# Assignment 3

**Question 1:**

1. Regular CNN
   - Learning rate: 0.001
   - Batch size: 64
   - Optimizer: Adam
   - Epochs: 7
   - Test Accuracy:  0.9897
   - Test Loss: 4.38

   - I varied the learning rate along with the number of epochs. When I put the learning rate as 0.01, the test accuracy was observed to be only 0.2. Best test accuracy was observed when I put learning rate as 0.001. When learning rate was kept at 0.0001, the accuracy further decreased. Hence, learning rate was decided at 0.001.

   - As for the epochs, I chose 7 as the optimal value. It gave a test accuracy of 0.9897, and test loss of 4.38.

2. Inverted CNN
   - Learning rate: 0.01
   - Batch size: 128
   - Optimizer: Adam
   - Epochs: 1
   - Test Accuracy:  0.9866
   - Test Loss: 6.05

   - I varied the learning rate along with the number of epochs. When I put the learning rate as 0.01, the test accuracy was observed to be only 0.97. However, the test loss was at 7.85. Best test accuracy was observed when I put learning rate as 0.01.

   - As for the epochs, when I chose 1 epoch, it gave a test loss of 6.64. However, using 2 epochs, test loss was at 6.05

3. Hour-glass CNN
   - Learning rate: 0.01
   - Batch size: 128
   - Optimizer: Adam
   - Epochs: 1
   - Test Accuracy:  0.9811
   - Test Loss: 9.22

- I varied the learning rate along with the number of epochs. When I put the learning rate as 0.01, the test accuracy was observed to be only 0.63, and test loss was 160.276. However, when learning rate was 0.01, the test loss was at 9.22, and accuracy 0.9811. Best test accuracy was observed when I put learning rate as 0.01.

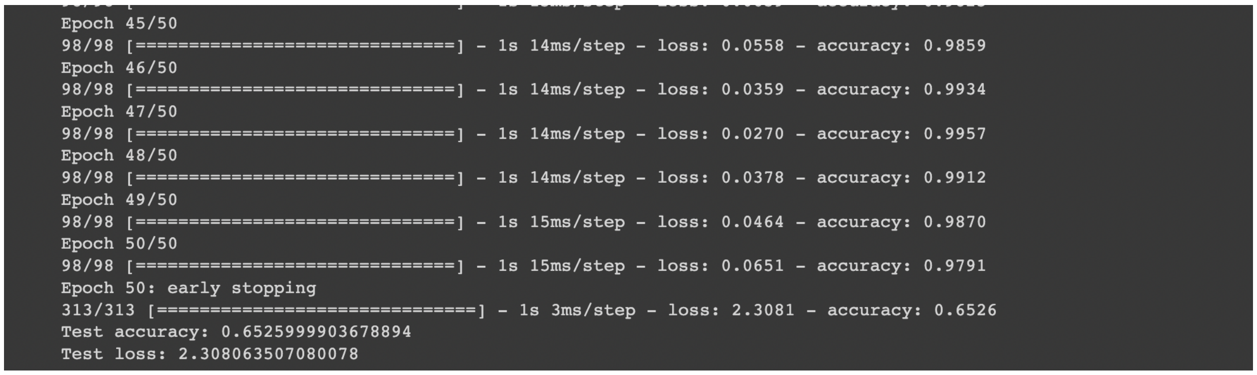
- As for the epochs, when I chose 1 epoch, it gave a test loss of 14.15. However, using 2 epochs, test loss was at 9.22

**Question 2:**

1. Learning rate didn't impact the training immensely. In fact, changing the learning rate values to 0.01, 0.001, or 0.0001 – all had similar test accuracy. I went ahead with learning rate as 0.001
2. I used {64,128,512} to test the performance on the model. I observed that batch size of 512 performed the best, which is why I chose 512 as the batch size.
3. As learning rate didn't impact performance that vastly, I decided to vary the epochs to observe changes in performance

For 25 epochs:
- Test accuracy: 0.67
- Test loss: 1.077

For 50 epochs:
- Test accuracy: 0.65
- Test loss: 2.30



```
Epoch 45/50
98/98 [==============================] - 1s 14ms/step - loss: 0.0558 - accuracy: 0.9859
Epoch 46/50
98/98 [==============================] - 1s 14ms/step - loss: 0.0359 - accuracy: 0.9934
Epoch 47/50
98/98 [==============================] - 1s 14ms/step - loss: 0.0270 - accuracy: 0.9957
Epoch 48/50
98/98 [==============================] - 1s 14ms/step - loss: 0.0378 - accuracy: 0.9912
Epoch 49/50
98/98 [==============================] - 1s 15ms/step - loss: 0.0464 - accuracy: 0.9870
Epoch 50/50
98/98 [==============================] - 1s 15ms/step - loss: 0.0651 - accuracy: 0.9791
Epoch 50: early stopping
313/313 [==============================] - 1s 3ms/step - loss: 2.3081 - accuracy: 0.6526
Test accuracy: 0.6525999903678894
Test loss: 2.308063507080078
```

4. Next, I implemented the feedforward network.
   a) I observed the performance as
      - Test accuracy: 0.10
      - Test Loss: 2.30

b) There are 697,046 parameters in the LeNet CNN as compared to 31,604 parameters in the feed forward network. The parameters in LeNet CNN even though they are way higher than normal feed forward, I think it is worth it because the performance in LeNet is way better. The test accuracy is 65% as compared to 10%

**Question 3:**

1. Matrix size = (6,6,1). The size of the filter is (3,3,1). Given depth = 1

Then,

Number of parameters = [{filter_size*filter_size*d}+1))*no of filters]
= (3*3*1)+1*1
= 10 parameters

**Number of parameters** is 10.

2. Part2Part3.pdf. It is a handwritten work.

3. Part2Part3.pdf. It is a handwritten work.