**1. Theory**

**Q 1.1**

Note that the SoftMax function is

Because calculation can be very unstable when is very large, it is often recommended that we subtract the maximum value of first.

**Q 1.2**

,

One could say that “SoftMax takes an arbitrary real valued vector x and turns it into a probability distribution”.

The SoftMax function is trying to approach the categorical distribution of events. The first step is calculating the weight of a certain class, the second step is calculating the normalization factor of the distribution, and finally, normalize the distribution by the factor to get a proper probability distribution.

**Q 1.3**

The linear activation function has a derivative of , which is constant and not dependent on the input . Also, the activation of one layer in turn goes into the next layer as input and the second layer calculates weighted sum on that input and it in turn, fires based on another linear activation function. No matter how many layers we have the final activation function of last layer is nothing but just a linear function of the input of first layer. So, without non-linearity, the network is similar to linear regression, only with different numerical precision.

**Q 1.4**

Derivative of sigmoid function:

**Q 1.5**

**Q 1.6**

1.

Because the maximum value of the derivative of sigmoid function is and the nature of chain rule, when and the network is very deep, it is likely to have vanishing gradient.

2.

The output range of sigmoid and tanh function are the same But the maximum value of the gradient of tanh function is , it is less likely to have vanishing gradient problem. And because is usually less than , exploding gradient is rare.

3.

The gradient of tanh function has a range of . It is less likely to have vanishing gradient problem.

4.

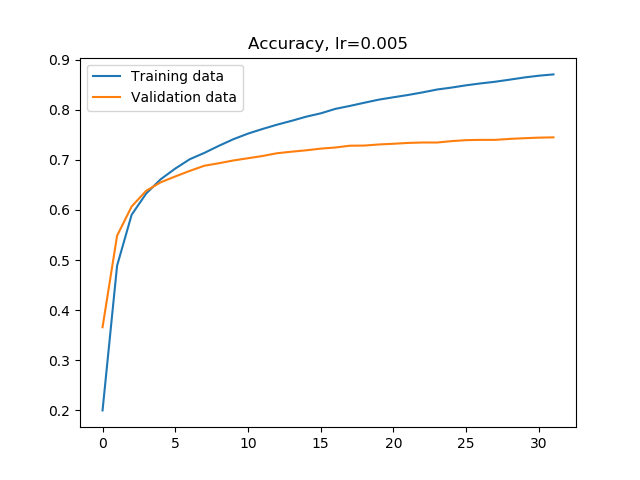
2.1

If all weights are initialized to the same value, no matter what the input is, al units in hidden layer will be the same---the network gets stuck. Moreover, if all weights are set to zero, the gradient descent algorithm will be slow at the beginning because the gradients will be small.

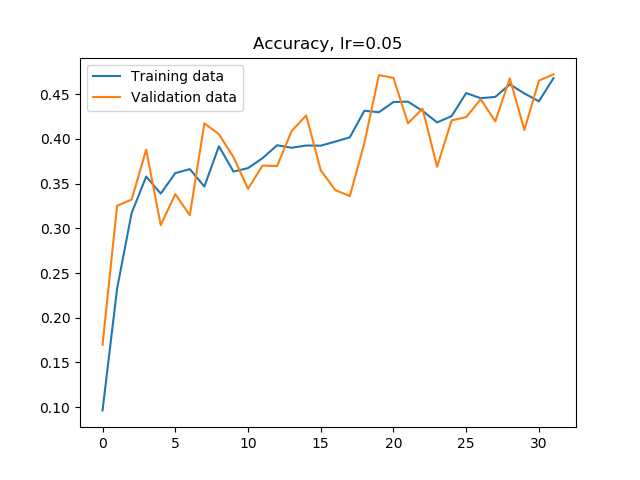
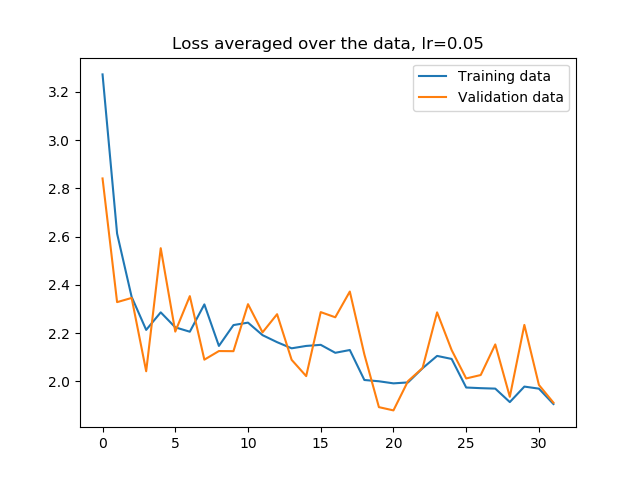
2.2

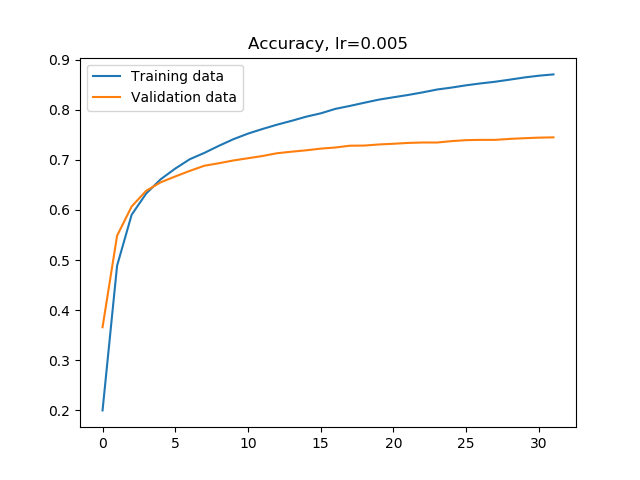
Gradient descent requires that the weights of the network are initialized to small random. Randomness is also used during the search process in the shuffling of the training dataset prior to each epoch, which in turn results in differences in the gradient estimate for each batch. Typically, we want the variance of input layer and output layer to be the same, so that the learning process is less likely to get stuck in saturated region of the activation function.

