ABHINAV AGRWAL

ROLL NO. : 14011

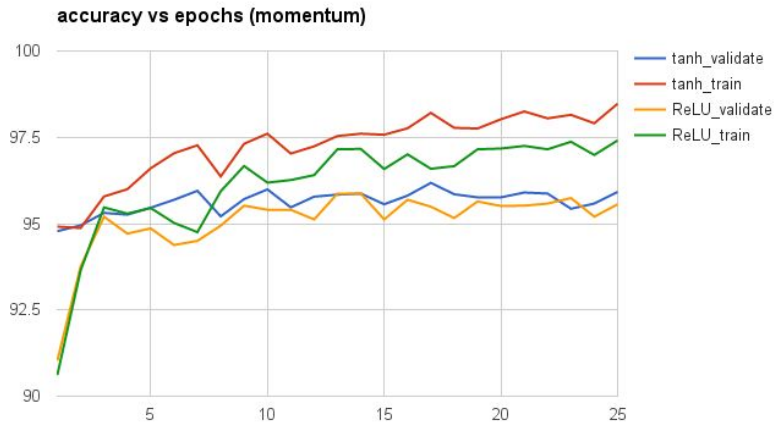
CS698A - ASSIGNMENT 2

Part1:

Compare the results from the previous assignment and current implementations reveal that :

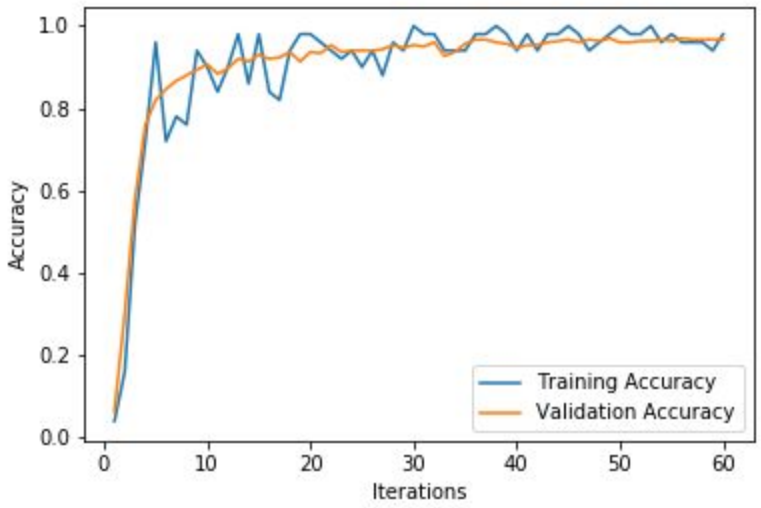
- Fully connected layer network of the assignment 1 required more epochs to reach the same level of accuracy as that of the CNN implementations.
  Also it overfits data after some epochs and thus has limited “capacity” as a network
- Tensorflow implementation was fast and required less epochs to reach the highest accuracy among the three models.
- Our Numpy implementation is naive and thus takes more time as compared to a tensorflow implementation. Although it achieves the desired results without overfitting the data.

Results from assignment 1 for tanh and ReLU with momentum update.



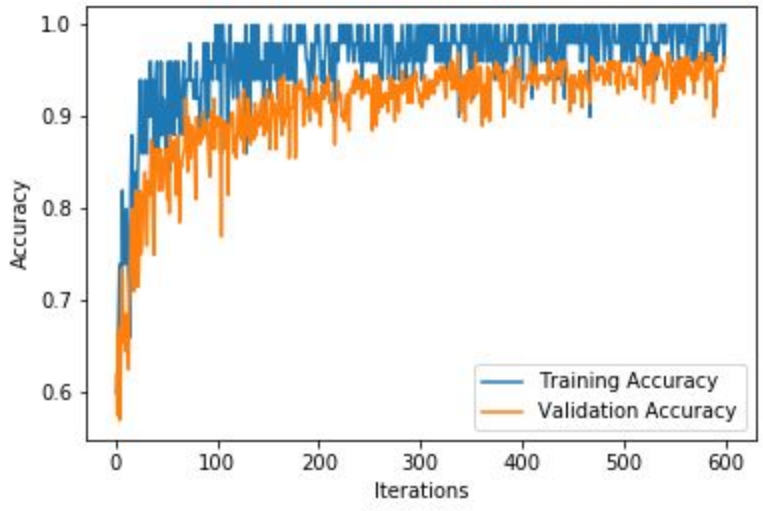
This plot is for training and validation accuracies vs Epochs. The data set of only 50000 images was used for this. The momentum was 0.5 and learning rate = 0.1 with a mini-batch-size of 30.

