Assignment-3
ELL409

Submitted By:

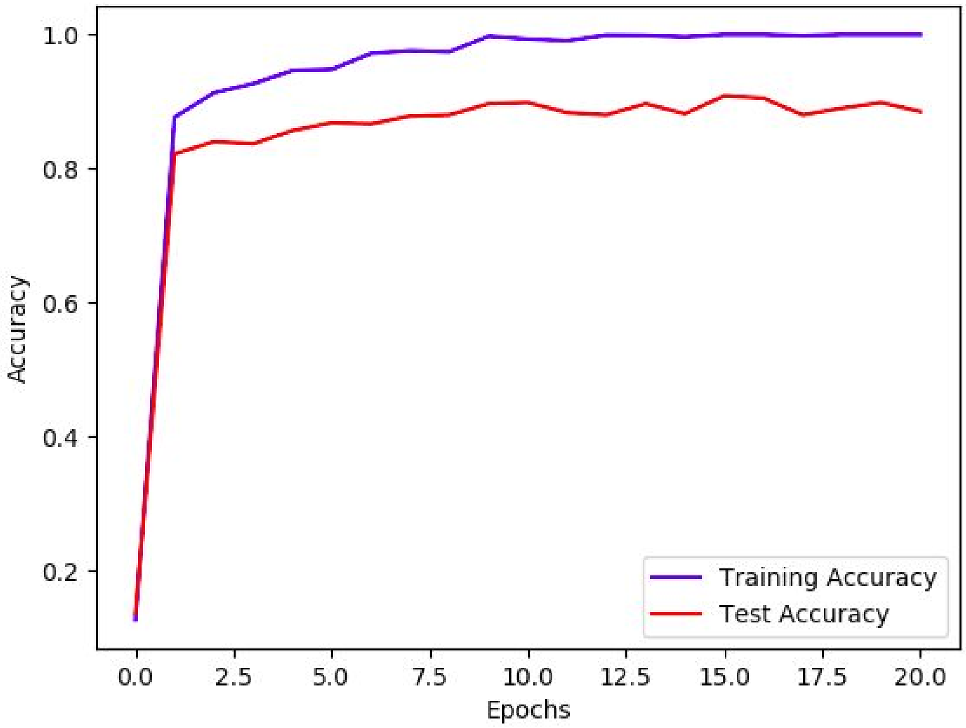
Ritik Agrawal (2017EE10482)

