

Assignment#3 (Digit Classification)

Anand Kumar Verma (2016EE10459)

Part a | Implementation of basic FCNN

1. From Scratch (Without using any Neural Network Library):

Flexibility included –

- No. of hidden layer – multi (based on user choice)
- Unit in each layer – variable (based on user choice)
- Gradient descent parameters – learning rate, batch size, convergence criteria
- Activation function – Sigmoid, ReLu
- Mode of regularisation – L2 regularisation, early stopping
- Loss type – Cross entropy

2. Using Keras:

Implemented with same parameter as above implementation.

Part b | Standard backpropagation neural net:

Output Layer:

Activation function – Sigmoid

Loss Type – Cross Entropy

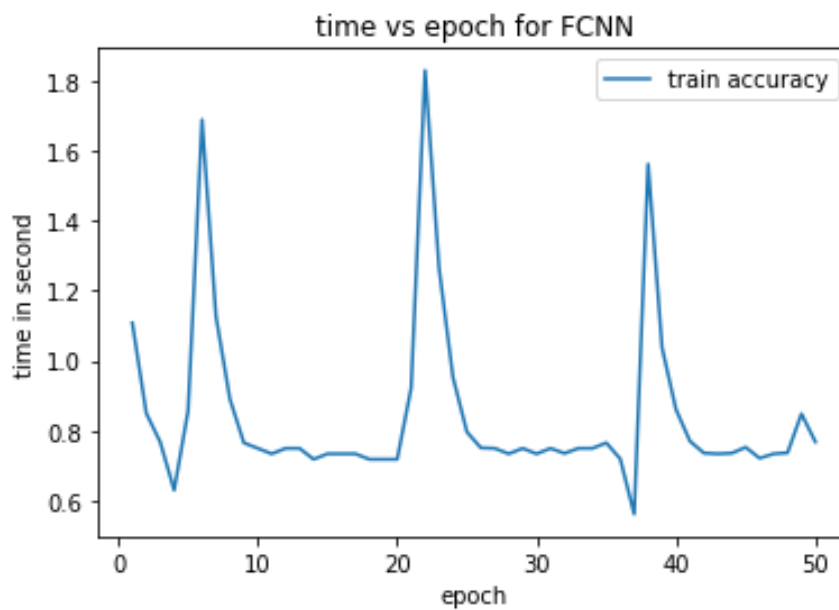
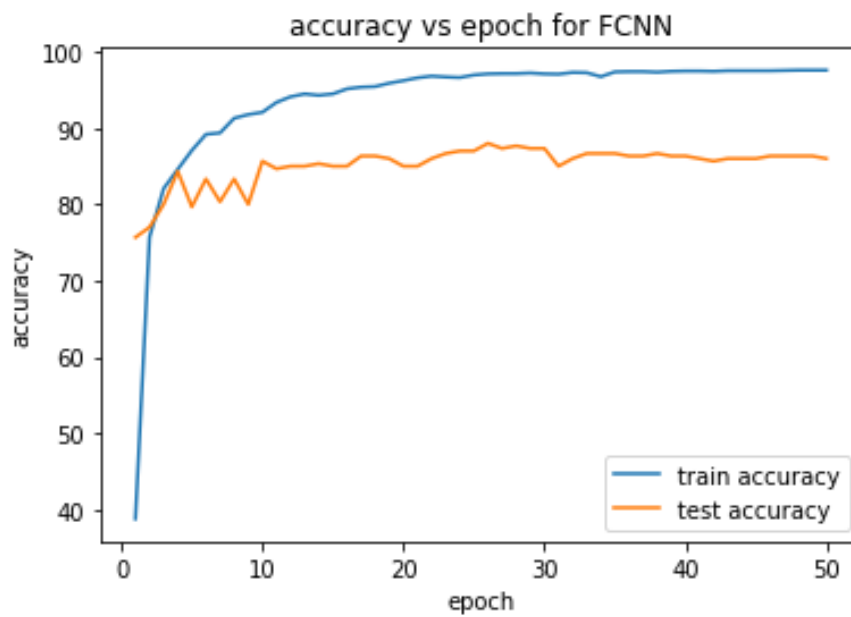
No. of output Node – 10

Model 1 (Input (784) + ReLu(36) + Sigmoid(10)):

- Learning rate = 0.1
- Batch size = 1
- Convergence criteria = 99% train accuracy
- Max Epoch size = 50
- lam = 0

