Assignment

BASICS OF ANN

1. write the purpose of weights in the Artificial N.N (ANN)

→ 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 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

→ 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)

use: 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)

use:- widely used in hidden layers of deep neural

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 +ve inputs

Limitations

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

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 +ve value (e.g. 0.01)

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

Use :- 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

use :-

- Commonly used in the output layer for multi-class classification task

- 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

→ The simple 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 +ve constant)

ii) Input and Output

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

iii) Training Loop

- For each training Epoch
  - For each (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)

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

$$b = b + \eta e x_i$$

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.

④ Distinguish the following Threshold, Learning Rate, Activation

→ Threshold:- The threshold is a parameter that determines whether the output of neuron is activated

- It is used in conjection with a step function or similar activation function.

2) Learning Rate:-

A hyperparameter that controls the step size in weight updates. control how much the weights are updated during training.

3) Activation Function:-

→ used to introduce non linearity into the model, Enabling it to learn complex patterns and relationships.

→ Determine the output of a neuron based on the weighted sum and bias.

ex: output = $f(wx+b)$

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

Name : Karthik. E

Reg. No : 24105 8016

Sub : MLDB