Question 2 Report

# Question 2.A

## Performance with ReLU:

Train set: Average loss: 0.0548, Accuracy: 58989/60000 (98%)

Test set: Average loss: 0.0529, Accuracy: 9840/10000 (98%)

## Performance with Sigmoid:

Train set: Average loss: 2.1522, Accuracy: 16994/60000 (28%)

Test set: Average loss: 2.1483, Accuracy: 2814/10000 (28%)

## Observations:

The non-linearity introduced by ReLU is important in finding better local minima. Sigmoid activation performs bad at least in hidden layers for deep neural network.

# Question 2.B

## Drop out

* Probability = 0.25

Train set: Average loss: 0.0365, Accuracy: 59323/60000 (99%)

Test set: Average loss: 0.0378, Accuracy: 9872/10000 (99%)

* Probability = 0.5

Train set: Average loss: 0.0548, Accuracy: 58989/60000 (98%)

Test set: Average loss: 0.0529, Accuracy: 9840/10000 (98%)

* Probability = 0.75

Train set: Average loss: 0.1071, Accuracy: 58093/60000 (97%)

Test set: Average loss: 0.0970, Accuracy: 9703/10000 (97%)

* Probability = 1

Train set: Average loss: 2.3272, Accuracy: 6742/60000 (11%)

Test set: Average loss: 2.3283, Accuracy: 1135/10000 (11%)

## Observations:

* The model becomes very robust as each neuron learns independently.
* Usual default is 0.5 but in this case at p=0.25, the model gives best performance.
* At p=0.75 very less parameters are initialized in every iterations and the model underfits the dataset.
* At p=1, no parameters from dropout layers are initialized and hence the model highly underfits the dataset.

# Question 2.c

## With batch-norm and dropout (after 10 epochs):

Train set: Average loss: 0.0230, Accuracy: 59614/60000 (99.3567%)

Test set: Average loss: 0.0296, Accuracy: 9899/10000 (98.99%)

## With batch-norm only (after 10 epochs):

Train set: Average loss: 0.0134, Accuracy: 59824/60000 (99.7067%)

Test set: Average loss: 0.0299, Accuracy: 9907/10000 (99.07%)

## Observations:

In this case, using only batch norm performed slightly better on both train and test accuracy.

It was as expected that batch norm will do better than any combination of dropout.

# Question 2.d

## Xavier’s uniform weight initialization:

Train set: Average loss: 0.0203, Accuracy: 59714/60000 (99.5233%)

Test set: Average loss: 0.0337, Accuracy: 9889/10000 (98.8900%)

## Kaiming’s uniform weight initialization:

Train set: Average loss: 0.0304, Accuracy: 59509/60000 (99.1817%)

Test set: Average loss: 0.0423, Accuracy: 9862/10000 (98.6200%)

## Observations:

Xavier’s initialization on conv2d and fully connecter layer performs better than Kaiming’s initialization.

The important point is that better initialization results in faster reduction in error rate compared to default initialization.