Train accuracy – 97.591%

Test accuracy – 86%

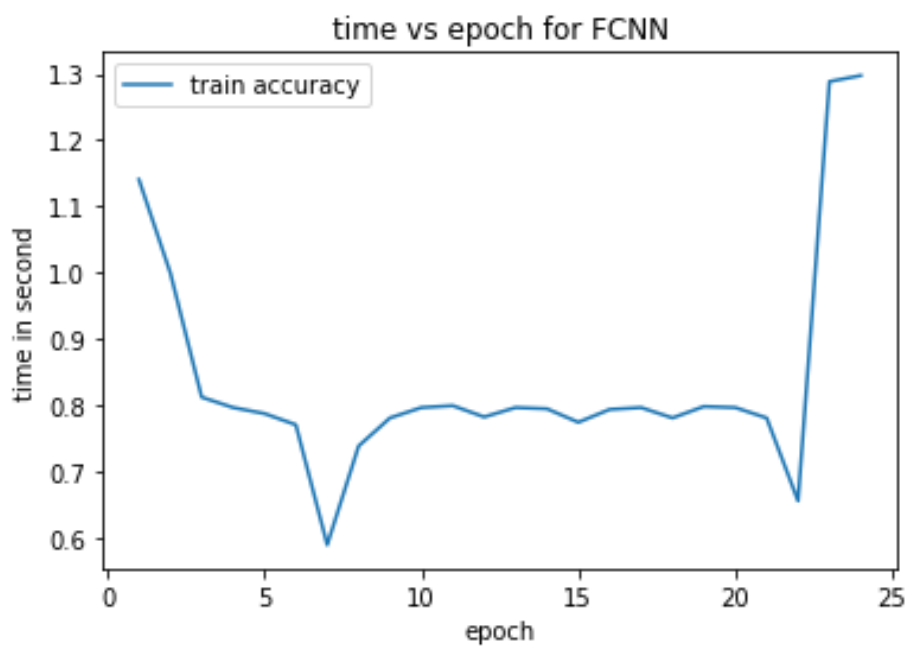
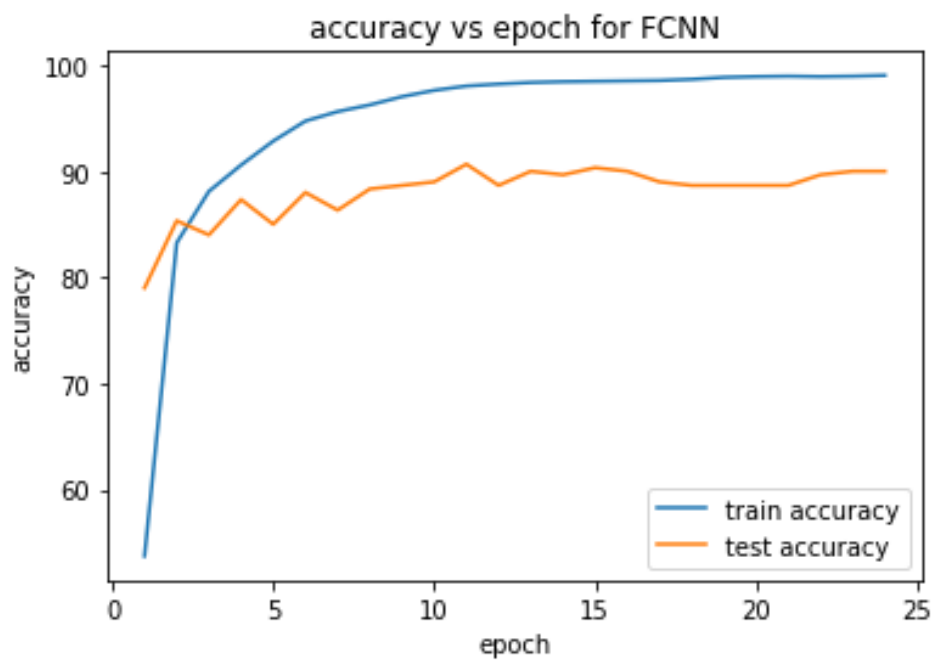


Model 2 (Input (784) + Sigmoid(36) + Sigmoid(10)):

- Learning rate = 1
- Batch size = 1
- Convergence criteria = 99% train accuracy
- Max Epoch size = 50
- lam = 0

