

# Activation functions Assignment

**Q1.** Explain the role of activation functions in neural networks. Compare and contrast linear and nonlinear activation functions. Why are nonlinear activation functions preferred in hidden layers?

**ANS:**

Activation functions introduce non-linearity to neural networks, which allows the model to learn and represent complex patterns in data. Without them, the network would behave like a linear model, no matter how many layers it had, limiting its ability to model complex real-world problems. Activation functions help control the output of neurons by determining whether a neuron should be activated or not based on the input it receives.

## Linear vs. Nonlinear Activation Functions:

**Linear Activation Function:** A linear function does not introduce non-linearity. For instance, a function like  $f(x)=x$  produces the output proportional to its input. It's not suitable for complex data because the composition of linear transformations is still linear, limiting the learning capacity of the network. **Nonlinear Activation Function:** Introduces non-linearity to the model, allowing the network to learn from complex data. Examples include Sigmoid, ReLU, and Tanh functions. Nonlinearity is necessary for hidden layers, enabling the network to stack multiple layers to solve problems like image classification, natural language processing, etc. **Why Nonlinear Activation Functions are Preferred in Hidden Layers:** Nonlinear activation functions allow for more expressive power in the neural network. With nonlinearities, neural networks can approximate complex functions, perform hierarchical feature extraction, and learn complex patterns in data. Linear activation in hidden layers would make the entire network equivalent to a linear function, regardless of depth.

**Q2.** Describe the Sigmoid activation function. What are its characteristics, and in what type of layers is it commonly used? Explain the Rectified Linear Unit (ReLU) activation function. Discuss its advantages and potential challenges. What is the purpose of the Tanh activation function? How does it differ from the Sigmoid activation function?

**ANS:**

The Sigmoid function is defined as:  $f(x)=\frac{1}{1+e^{-x}}$   $f(x)=\frac{1}{1+e^{-x}}$

Characteristics:

Output range: (0, 1), making it useful for models requiring probability-based output. It squashes large positive or negative values into a small range, which helps in interpreting the output as probabilities. Derivative: The gradient becomes very small for values far from zero, leading to the vanishing gradient problem in deep networks. Common Usage:

Used in output layers for binary classification problems (logistic regression). Rarely used in hidden layers due to the vanishing gradient issue. Rectified Linear Unit (ReLU) Activation Function: The ReLU function is defined as:  $f(x)=\max(0,x)$   $f(x)=\max(0,x)$

Advantages:

Efficient Computation: ReLU is computationally efficient because it only needs to compare with zero. Solves Vanishing Gradient Problem: Unlike Sigmoid and Tanh, ReLU avoids the vanishing gradient problem for positive values. Sparse Activation: Since ReLU outputs zero for negative inputs, it allows for sparsity in the network, which can improve generalization and reduce computation. Challenges:

Dying ReLU Problem: Neurons can "die" if they always output zero for negative inputs, causing some neurons to never activate and thus not contribute to learning. Tanh Activation Function: The Tanh function is defined as:  $f(x)=\tanh(x)=\frac{e^x-e^{-x}}{e^x+e^{-x}}$   $f(x)=\tanh(x)=\frac{e^x-e^{-x}}{e^x+e^{-x}}$

Purpose:

The Tanh function maps input to a range of (-1, 1), which can center the data better than Sigmoid, making gradients flow better during training. Differences from Sigmoid:

The range of Tanh is (-1, 1), whereas Sigmoid outputs between (0, 1). This centering effect in Tanh helps mitigate some issues related to training deep networks. Tanh still suffers from the vanishing gradient problem, though less severe than Sigmoid.

**Q3.** Discuss the significance of activation functions in the hidden layers of a neural network.

**Ans:**

Activation functions in the hidden layers help the network learn complex representations of data. Nonlinear activation functions (like ReLU, Tanh, and Sigmoid) allow the network to capture non-linear relationships between input features and output labels. Without non-linear activation functions, a neural network would not have the expressive power to model intricate patterns like those found in images, speech, or text.

Additionally, different activation functions have different properties in terms of gradient flow and optimization efficiency, which directly impacts how well a network can learn over time.

**Q4.** Explain the choice of activation functions for different types of problems (e.g., classification, regression) in the output layer.

**Ans:**

For Classification Problems:

Sigmoid: Common in binary classification (output layer) as it maps outputs to probabilities between 0 and 1.

Softmax: Used in multi-class classification problems as it generalizes Sigmoid by outputting a probability distribution over classes.

For Regression Problems:

Linear: In the output layer, a linear activation function is used when predicting continuous values. ReLU/Tanh: In hidden layers, ReLU or Tanh are typically used for their non-linear properties, which help the network capture complex relationships.

**Q5.** Experiment with different activation functions (e.g., ReLU, Sigmoid, Tanh) in a simple neural network architecture. Compare their effects on convergence and performance.

**Ans :**

To compare the effects of activation functions on a simple neural network, you can implement a basic feedforward network with different activation functions (e.g., ReLU, Sigmoid, Tanh) and observe their effects on convergence and performance. The architecture might include:

Input layer (for features) Hidden layers with various activation functions (ReLU, Sigmoid, Tanh) Output layer (depending on the task, e.g., softmax for classification or linear for regression) You can observe:

Convergence speed: ReLU usually converges faster than Sigmoid and Tanh due to the absence of the vanishing gradient problem. Accuracy/Performance: ReLU often outperforms Sigmoid and Tanh, especially in deeper networks. Gradient issues: Sigmoid and Tanh suffer from the vanishing gradient problem, leading to slower training and potentially worse performance in deep networks. By running the network through different tasks, you can observe how these functions impact training stability, convergence time, and generalization ability.