a Model:
- Transforms the input signal into an output signal based on the chosen function:

$$output = f(w \cdot x + b)$$

- $f(z)$ could be linear, non-linear or threshold-based.

Types of Activation functions:
- Linear: Directly passes the input (used in regression)
- Non-Linear: Includes sigmoid, ReLU, tanh, etc. which enables the model to solve complex, non-linear problems.