Train accuracy – 99.03668025194517%

Test accuracy – 90%

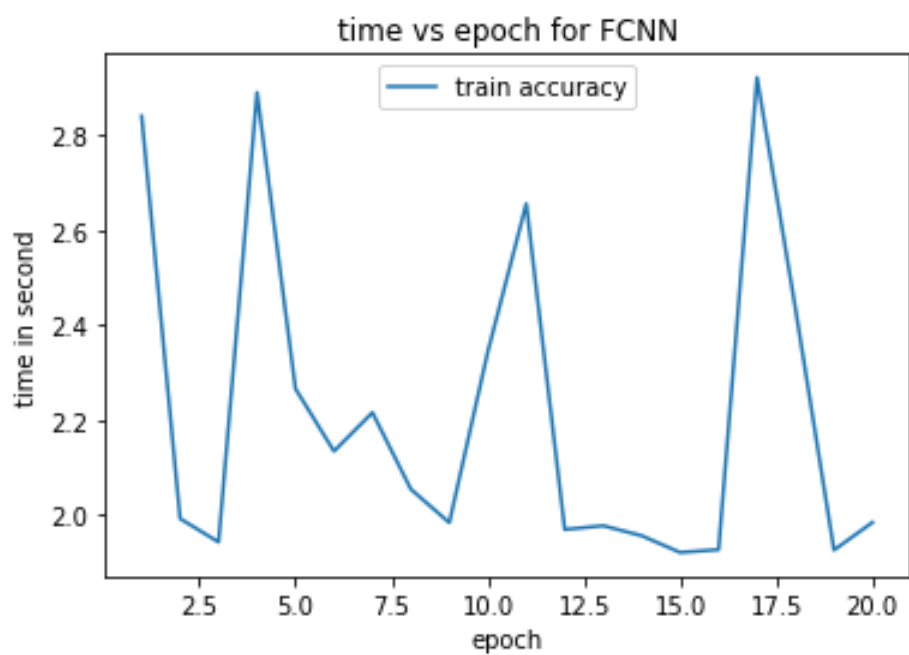
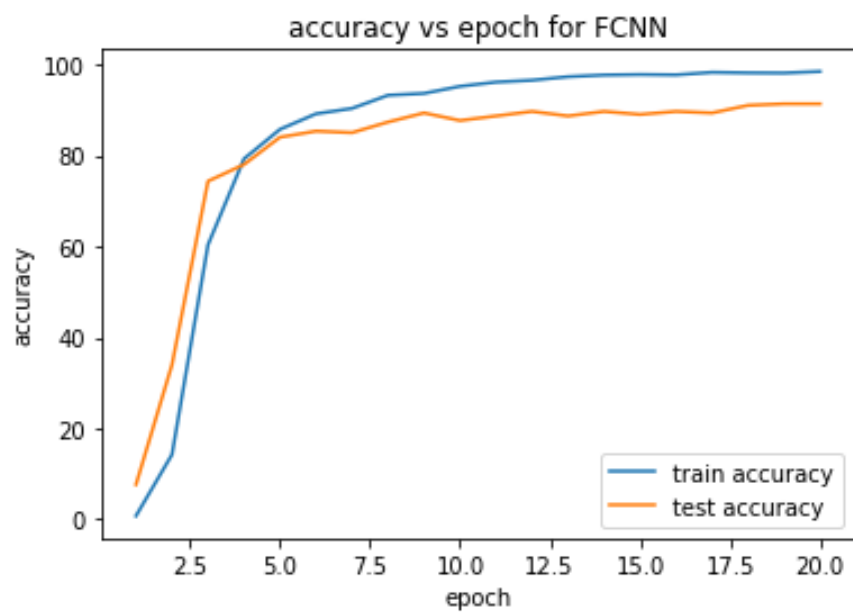


Model 3 (Input (784) + ReLu(100) + Relu(36) + Sigmoid(10)):

- Learning rate = 0.1
- Batch size = 1
- Convergence criteria = 99% train accuracy
- Max Epoch size = 20
- lam = 0

Train accuracy – 98.44386809929604%

Test accuracy – 91.33333333333333 %

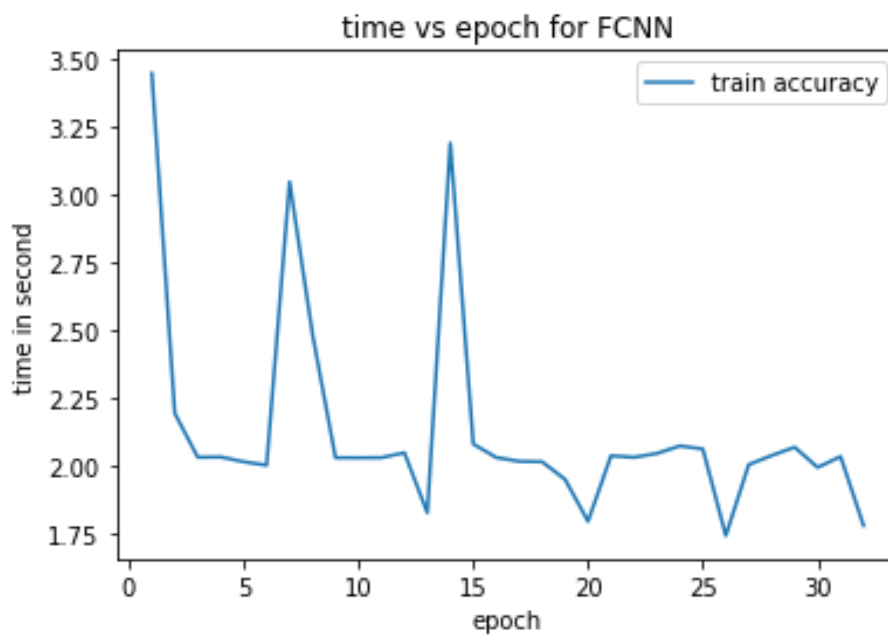
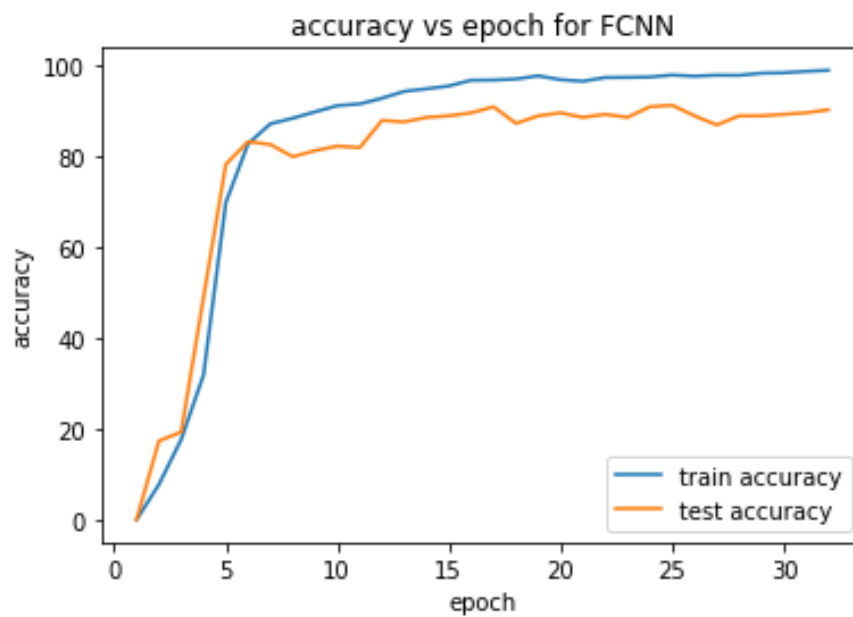