3.1.1



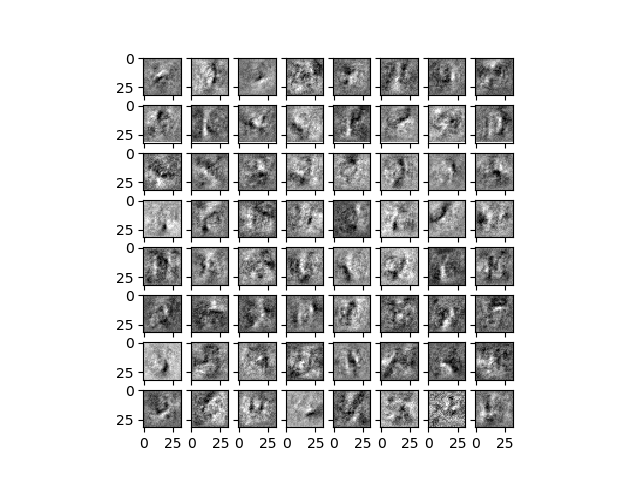
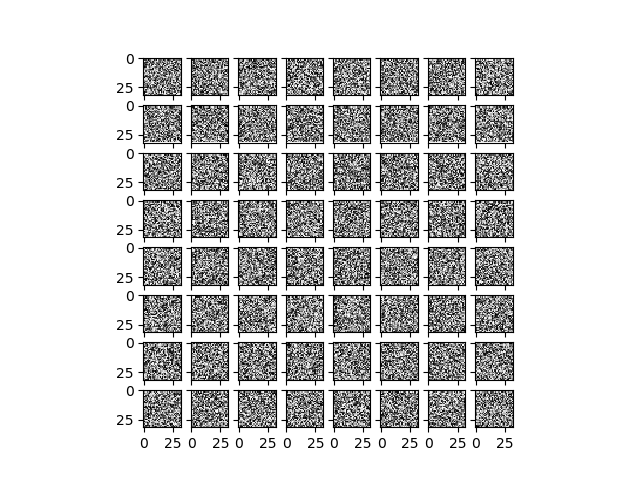
3.1.2





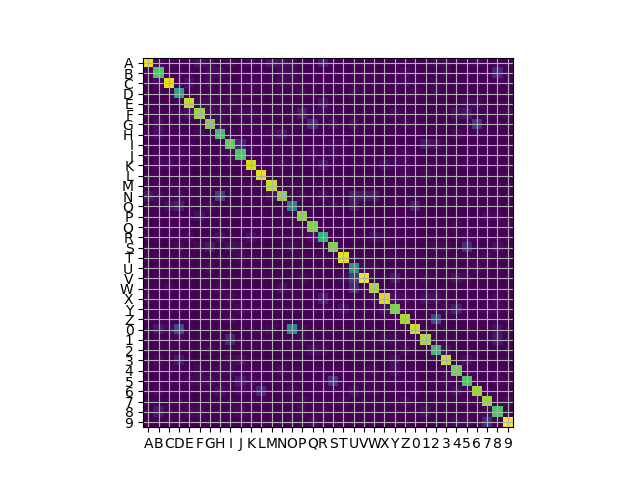
Large learning rate will speed up the training at the beginning, but could lead to over shoot, meaning that the gradient is not necessary going down at the right direction. Small learning rate could be too slow and easily get stuck in local minimum. The best learning rate is , and the average accuracy after 30 epochs is .

3.1.3



The weights become more organized and seem to capture edge information at all directions.

