EE5179: DEEP LEARNING IN IMAGING

ASSIGNMENT 3: RNN

REPORT

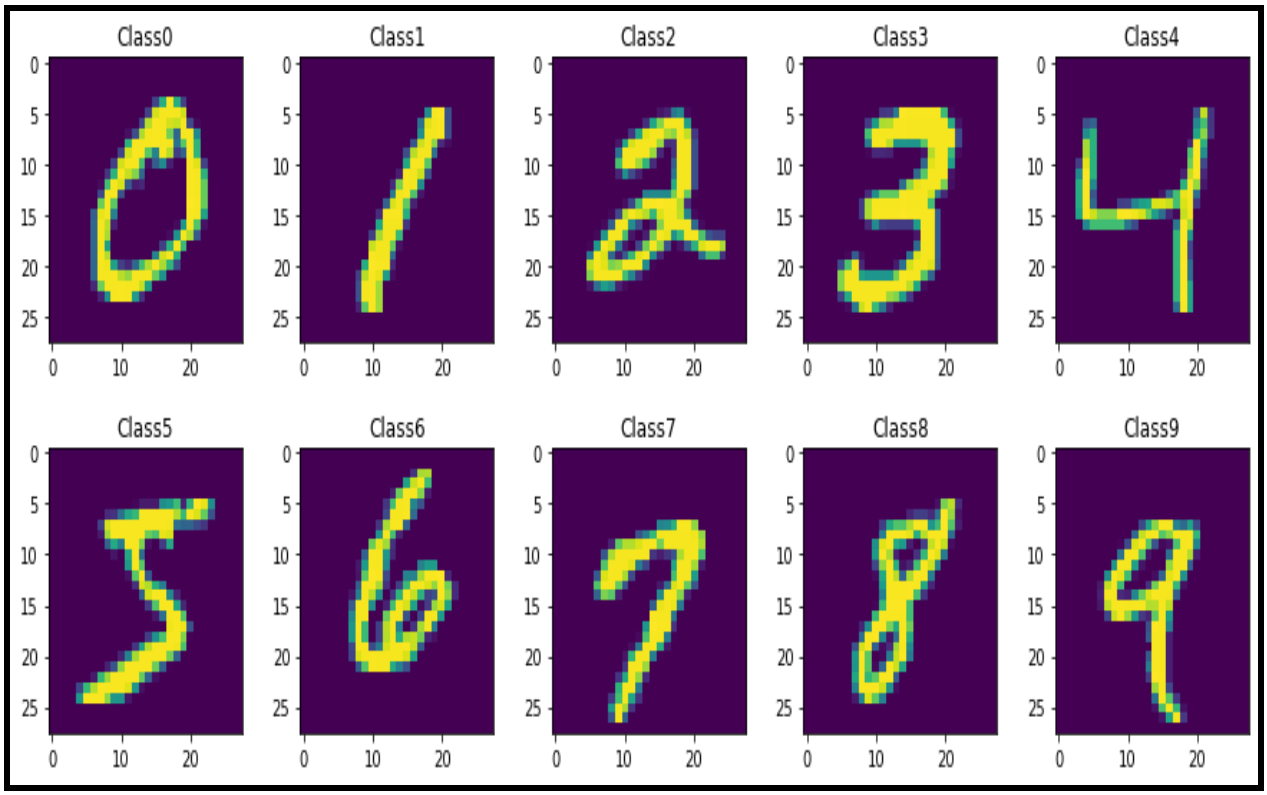
RAVEENA RAI (ED21S006)

Resources used:

1. Google Colab
2. MNIST Dataset of handwritten digits
3. Tutorials provided in the class of the course ‘Deep Learning for Imaging’
4. Online references.

About the dataset:

1. The dataset used in the algorithm is the MNIST Digits Recognition dataset.
2. It comprises handwritten digits, pre-processed to ensure that the digits are centered and size normalized.
3. The train set consists of 60,000 images, and the test set consists of 10,000 images.
4. The training dataset was divided into two sets: training (50000 images) and validation(10000 images).

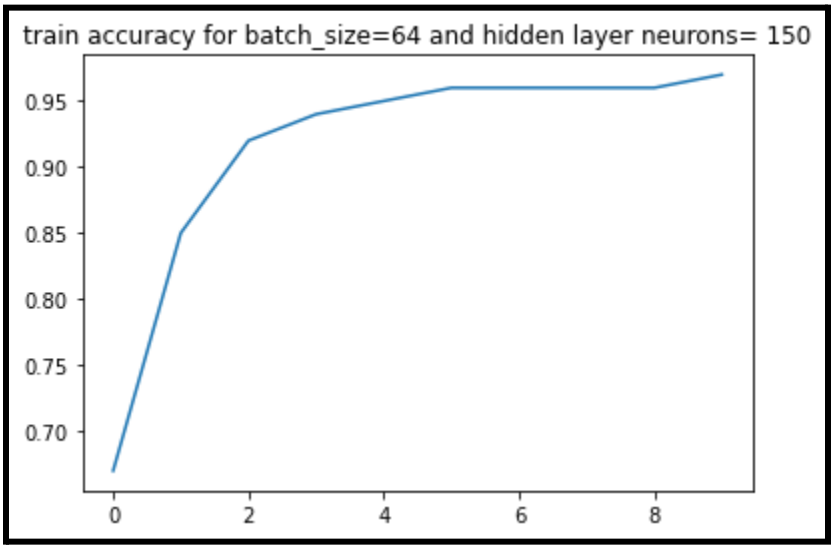

Class0
Class1
Class2
Class3
Class4
Class5
Class6
Class7
Class8
Class9

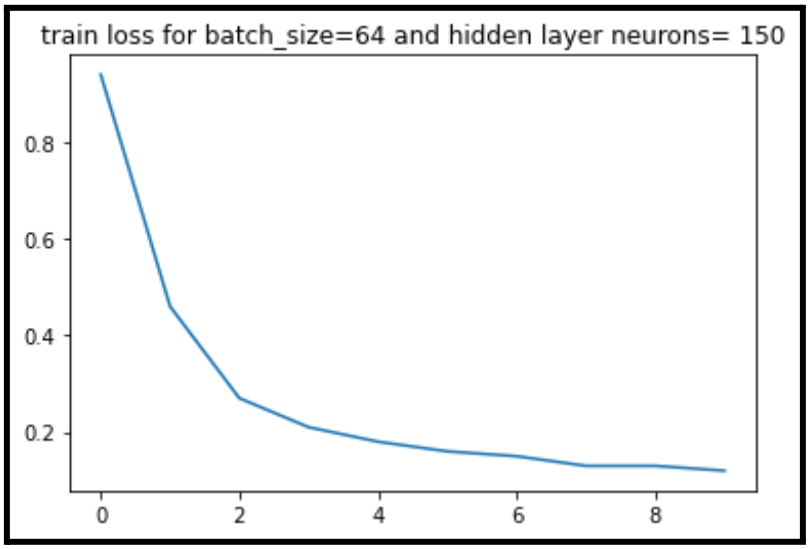
Q-1 :

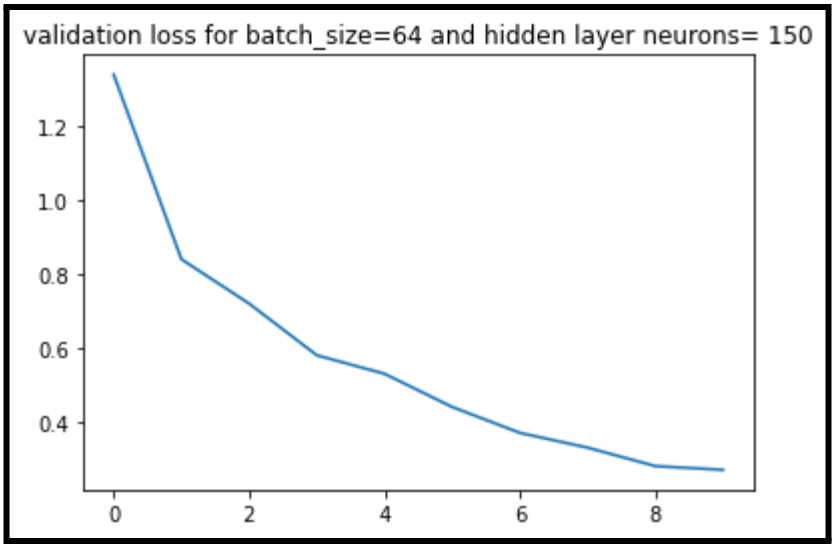
1. Input to the RNN network is a digit image (28 × 28) which will be converted to a sequence to enable the usage of RNN.
2. This is many to one network.
3. Each image will be a sequence of 28 vectors, each of dimension 28 (these can be
4. columns or rows of the image). Hence, the RNN has to be unrolled for 28 steps.
5. Each input vector is fed to the first hidden layer of an RNN in a fully connected manner.
6. The output unit consists of 10 units for each digit.
7. The dataset is divided into training (50000), validation(10000), and test (10000) datasets.
8. The output activation function is softmax to do classification, which will get
9. probabilities of digits belonging to 10 classes.
10. Starting with a network with a hidden state size (state vector size) of 128.
11. Adam optimizer is used.
12. Model is trained for 10 epoch in each case.

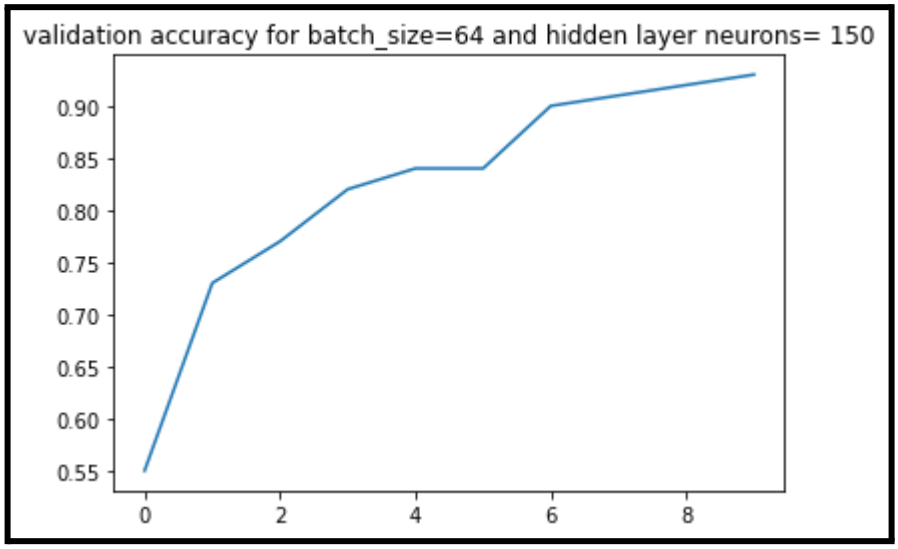
Vanilla RNN:

1. batch_size=64
2. hidden layer neurons= 150
3. Epoch =10
4. training (50000), validation(10000), and test (10000) datasets.





