Uzair Mustafa 21100049
Report for MyNetwork

In this function, I implemented a neural network, which recognised handwritten digits. For that, I had training data containing 60000 images. Each image was 28 x 28 pixels meaning each image consisted of a total of 784 pixels that made the first layer of the neural network. It was a 3 layered neural network. The hidden layer consisted of 30 elements and the output layer consisted of 10 elements because the handwritten digits were from 0 - 9. I read the pixels and normalised by dividing each pixel by 255 and then read the labels and applied one hot encoding to each label for the purpose of comparing with argmax of activations in the future. The one first step is that we used random predictions for the results and using forward pass, we sent those activations to the output layer to compare how far we are from the actual results. Then the second steps included partial derivatives of the difference in weights between actual and predicted with respect to the sigmoid functions. Through these partial derivatives, the amount of weight that needed to be adjusted from the random weights that we calculated were done and the weights were then updated based on those partial derivatives. Sigmoid function helped us converting sum of w1*a1 + bias between 0 and 1. Now that the weights have been adjusted, it was time to evaluate the training data under two epochs. Actual labels and calculated activations were sent to the accuracy function. If the argmax of each column matched then accuracy was incremented. With the updated weights, test data was evaluated. The accuracy came out to be 93%, which shows the model was correct