Student id: i6358518

Abstract

This report presents the implementation and evaluation of a simple perceptron algorithm designed to learn and classify logical functions. The perceptron is implemented in Python and tested on the logical AND function. Additionally, adjustments are made to handle a linear function y=2x+1, and the impact of adding more target values is analyzed.

Introduction

Artificial Neural Networks (ANNs) are computational models inspired by the human brain's neural networks. They consist of interconnected nodes, or neurons, that process data and learn to perform tasks by adjusting the connections, or weights, between them. One of the simplest types of ANN is the perceptron, which is a binary classifier that maps input features to an output using a linear decision boundary.

This report focuses on implementing a simple perceptron to classify logical functions such as AND and linearly separable functions like y=2x+1. The perceptron algorithm is fundamental in understanding how more complex neural networks operate and serves as a building block for deeper learning models. By examining the perceptron's performance on these tasks, we gain insights into its strengths and limitations.

Implementation

The perceptron algorithm is implemented in Python, using NumPy for efficient numerical computations. The perceptron is trained on the logical AND function with the following steps:

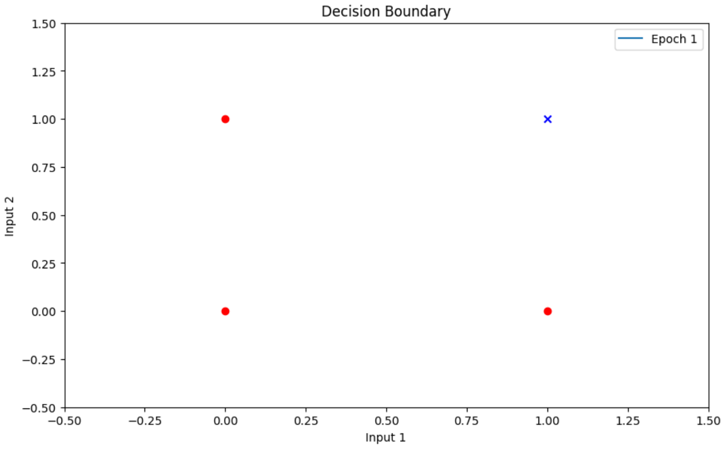
1. Initialised weights randomly, including a bias term.
2. Use the step function as the activation function.
3. Adjust weights iteratively based on the error until all target values are correctly classified or the maximum number of epochs is reached.
4. Print weights and errors during the training process for transparency and debugging.
5. Visualize the decision boundary at each iteration and plot the training error over epochs.

The algorithm is then modified to handle the function y=2x+1 by interpreting the perceptron as a linear classifier. The effect of adding more target values is also explored.

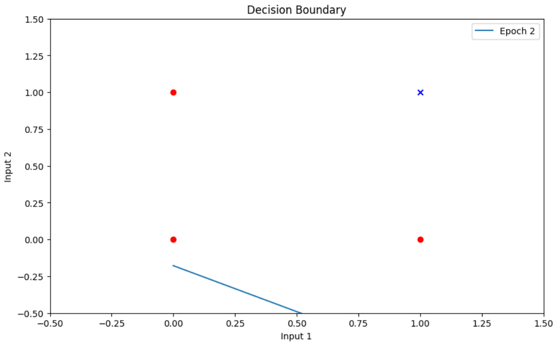
Results

1. **AND Function**: The perceptron successfully learned the AND function within a few iterations, with the final weights and number of iterations printed out. The outputs for the input vectors matched the expected AND outputs.

