Linear Limitations

An overview of the limitations of a single layer perceptron model.

Chapter Goals:

- Understand the limitations of a single layer perceptron model

A. Linear decision boundary

The input data we've been using to train and evaluate the single layer perceptron model has been pairs of (x, y) points with labels indicating whether the point is above (labeled **1**) or below (labeled **0**) the $y = x$ line. We trained the pretrained model on the generated data for 100,000 iterations at a batch size of 50.

Running the model on some randomly generated 2-D points will give a plot like this:
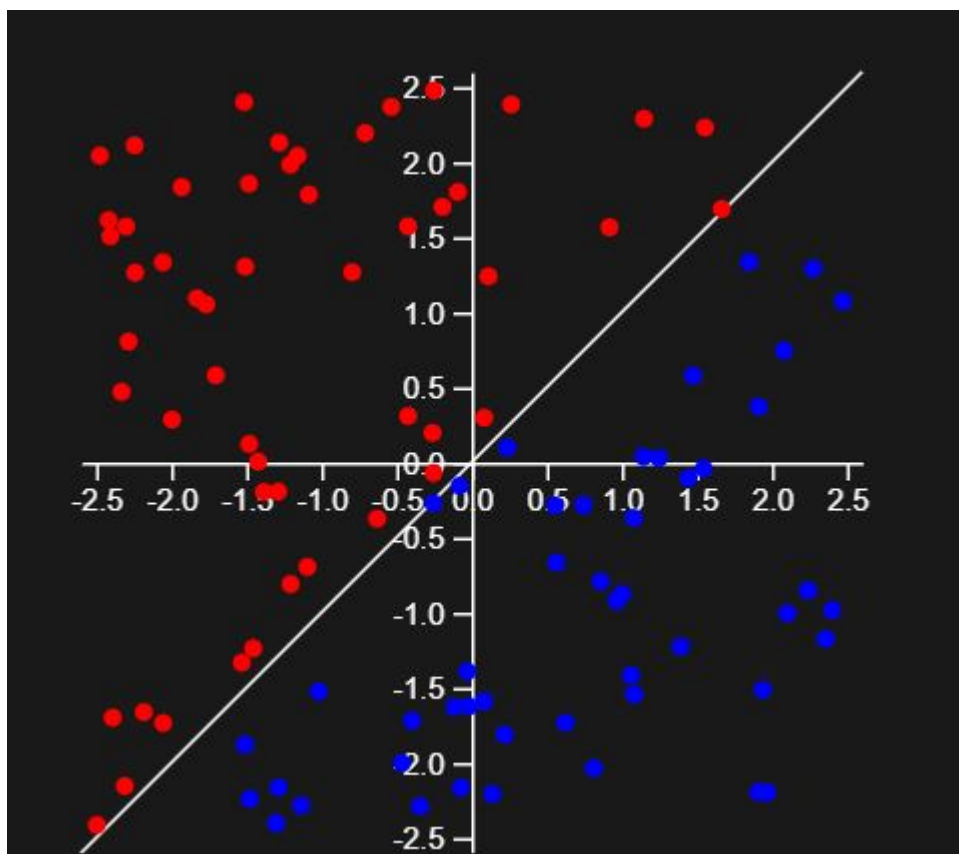
In the plot above, there are 100 randomly generated 2-D points. The model

was tasked with classifying whether or not each point is above or below the *y*

= *x* line (red for above, blue for below). As you can see, the model classifies each of the points correctly.

However, if we instead train the single layer perceptron on 2-D points being inside or outside a circle, the plot will look something like this:
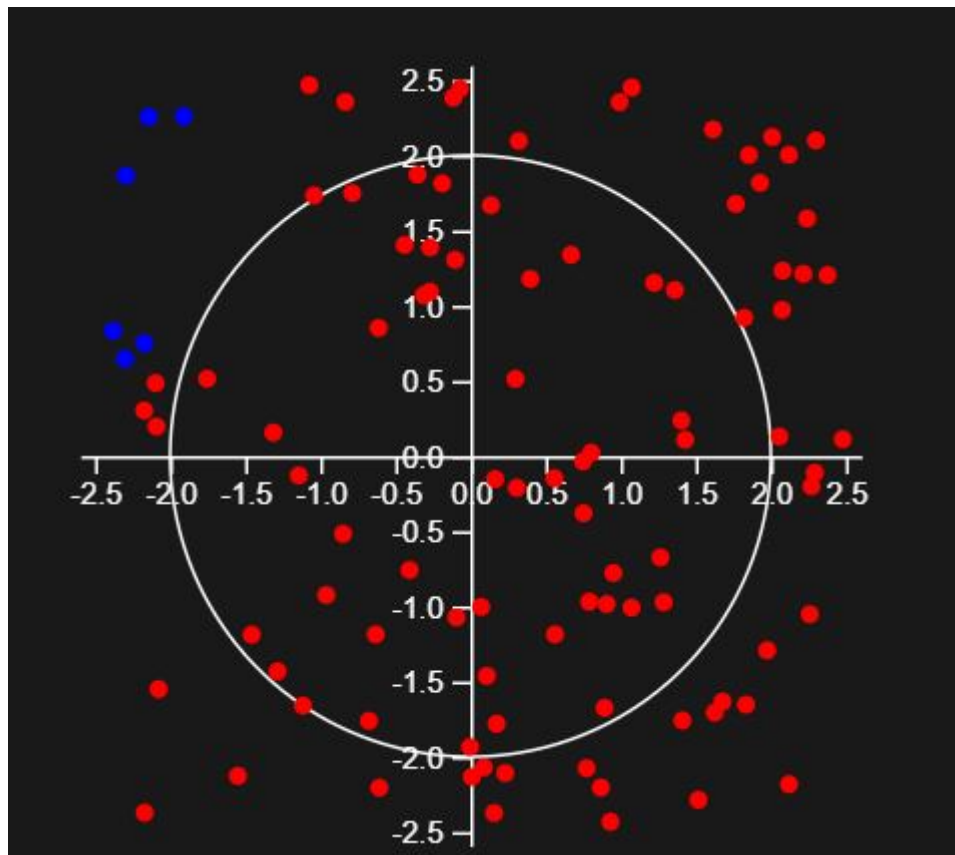
In this case, red means the model thinks the point is inside the circle, while blue means the model thinks the point is outside the circle. In this case, the single layer perceptron does a very poor job of classifying the points, despite being trained on the circle dataset until the loss converged.

The lack of performance from the model is not due to undertraining, but instead due to an inherent limitation in the abilities of a single layer perceptron. With just a single fully-connected layer, the model is only able to learn linear decision boundaries. So for any set of points that are divided by a line in the 2-D plane, the model can be trained to correctly classify those points. However, for non-linear decision boundaries, such as this circle example, no matter how much you train the model it will not be able to perform well in classification.

In the next chapter, we delve into how we can create perceptron models that

are able to learn non-linear decision boundaries.