Results from Tensorflow implementation:



This was done with mini-batch-size=50 and momentum update with momentum=0.5 and learning rate=0.01. The plot depicts the accuracy of the current training mini-batch and a randomly selected validation set of 300 images.

Results from Numpy implementation:



This was done with mini-batch-size=50 and momentum update with momentum=0.5 and learning rate=0.1. The plot depicts the accuracy of the current training mini-batch and a randomly selected validation set of 300 images.

Part2:

Time taken by Convolution and FC layers

TIME FORWARD PASS(5000 IMAGES) :

Convolution layers

Total time/example=(16.6823408604+26.6283721924)/5000=0.00866214261

FC layers

Total time/example=(0.1451020+0.089962959+0.088077068)/5000=6.46e-5

TIME BACKWARD PASS(5000 IMAGES) :

FC layers

Total time/example=(0.113877058029 + 0.249548912048 +0.6888098)/5000 = 2.1e-4

Convolution layers

Total time/example=(199.872503042 +16.24345994)/5000 = 0.0432231926

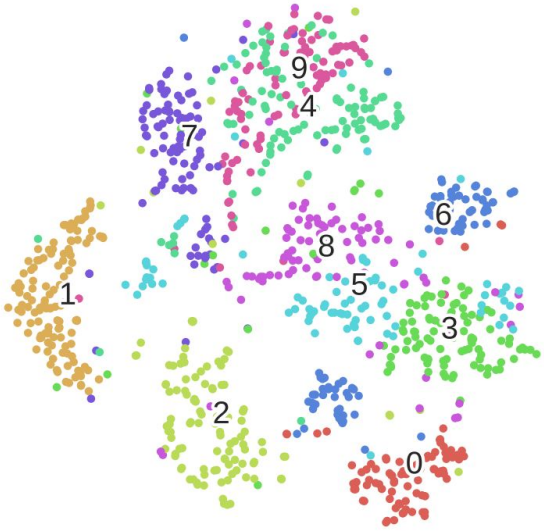
Number of Parameters in the Convolution and Fully Connected Layers

Params in CONV: 6*1*5*5 + 6 + 16*6*5*5+16 = 2572

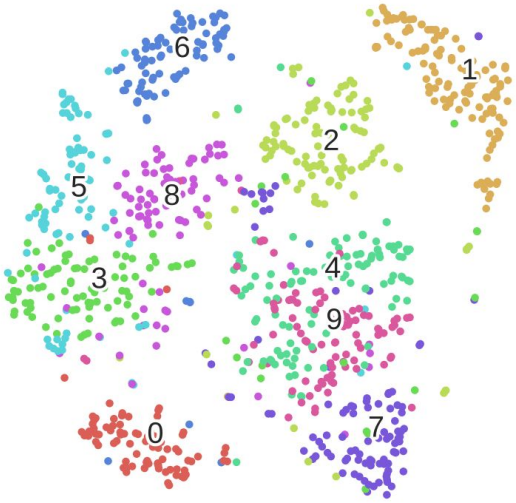
Params in FC: 401*120 + 121*84 + 85*10 = 59134

