# Two Moons Binary Classification with TensorFlow

Name: Skyler Choplin

## Implementation

The experiment aimed to classify the two moons dataset using a simple perceptron. As indicated in the instructions, the make moons function was used from Scikit Learn to generate the initial dataset for the experiment. The dataset was then split 80/20 training/test data, scaled using the StandardScaler from TensorFlow to normalize ranges and stabilize training convergence.

Once the dataset has been processed, the model is defined using a Sequential structure from Keras to arrange the layers in linear stacks. Granted, it is only a one layer, one perceptron model. Then once the structure has been defined the model is then compiled using the Adam optimizer, while setting flags to keep track of accuracy and what function will be used as the loss function. Then the actual training happens using the TensorFlowTrainer class implementation for fitting data.

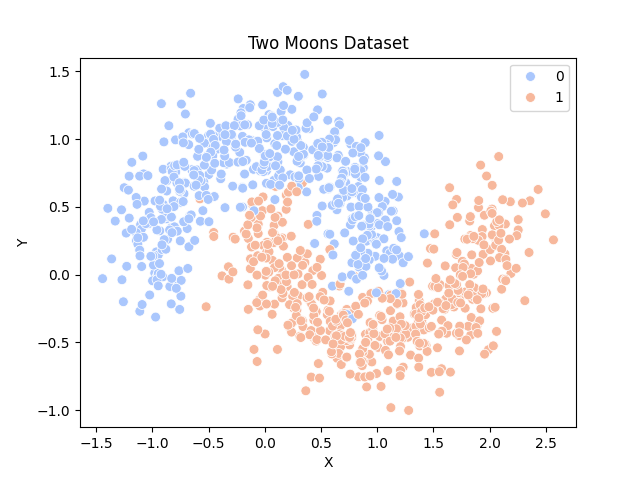


Figure 1: Two Moons Dataset Distribution

## Results and Observations

Across multiple training runs, performance was primarily influenced by the number of epochs and sample size. When trained for 50 epochs, test accuracy fluctuated between 77% and 88%, with most runs near the 87–88% mark. Extending training to 60 epochs yielded a final accuracy of 81%, with the model beginning to slightly plateau and potentially overfit, as indicated by narrowing gaps between training and validation loss curves. Throughout runs, the random state was set to 42 for consistent generation of data and consistency of results.

A secondary experiment was conducted using 10,000 samples, where training ran for 50 epochs. This larger dataset improved overall generalization, achieving approximately 89% accuracy. The learning curves demonstrated rapid convergence within the first 10 epochs, followed by a gradual tapering in both loss and accuracy improvements. The correct vs. incorrect prediction scatter plots for both the 1,000 and 10,000-sample runs illustrate that the model reliably captures the overall structure of the moons but still misclassifies points near the class boundaries — regions where the two distributions overlap due to noise.

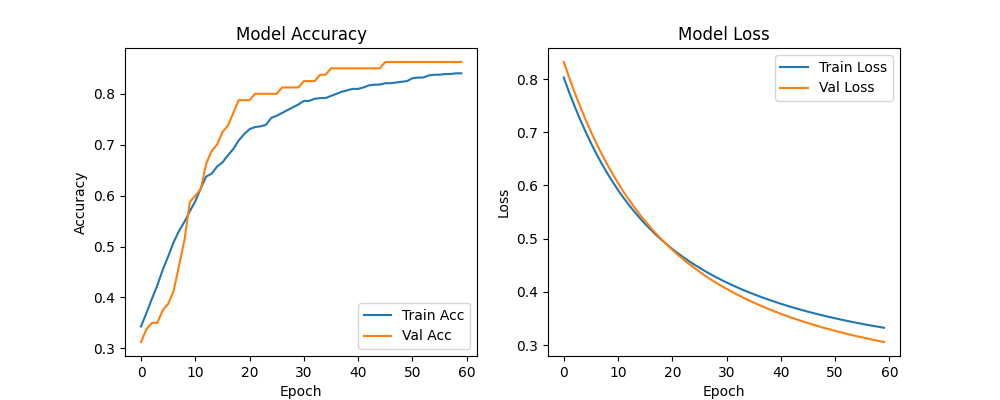


Figure 2: Accuracy and Loss Curves (1k Sample, 60 Epochs)