Standard Library used- Keras

Part-1

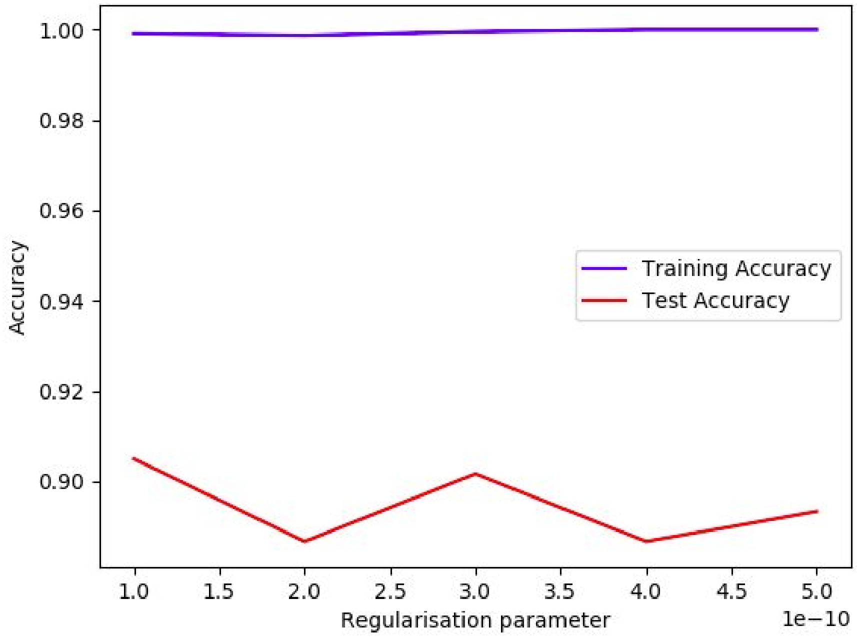
Observations and Plots-

A) Early Stopping-



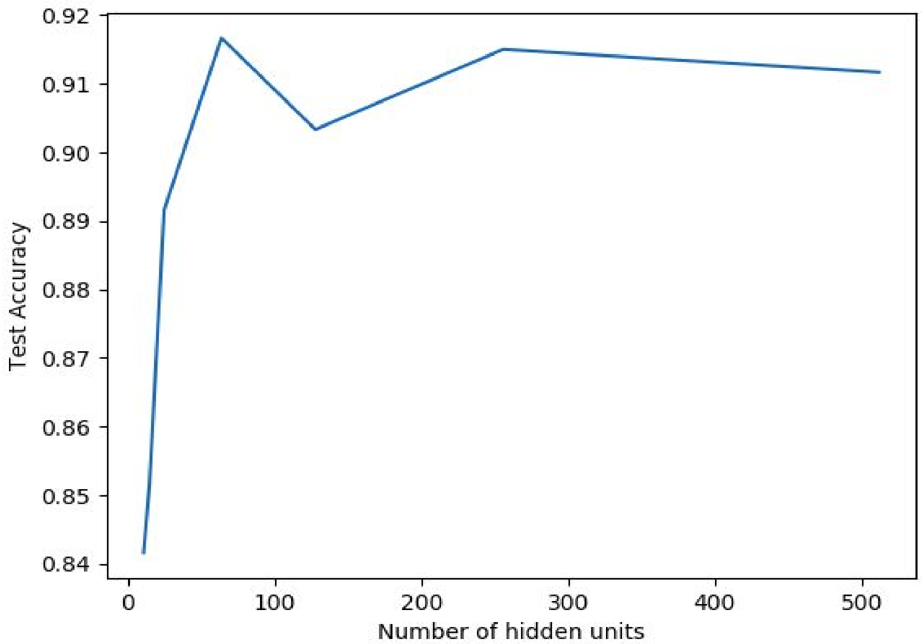
Number of epochs= 16

B) L2- Regularisation-



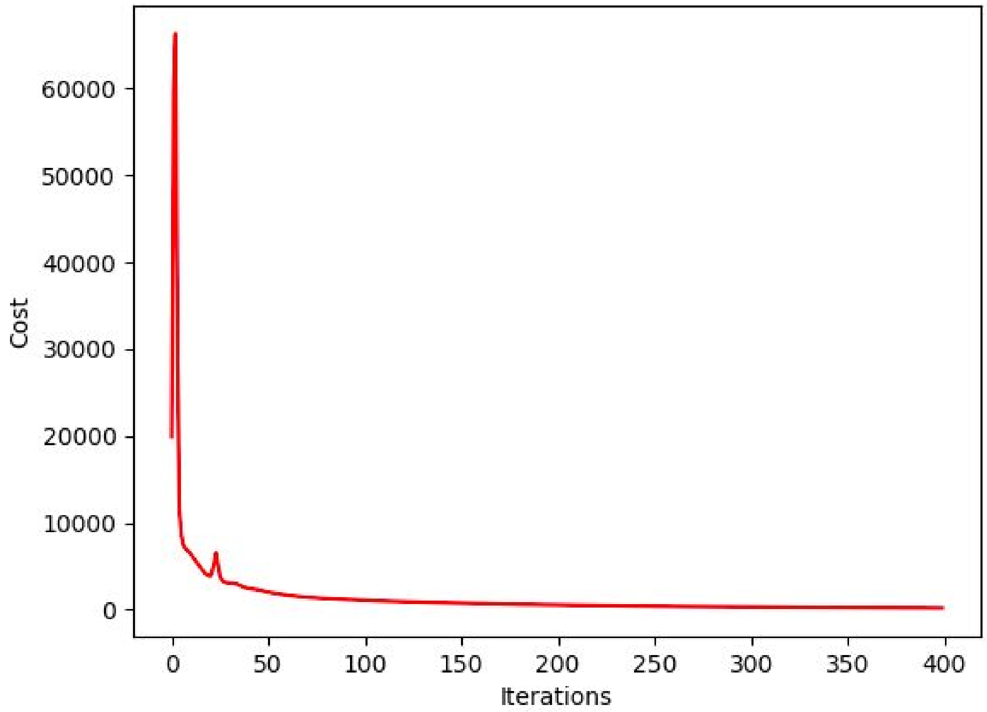
<u>Regularisation constant= 3e-10</u>

C) Number of hidden units-



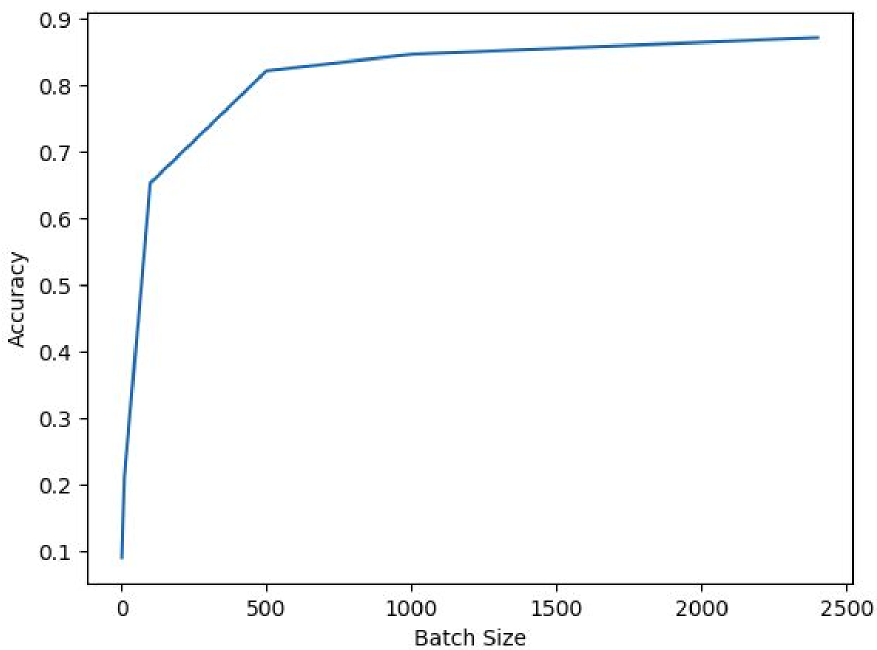