t-SNE Visualizations

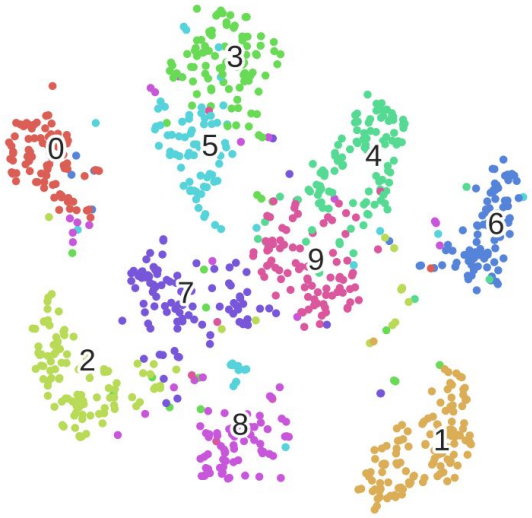
Features extracted from layer 1:



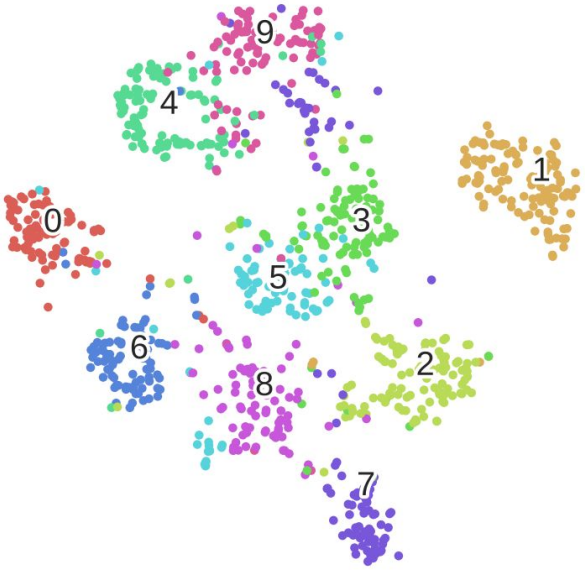
Features extracted from layer 2:



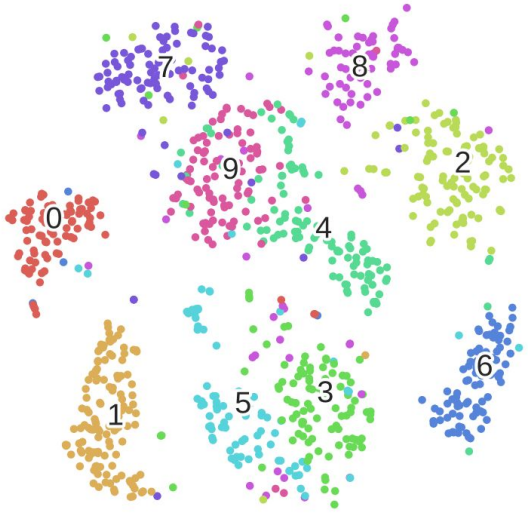
Features extracted from layer 3:



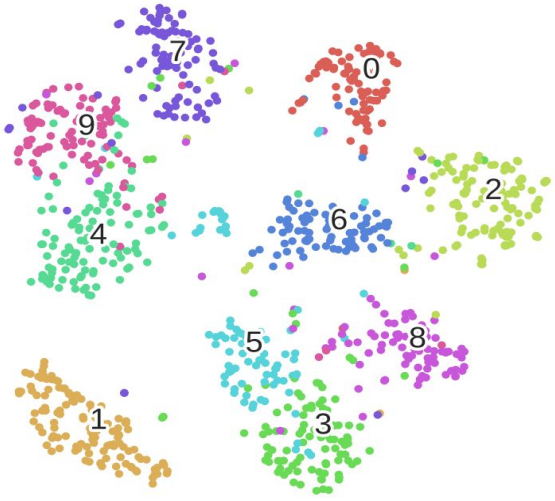
Features extracted from layer 4:



Features extracted from layer 1-2-3-4 concatenated:

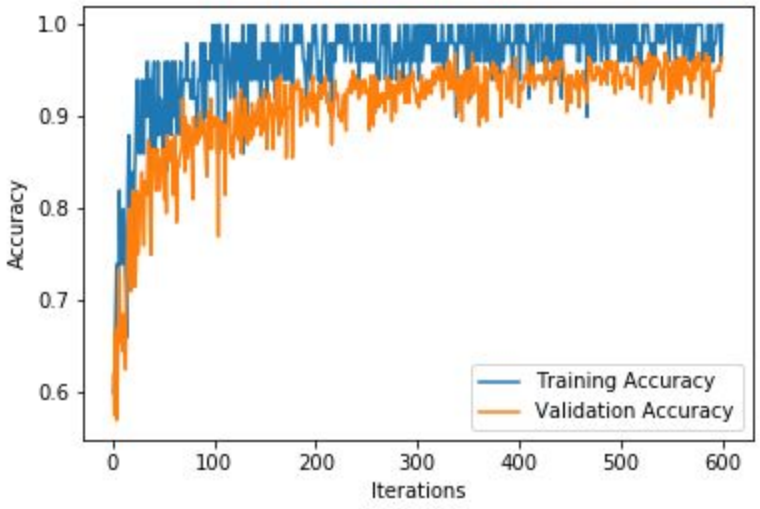


Features extracted from layer 2-4 concatenated:



Plots for Training and Validation Error:

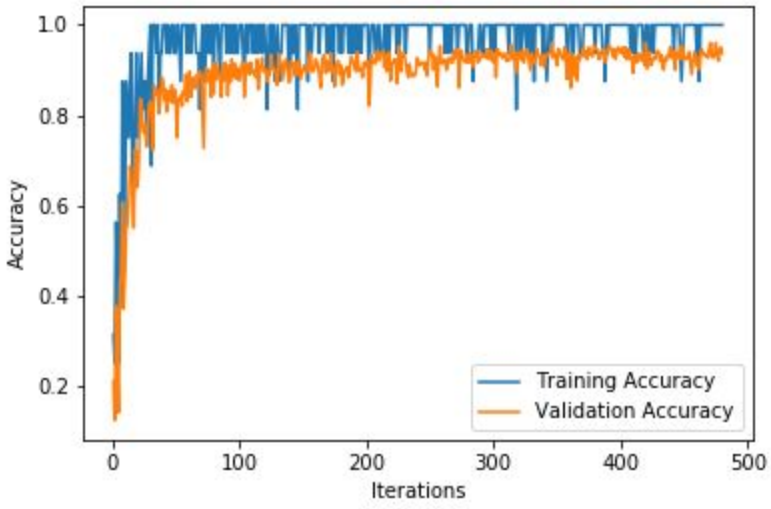
The net was trained with full 50000 image training data set with 3 epochs and mini-batch size 50. The accuracy over each mini-batch and a randomly sampled 200 image validation set were plotted at every 5th iteration.



Effect of different mini batch sizes:

The net was trained with 12800 images to create equally sized mini-batches for all the choices of mini-batch-size given. Results show that mini-batch-size of 32 was the best one as it concluded in comparatively faster time and with better accuracy.