3.1.4



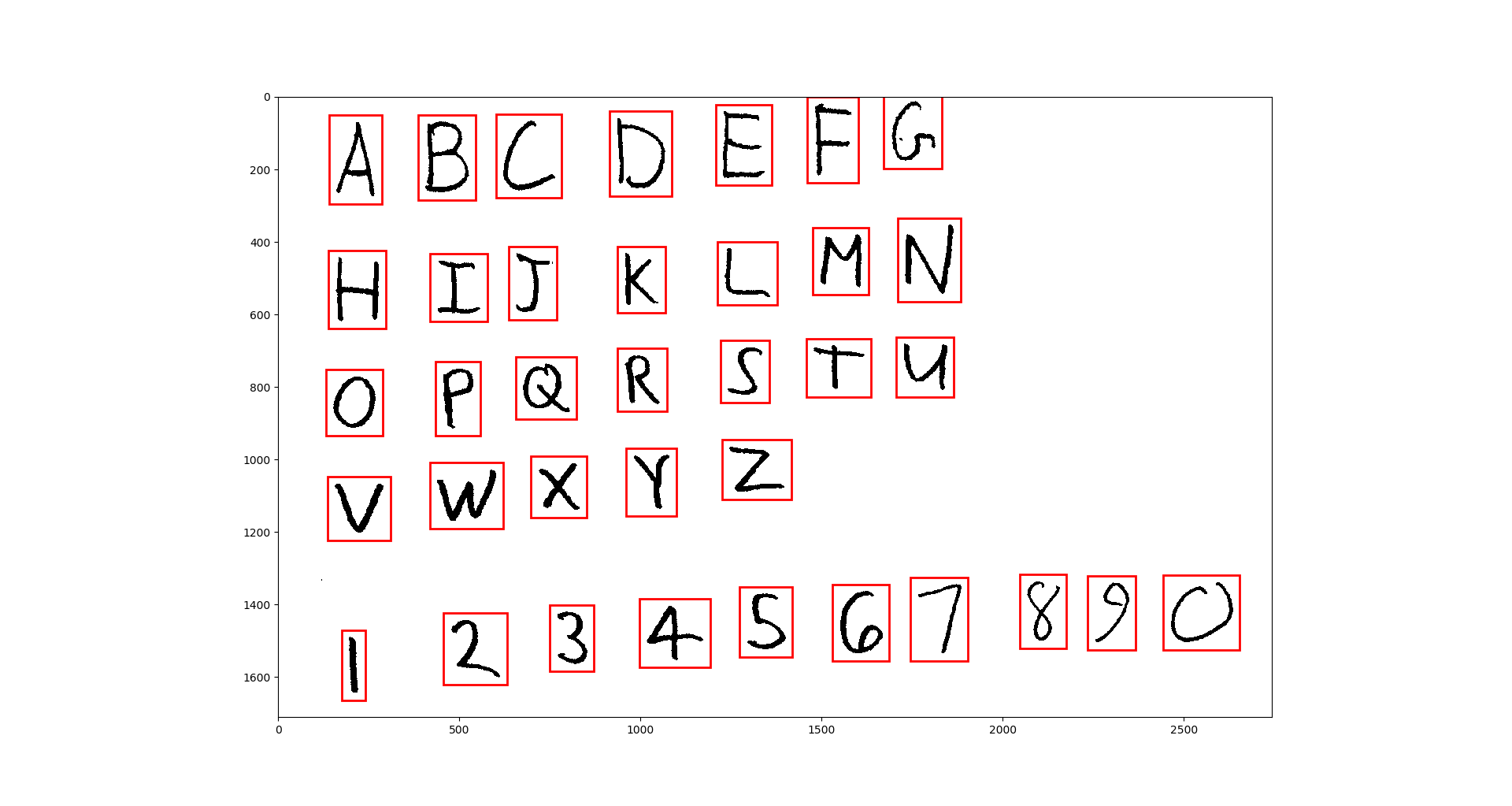
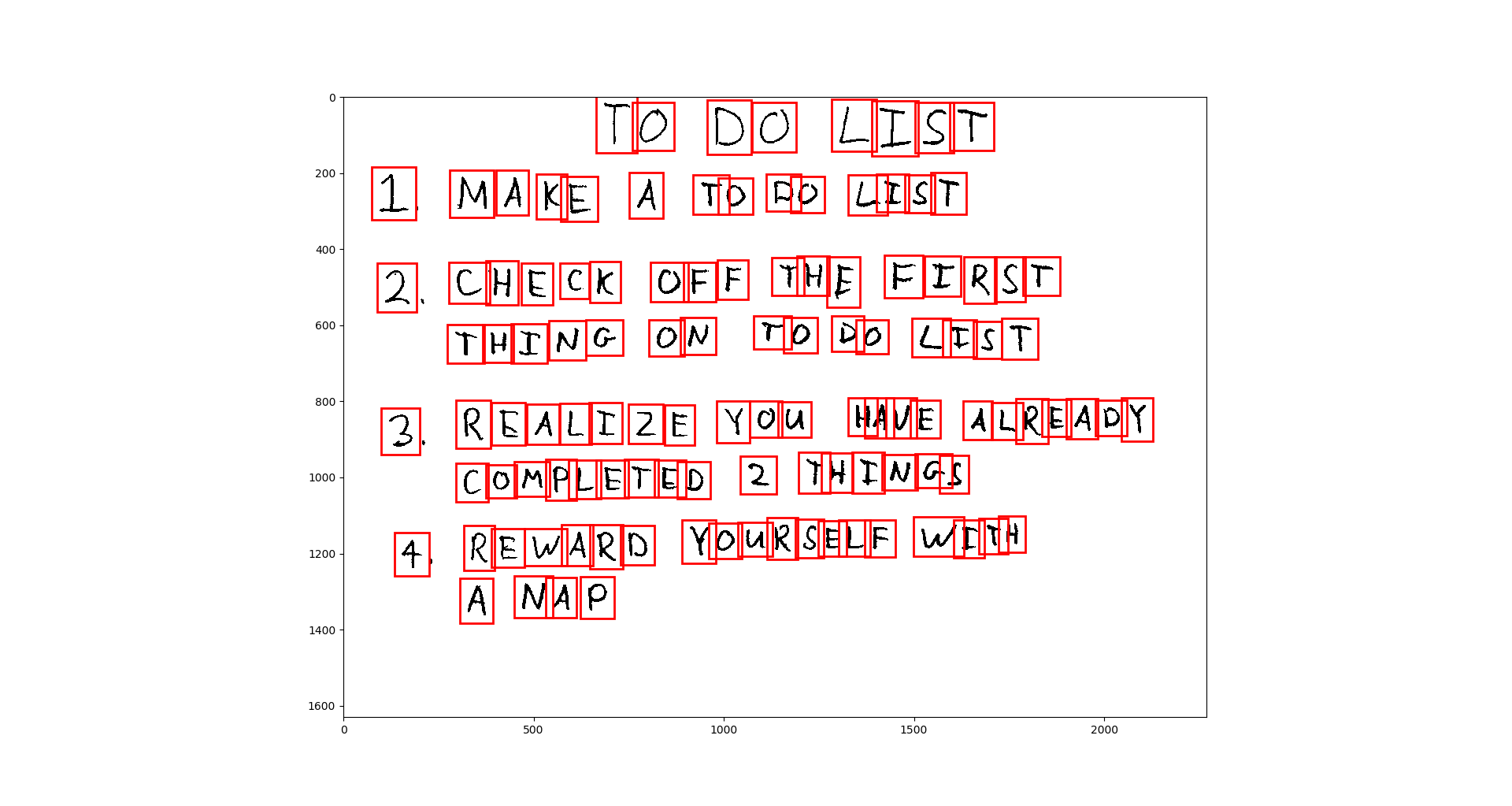
The most commonly confused pairs are 0 and O, U and V, 2 and Z, N and H, which make sense, because they are similar to each other.