Model 4 (Input (784) + Sigmoid(100) + Sigmoid(36) + Sigmoid(10)):

- Learning rate = 1
- Batch size = 1
- Convergence criteria = 99% train accuracy
- Max Epoch size = 50
- lam = 0

Train accuracy – 99.03668025194517%

Test accuracy – 90.33333333333333%

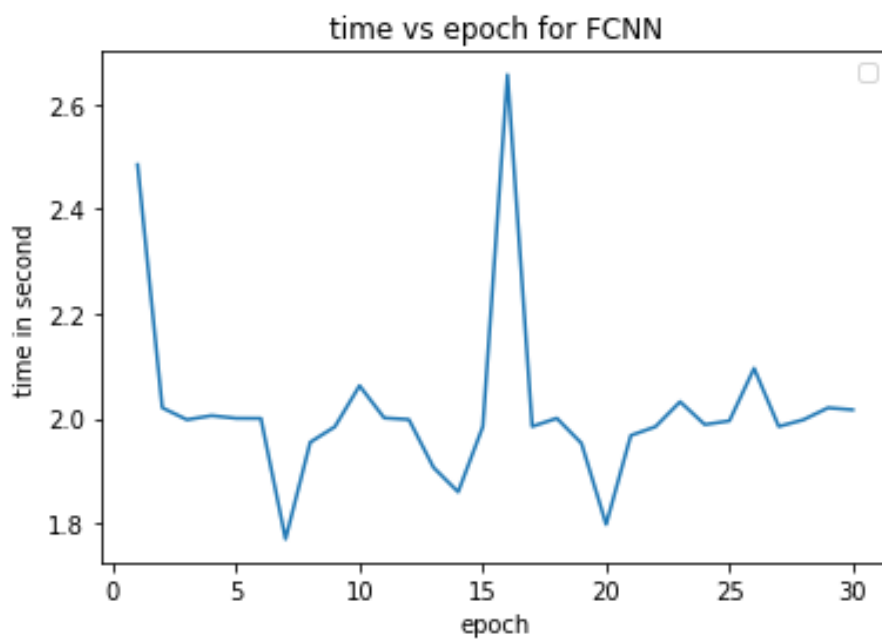
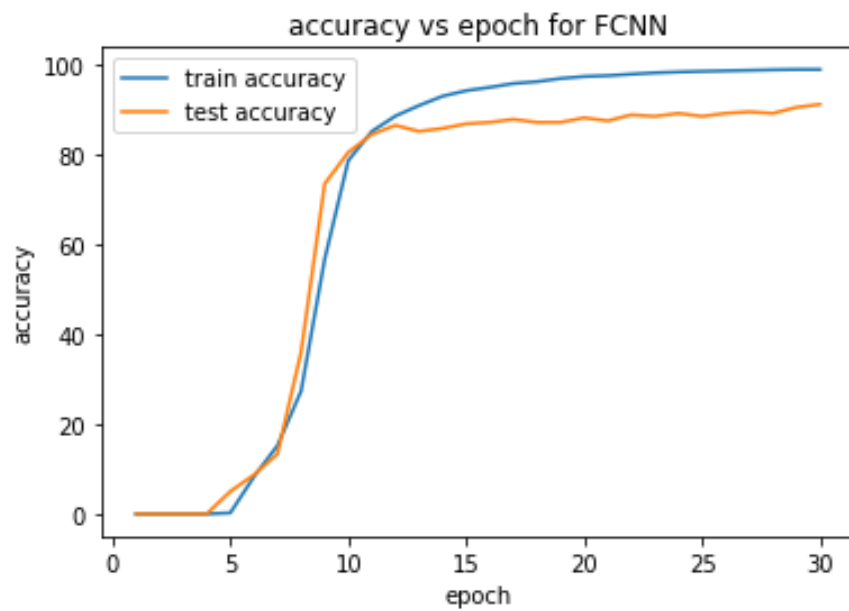