<u>Number of hidden units= 64</u>

D) Cost vs Iterations-

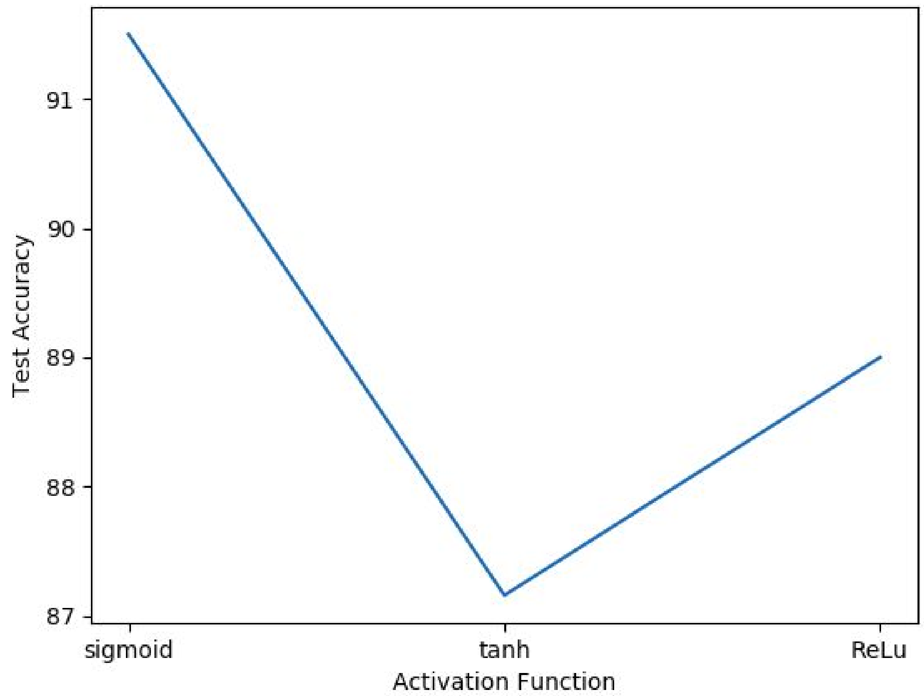


The convergence criterion is defined when the cost reaches below 0.0001

E) Batch Size-

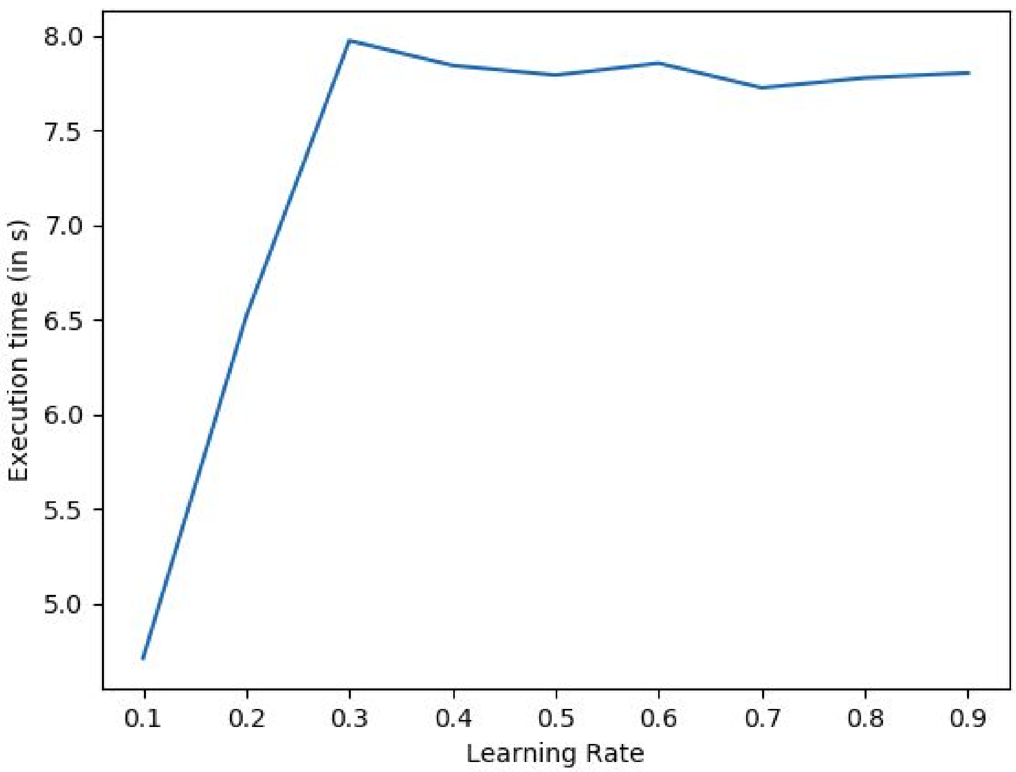


F) Activation Function-

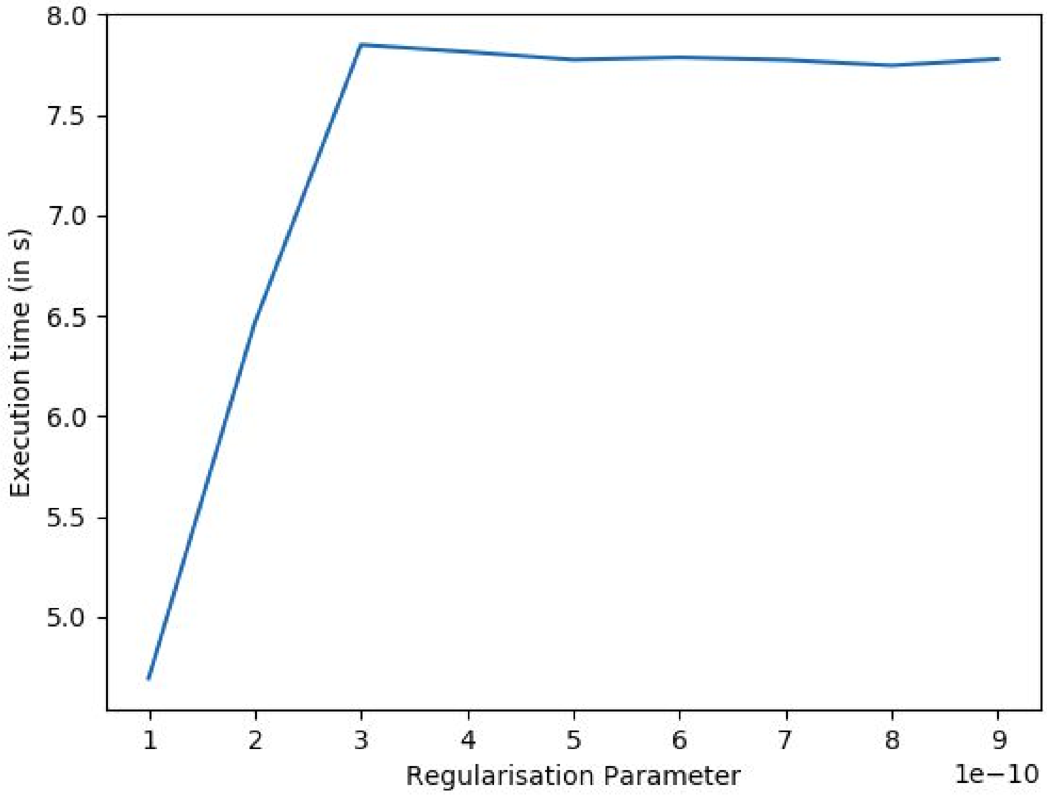


G) Time of Execution-

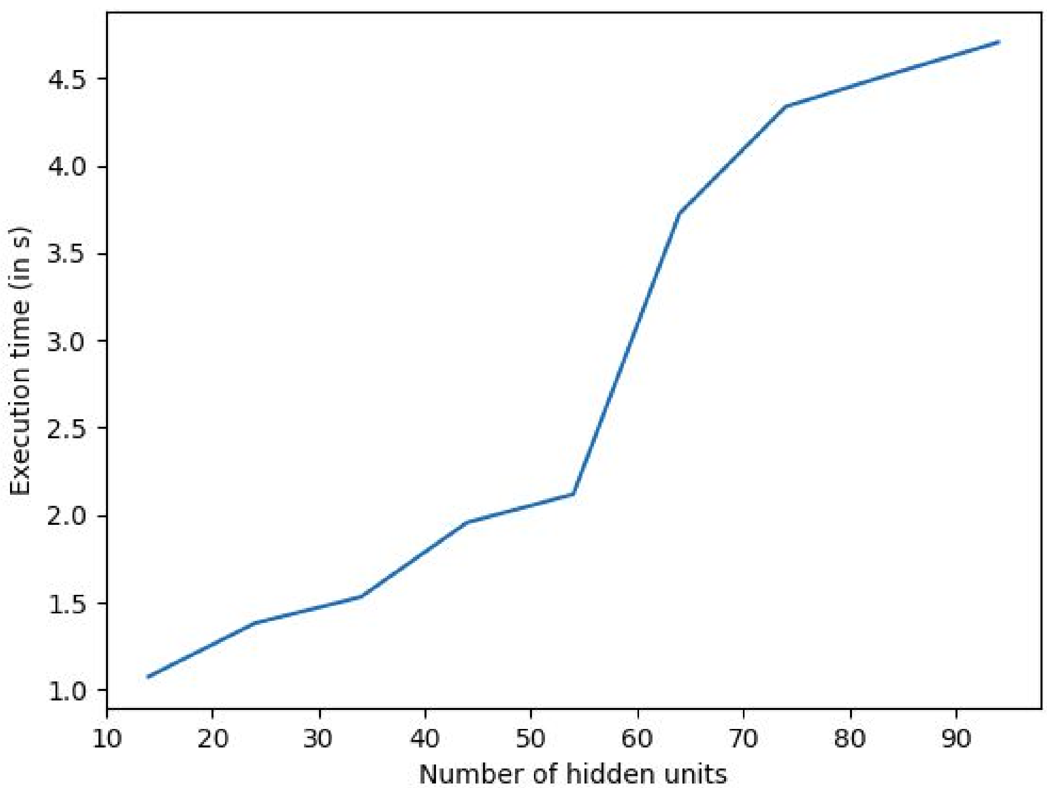
i)