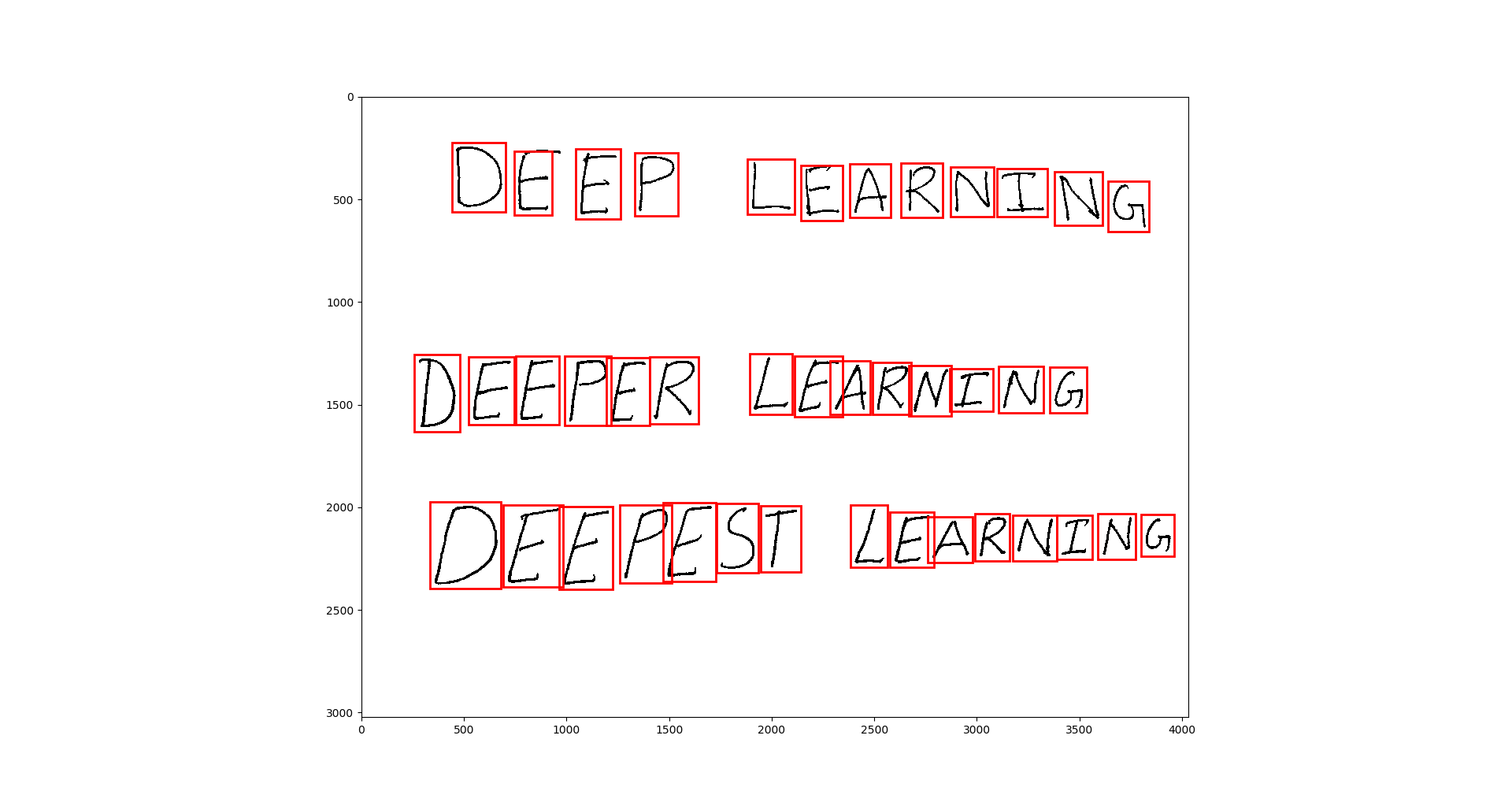
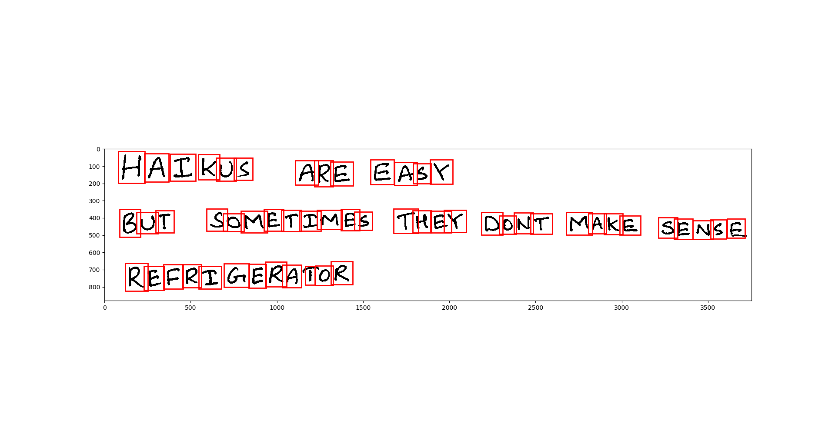
Q 4.1