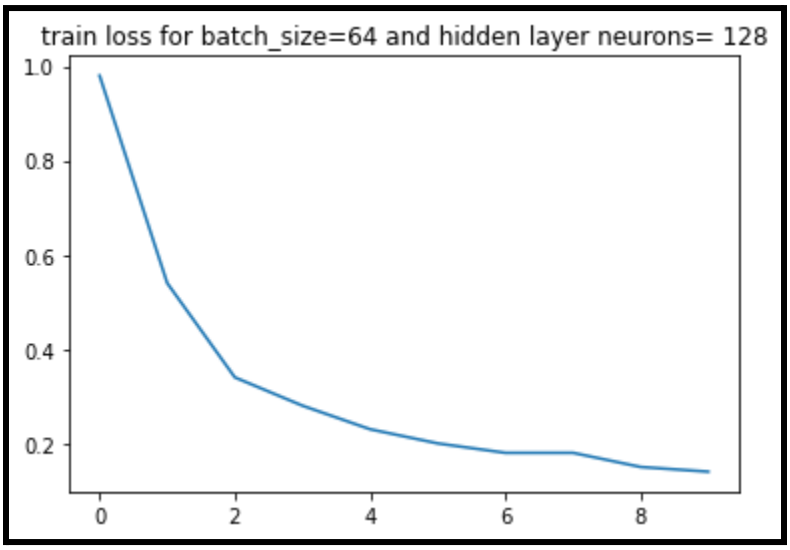


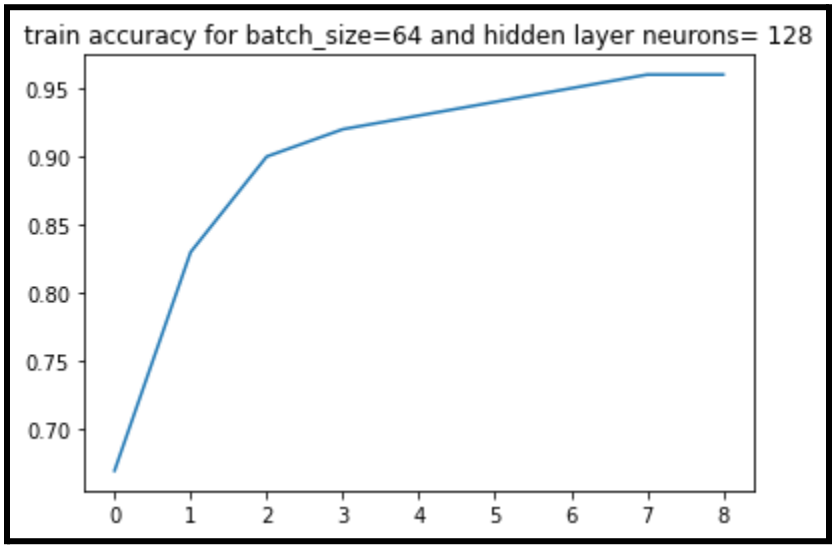


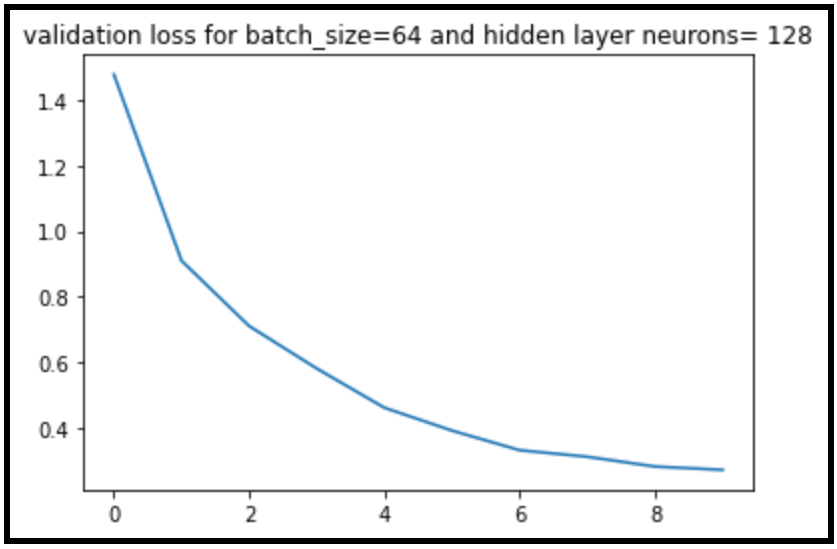
Test accuracy=97.07%

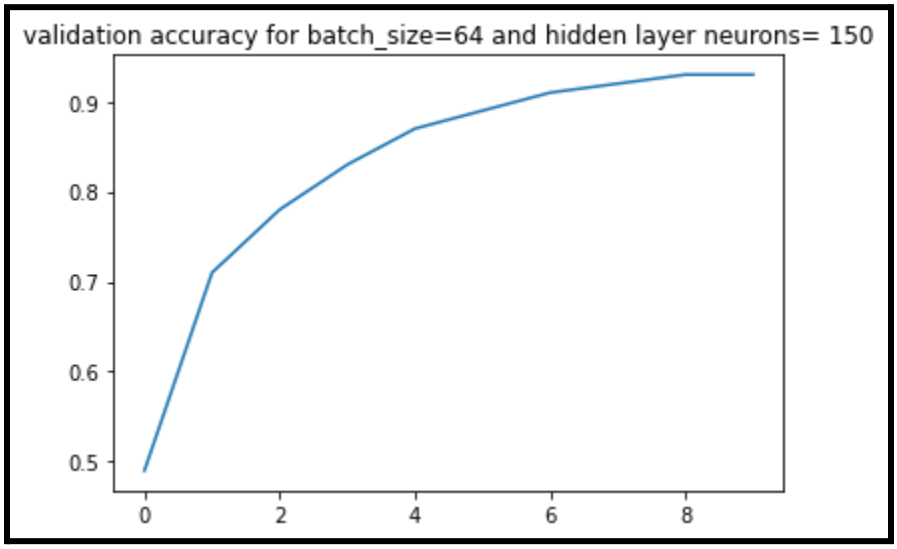
2.

1. batch_size=64
2. hidden layer neurons= 128
3. Epoch =10
4. training (50000), validation(10000), and test (10000) datasets.





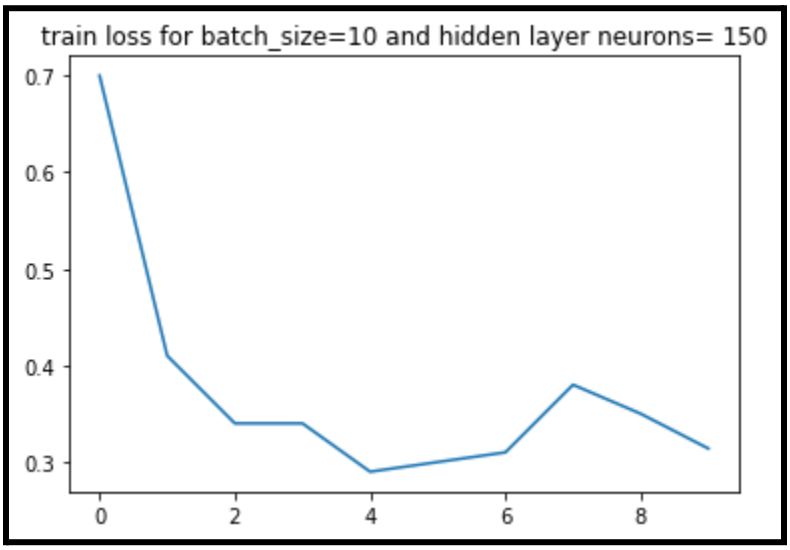


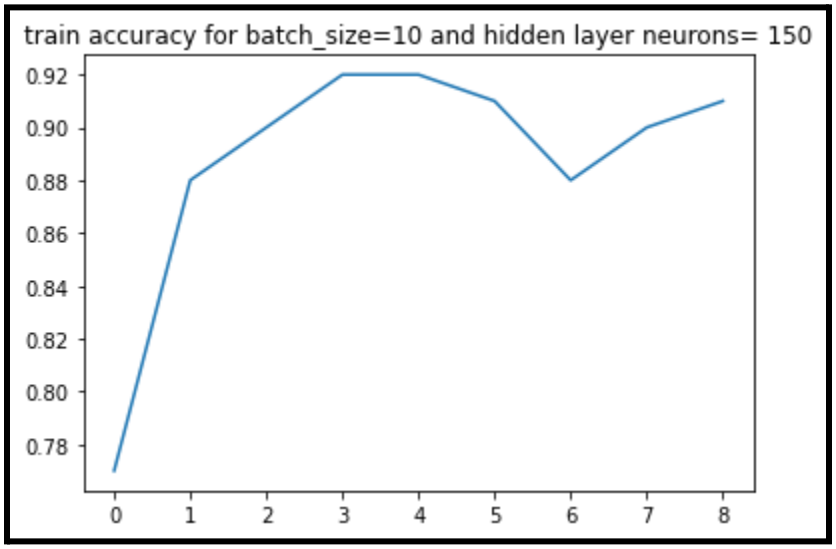


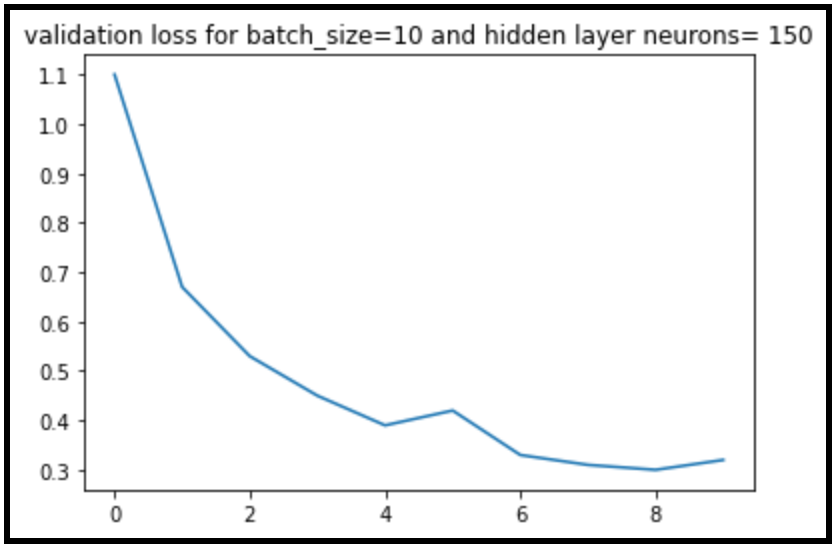
Test accuracy= 96.43%

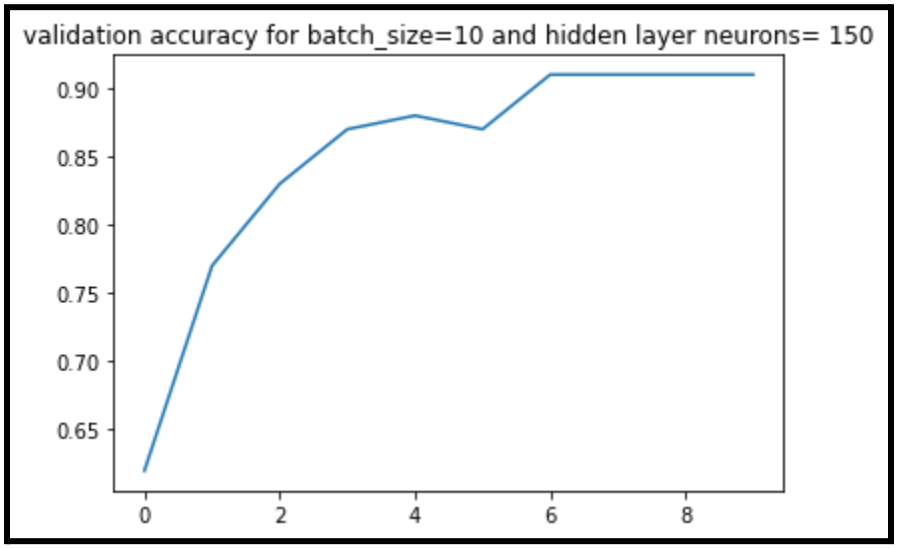
3.