Model 5 (Input (784) + ReLu(100) + Sigmoid(36) + Sigmoid(10)):

- Learning rate = 0.1
- Batch size = 1
- Convergence criteria = 99% train accuracy
- Max Epoch size = 30
- lam = 0

Train accuracy – 98.7402741756206%

Test accuracy – 91.0%

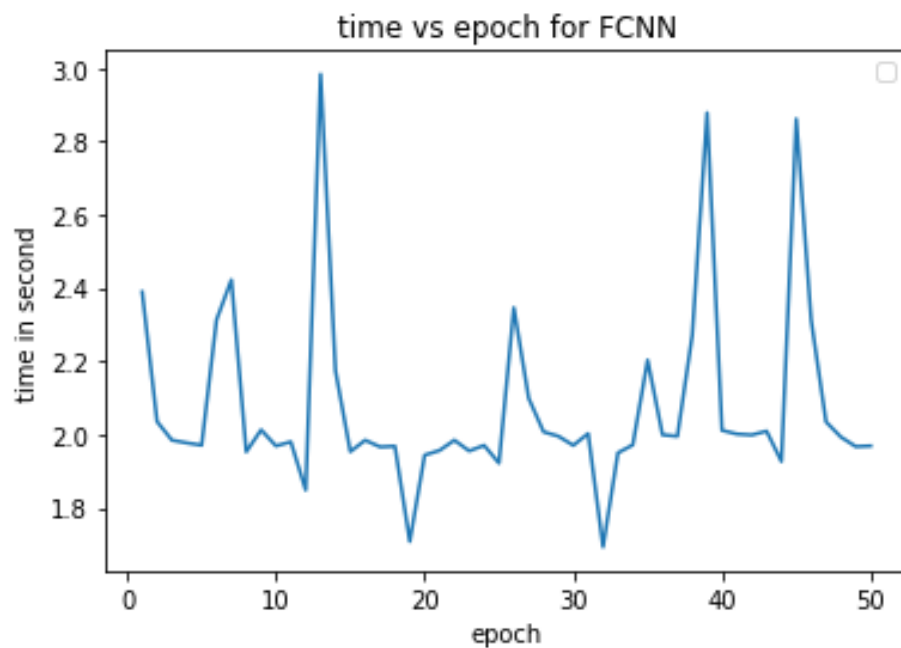
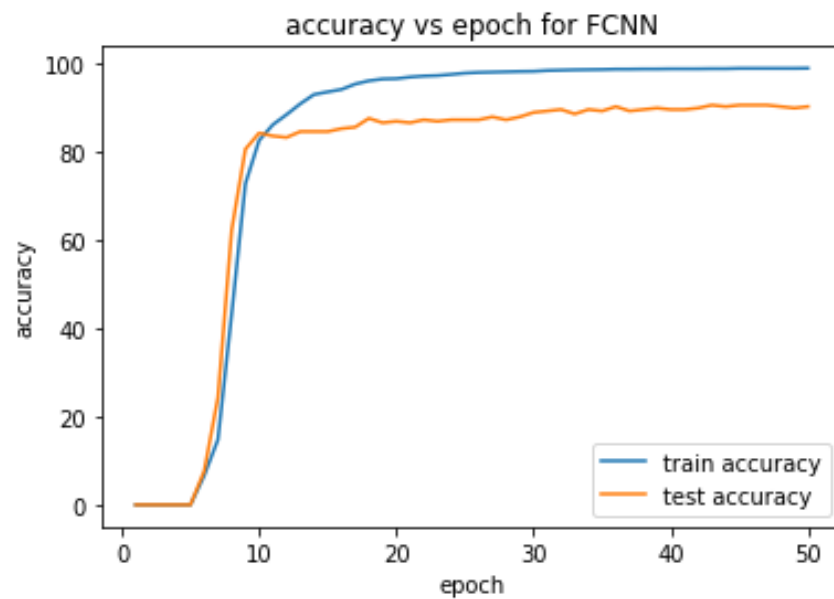


Model 6 (Input (784) + ReLu100) + Sigmoid(36) + Sigmoid(10)):

- Learning rate = 0.1
- Batch size = 1
- Convergence criteria = 99% train accuracy
- Max Epoch size = 50
- $\text{lam} = 10^{-5}$

Train accuracy – 99.03668025194517%

Test accuracy – 90.33333333333333%



Summary :

- Inclusion of hidden layer leads to more train and test accuracy. But, it also take more time to train, so more computational cost.
- Too small learning rate make learning slow and under fitting and too big learning rate leads to divergence.
- Too large epochs lead to over fitting.
- Here, lambda (L2 regularisation) didn't help.

