Name - Rahul Sailesh Wadhwa

Student ID - 862309846

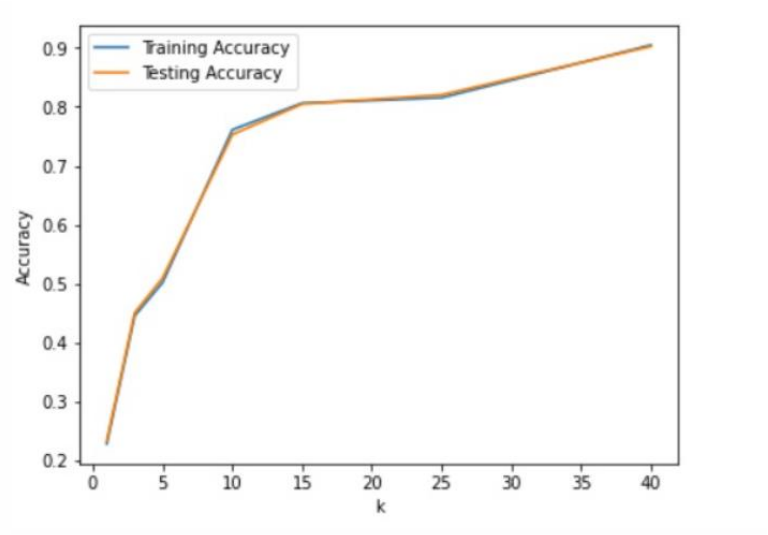
Problem Set 4 Report

1. **Setup your code so that you can run multiple MNIST models for varying choices of k and p automatically, Specifically, you need two for loops (one for k and one for p) and within the loop, you call TensorFlow/PyTorch.**

   Implemented in the notebook.

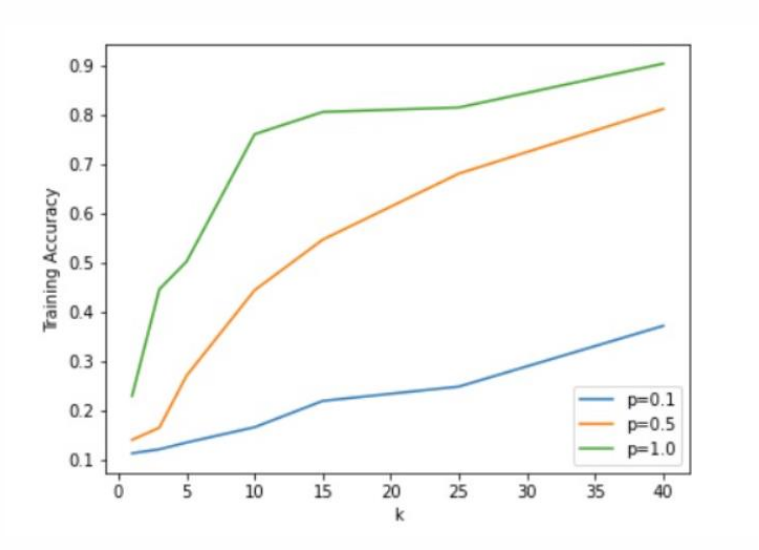
2. **Pick the width grid K = [1, 3, 5, 10, 15, 25, 40] and dropout grid P = [0.1, 0.5, 1.0]. Run MNIST models over these grids with Adam optimizer for 80 epochs. Store the test/train accuracy and loss.**
   - Fix p = 1.0 which is the case of "no dropout regularization". Plot the test and training accuracy as a function of k. As k increases, does the performance improve? At what k, training accuracy becomes 100%?

     As we can see from the graph that as the value of k increases, the performance of the model improves slightly and at k=40 a training and testing accuracy of around 90% is achieved.



   - Plot the training accuracy as a function of k and for different p ∈ P on the same plot. What is the role of p on training accuracy? When p is smaller, is it easier to optimize or more difficult? For each choice of p, determine at what choice of k, training accuracy becomes 100%.

