HW 6 Write-up

- **Q4** Visualize the output of the second layer after 30 iterations of training. Note that the output of first layer is simply the image itself (because the first layer is data layer). The output of the second layer (convolution) is 20 images, each of size 24 24, hence 24 24 20. Show 20 images from the layer on a single figure file.

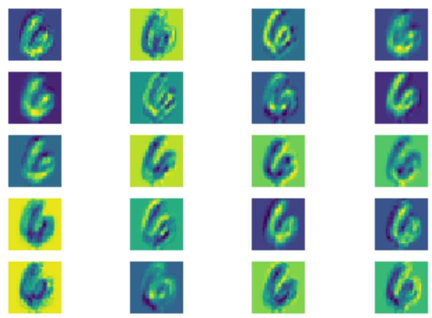

Figure 1: Output of 2nd layer after 30 iterations

- **Q5** Plot the train cost and test cost as a function of training iterations. Include a point every 100 iterations. Notice these numbers are printed to the screen when you run the code. Note also the test cost is computed over the 10,000 test examples, and the training costs are computed over minibatches of 64 examples. Label each line on your plot.

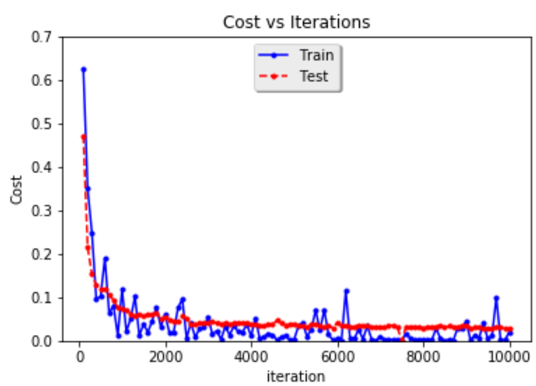

Figure 2: Train cost and Test cost as a function of iteration

- **Q6** Plot the train accuracy and test accuracy as a function of training iterations. Include a point every 100 iterations. Note also the test accuracy is computed over the 10,000 test examples, and the training costs are computed over minibatches of 64 examples. Label each line on your plot.

  Note: I was getting 0 accuracy for iteration number 7500. This was because of the selection of batchisize 64 and train size 60,000. Due to this part some iterations will be skipped. Under 10,000 iterations those numbers are 1875, 3750, 5625, 7500 and 9375. Here the 7500th iteration is important because we need to record the accuracy and cost for it but unfortunately it's skipped. In this iteration, start idx becomes 59936 and end idx becomes 0 thereby resulting in start idx greater than end idx and skipping of evaluation. As suggested by an instructor on piazza, for 7500th iteration I have taken the cost and accuracy to be same as 7499th iteration

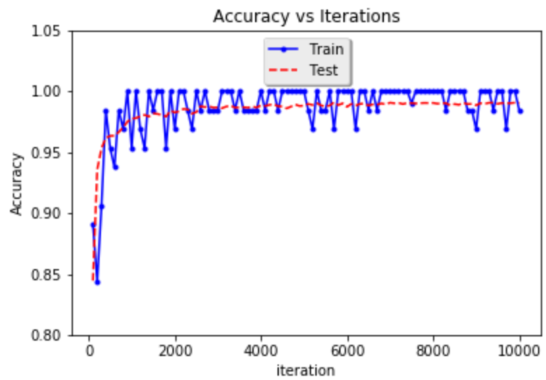

Figure 3: Train accuracy and Test accuracy as a function of iteration

- **Q7** Report the test accuracy and time taken by testLeNet to run the training procedure for 10,000 iterations

  Test accuracy: 0.9903 (i.e., 99.03 %)
  Time take: 220 minutes (i.e., approx 3.5 hours)

- **Q8** Visualize the output of the second layer after you finish training (10,000 iterations). Show 20 images from the layer on a single figure file.