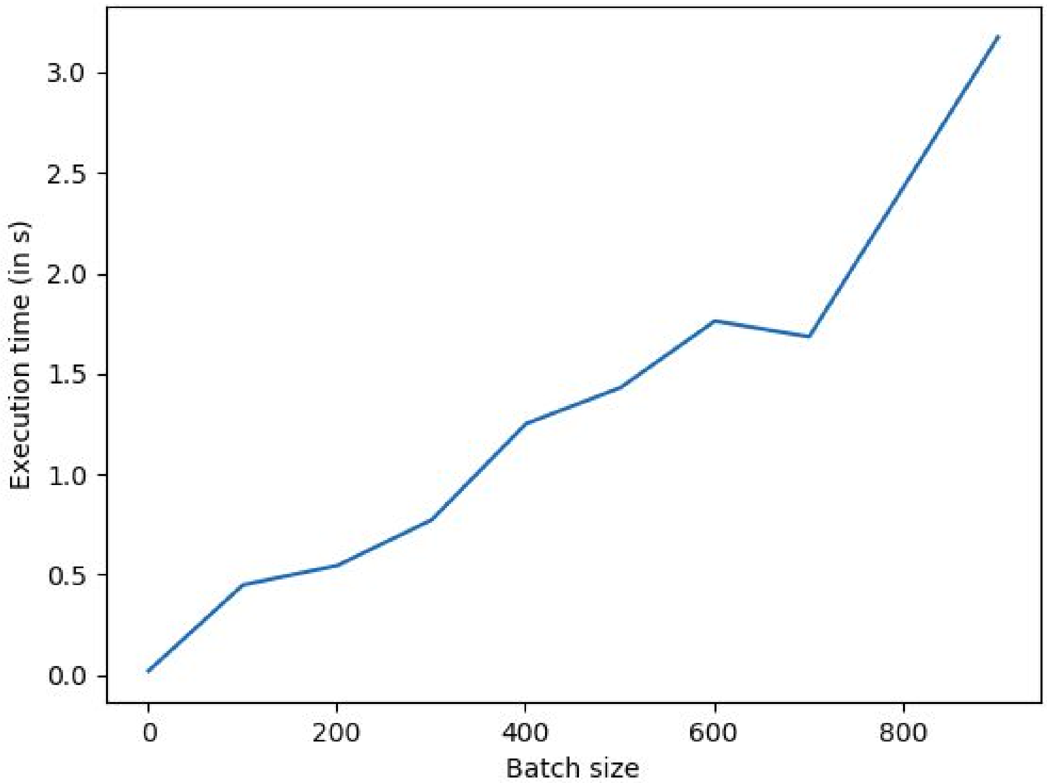
ii)


Execution time (in s)
Regularisation Parameter
1e−10
8.0
7.5
7.0
6.5
6.0
5.5
5.0
1
2
3
4
5
6
7
8
9

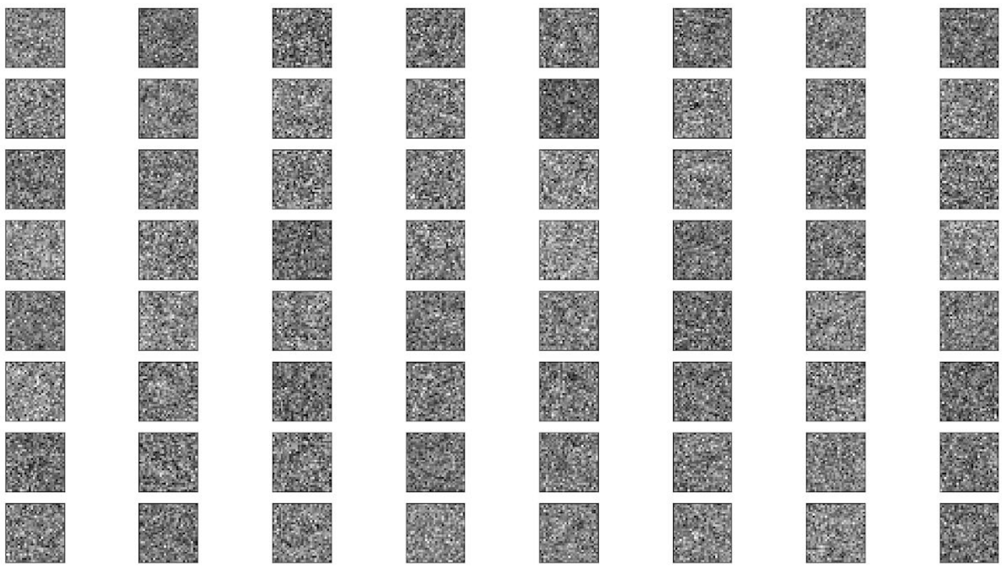
iii)


Execution time (in s)
Number of hidden units
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
10
20
30
40
50
60
70
80
90

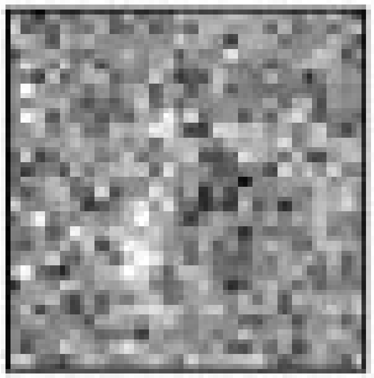
iv)