     We can observe that dropout regularization is a cheap way of regularizing the neural network and helps in preventing the overfitting of the model to the data.
     We also observe that a higher dropout rate causes the model to achieve a better accuracy, and it becomes easier to optimize the data
     It can be observed from the image shown below for the different network width (k) values that the training accuracy obtained is higher at smaller values of p.



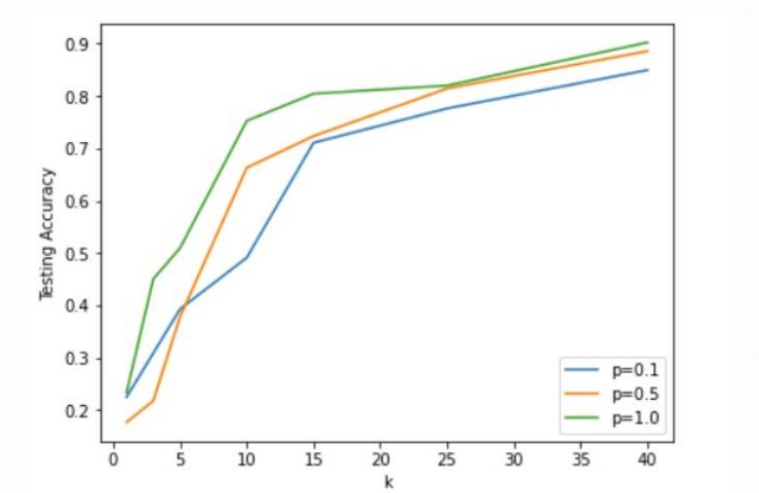
When p=0.1,
It can be seen that a network width k=40 or more, the training accuracy reaches around 35%.

When p=0.5.
It can be seen that a network width k=40, the training accuracy reaches around 80%.

When p=1.0,
It can be seen that a network width k=40 or more, the training accuracy reaches around 90%.

- Plot the test accuracy as a function of k and for different p ∈ P on the same plot. Does dropout help with the test accuracy? For which (k, p) configuration do you achieve the best test accuracy?

  It can be observed that dropout helps to increase the testing accuracy in cases where the network width is higher and it helps to prevent the overfitting of the model to the training data.
  At k=40 and p=1.0, the highest testing accuracy of around 90% was obtained.

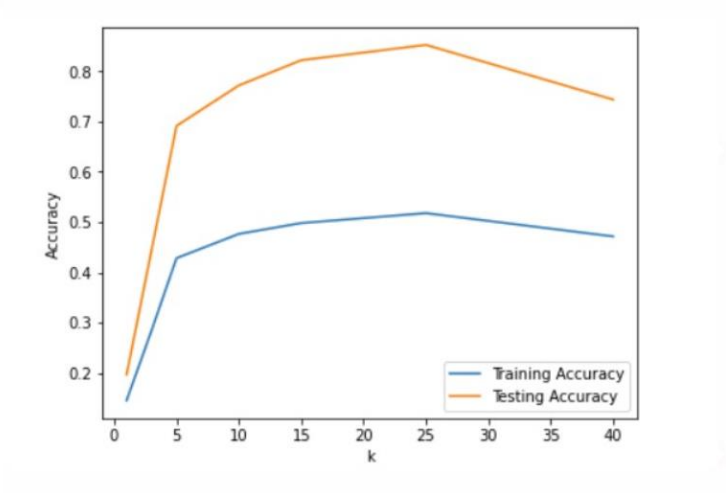


3. **We will spice up the problem by adding some noise to labels. Pick 40% of the training examples at random. Assign their labels at random to another value from 0 to 9. For instance, if the original**