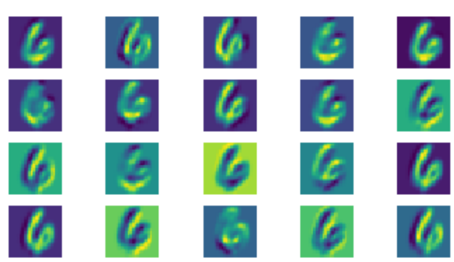

Figure 4: Output of 2nd layer after 10000 iterations

- **Q9** Interpret the differences between the visualization results obtained in Q4 and those obtained in Q8.

> Lower layers are responsible for learning local features of the image like edges, curves etc. The difference between Fig 1 and Fig 4 is that convolution layer after 10000 iterations has learned these local features much better than the one in just 30 iterations. As such, these edges and curves are easily identified in Fig 4 but are blurred in Fig 1.

- **Q10** Visualize the output of the second and third layer after 10,000 iterations. The output of the third layer (max pooling) is of size 12 12 20. Show 20 images from each layer on a single figure file.

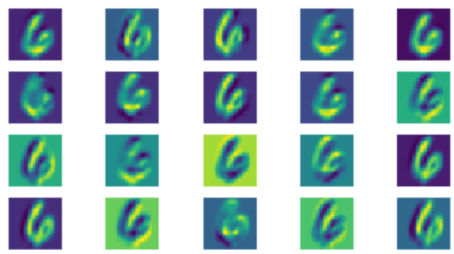

Figure 5: Output of 2nd layer after 10000 iterations

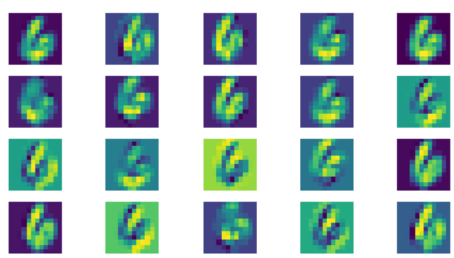

Figure 6: Output of 3rd layer after 10000 iterations

- **Q11** Answer the following questions based on your visualizations:
  - a Compare the output of the second layer (from Q9) and the original image (output of the first layer), what changes do you find? Explain your observations.

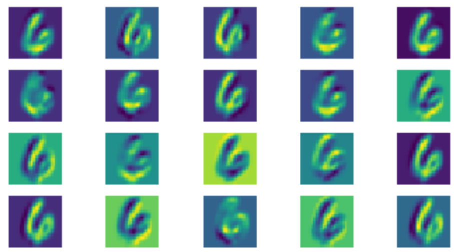

Figure 7: Output of 2nd layer after 10000 iterations

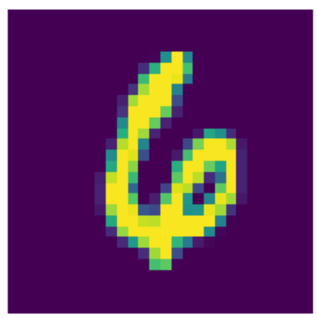

Figure 8: Original Image

Second layer is responsible for learning the local features of the image. In this case, it is learning the edges and curves of the digit 6. All the 20 images shown in Fig 7 are trying to learn one or the other local feature of original image.

– b Compare the output of the third layer (from Q9) and the output of the second layer (from Q9), what changes do you find? Explain your observations

The difference between third layer and second layer is that third layer samples the output of second layer. In this case, since we are doing a maxpool, it takes just the maximum intensity from each 2x2 window. As such, the fine learning of local features like edges and curves demonstrated by layer 2 is reduced in quality when it is passed through layer 3. This is evident from Fig 6 where the images are blurred and have lesser resolution compared to those in Fig 5. However, this will not result in data loss because the local gradients of images are relatively small. So it is safe to pool and reduce the image size.

- **Q12** Did you receive any help whatsoever from anyone in solving this assignment?

No

- **Q13** Did you give any help whatsoever to anyone in solving this assignment?

Yes, I helped Ojas in the visualization part

- **Q14** Did you find or come across code that implements any part of this assignment?

No