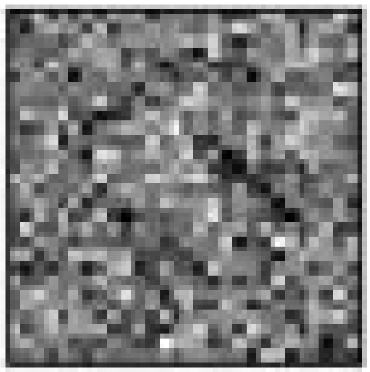
H) Hidden Layer Visualization-



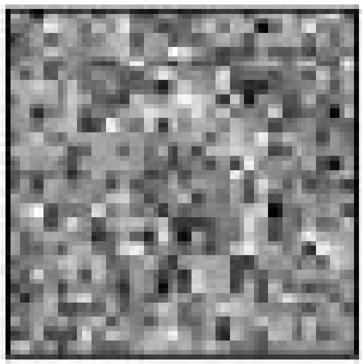
a) Neuron visualizing 8-



b) Neuron visualizing- 3



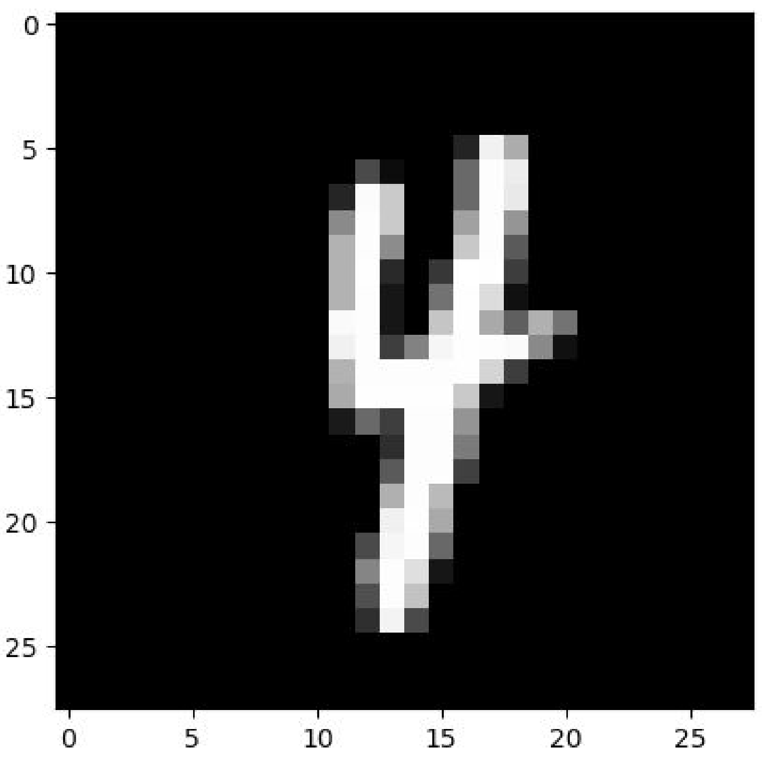
c) Neuron visualizing- 0



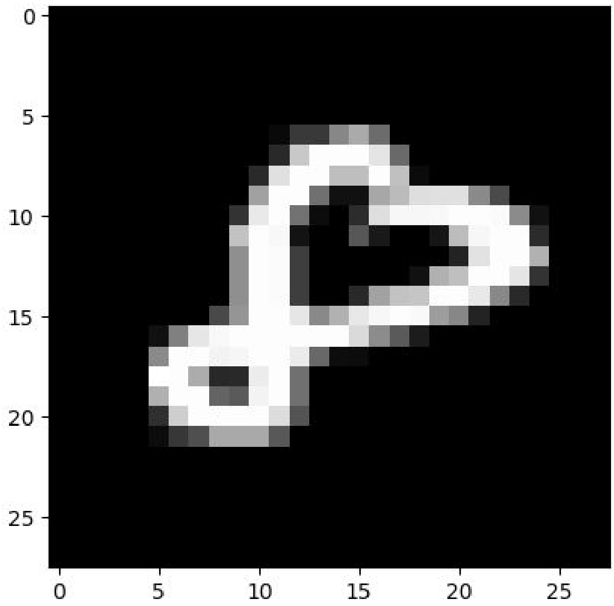
I) Confusion Matrix-

```
[[56  0  0  0  0  3  0  1  0  0]
 [ 0 55  2  0  0  1  1  0  0  1]
 [ 3  0 47  1  3  1  2  3  0  0]
 [ 0  2  2 50  0  6  0  0  0  0]
 [ 1  0  1  2 51  0  0  0  0  5]
 [ 4  0  1  7  1 43  1  1  2  0]
 [ 1  1  0  0  0  1 56  0  1  0]
 [ 1  1  0  2  1  0  0 51  0  4]
 [ 2  2  3  2  0  2  2  0 46  1]
 [ 0  0  0  2  4  0  0  2  0 52]]
```

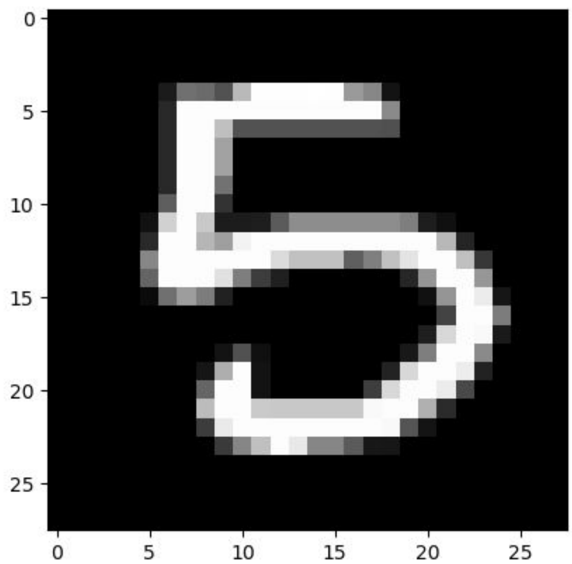
a) True label- 4, Predicted Label- 9



b) True label- 8, Predicted Label- 0



c) True label-5, Predicted label- 6



Results-

i) Using 1 hidden layer-

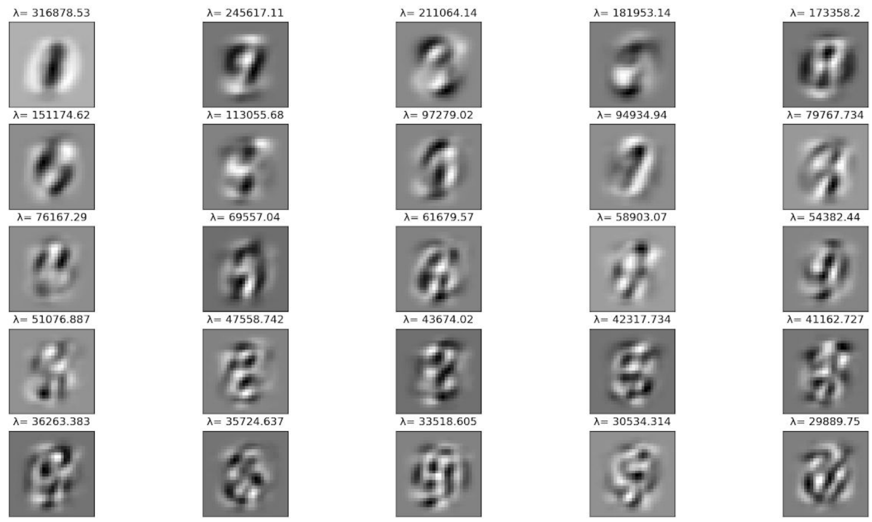
A) Accuracy using my implementation- ~88-90%
B) Learning Rate- 0.5 for sgd, 0.005 for full batch
C) Number of hidden units- 64
D) Number of epochs- 16
E) Regularisation constant- 3e-10
F) Accuracy using standard library- ~89-90%

ii) Using 2 hidden layers-

A) Accuracy using my implementation- ~90-91%
B) Learning Rate- 0.5 for both sgd, 0.005 for full batch
C) Number of hidden units- [40, 25]
D) Number of epochs- 400
E) Accuracy using standard library- ~88-90%