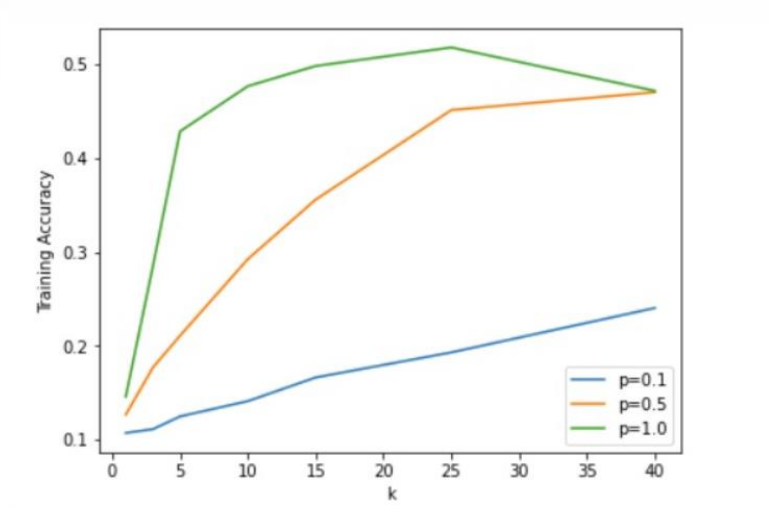
image is 0 and its label is 0, then you will assign its label to a number from 1 to 9 at random. Thus 60% of the training examples remain correct and 40% will have incorrect labels. Repeat the previous step with this noisy dataset.

- Fix p = 1.0 which is the case of "no dropout regularization". Plot the test and training accuracy as a function of k. As k increases, does the performance improve? At what k, training accuracy becomes 100%?

  As it can be seen from the graph shown below that for a network width (k) of 25, and fixing the dropout rate as 1.0 we observe that although the model has a relatively low training accuracy, it still achieves a decent testing accuracy (85%) as making use of dropout is similar to stringing together a group of weak classifiers which when combined together can be used to achieve a relatively good training accuracy.



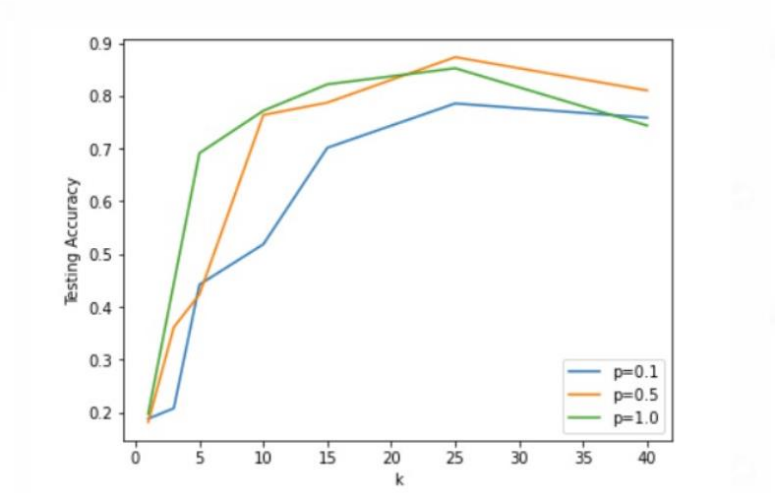
- Plot the training accuracy as a function of k and for different p ∈ P on the same plot. What is the role of p on training accuracy? When p is smaller, is it easier to optimize or more difficult? For each choice of p, determine at what choice of k, training accuracy becomes 100%.



  It can be observed from the image shown above for the different network width (k) values that the highest training accuracy was obtained for p=1.0 and the network width k=25.

For all the given values of p (0.1, 0.5 and 1.0) the highest accuracy was achieved when the model had a higher value for the network width (k=25).

- Plot the test accuracy as a function of k and for different p ∈ P on the same plot. Does dropout help with the test accuracy? For which (k, p) configuration do you achieve the best test accuracy?



It can be observed from the image shown above for the different network width (k) values that the highest testing accuracy was obtained for p=0.5 and the network width k=25.

This could be because the higher dropout rate for a model with noisy data will help to prevent the model from overfitting the noisy instances of the data.

4. Comment on the differences between Step 2 and Step 3. How does noise change things? For which setup dropout is more useful?

We can observe from the experiment of adding noise to the data that a higher dropout rate (of around 0.5) is preferred in the case of noisy data as it can help us to prevent the model from overfitting these noisy cases and make the model more versatile. Dropout is a form of regularization that is very cheap and effective when used to the right extent. Using dropout for regularization is similar to the idea of stringing together a group of weak classifiers which when combined together can be used to achieve a relatively good accuracy.