Under Fitting: Low learning rate

Over fitting: large no. of epochs

Divergence: Large learning rate

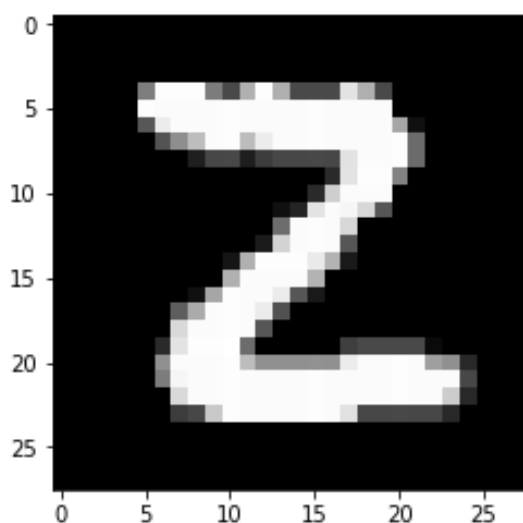
Best model till now – ReLu(100)+ReLu(36)+Sigmoid(10)

- Learning rate = 0.1
- Batch size = 1
- Convergence criteria = 99% train accuracy
- Max Epoch size = 20
- lam = 0
- Train accuracy – 98.44386809929604%
- Test accuracy – 91.33333333333333 %

Visualisation of representation learnt by hidden layer:

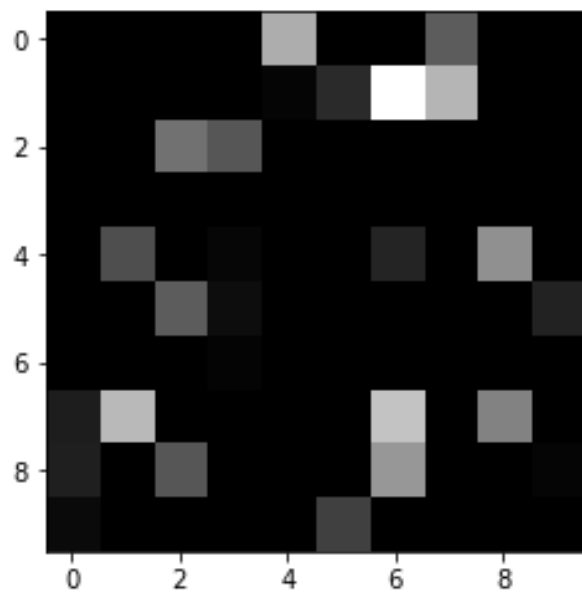
Input image (2) :

Pixel size : 28*28



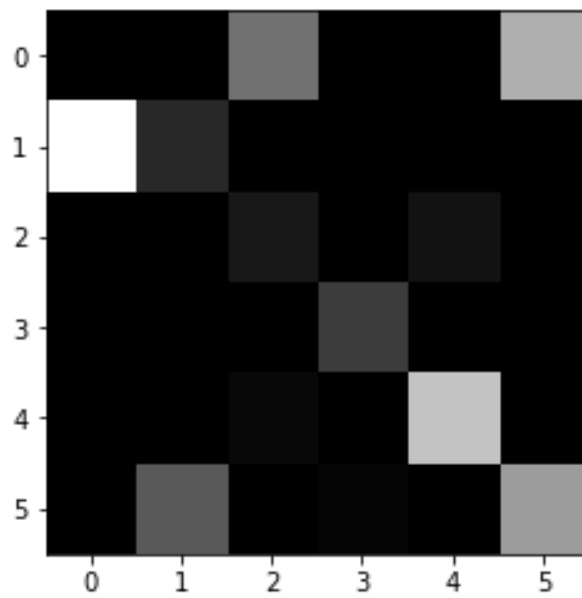
Representation learnt by 1st hidden layer (ReLU):

Pixel Size: 10*10

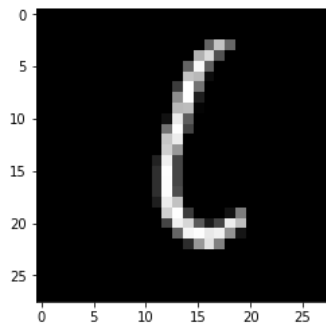


Representation learnt by 2nd hidden layer (ReLU):

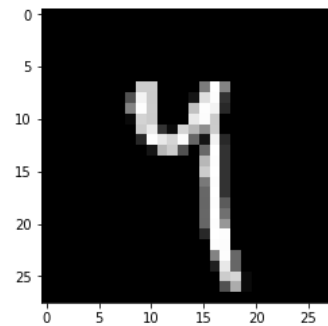
Pixel Size: 6*6



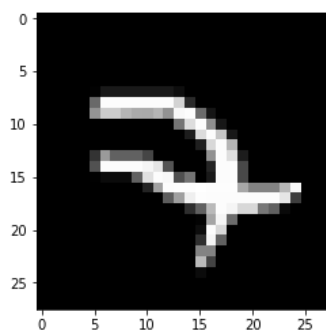
Example of Misclassified Images :



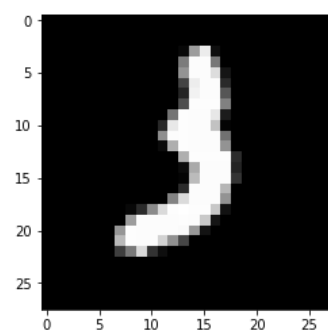
Actual label – 6



Actual label - 4



Actual label – 7



Actual label - 3

Implementation of above model using Keras library

Train accuracy – 100%

Test accuracy – 93.666667%

Conclusion: Results using Keras library are better. It has more accuracy for both train and test.