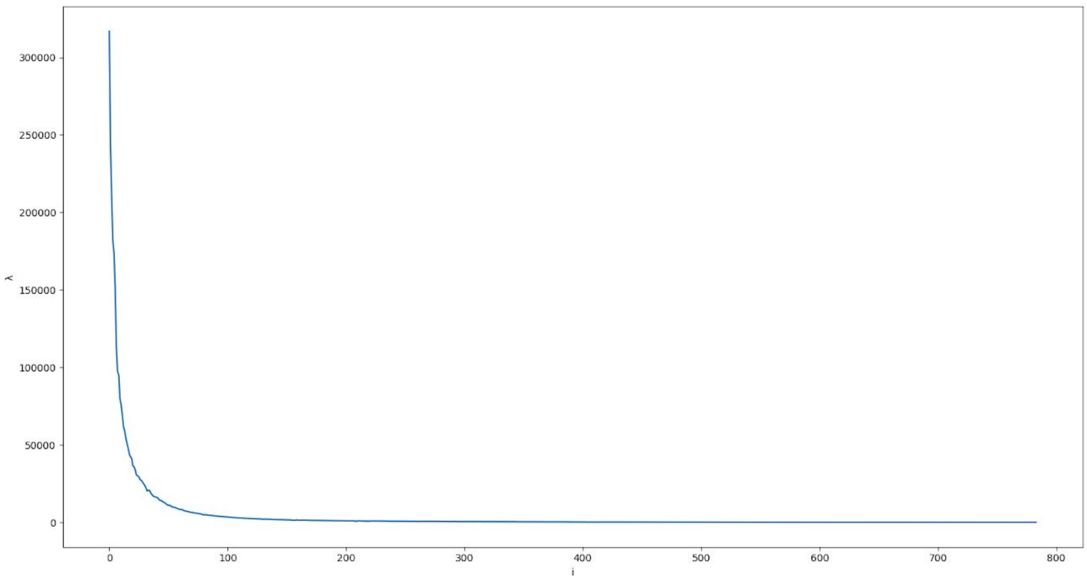
Part-2

A) PCA visualization-



Here, we can observe that the PCA eigenvectors are learning a better representation than in the case of standard neural nets. This shows that intrinsically the dimension of the input space is much lower than 784.

B) Eigenvalue Spectrum-



Results and Conclusions-

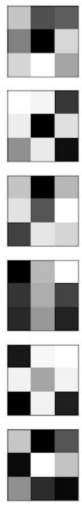
A) PCA with no hidden layer- Accuracy is 85.16%
B) PCA with 1 hidden layer- Accuracy is 88.16%

- The accuracy of the model by using PCA features alone with no hidden layer came out to be lower than the standard neural nets. This emphasises the fact that the discriminative information that distinguishes one class from another might be in the low variance components, so using PCA can lower the performance.
- Upon adding another hidden layer, the performance improves slightly, indicating the fact that the data is still non-linearly separable, in reduced dimensionality as well. It achieves similar accuracy as our standard neural nets.