All letters are horizontally arranged on blank paper and have great contrast with respect to the background. Only letters are on the paper.

All letters are about the same size and placed at the same depth, they are connected

Q 4.3





Q 4.4

T0 D0 LIST

I NAXE A TQDQLIST

2 CHERK DFFTHEFIRST

THINGQN T0D0CIST

3 RXALI2EYDU HUVEALREADY

C0MPLET2D 2THINGS

4 R8WARD Y0URSELFWITH

ANAP

A BC B E F G

H IJ K L M N

Q F Q R S T U

V W X Y Z

8 Z 3 F S G7 X9 0

HAIWUS ARE EASY

BUT SQMETIMES THEX D0NT MAKE SENGG

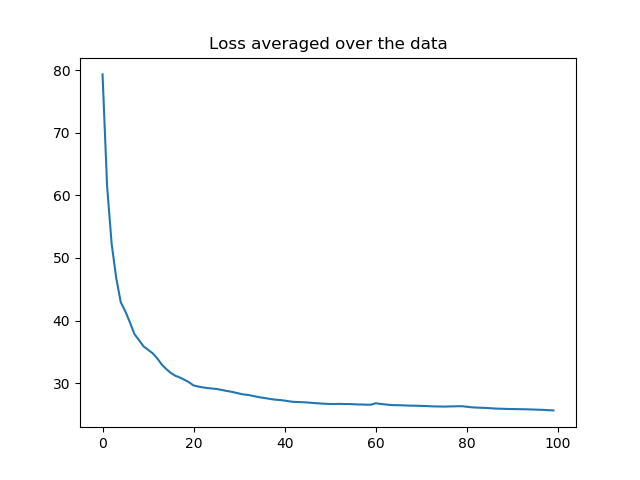