1. Batch size=10
2. Hidden layer neurons= 150
3. epoch=10





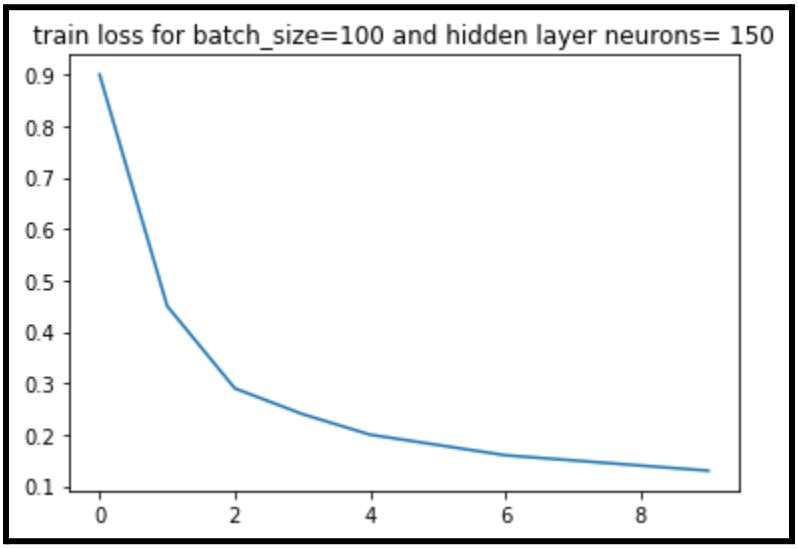


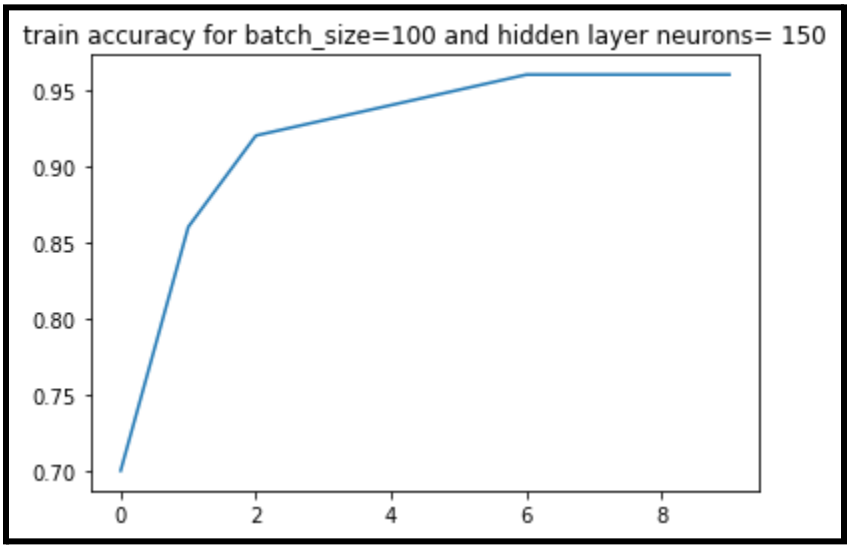


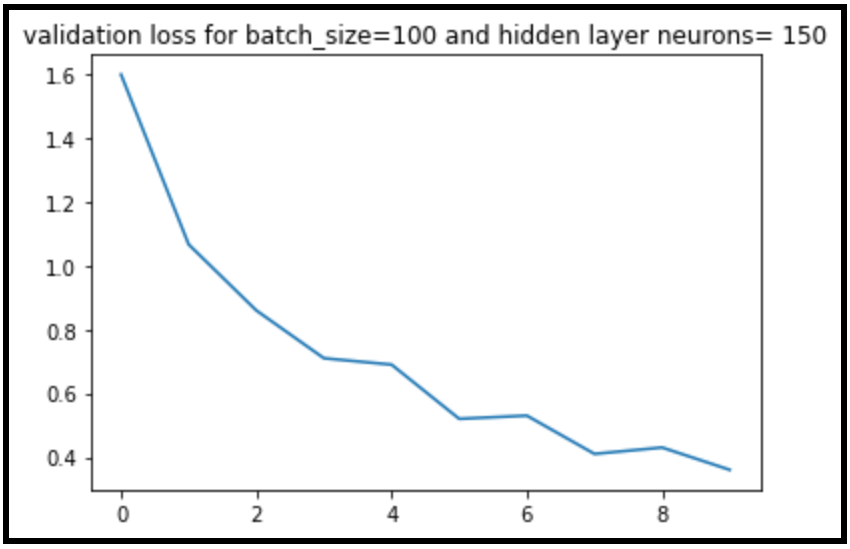
Test accuracy = 91.67%

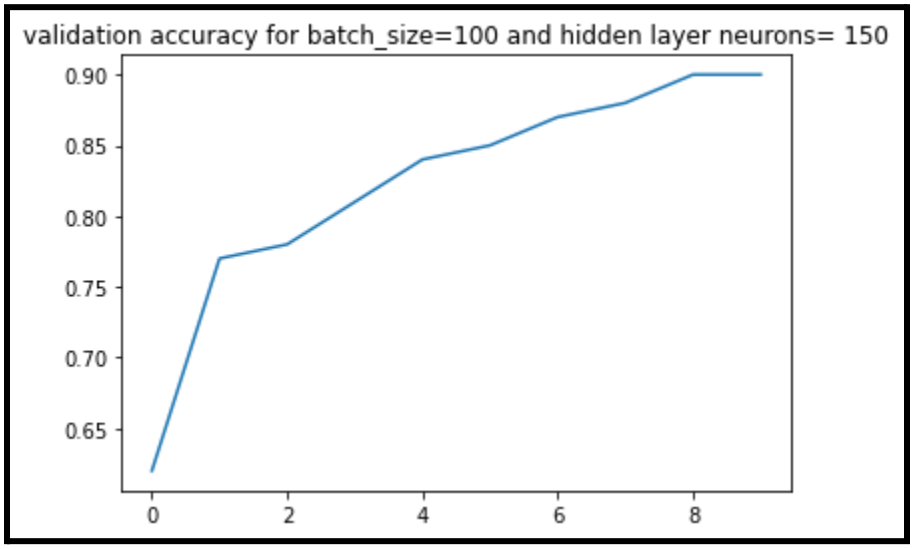
4.

1. Batch size=100
2. Hidden neuron size = 150
3. epoch= 10







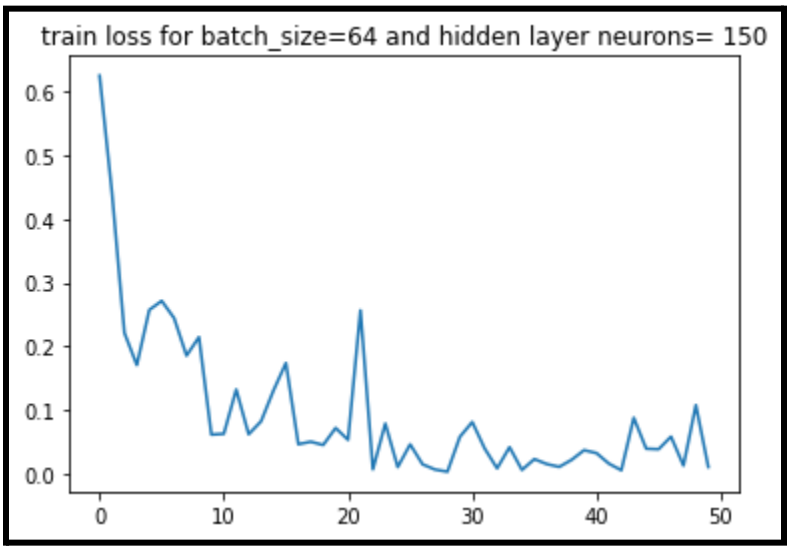


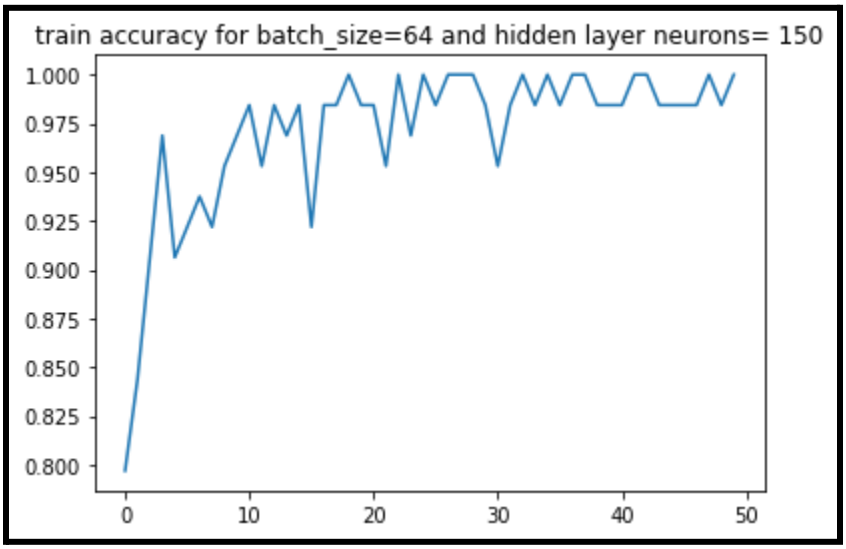
Training accuracy= 96%

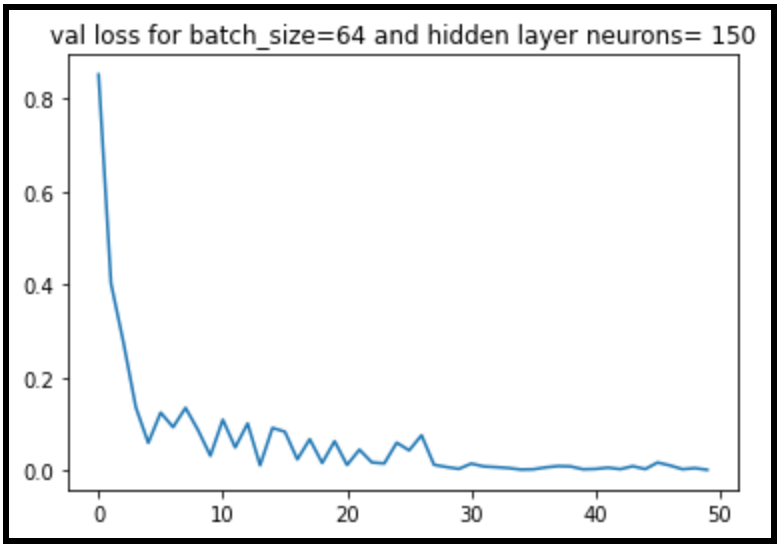
Test accuracy= 95.42%

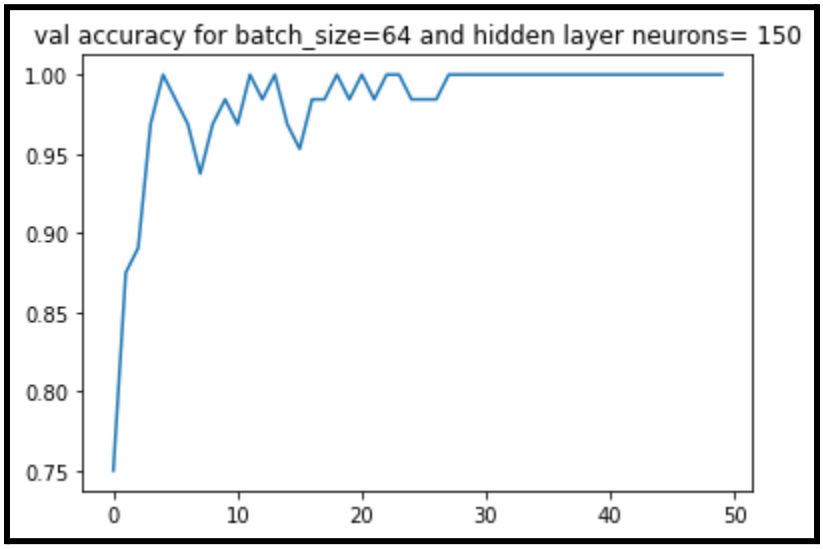
LSTM:

1. batch_size= 64
2. Hidden layer neurons= 150
3. Epoch = 10



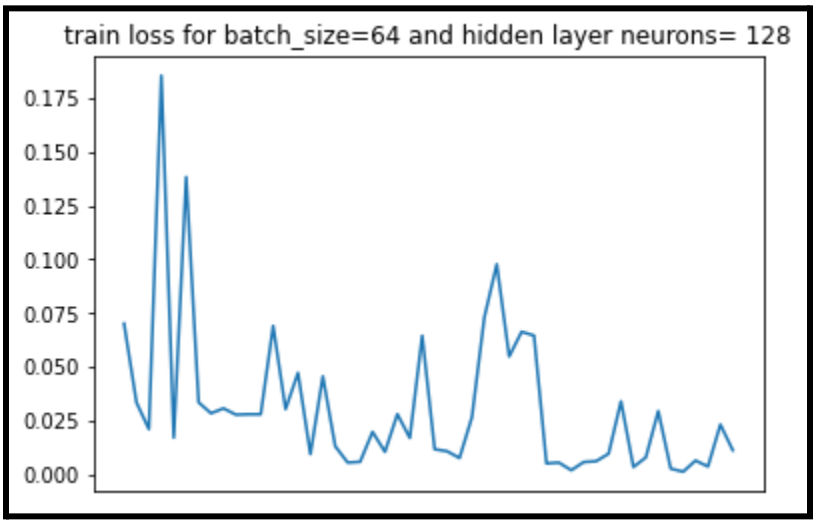


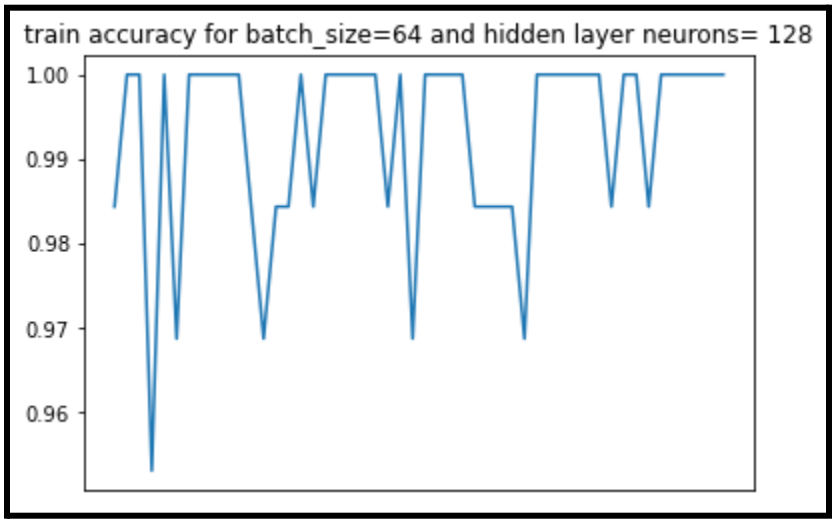


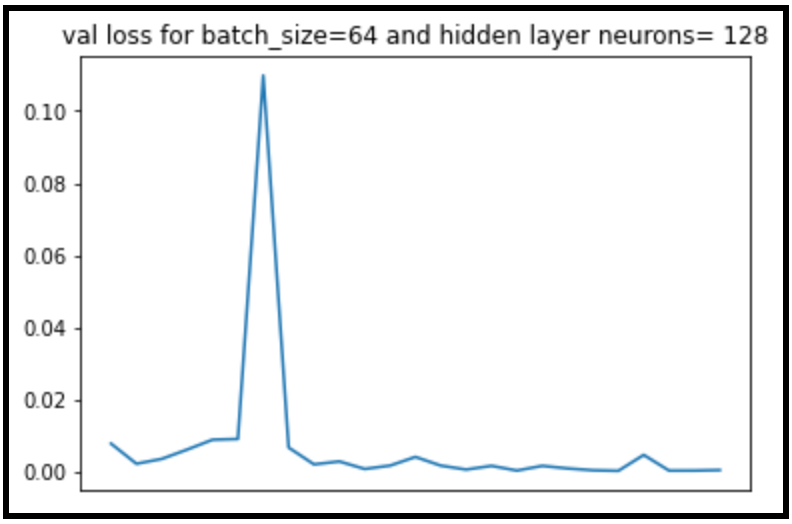


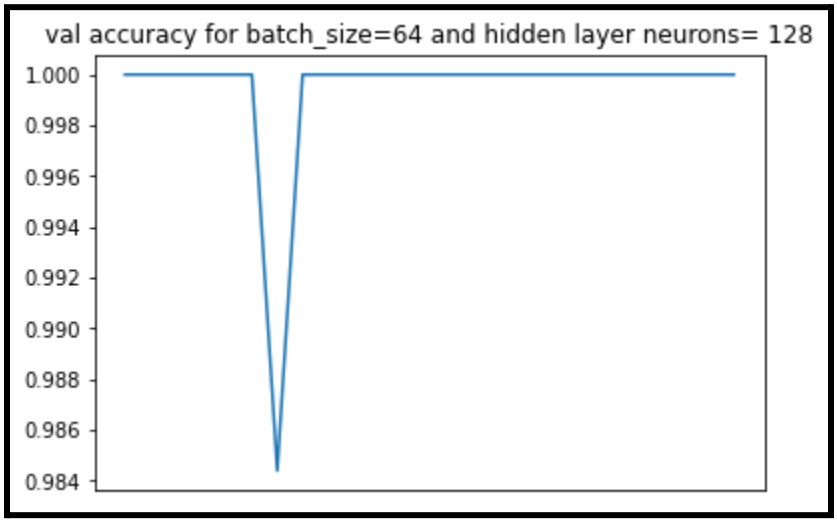
Test accuracy = 98.23%

2.
    1. Batch size= 64
    2. epoch= 10
    3. Number of hidden layer neurons= 128