Possible reasons –

- Better initialisation of weight matrix
- Use of modified ReLu
- More efficient coding

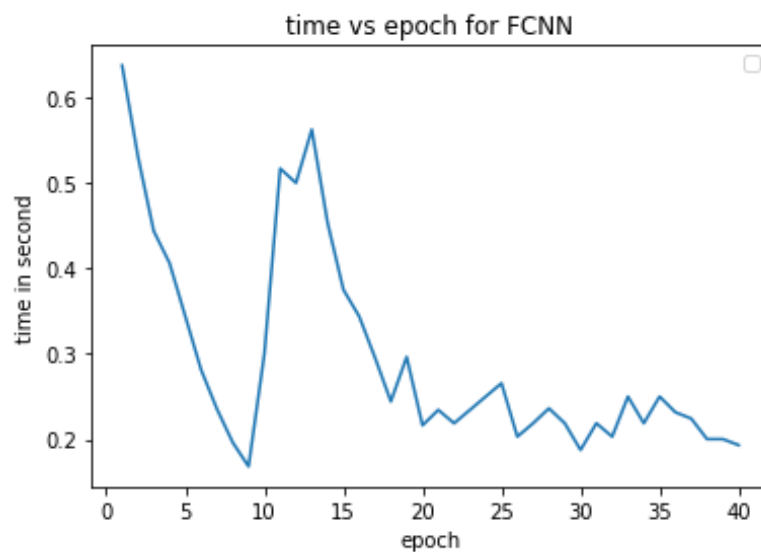
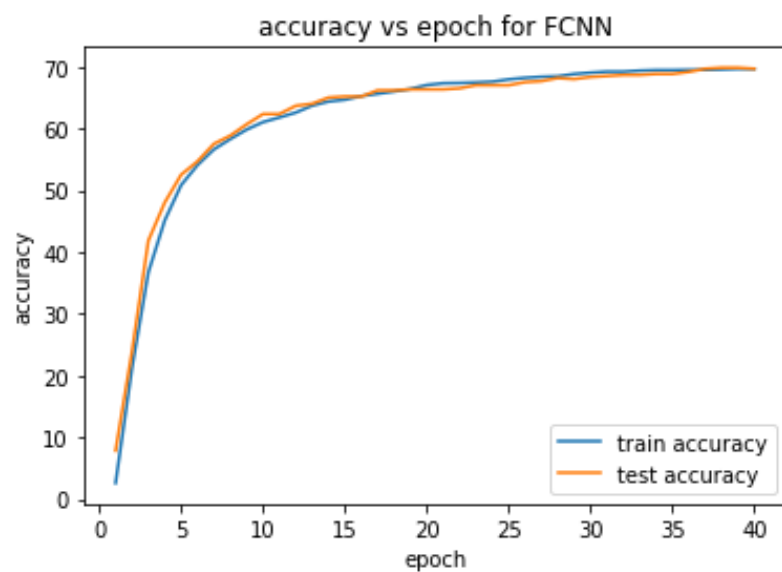
Part c | Comparison with PCA features:

Model 1 (without hidden layer: Sigmoid (10)):

- Learning rate = 1
- Batch size = 1
- Convergence criteria = 99% train accuracy
- Max Epoch size = 40
- lam = 0