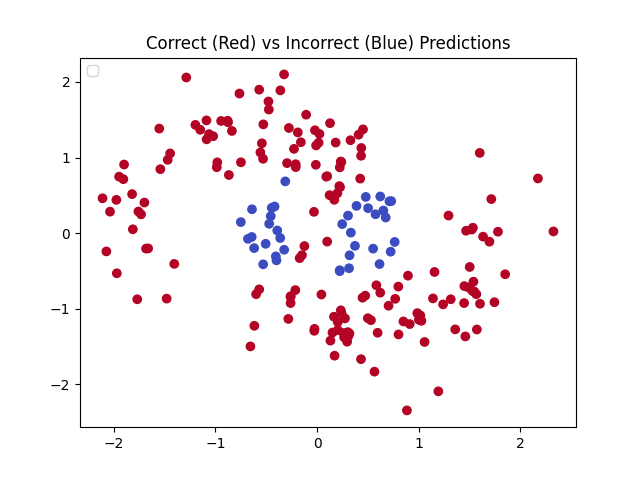


Figure 3: Correct (Red) vs Incorrect (Blue) Predictions (1k Sample)

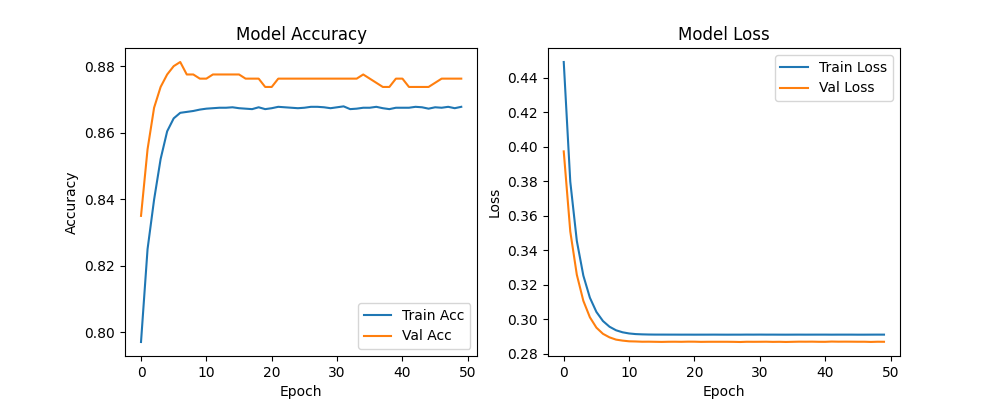


Figure 4: Accuracy and Loss Curves (10k Sample, 50 Epochs)

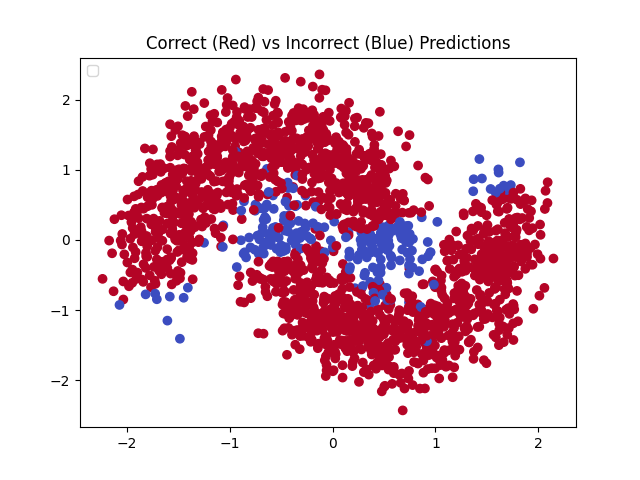


Figure 5: Correct (Red) vs Incorrect (Blue) Predictions (10k Sample)

## Conclusion

This experiment highlights how both dataset size and epoch count directly influence model performance and stability. While a simple perceptron can achieve reasonably high accuracy on a non-linear dataset; it struggles with boundary precision, which is an expected limitation of its linear decision surface. Increasing the number of samples noticeably improved performance and reduced variance in results.

If time permitted, further experimentation with different scalers such as MinMaxScaler and RobustScaler to see the effect on performance would have been interesting for seeing the effect the scaler will have on the outcome of the perceptron.