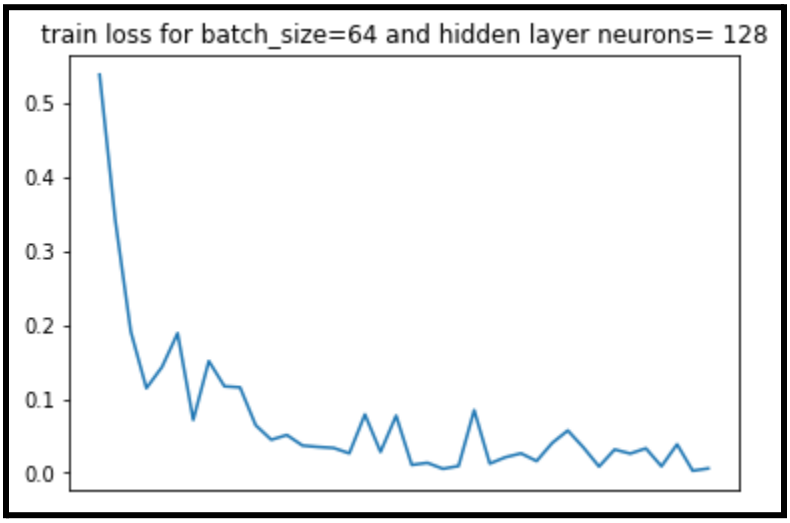


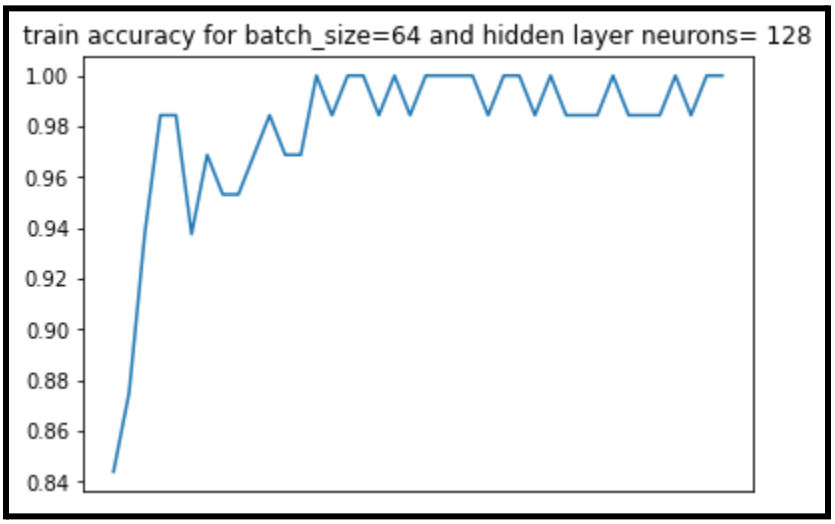




Test accuracy: 98.06%

3.
Batch size=64
Number of hidden layer neurons=150
epoch=20

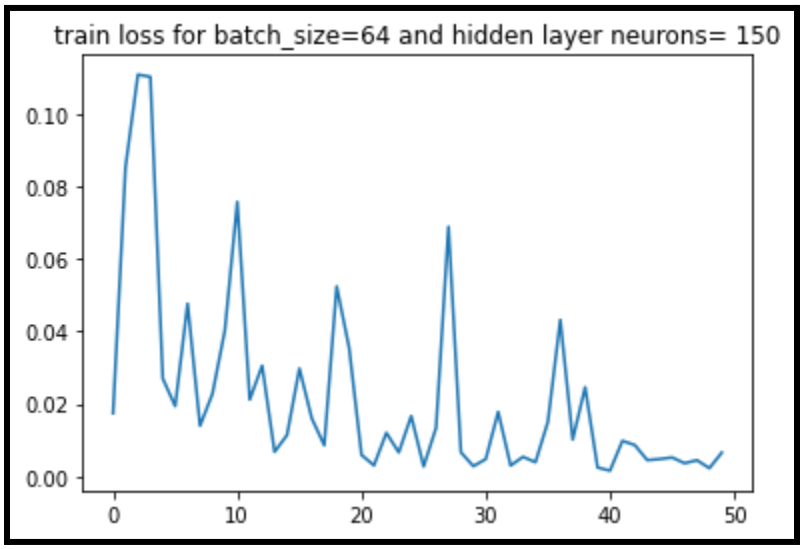


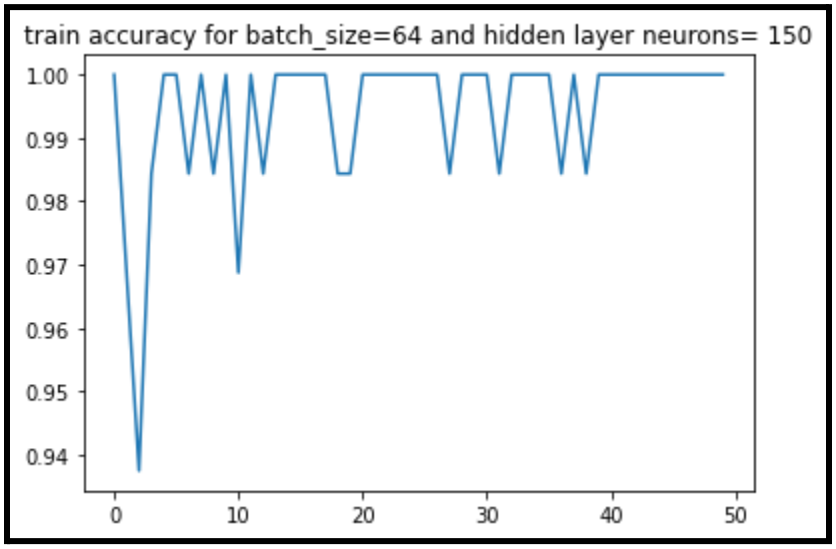


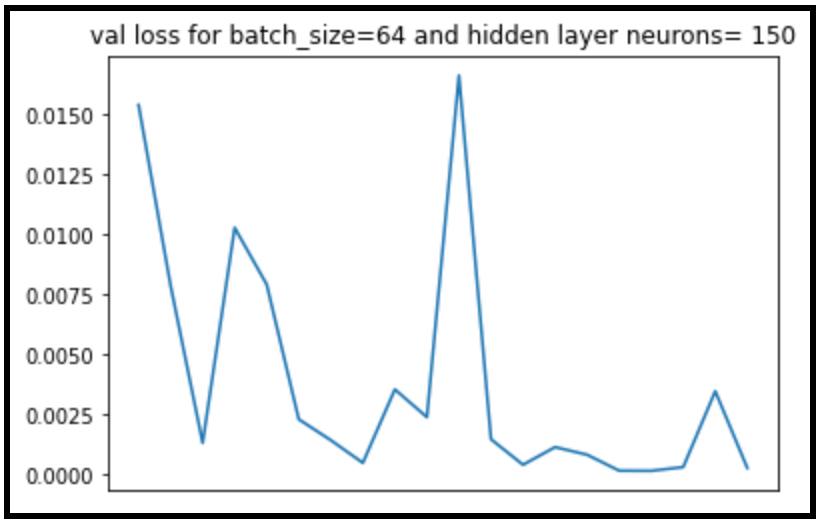
Test accuracy = 97.34%

Bidirectional lstm:

1. Batch size=64, epoch=10, hidden layers neuron number = 150

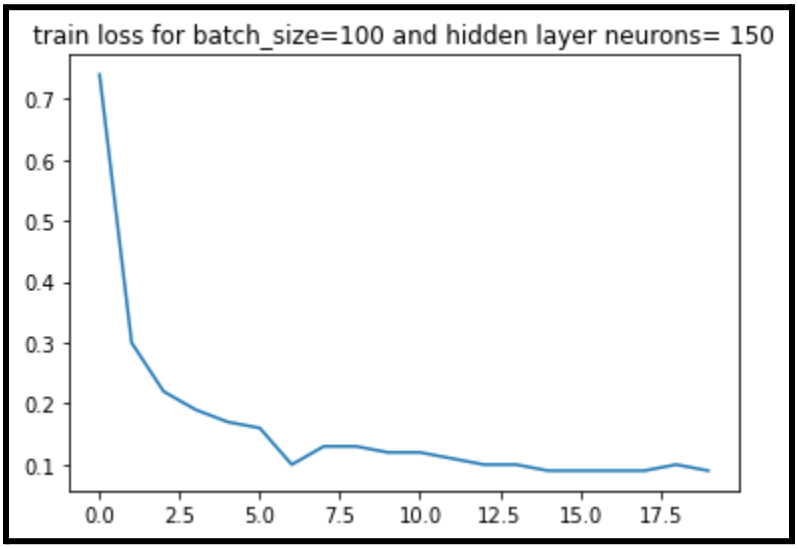


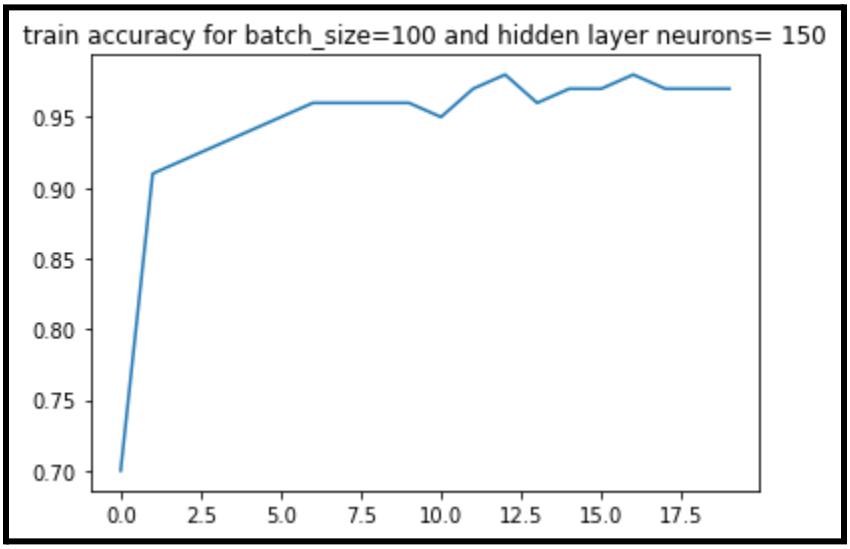


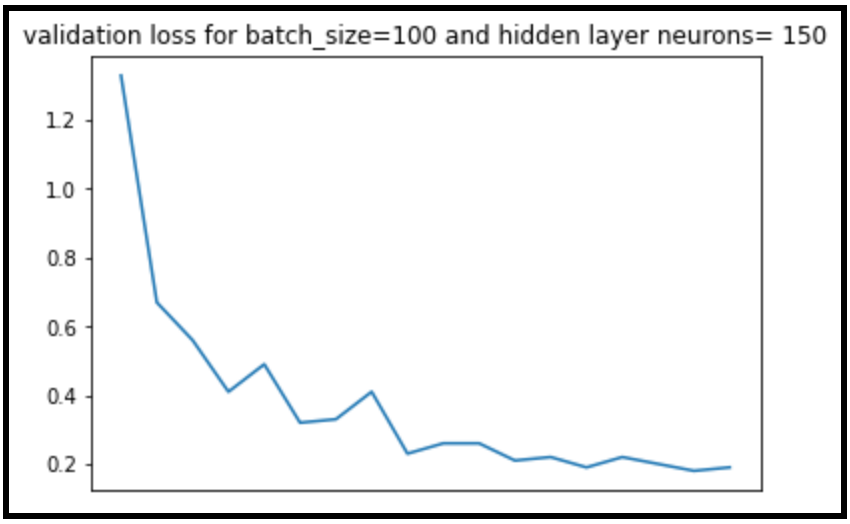


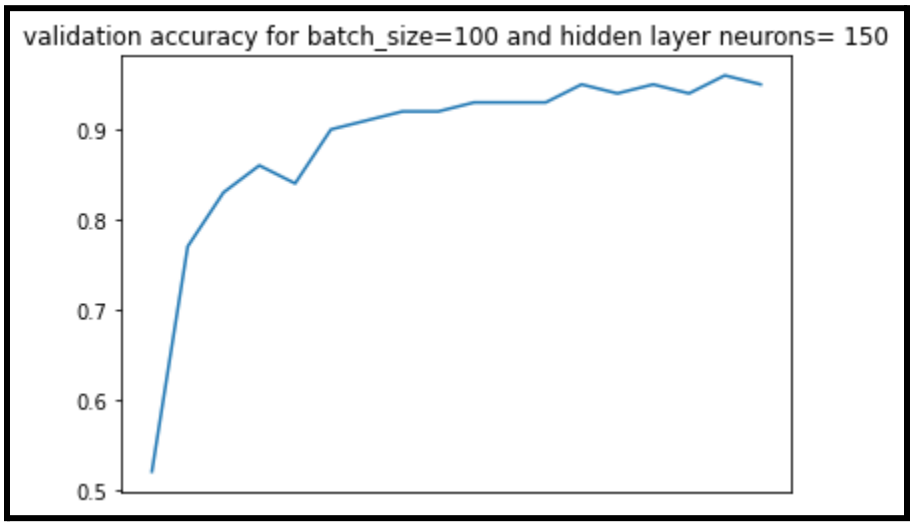
Test accuracy: 98.47%

2.
Batch size=64
Number of hidden layers neuron=150
epoch=20



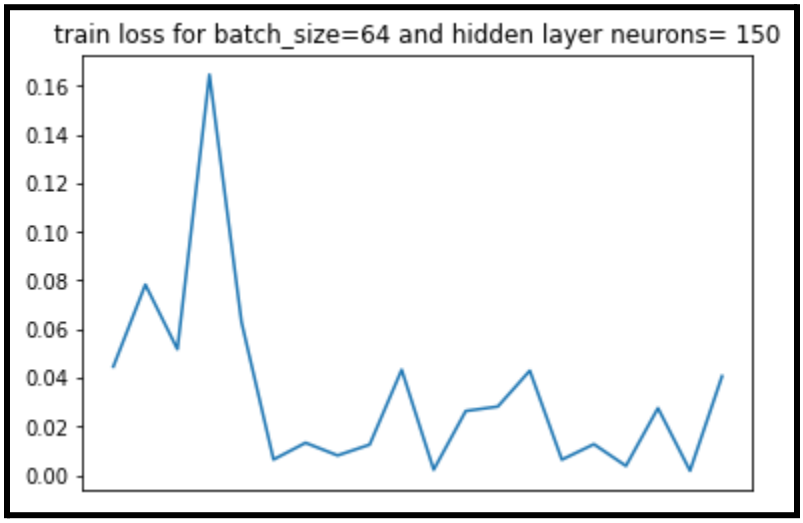


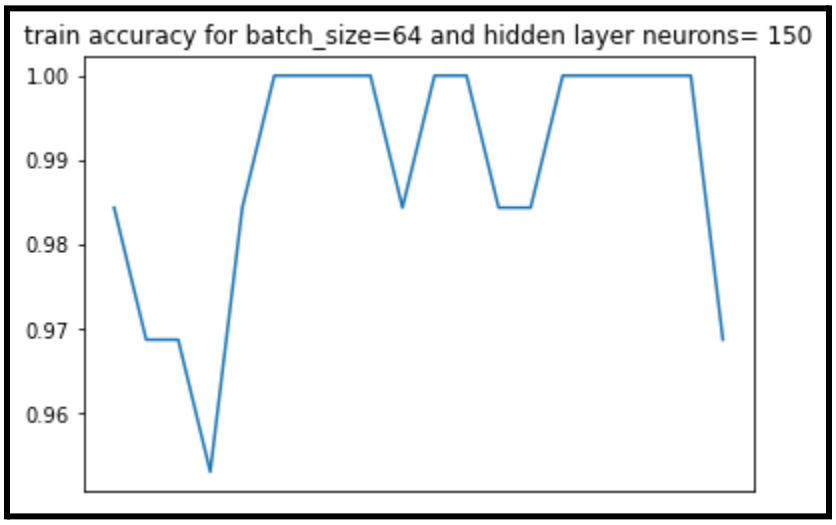


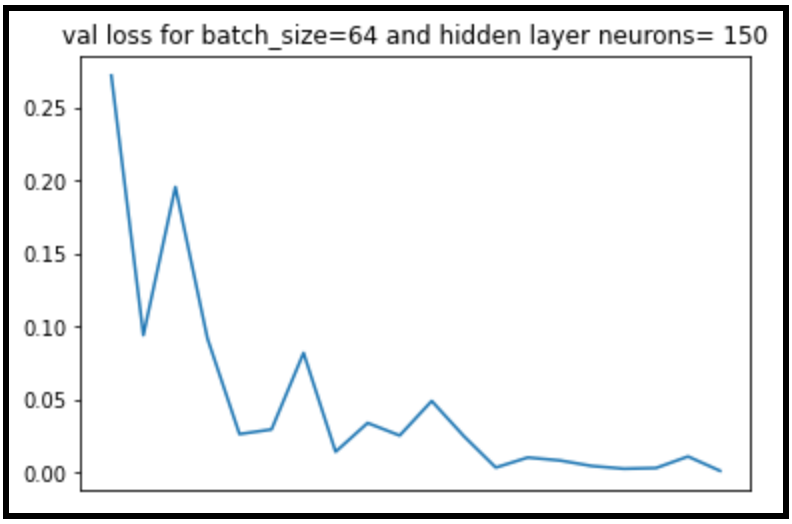


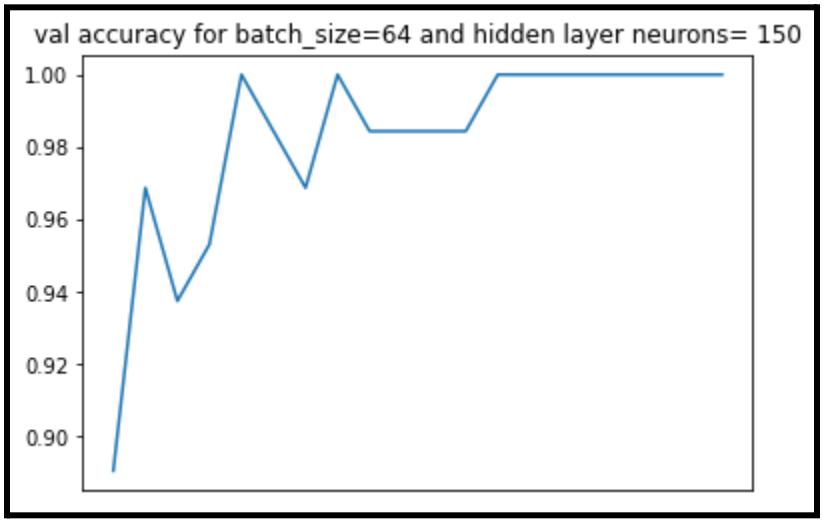
Test accuracy= 97.40%

2.
Batch size = 64
Number of hidden layer neurons= 128
epoch= 10









Test accuracy = 98.28%

Observations:

1. The model using vanilla rnn and different configurations of number of hidden layer neuron and batch size is shown:

	Batch size	Number of Hidden layer neurons	Test accuracy (%)
1.	64	150	97.07
2.	64	128	96.4
3.	10	150	91.6
4.	100	150	95.42

2. With the help of the table, it can be seen that increase in batch size has a positive impact on the test accuracy whereas among the two values of number of hidden layer neurons tried, a higher value seems to be beneficial.
3. Increasing the number of epoch do not seem to affect the accuracy much.

4. The model using lstm and different configurations of number of hidden layer neuron and batch size is shown:

	Batch size	Number of Hidden layer neurons	Test accuracy (%)
1.	64	150 (epoch=10)	98.43
2.	64	128 (epoch=10)	98.06
3.	64	150(epoch=20)	97.34

5. Lstm has better accuracy than vanilla rnn for the same epoch.
6. More neurons in the hidden layer are seen to be giving better results.

7. The model using bidirectional lstm and different configurations of number of hidden layer neuron and batch size is shown:

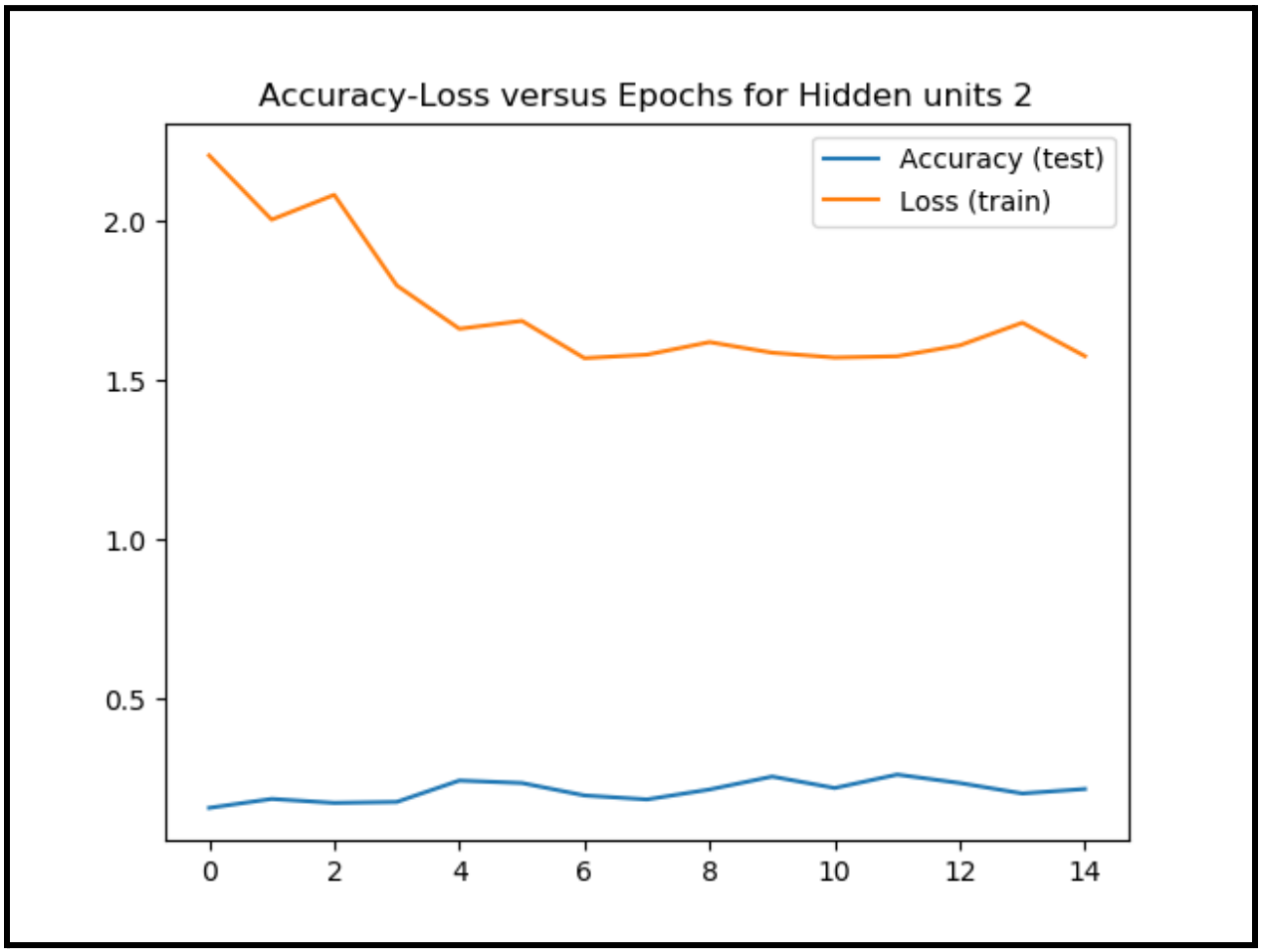
	Batch size	Number of Hidden layer neurons	Test accuracy (%)
1.	64	150 (epoch=10)	98.47
2.	64	128 (epoch=10)	98.28
3.	64	150 (epoch=20)	97.40