Train accuracy – 69.6540225093789%

Test accuracy – 69.6666666666667%

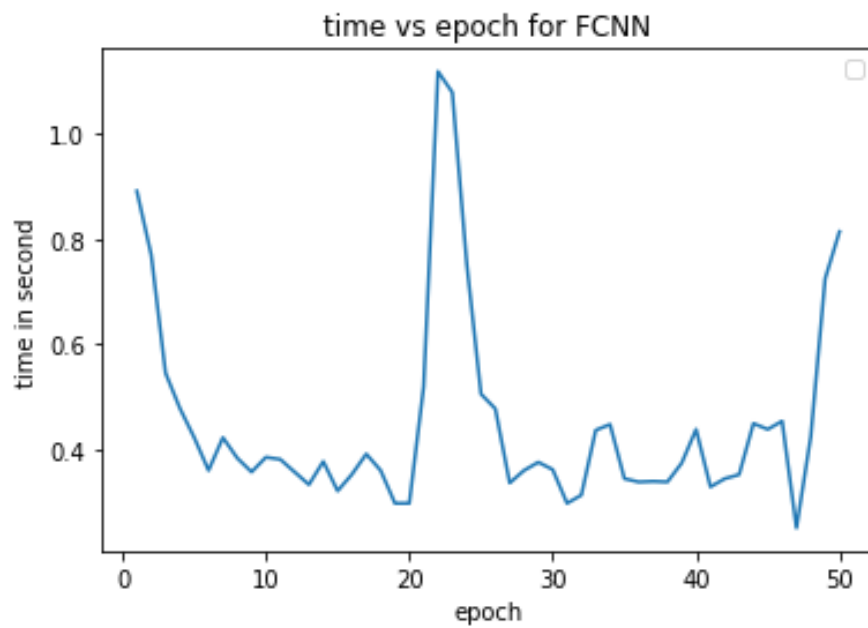
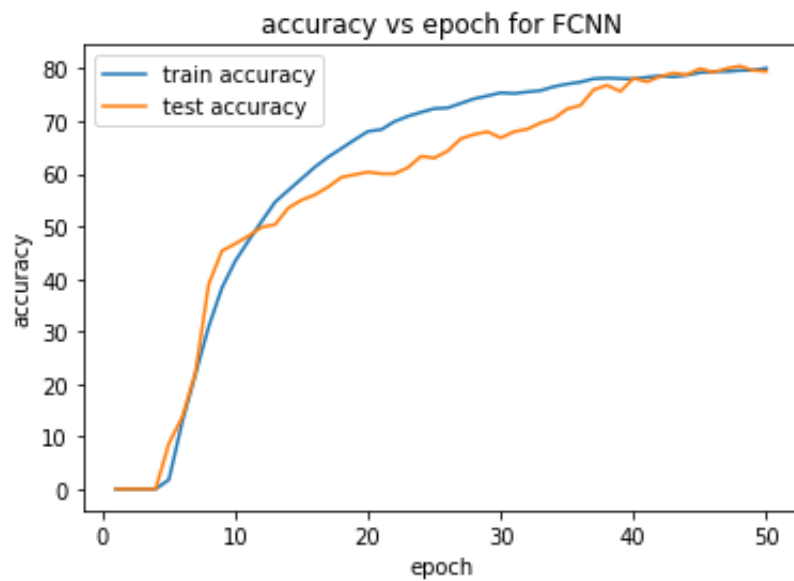


Model 2 (Sigmoid(25)+ Sigmoid (10)):

- Learning rate = 1
- Batch size = 1
- Convergence criteria = 99% train accuracy
- Max Epoch size = 40
- lam = 0

Train accuracy – 80.07503126302626%

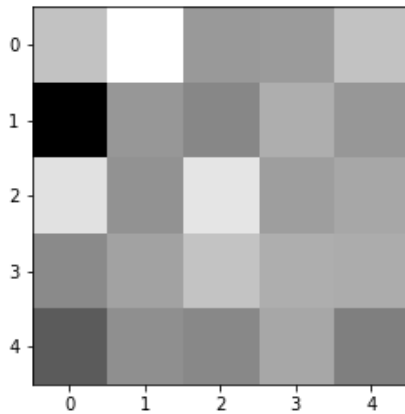
Test accuracy – 79.5%



Visualisation of PCA input features:

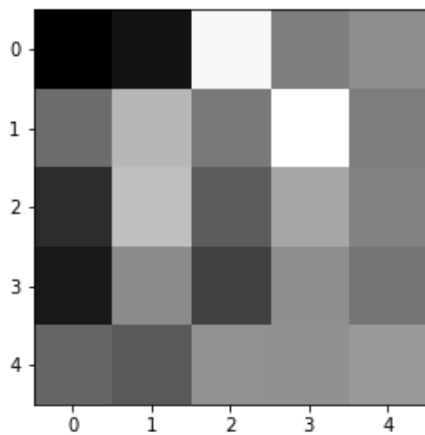
Label – 2

Pixels: (5*5)

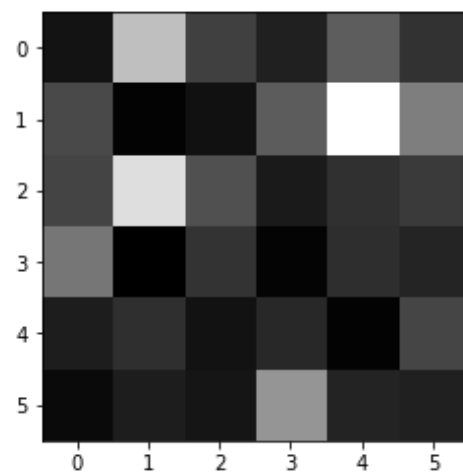


Label – 6

Pixels: (5*5)



Input Layer (5*5)



Hidden Layer (6*6)

Conclusion:

- Addition of hidden layer increased accuracy, as it allow features to be represented in higher dimension.
- Accuracy obtained from PCA feature are not as good as accuracy obtained from raw pixels.