REFRIGERAT0R

CC FF LCAKMIXG

DFHFFK LFAKNING

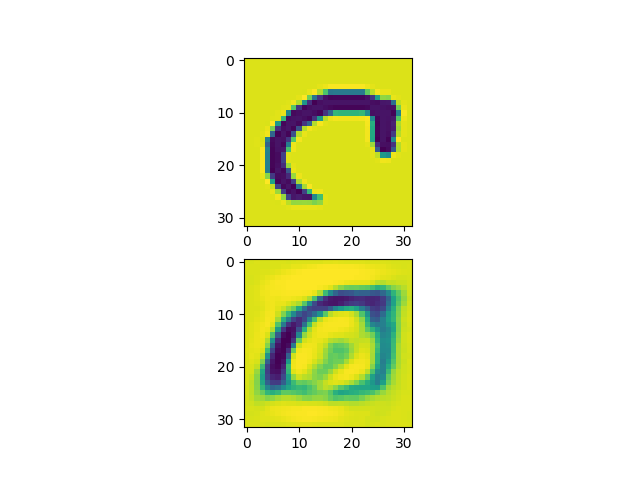
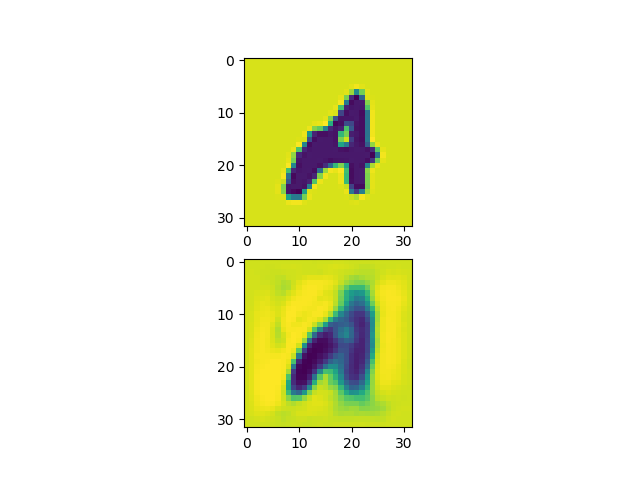
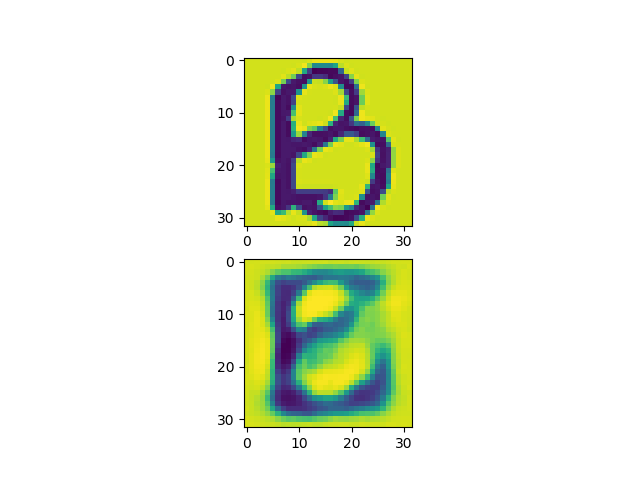
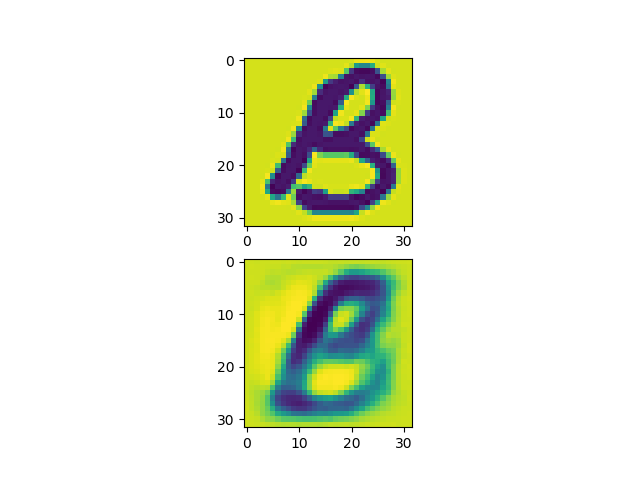
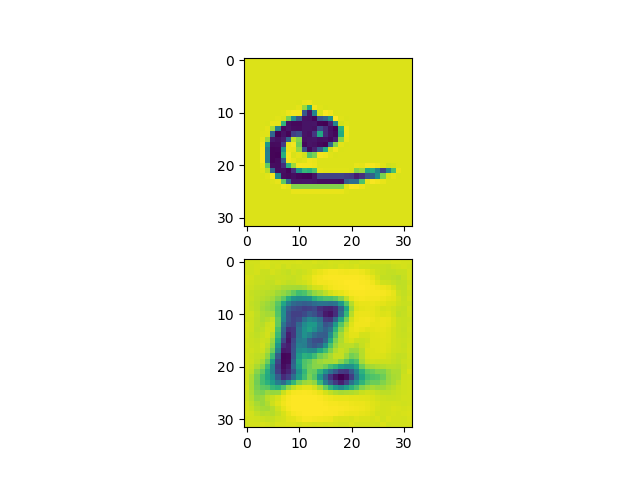
JFFFFST LFARNING

Q 5.2



The overall loss plot is smooth. The loss drops fast at the beginning because of large momentum.

Q 5.3.1



The reconstruction image is similar to the input image.

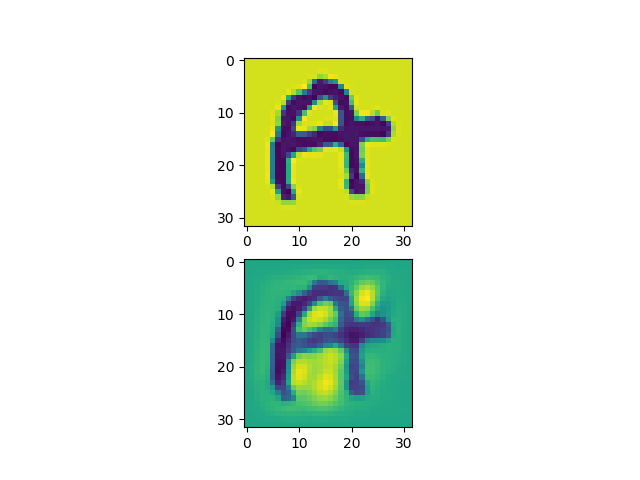
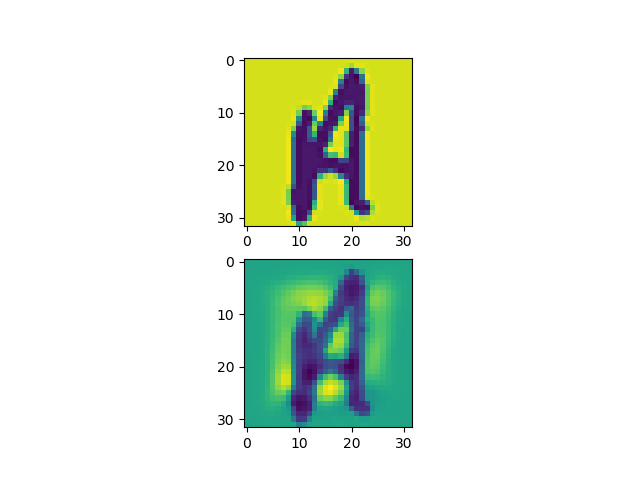
Q 5.4

