Yash Sharma
Binary Classification Assignment

Short Writeup

For this assignment, the goal was to perform binary classification on the spirals dataset using a multilayer perceptron.

After learning how to construct a network in Tensor Flow, I began the tuning process. I ended up becoming convinced that this assignment was not possible, believing that only a incredibly deep neural network that couldn't possibly run on my laptop could learn a spiral decision boundary.

I realized later that I might have had such trouble tuning the standard neural network parameters and hyperparameters due to the fact that my spiral data had 3 turns rather than the 1.5 turns required for the assignment.

Because of my trouble, I ended up giving up and starting over, and working on my initial hunch that using standard activation functions would not work, and possibly a sine activation function would, as the spirals are periodic.

With this, I began researching how one would use sine as an activation function for binary classification. I found that neural nets using sine train quickly, and are normally quite simple, and hence I lowered the number of epochs to about 50, the learning rate to around 1e-3, and set one hidden layer.

I then found that a “good” property of the sine activation function was it's ability to produce odd-like functions, but this only occurs if the bias is removed, and hence I did just that. Lastly, I found that what really determines the fit is the initial weights provided, and I began to tune on that.

I first started with a random uniform distribution at [-1,1] and quickly iterated to a range closer to [-1/2,1/2]. Then I began to see differences in tuning the individual layers. Tuning the hidden layer's weights upward began to see results, and while keeping the output layer's weights at [-1/2,1/2], I finally saw a spiral-like pattern with the hidden layer's weights at [-7/2,7/2]. Tuning the output layer's weights downward seemed to improve the spiral fit, and I finally ended up with the hidden layer's initial weights at [-7/2,7/2] and the output layer's initial weights at [-0.001/2,0.001/2].

The model does have a non-negligible variance, as the spiral fit is obtained every time, but sometimes the fit is a bit noisy. Furthermore, outside of the datapoints area, there is not even close to a fit. However, the task was to fit the spirals dataset, and that was achieved here with a single hidden layer with 16 neurons, fitting not only 1.5 turns but 3 turns.

There probably won't be another use for the sine activation function, but I'm glad I was able to succeed in this task, fitting such a complex decision boundary with such a simple model!