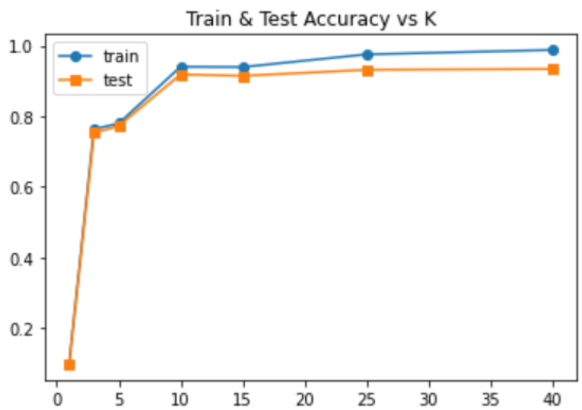
Overparameterization and Dropout

- Bhagyesh Gaikwad.

- Note :
  - Network width denoted by k and dropout rate by p.
  - Results are reported for running the model for 80 epochs with batch size 8

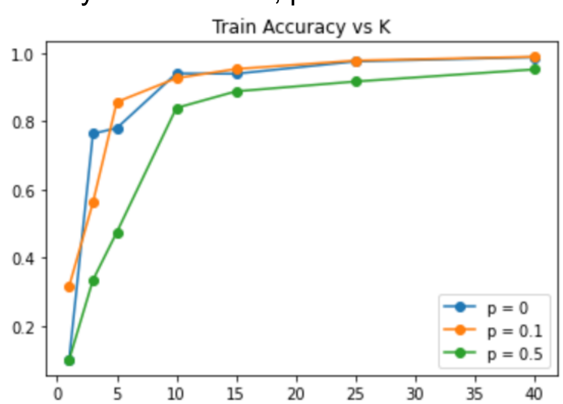
Foe width grid K = [1; 3; 5; 10; 15; 25; 40] and dropout grid P = [0:1; 0:5; 1:0].:

- For p=0 i.e no dropout regularization :

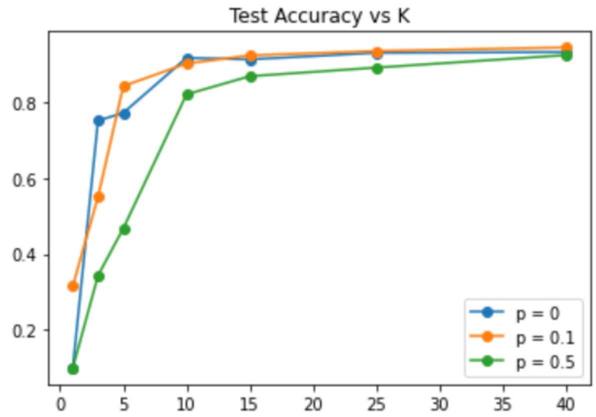


  - As k increases performance improves but the margin is very low.
  - At k = 40 the train accuracy reaches 99.85% (~100)

- Train Accuracy for different k, p values:



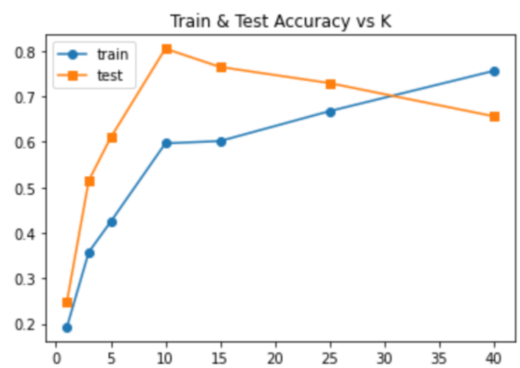
  -
  - With more dropout the same training accuracy is achieved after more epochs, as seen in the graph above p=0.5 nears the training accuracy but would have to be trained for more epochs. It is easier to optimize for smaller dropouts.
  - P = 0 training accuracy reaches 99.76% for k=40
  - P = 0.1 training accuracy reaches 99.31% for k=40
  - P = 0.5 training accuracy reaches 98.14% for k=40

- Test Accuracy for different k, p values:
    - 

    - Dropout with p=0.1 seems to help with the test accuracy as seen in the above graph.
    - The best test accuracy of 89.78% is achieved with k=40 and p=0.1.

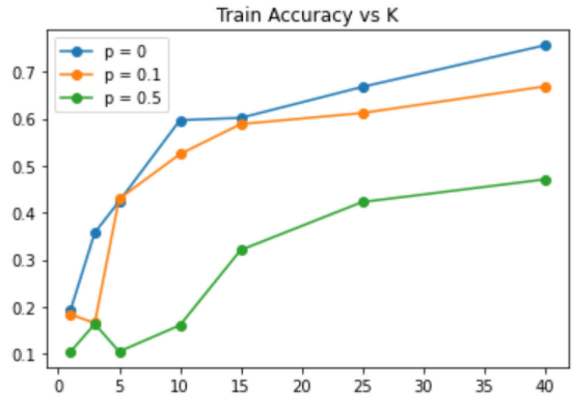
Dropout: Taking 40% of the training examples at random. Assign their labels at random to another value from 0 to 9.

- Part 1:
    - For p=0 i.e no dropout regularization :

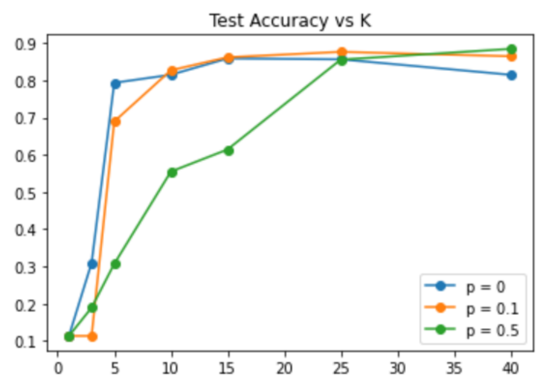


    - As k increases train accuracy improves but test accuracy decrease after reaching a peak.
    - Train accuracy doesn’t reach 100 for batch 8 an 80 epochs

- Part 2:

- 

  - We can see above that the least dropout has achieved more training accuracy i.e at p=0. It is easier to optimize for smaller dropouts.
  - None of the (k, p) combinations give 100% accuracy.

- Part 3:
  - 

  - Dropout with p=0.5 seems to help with the test accuracy as seen in the above graph.
  - The best test accuracy of 89.65% is achieved with k=40 and p=0.5.

Differences between initial results and Dropout:

- We see a drop in training accuracy from step 2 to step 3, this is an obvious result since the data has more noise in step 3 which makes it perform poorly w.r.t step 2.
- We observe almost the similar accuracy at peak for test data, but it should be noted that it is achieved only for one of the (k,p) configurations in step 3. Whereas, in step 2 the test accuracy is better for all the k's from 10 to 40.
  - The interesting part here to note is that even with the noisy train data, we can see good test accuracy with the given model. We can thus find the best (k,p) configuration even if we have some noise in the data in the real world.
- The test accuracy crosses the others and gains maximum for p=0.5 for step 3 while setup 2 has max for 0.1, it seems that setup 3 benefits more from the dropout than setup2.