Challenge #6

1. Implement a simple neuron (a.k.a. perceptron) with two inputs and a sigmoid activation function.
Hints: https://machinelearningmastery.com/a-gentle-introduction-to-sigmoid-function
2. Use the perceptron learning rule (Google or LLM it) to train the neuron to realize the following binary logic functions:
a. NAND
b. XOR
3. Good video resources:
a. A Gentle Introduction to Neural Networks:
https://www.youtube.com/watch?v=b7oYqAlX_Bo
b. But what is a neural network? https://www.youtube.com/watch?v=aircAruvnKk

Melaku Desalegn, ECE510-2025

Challenge #6

1. Implement a simple neuron, also known as a perceptron, with two inputs and a sigmoid activation function.

Hints: https://machinelearningmastery.com/a-gentle-introduction-to-sigmoid-function

2. Implement the perception learning rule to train the neuron to recognize the following binary logic functions:

a. NAND

b. XOR

3. Recommended video resources:

a. A Gentle Introduction to Neural Networks:
https://www.youtube.com/watch?v=b7oYqAlX_Bo

b. But What Is a Neural Network? https://www.youtube.com/watch?v=aircAruvnKk

Title: Implementation and Analysis of Logic Gates Using Multi-Layer Perceptron with Sigmoid Activation Function

This paper examines the implementation of a basic neural network, specifically a Multi-Layer Perceptron (MLP), to realize fundamental binary logic gates including AND, OR, NAND, and XOR. A single-layer perception is capable of successfully learning linearly separable gates (AND, OR NAND); however, a two-layer MLP architecture is required to learn the non-linearly separable XOR function. The sigmoid activation function facilitates smooth transitions, while the perception learning rule, alongside gradient descent, is utilized to update the weights and biases. An analysis of errors, training plots, and visualizations of decision boundaries illustrate the learning dynamics of the network and emphasize the necessity of hidden layers for modeling non-linear functions.

Keywords

Multi-Layer Perceptron (MLP), Neural Networks, XOR Gate, Logic Gates, Sigmoid Activation, Decision Boundary

Objectives / Research Questions

- Is a single-layer perception capable of learning both linearly separable and non-linearly separable binary logic gates?

- In what ways does the addition of a hidden layer enable the MLP to learn the XOR gate?

- What influence does the sigmoid activation function exert on the learning process?

Methods / Approach

The study involved the implementation of a simple MLP in Python utilizing NumPy. The network contains two input neurons, a hidden layer with two neurons, and a single output neuron. A sigmoid activation function was applied to both the hidden and output layers. The MLP was trained using the perceptron learning rule and gradient descent over 10,000 epochs, with Mean Squared Error (MSE) recorded for each epoch to facilitate an analysis of convergence. Decision boundary plots were generated to visualize the classification regions learned by the model.

Results / Findings

The AND, OR, and NAND gates converged within a few hundred epochs, confirming their linear separability. In contrast, convergence for the XOR gate was achieved exclusively with the introduction of a hidden layer, thus underscoring the requirement for non-linear transformations. Error plots indicated that the MSE decreased over time for all gates, with the XOR function necessitating a greater number of epochs to converge. The decision boundary plot for the XOR demonstrated the model's ability to form non-linear boundaries due to the transformative capacity of the hidden layer.

Discussion / Interpretation

The results corroborate the theoretical understanding of perceptron learning. Single layer perceptron’s are limited to learning linearly separable functions such as AND, OR, and NAND, while the XOR function necessitates a hidden layer to facilitate non-linear transformations. The sigmoid activation function contributes to stable learning by ensuring smooth gradients. The visualization of the decision boundary for the XOR gate validates the MLP's capacity to modify the input space adequately for correct classification of XOR outputs.

Conclusion

This study successfully demonstrated that a Multi-Layer Perceptron equipped with a hidden layer and sigmoid activation function can implement both linearly and non-linearly separable logic gates. The experiments highlight the significance of network depth in managing complex functions. Future research may investigate various activation functions, learn rates, or extend to larger neural network architectures for multi-bit logic and application to real-world datasets.

References / Bibliography

1. Machine Learning Mastery. "A Gentle Introduction to the Sigmoid Function" - https://machinelearningmastery.com/a-gentle-introduction-to-sigmoid-function

2. YouTube. "A Gentle Introduction to Neural Networks" - https://www.youtube.com/watch?v=b7oYqAlX_Bo

3. YouTube. "But What Is a Neural Network?" - https://www.youtube.com/watch?v=aircAruvnKk

Figures / Tables / Appendices

Figure 1: Training error curves for AND, OR, NAND, and XOR gates.

Figure 2: Decision boundary plot for the XOR gate following training.

Appendix:

- Python code snippets for MLP training and visualization of decision boundaries.

- Mathematical exposition of forward and backward propagation in matrix form.