Assignment 4
CMPT 310
James Young
301243937

This assignment we used Multi Layer Perceptrons to detect handwritten prime digits. In order to do that, we created a Neutral Network. We use this Neutral Network to learn an approximation. In this Neural Network we have an MLP with one hidden layer which each has 5 neurons. I implemented the forward and backward function for DenseLayer that contains hidden neurons for Sigmoid activation functions.

In this DenseLayer the functions compute_activations, compute_gradients, and update_weights are implemented using the output neurons and the gradient of the loss with respect to weight and bias.

In the NeuralNet class the same functions are implemented for each layer in the network. In compute_gradients, back propagation was used to compute each gradient of the loss layer and passing it to the previous layer.

After the implementation and testing the correctness, we can build the MLP and train. I use the toy_example_regressor.py script which loads the data and trains MLP. The network weights are saved to simple_net_weights.pk1/ The predicted function is saved to data_function.png and matches f(x) = x^2. The validation loss found is around 0.011.

Finally we can use the same layers to find the prime digits. The data is saved into prime_net_weights.pk1. And the accuracy found is around 97%.