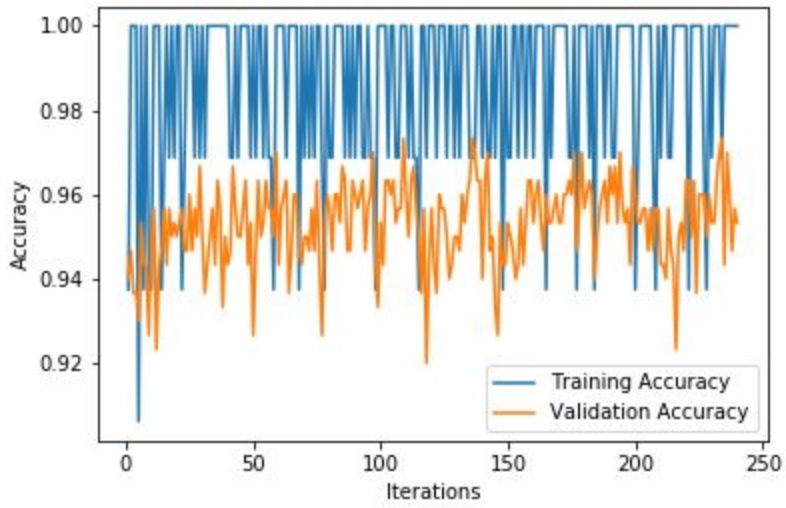
For mini-batch-size=16



Time : 0 epoch 1465.08369589
1 epoch 1535.02881384
2 epoch 1533.39292598

Test accuracy : 0.9563 Validation accuracy : 0.9573 Train accuracy : 0.9529

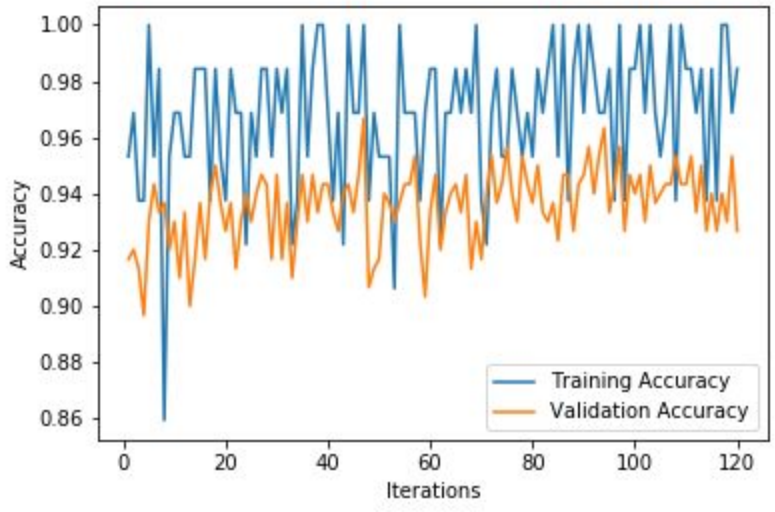
For mini-batch-size=32



Time : 0 epoch 1224.40599799
1 epoch 1197.81009912
2 epoch 894.275461912

Test accuracy :0.9672 Validation accuracy :0.9674 Train accuracy : 0.9670

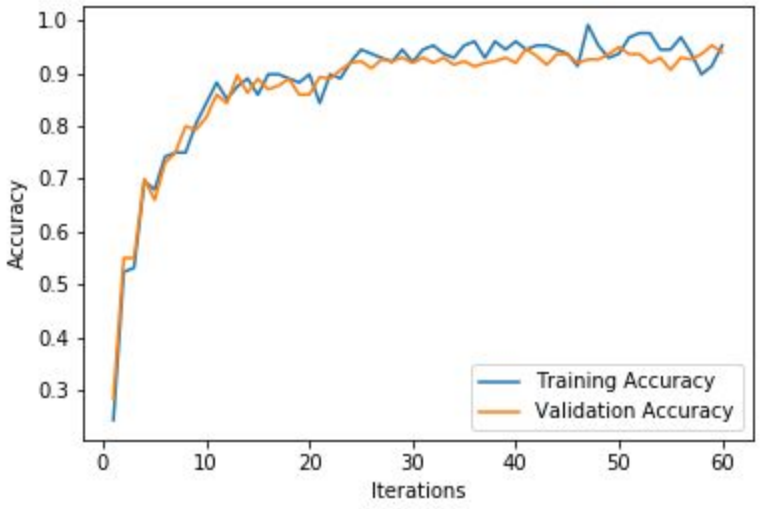
For mini-batch-size=64



time for 0 epoch 1078.84635496
time for 1 epoch 1052.40928292
time for 2 epoch 1084.69945192

Test accuracy :0.9492 Validation accuracy :0.9498 Train accuracy : 0.9475

For mini-batch-size=128



time for 0 epoch 1053.90425491
time for 1 epoch 1009.08536696
time for 2 epoch 1004.75643086

Test accuracy :0.9337 Validation accuracy :0.9368 Train accuracy : 0.9341