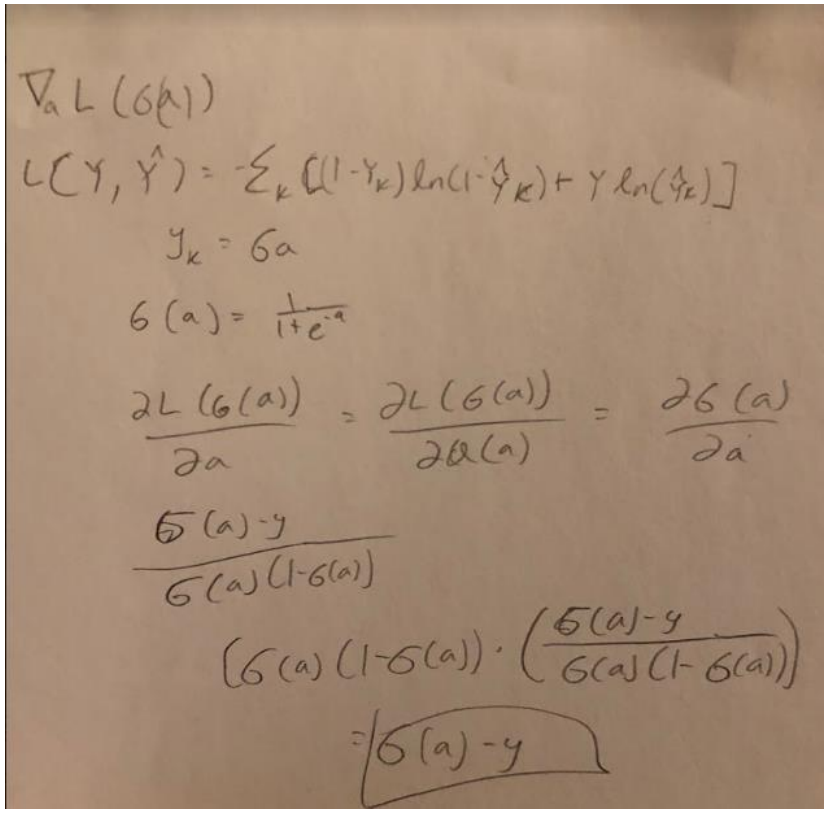
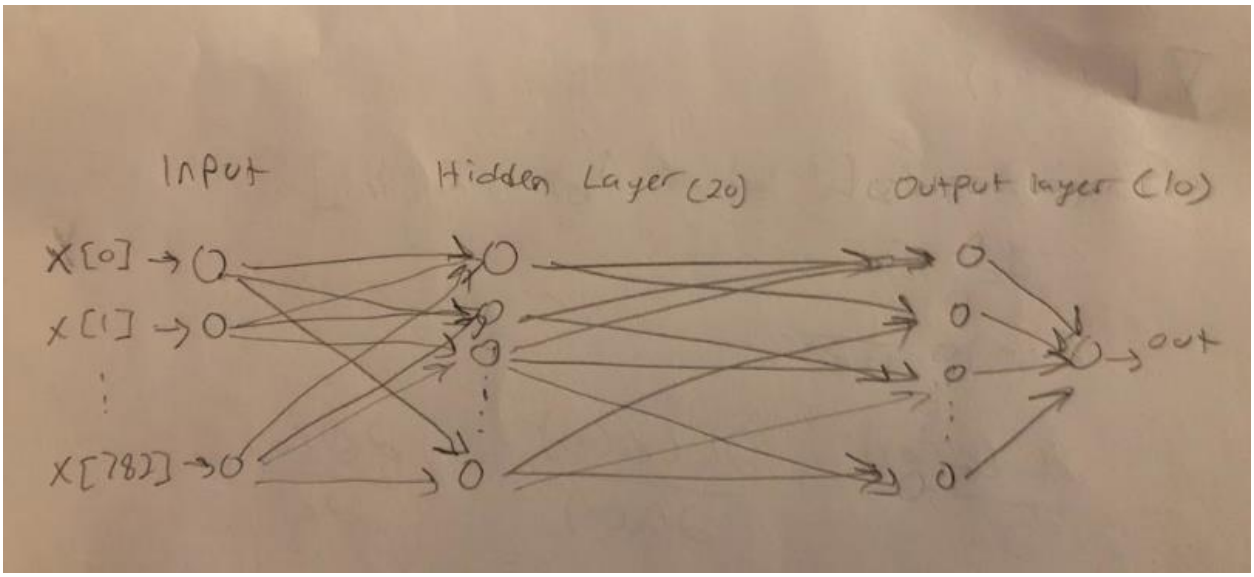
Abstract:

In this assignment, we were tasked to complete methods in a Python program that runs a forward & back propagation in a neural networks model. This assignment includes creating several methods required to successfully complete the neural network model. One method is the sigmoid function, which constricts the output of a non-linear function to between 0 and 1. Another is the back propagation method. This method uses weights and biases to determine a value for each feature in each layer of a network, sends each feature through the activation & loss functions, and outputs a prediction on the model. It does this by first forward propagating through each layer in the model, updating each feature with the corresponding weights and biases, and passing them through the activation function. Once it reaches the end layer, it then takes the gradient of each feature of the layers in reverse, determining the smallest error for each feature. This is then used to determine the most accurate prediction of a new set of inputs.

1) $\nabla_a L(\sigma(a))$,



2)

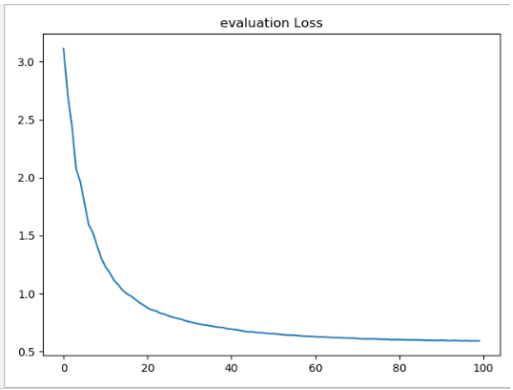


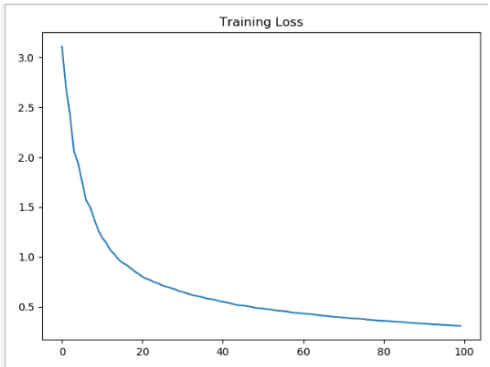
The network consists of 3 layers. The first being 783 features wide, the second being 20 features wide, and the third being 10 features wide. An example of this network is shown in the drawing above.

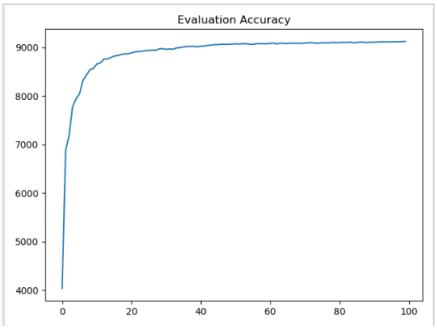
Hyperparameters of the network could include: the number of layers in the network, the number of features in each later, the number of epochs, etc. These parameters were not altered during this assignment.

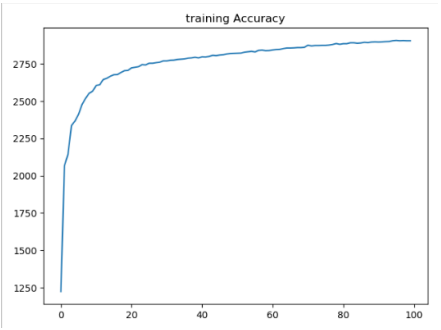
3)

The graphs below represent the Evaluation & training loss, and the Evaluation & Training accuracy of the network. This corresponds to 100 epochs in the model.









4) The test accuracy reported was 0.9054.