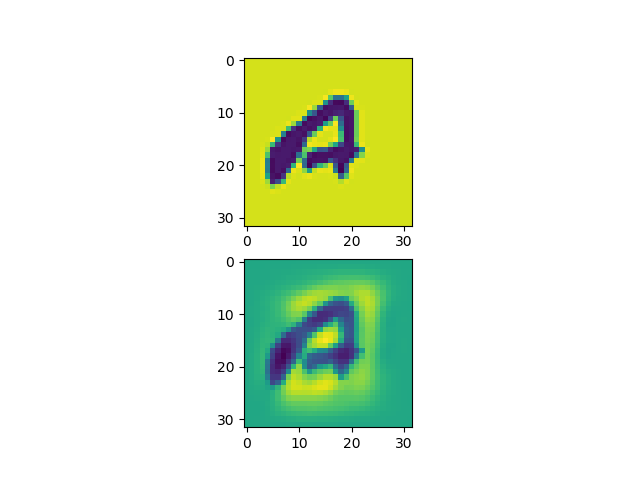
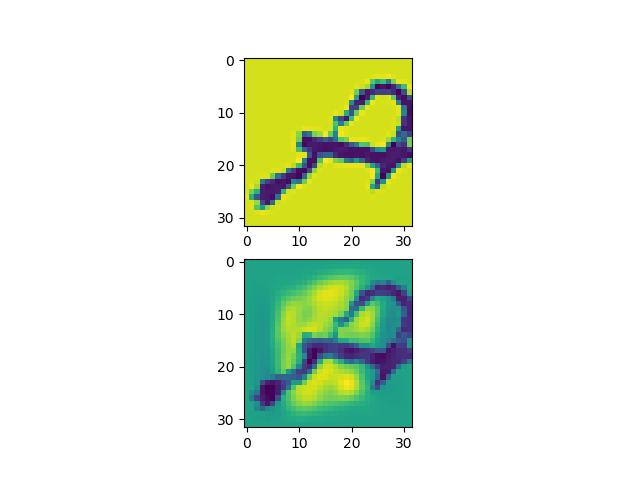
PSNR = 14.799982898096122

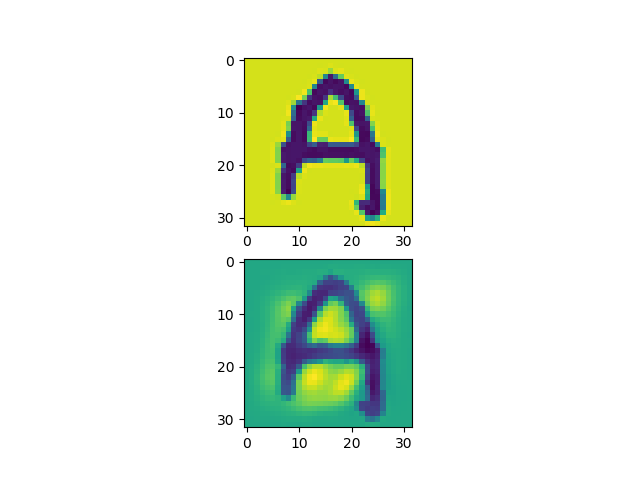
Q 6.1

For a PCA with only 32 principle components, the projection matrix is . It has a rank of .

Q 6.2







Eigenfaces capture the basic structure of the input image. It performs better than autoencoder. The autoencoder seems to capture the background information as well.

Q 6.3

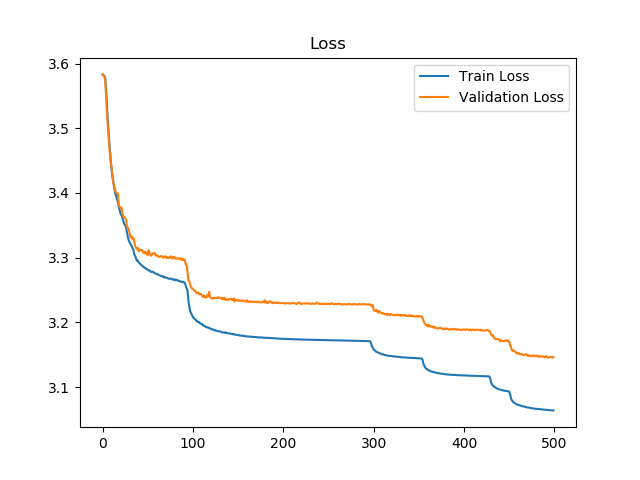
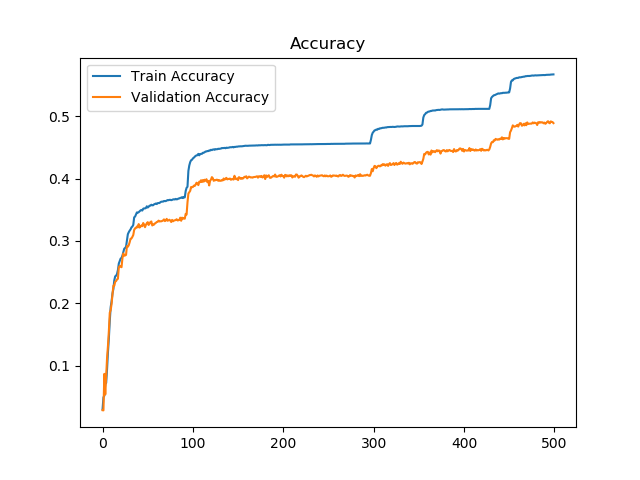
PSNR = 16.348422786979377.