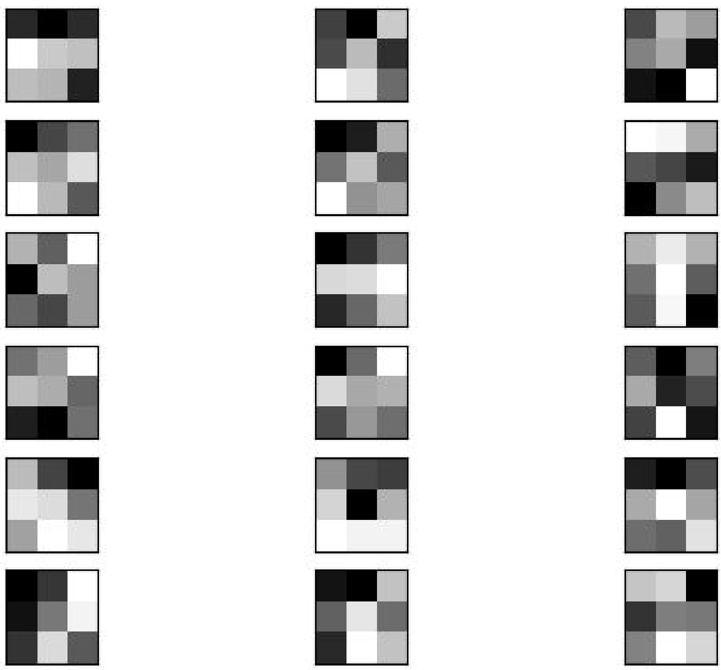
Part- 3

I. CNN-

A) Filter Visualization-



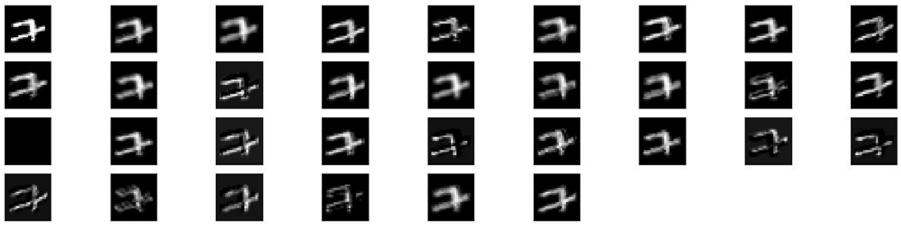
Filters of first convolutional layer



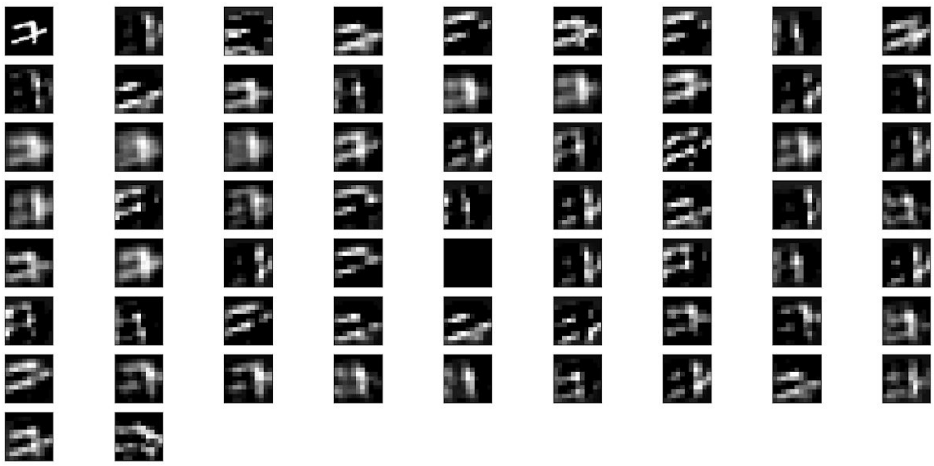
Filters of second convolutional layer

The dark squares indicate small or inhibitory weights and the light squares represent large or excitatory weights. Using this intuition, we can see that the filters detect a gradient on going from light region to dark region.When training an image, these weights change, and so when it is time to evaluate an image, these weights return high values if it thinks it is seeing a pattern it has seen before.

B) Feature Map Visualization-



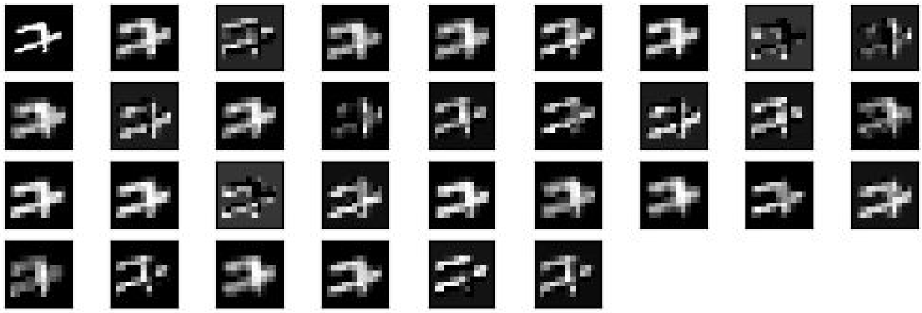
Output feature maps of first convolutional layer



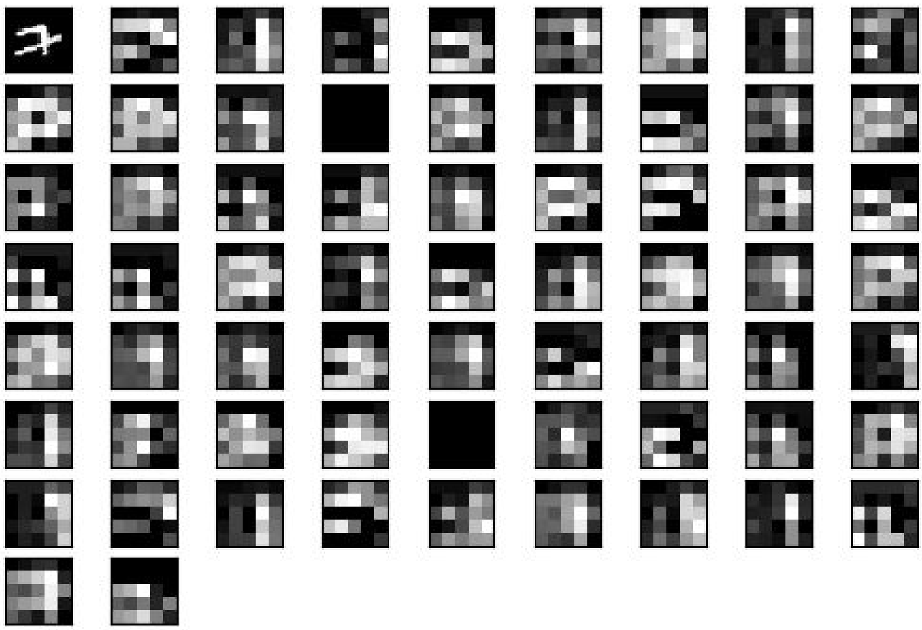
Output feature maps of second convolutional layer

Observations:

- We can observe that different versions of number 7 are highlighted, using the filters.Some tend to capture the lines, some focuses on the background portion, while some captures the boundary.
- We also notice that the feature maps nearer to the input capture fine details, but as we increase the convolutional layers, we are unable to interpret the feature maps.



Output feature maps of first max pooling layer

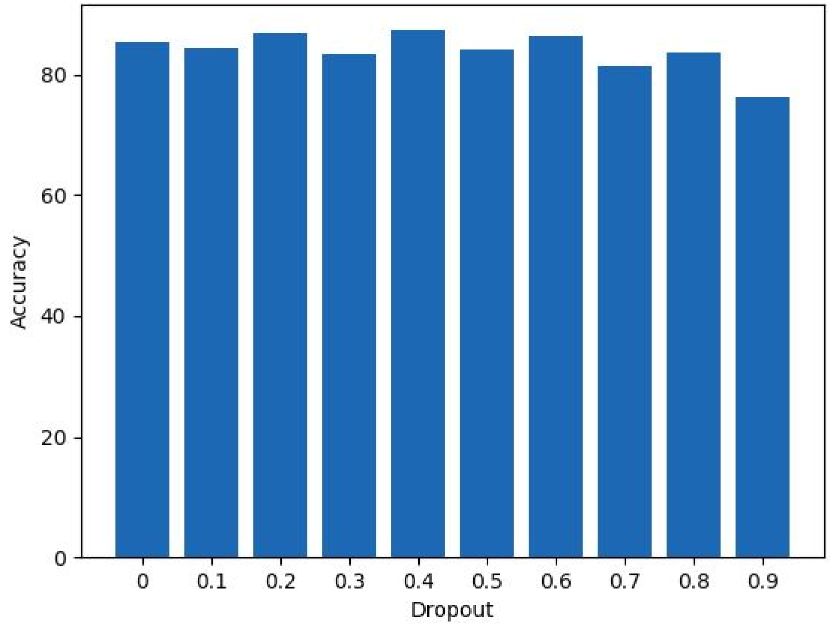


Output feature maps of second max pooling layer

Observations:

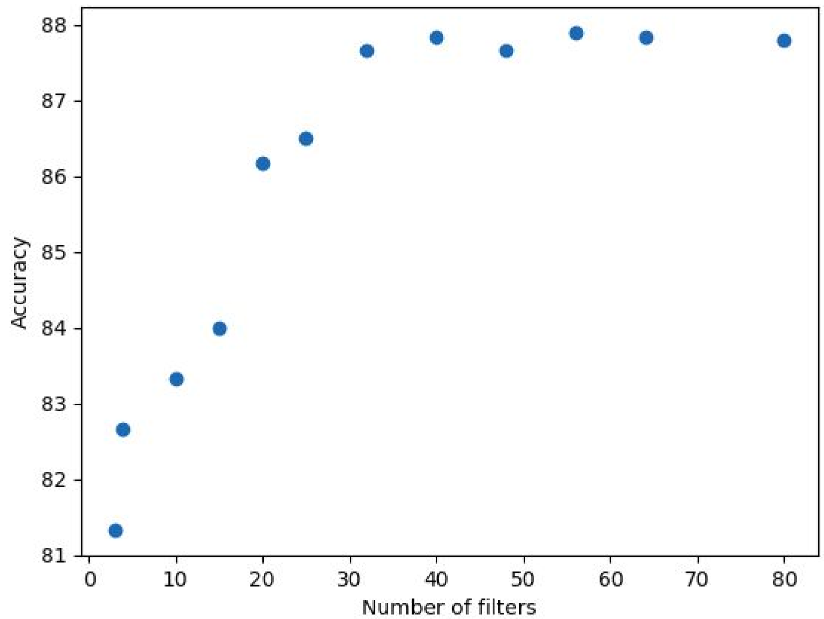
- A max-pooling layer keeps only the most active pixel values, and reduces the input dimension.
- As we increase the max pooling layers, they tend to activate the regions where the label is present, which helps in the classification process later.

C) Dropout-



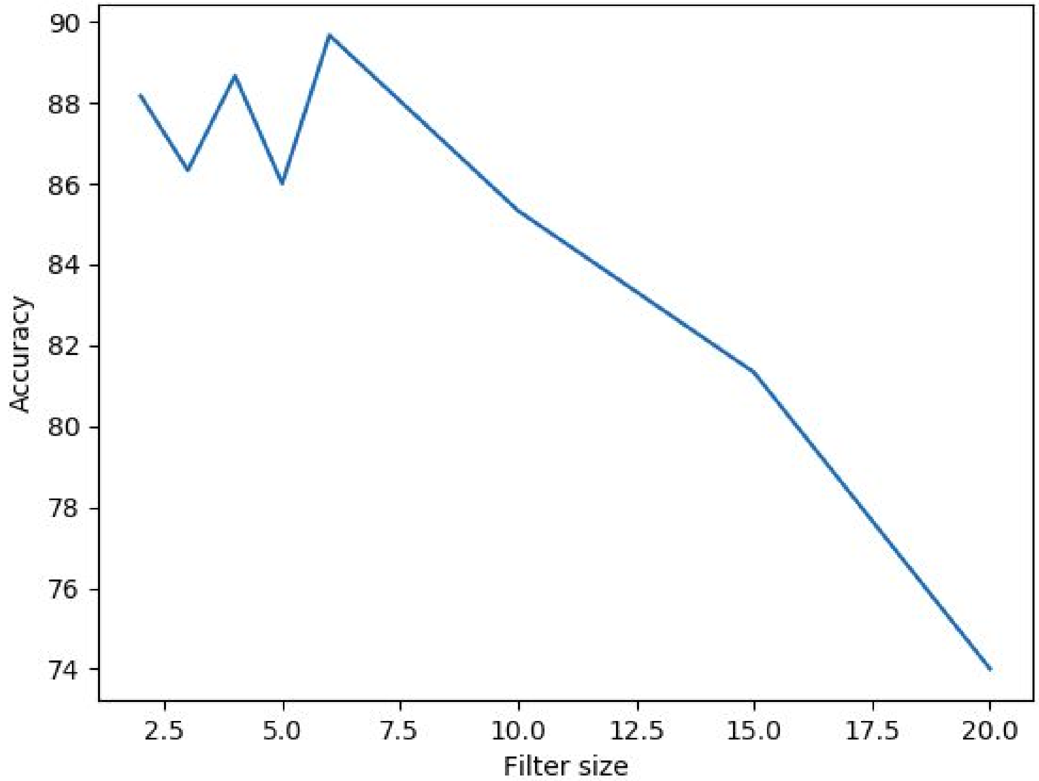
<u>Observation</u>- As we increase dropout, some neurons are dropped, forcing the model to learn more robust features, increasing the validation accuracy. But as we increased dropout beyond 0.4, the model is not able to learn properly, therefore leads to underfitting.

D) Number of filters-



After applying 32 filters, the accuracy doesn’t change much.

E) Filter size-



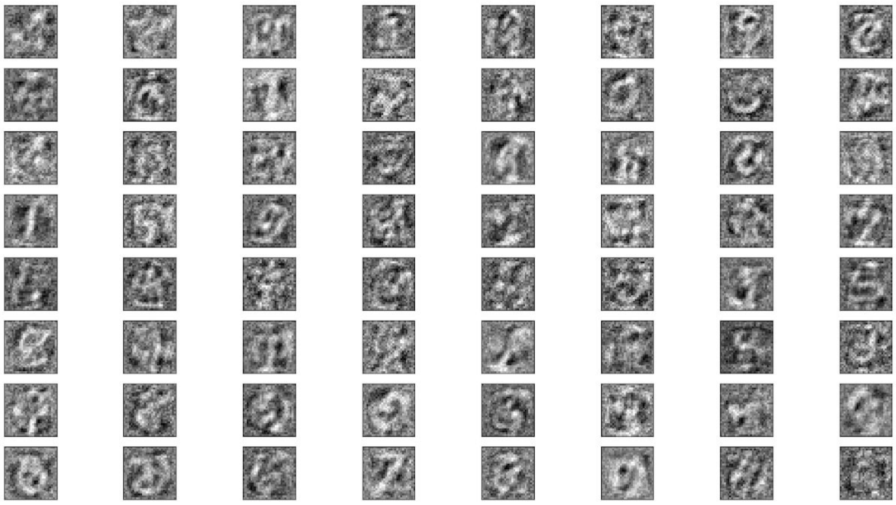
As we increase filter size, details of the image are captured, but as we increase the size too much, less information gets captured, hence accuracy decreases.

Results-

A) Number of layers- 7
B) Number of filters used- 32, 64
C) Number of convolutional layers- 2
D) Number of max pooling layers- 2
E) Activation function- ReLu
F) Accuracy- 96.5%

II) Sparse Autoencoders-

Compressed unit visualization



Sparse autoencoders have learnt a better representation than our standard neural nets. It has worked similarly as PCA, the difference is just that here it is using a nonlinear function to map the hidden unit, whereas in PCA we used linear mapping to get a compressed version of out input.

Results-

1) Accuracy- 87%
2) Number of hidden units- 64
3) L1-Regularizer- 10e-6
4) Activation function- ReLu