Matt Williams
4/13/2022
Computer Vision
Lab 6

Problem 2:

The Neural Network I made in the Tensorflow Playground software is based on the binary classification problem. The Neural Network I made is fully connected, meaning the outputs generated by each neuron in one layer will be passed to every neuron in the following layer as input. The model contains 6 layers: an input layer, an output layer and 4 hidden layers. The input layer of the model accepts 7 inputs. The number of neurons in each hidden layer is as follows: 7 neurons for the first, 5 neurons for the second, 3 neurons for the third, and 1 neuron for the final hidden layer. The model has a learning rate of 0.03. This number represents how fast our optimization function will “learn” the dataset. If the learning rate is too low, the model will take a long time to converge on a solution. If we need to stop the training after a certain number of epochs because the learning rate was too low, this can cause our model to underfit the data. On the other hand, if our learning rate is too high, the model might converge on a solution that isn’t the best solution. This could result in the model overfitting the data.

Each hidden layer in the Neural Network uses the ReLU activation function (except for the output layer which most likely uses some sort of entropy based function). The activation function is the final mathematical operation for a single neuron, where its goal is to take the output from multiplying the inputs with the weights as input, then using that input to produce some value within a certain interval. For the ReLU activation function, this interval is between [0,infinity). The goal of the ReLU function is to keep the output of a neuron greater than or equal to zero. The Neural Network also uses the L2 Regularization technique. Regularization techniques are used to help prevent overfitting in our models. Without going too much into the details of the algorithm, L2 Regularization works in conjunction with our loss function to help decrease the weights of features that aren’t significant to the classification of the data. For L2 regularization, the weights of insignificant features will be close to 0, but will never actually be set to zero (as opposed to L1 regularization). The regularization rate used in this model was 0.001.

The model used here uses a train-test split of 80% and 20% respectively. The model also uses a batch size of 10. This means that the model trains on 10 data point intervals before performing backpropagation in order to update the weights of the model. The model took around 600 epochs before converging on a solution for the data set. After training, we had a training loss of 0.7% and a slightly larger testing loss of 1%, which is to be expected.