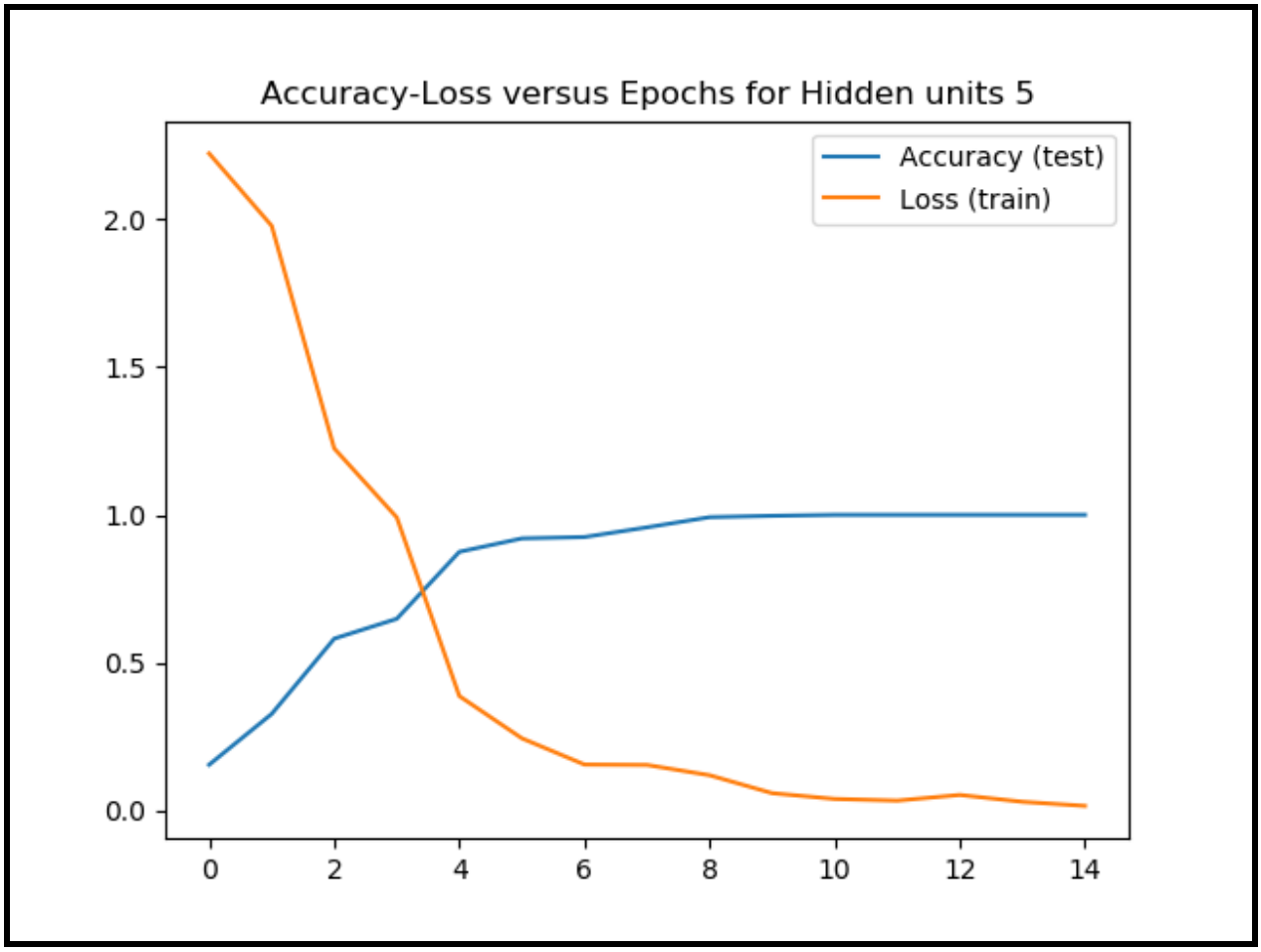
8. In a nutshell, it can be seen that number of hidden layer neurons has a positive correlation with the test accuracy and from the results of vanilla rnn, it can be said that batch size also has a positive correlation with test accuracy.
9. Number of epoch seems to have a negative correlation with accuracy.

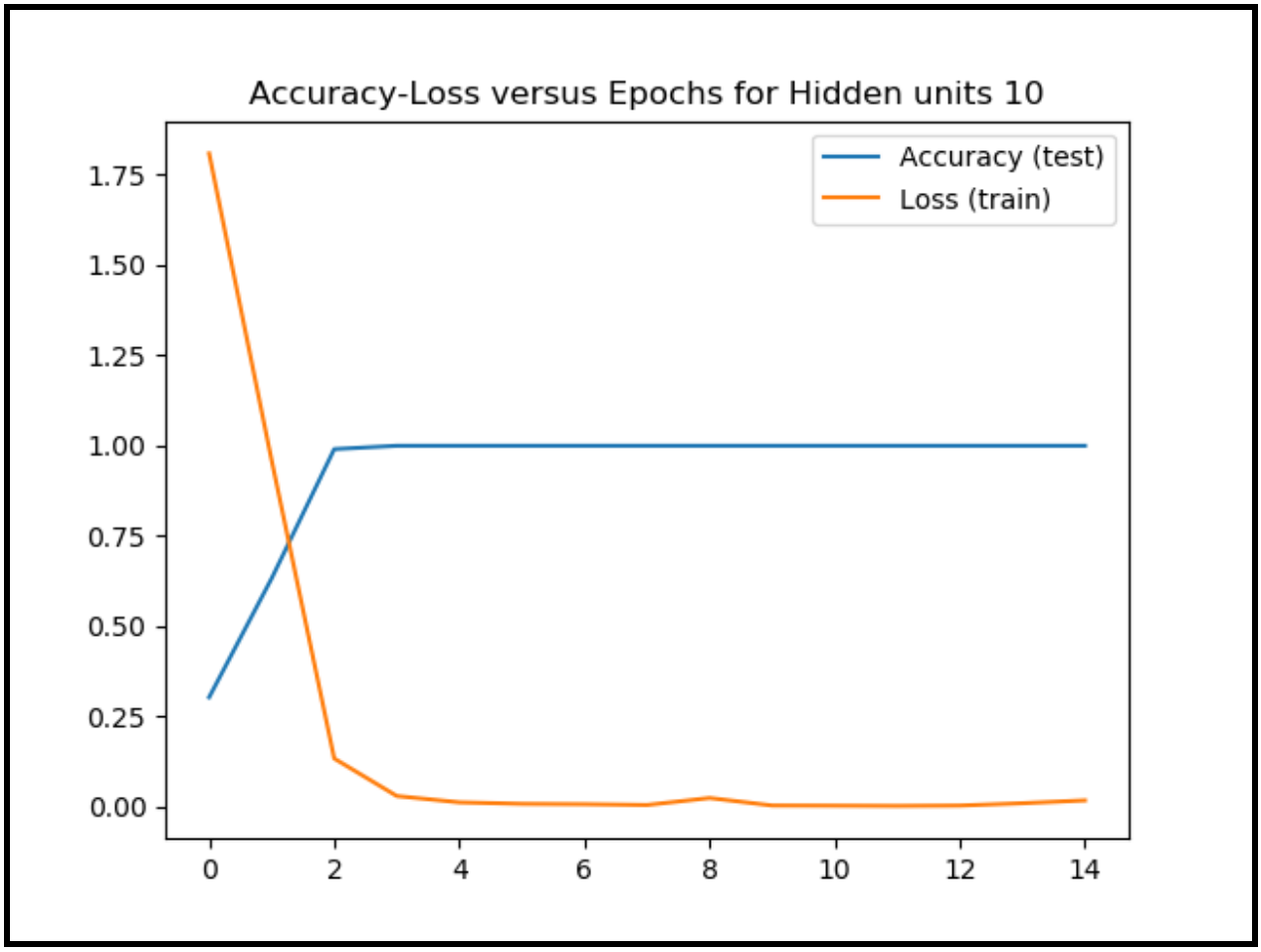
Q-2:

1. Increasing the hidden unit size increases the test accuracy due to better representation power.
2. The performance with hidden layer size of 10 performs better.
3. randomized testing and the results are shown as follows:

Input Sequence tensor([[2, 6, 5]], dtype=torch.int32)
Truth tensor([6])
Prediction 6




Accuracy-Loss versus Epochs for Hidden units 5
Accuracy (test)
Loss (train)
2.0
1.5
1.0
0.5
0.0
0
2
4
6
8
10
12
14


Accuracy-Loss versus Epochs for Hidden units 10
Accuracy (test)
Loss (train)
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
2
4
6
8
10
12
14

Q-3:

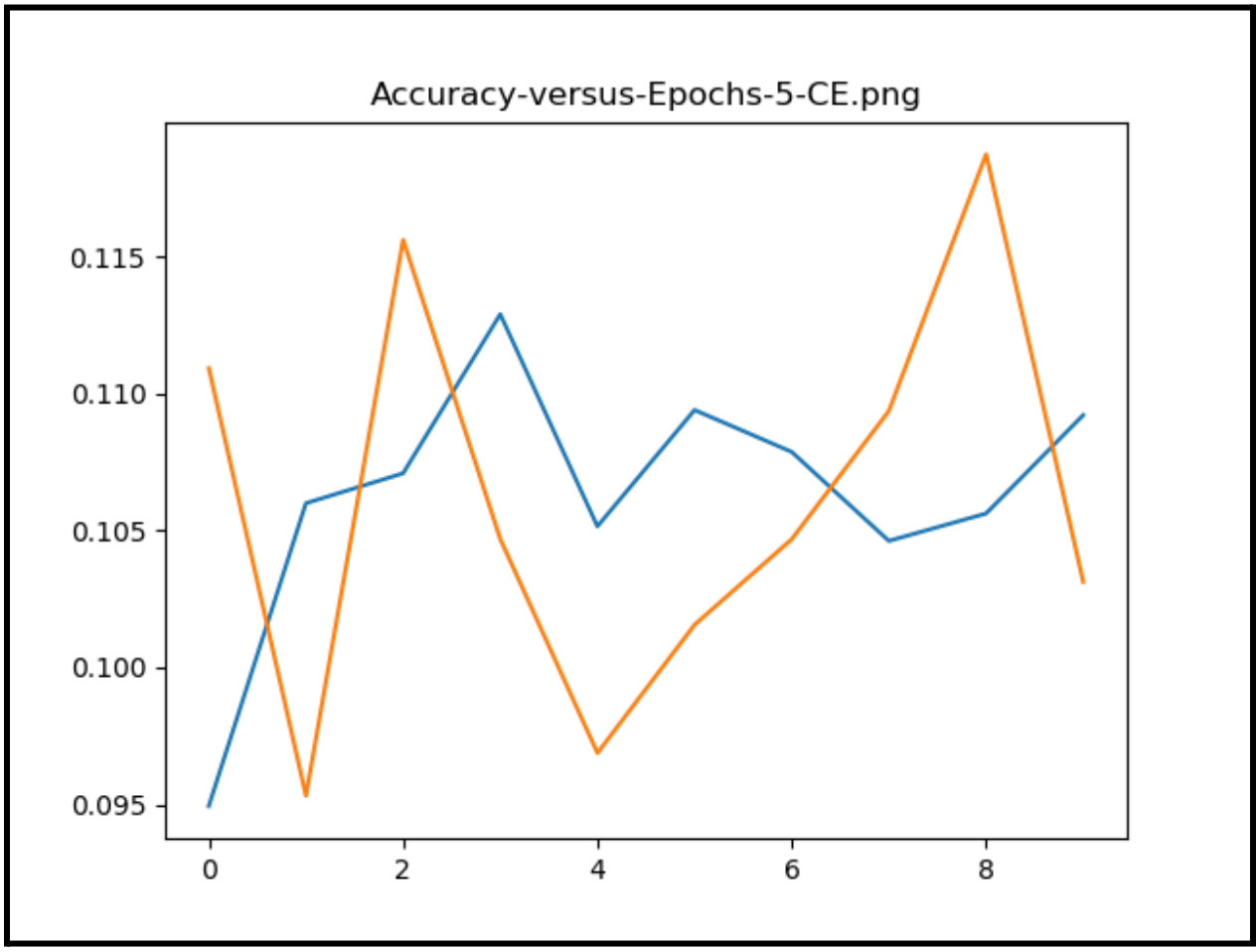

Accuracy-versus-Epochs-5-CE.png
0.115
0.110
0.105
0.100
0.095
0
2
4
6
8

Fig: a: L= 5; ce

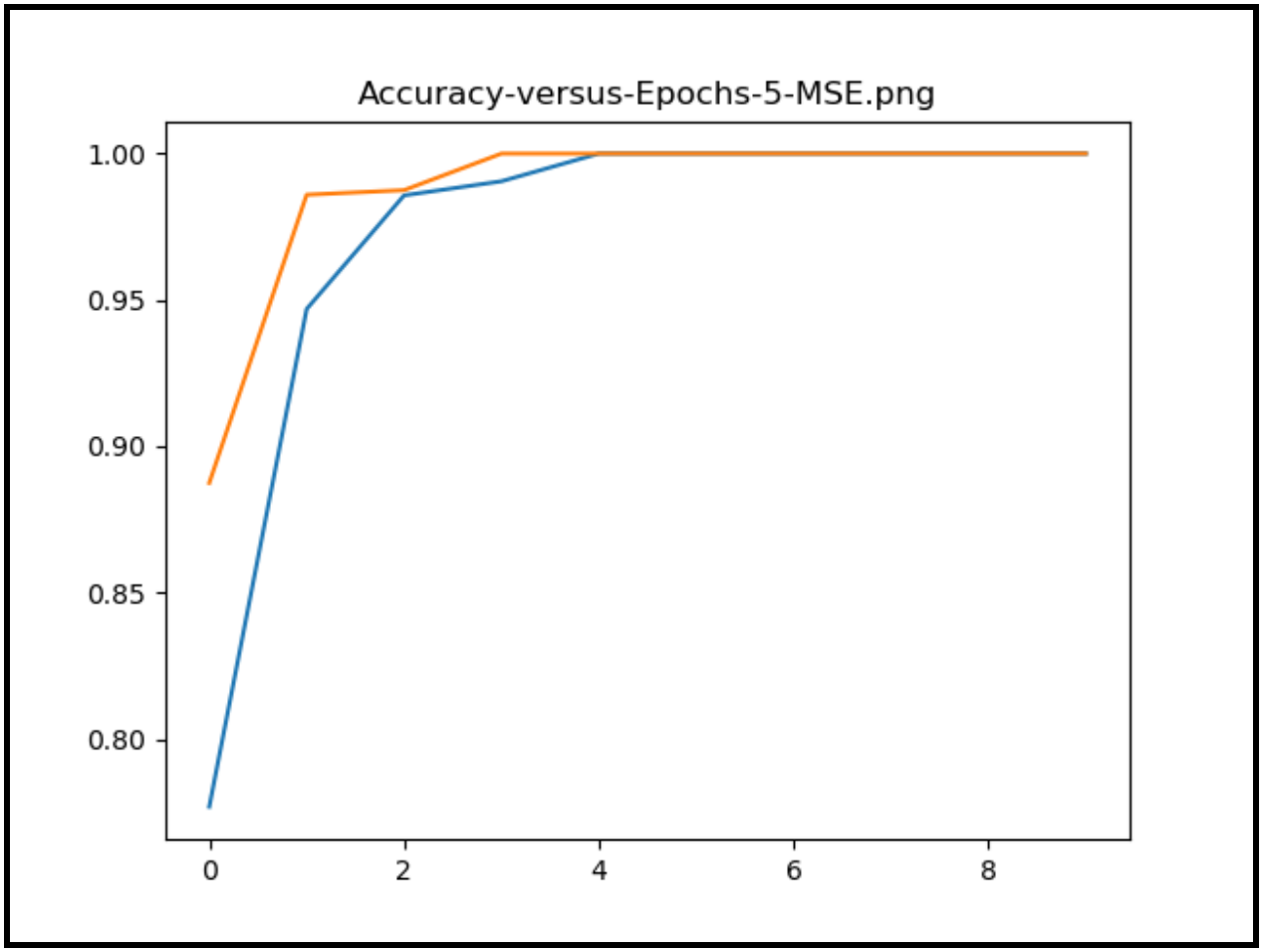
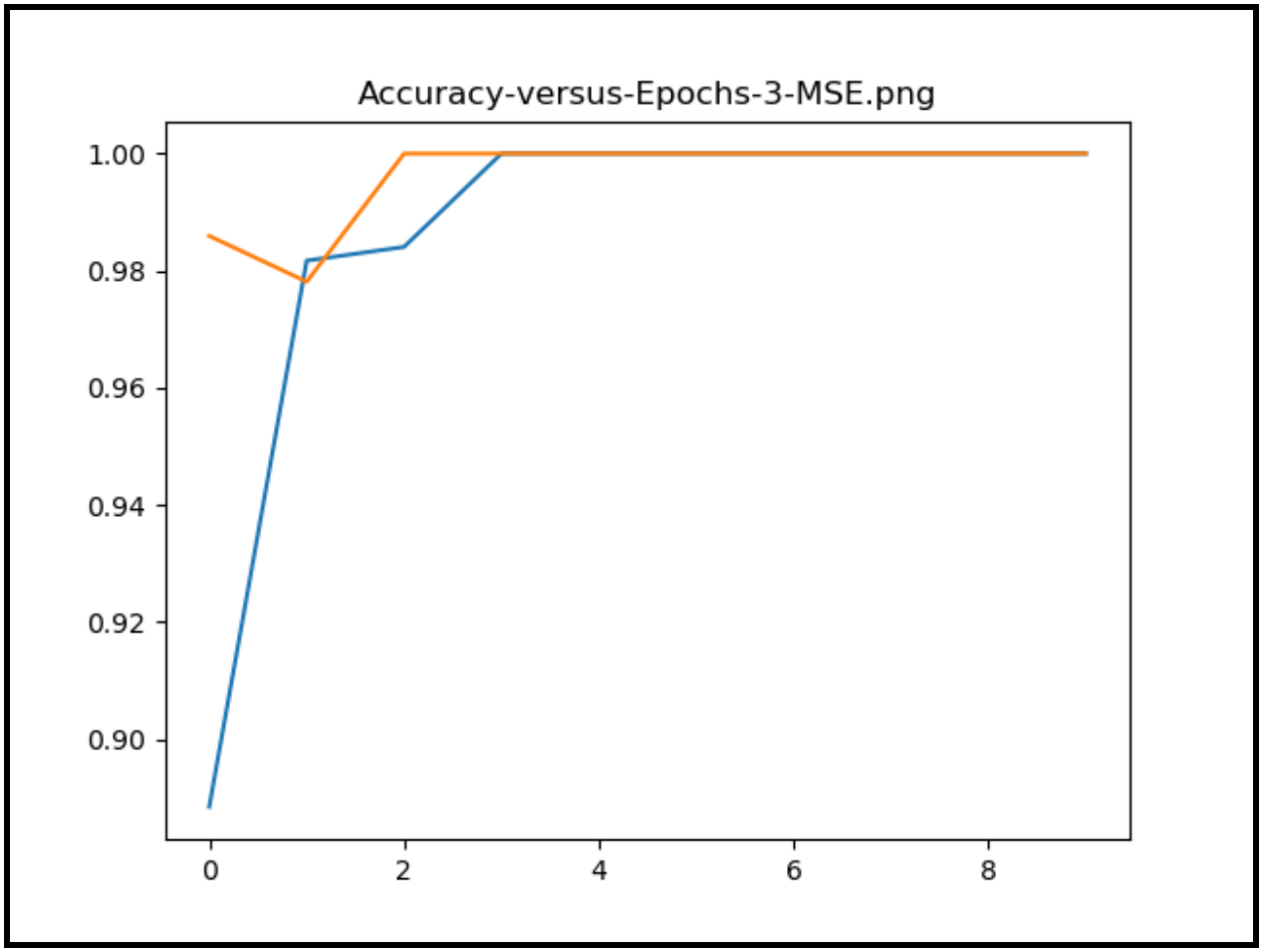
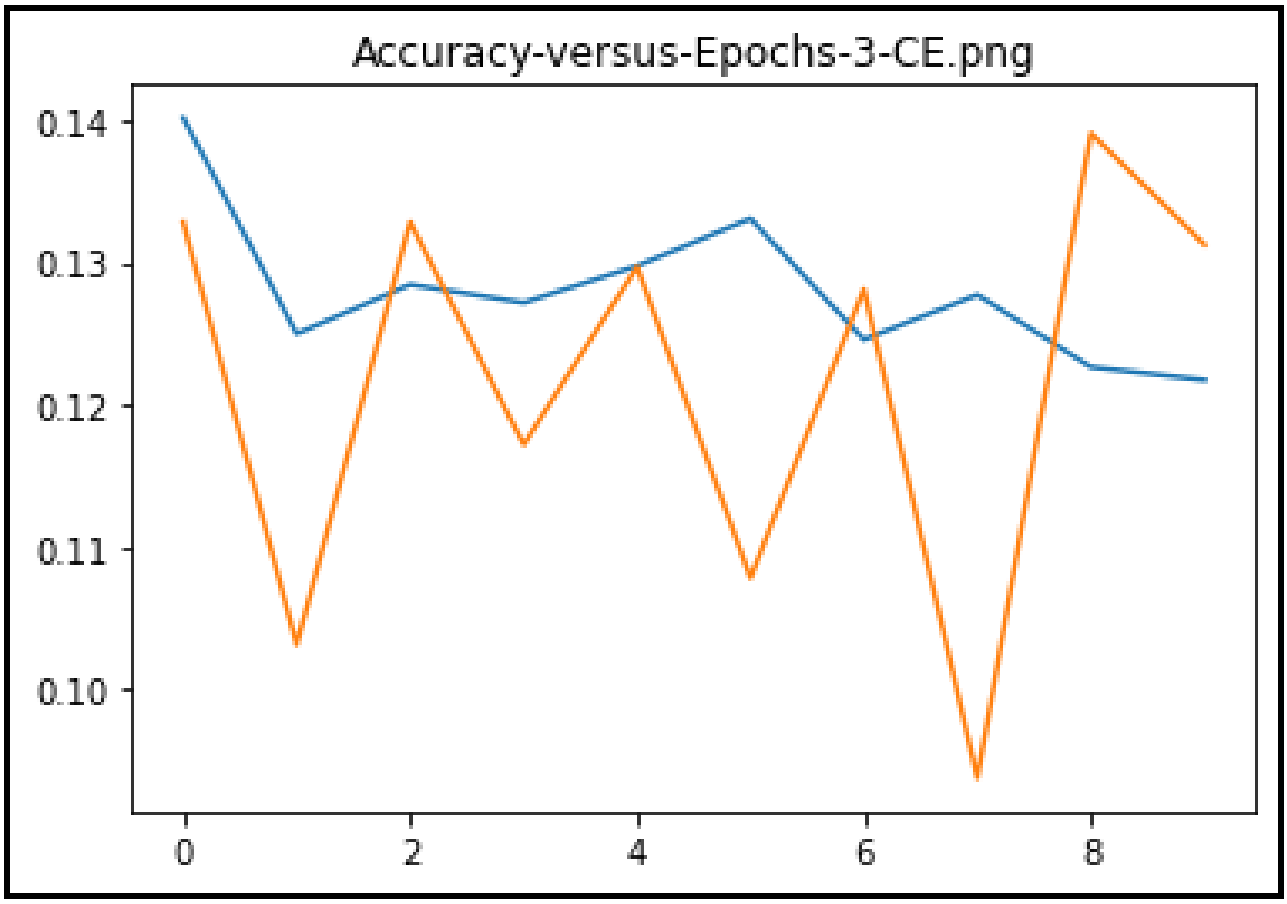