It performs better than autoencoder because PCA aims to minimize reconstruction error.

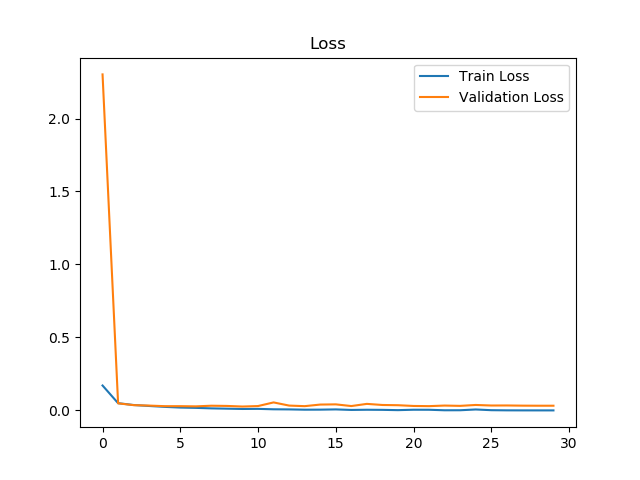
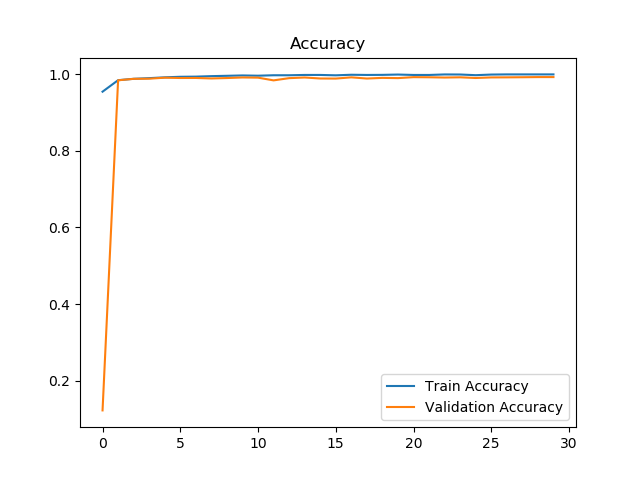
Q 6.4

Autoencoder has 1024\*32+32\*32+32\*32+32\*1024+32+32+32+1024 = 68704 parameters. PCA has only 32\*1024 = 32768 parameters. This means that feedforward neural network has lots of redundancy. These parameters do not necessarily improve the performance.

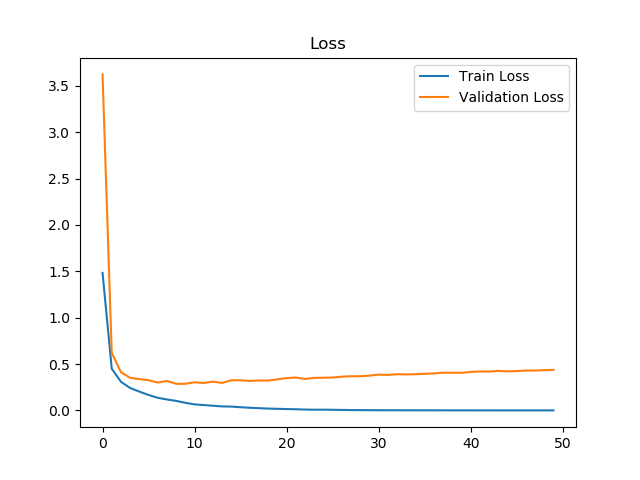
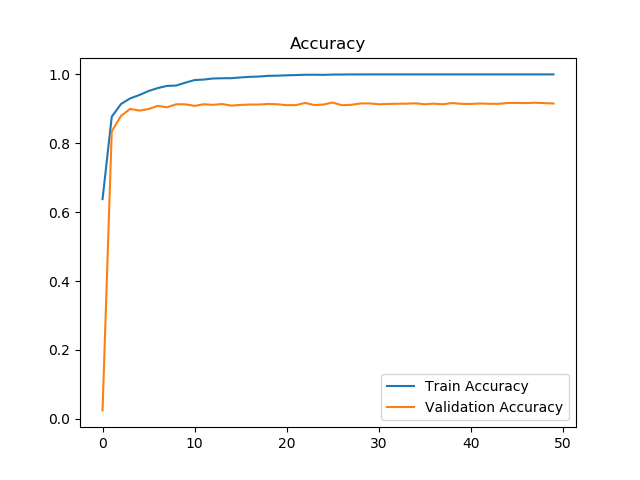
Q 7.1.1



Q 7.1.2



Q 7.1.3



Q 7.1.4

Please see q714.py. Convolutional Neural Network performs better than fully connected network. Here is the output

TO DO LI5T

1 MAKE A T3D0LtST

2 gHECK OFFTHEEIRST

TH5NG0N TODOLF5T

3 RBALtZEY0U HMUEALREAPY

COMPLETED 2THtNGS

4 REWARD YOUR5ELFWFtH

ANAP

A BC D E F G

H IJ K L M W

Q P Q R S T M

V W X Y Z

q 2 3 4 5 G7 8g 0

HAIKUS ARE EA5Y

BUT SOMETFMES THEY D0NT MAKE SENSE

REFRFGERATDR

DE EP LEARNfMG

DEEPER LEARNING

DFEHFST LEARMING

Q 7.2