## Part 1

<https://towardsdatascience.com/exploring-activation-functions-for-neural-networks-73498da59b02>

using this link along with the lecture slides to understand the layers and activations, I utilized relu and softmax to increase my accuracy. Relu’s gradient is ALWAYS equal to 1, allowing me to pass the maximum amount of error through the network. Softmax assigns decimal probabilities to each class that MNIST contains, which adds up to 1.0. This constraint helps training converge faster than normal.

From the slides, we increase the number of epochs to decrease underfitting and tend closer toward optimal curve, while not increasing it too high to overfit

Got 99.8% accuracy with 0.01 loss on the training model

Got 98.2% accuracy with 0.07 loss on the test model

## Part 2

## Part 3