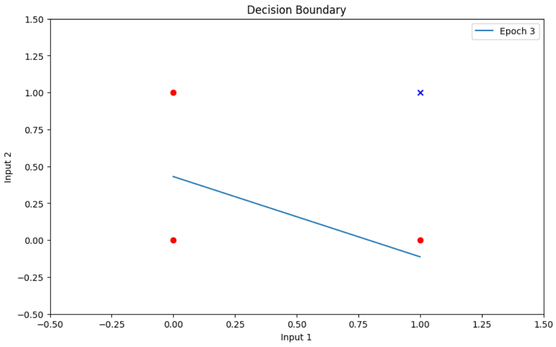
The decision boundary at each iteration is visualised below: Epoch 1



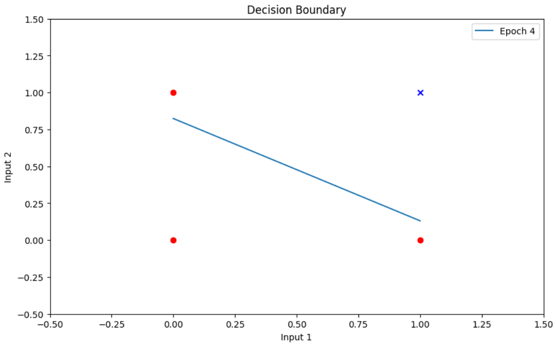
Epoch 2;



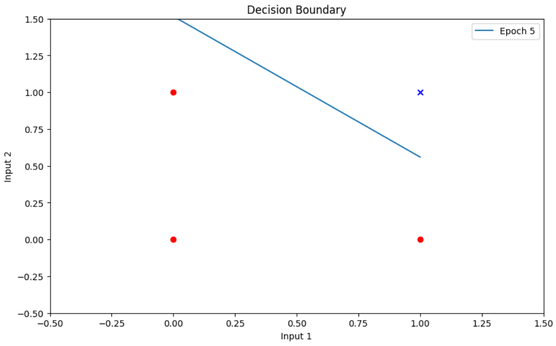
Epoch 3

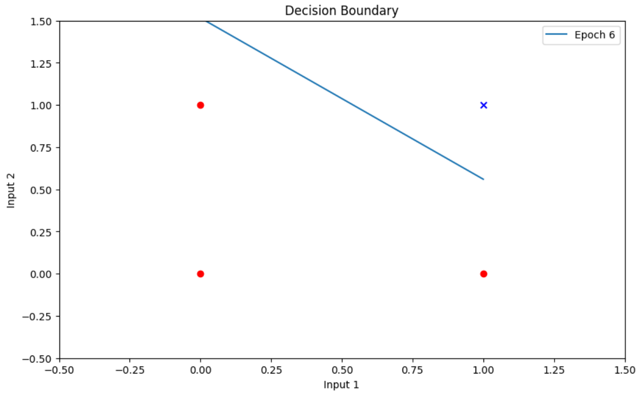


Epoch 4

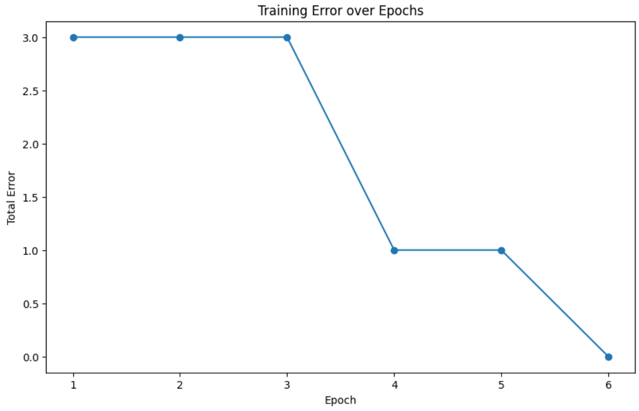


Training Error Over Epochs: The training error over epochs is plotted, showing the reduction in error as the perceptron learns to classify the inputs correctly.





Linear Function y=2x+1:
The perceptron was adjusted to interpret the linear function, and it correctly classified the inputs based on the threshold defined by the function.



Adding More Target Values: Increasing the number of target values to test the perceptron showed that while the perceptron can handle additional points, it remains limited to linearly separable data. Additional target values that do not follow a linear separation might not be classified correctly.

Conclusion

The perceptron successfully learned and classified the logical AND function and a linear function y=2x+1. The implementation demonstrated the perceptron's capability to learn linearly separable data but highlighted its limitations when dealing with non-linear separable data. Future work could involve exploring multi-layer perceptrons or other neural network architectures to handle more complex datasets.