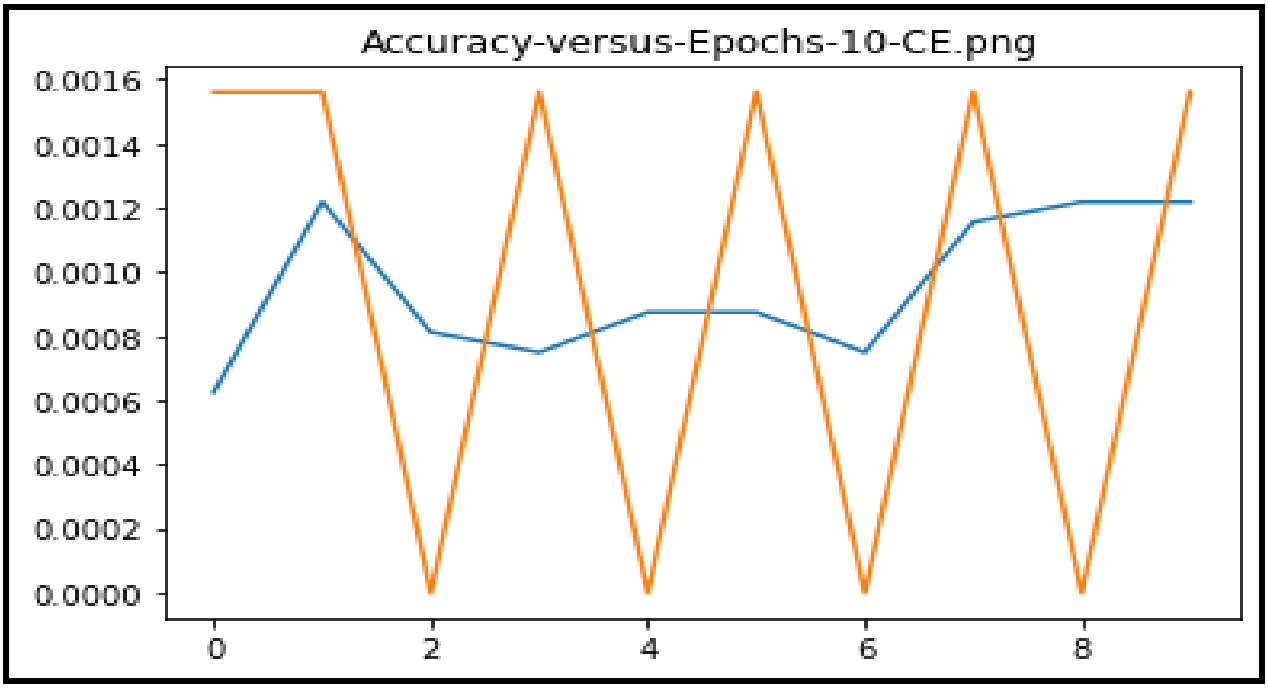
Fig: L= 5; mse

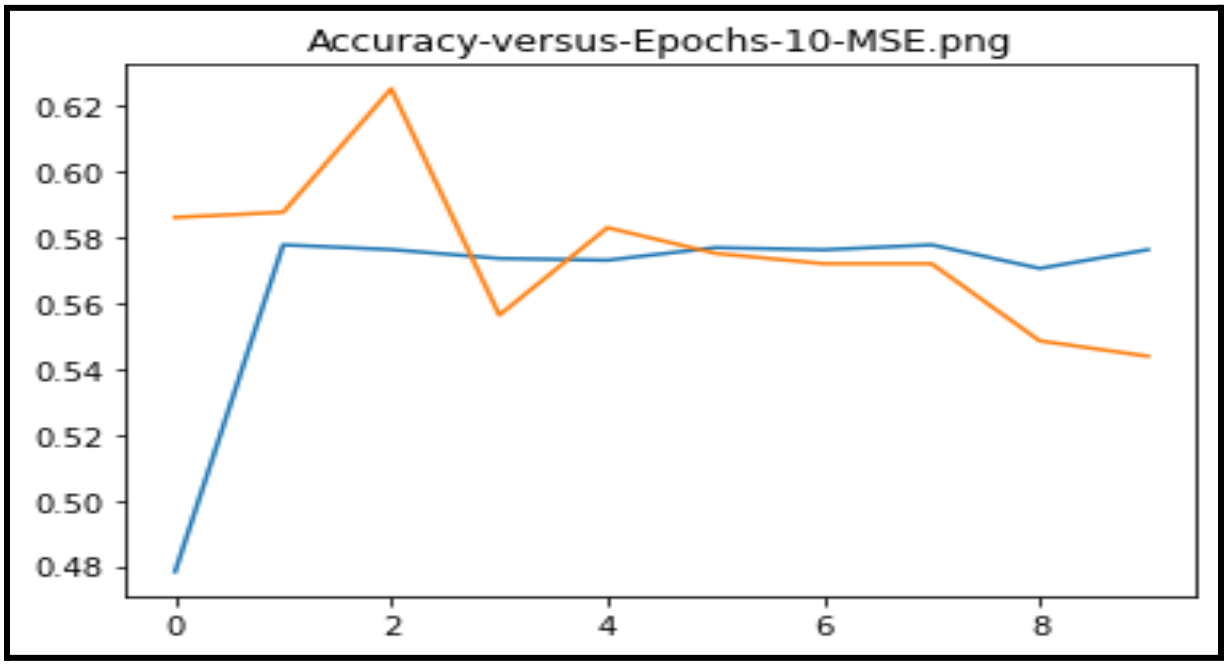

Accuracy-versus-Epochs-3-MSE.png
1.00
0.98
0.96
0.94
0.92
0.90
0
2
4
6
8

L= 3; mse

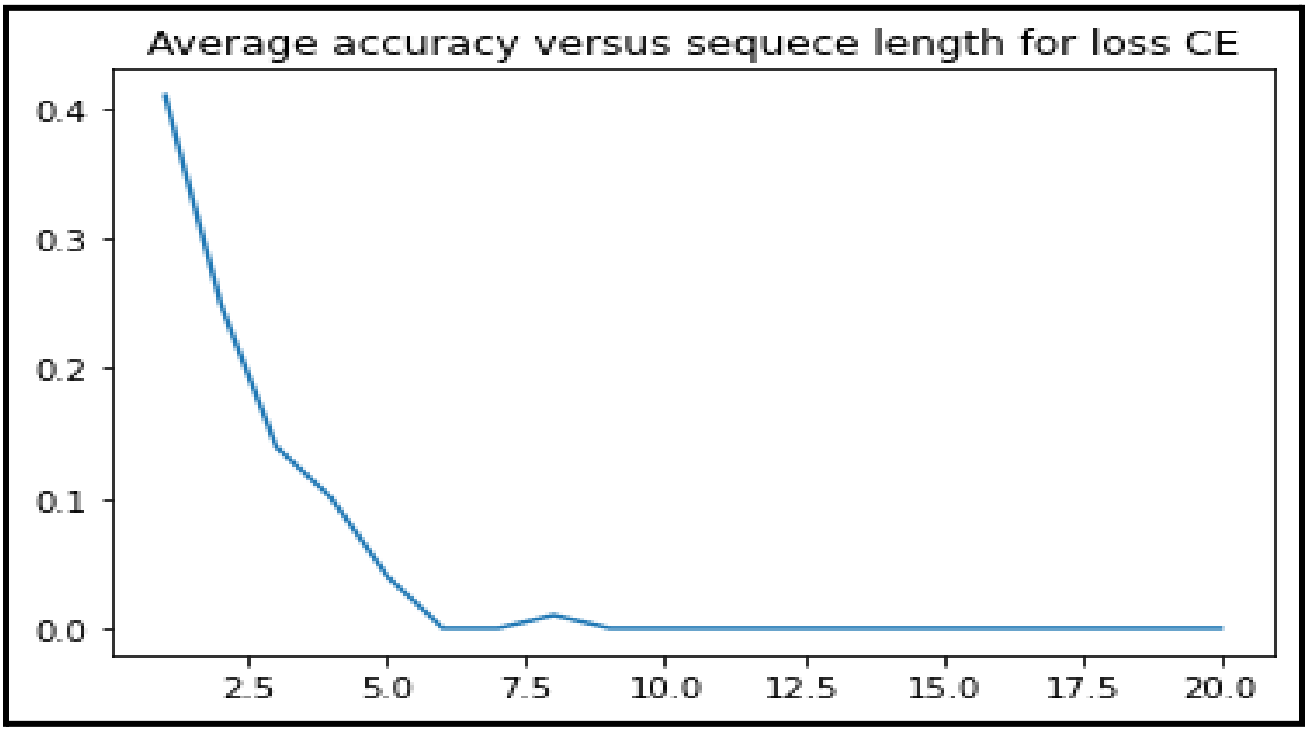

Accuracy-versus-Epochs-3-CE.png
0.14
0.13
0.12
0.11
0.10
0
2
4
6
8

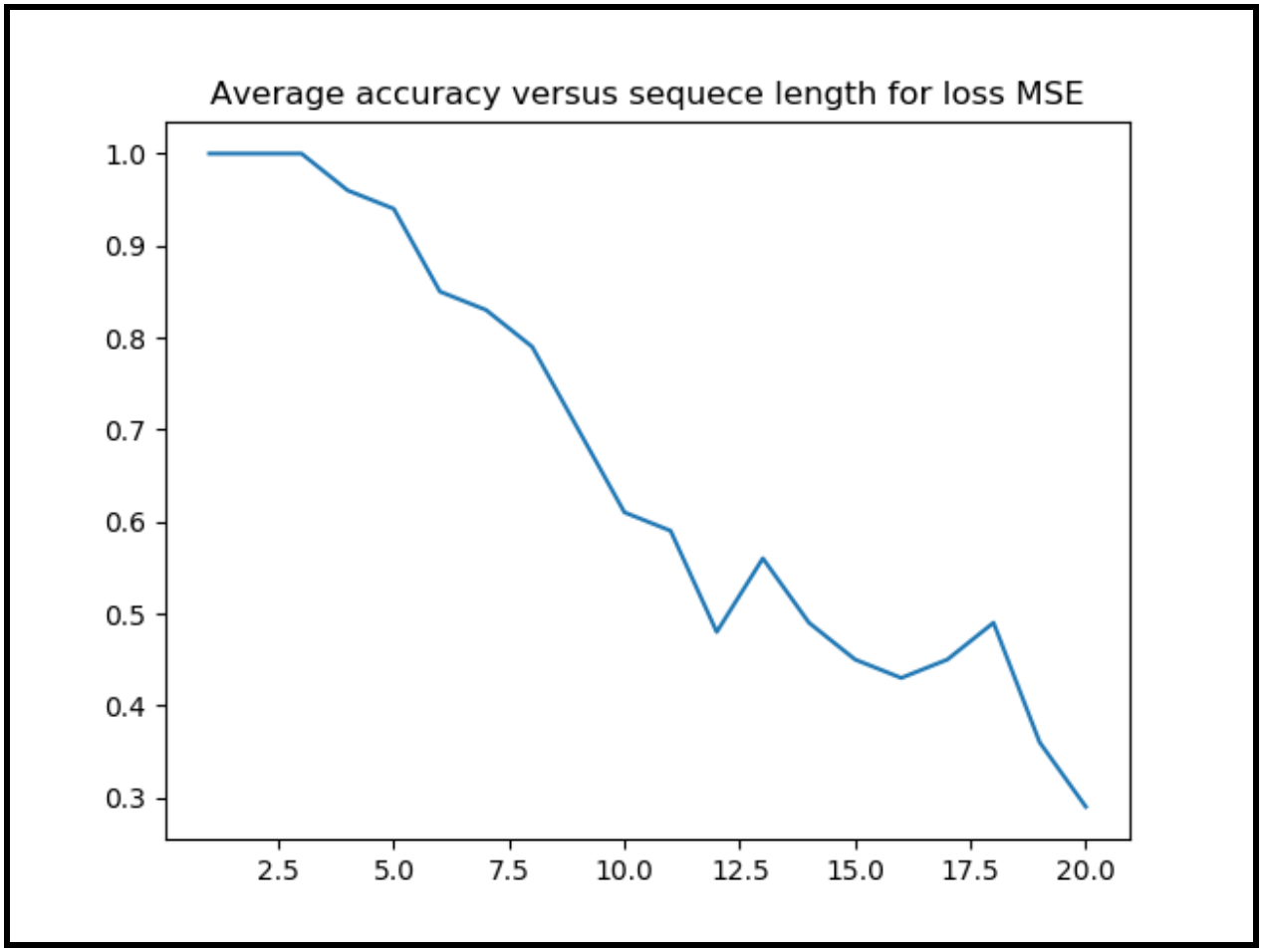
L= 3; ce

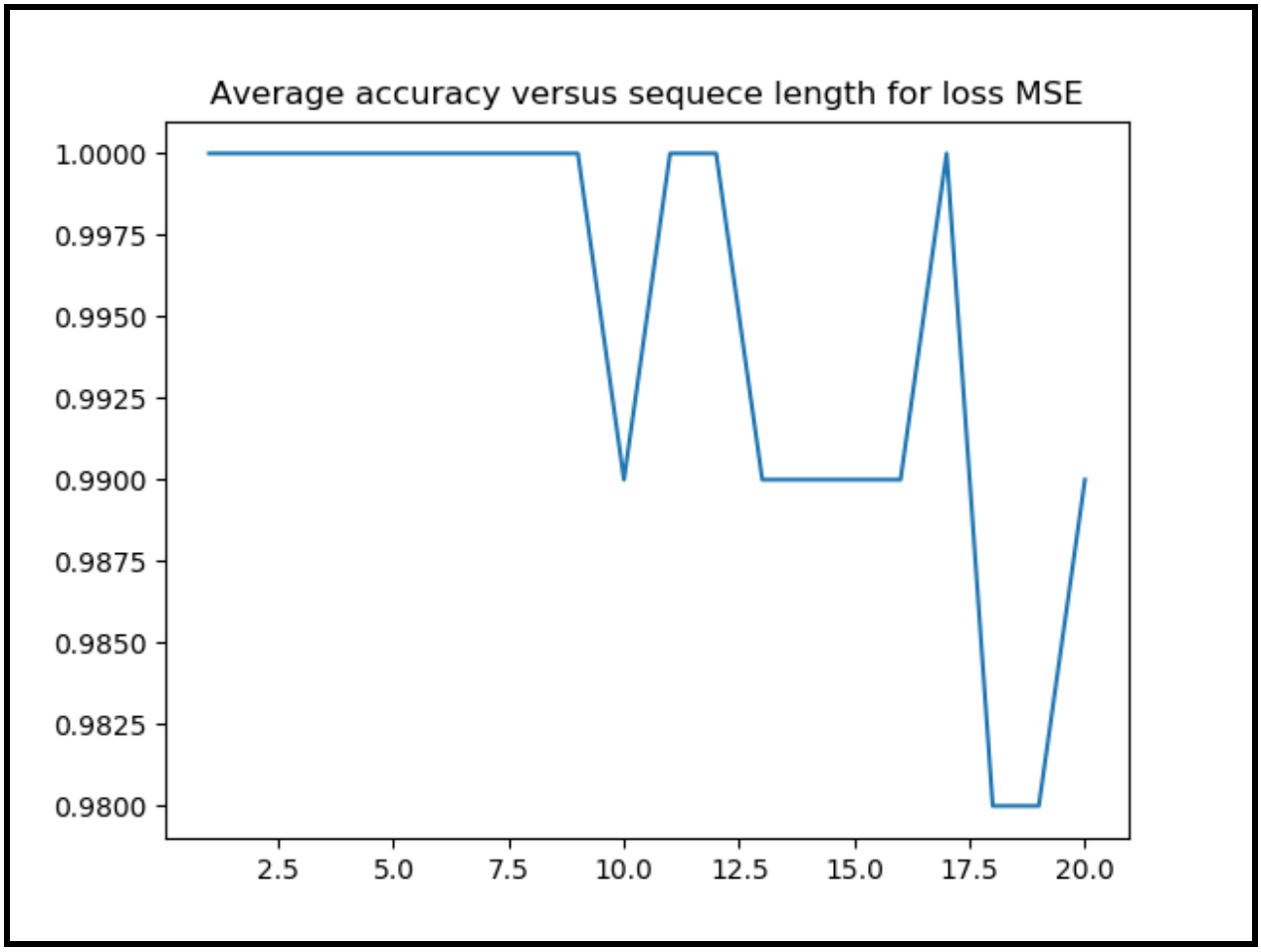

Accuracy-versus-Epochs-10-CE.png
0.0016
0.0014
0.0012
0.0010
0.0008
0.0006
0.0004
0.0002
0.0000
0
2
4
6
8

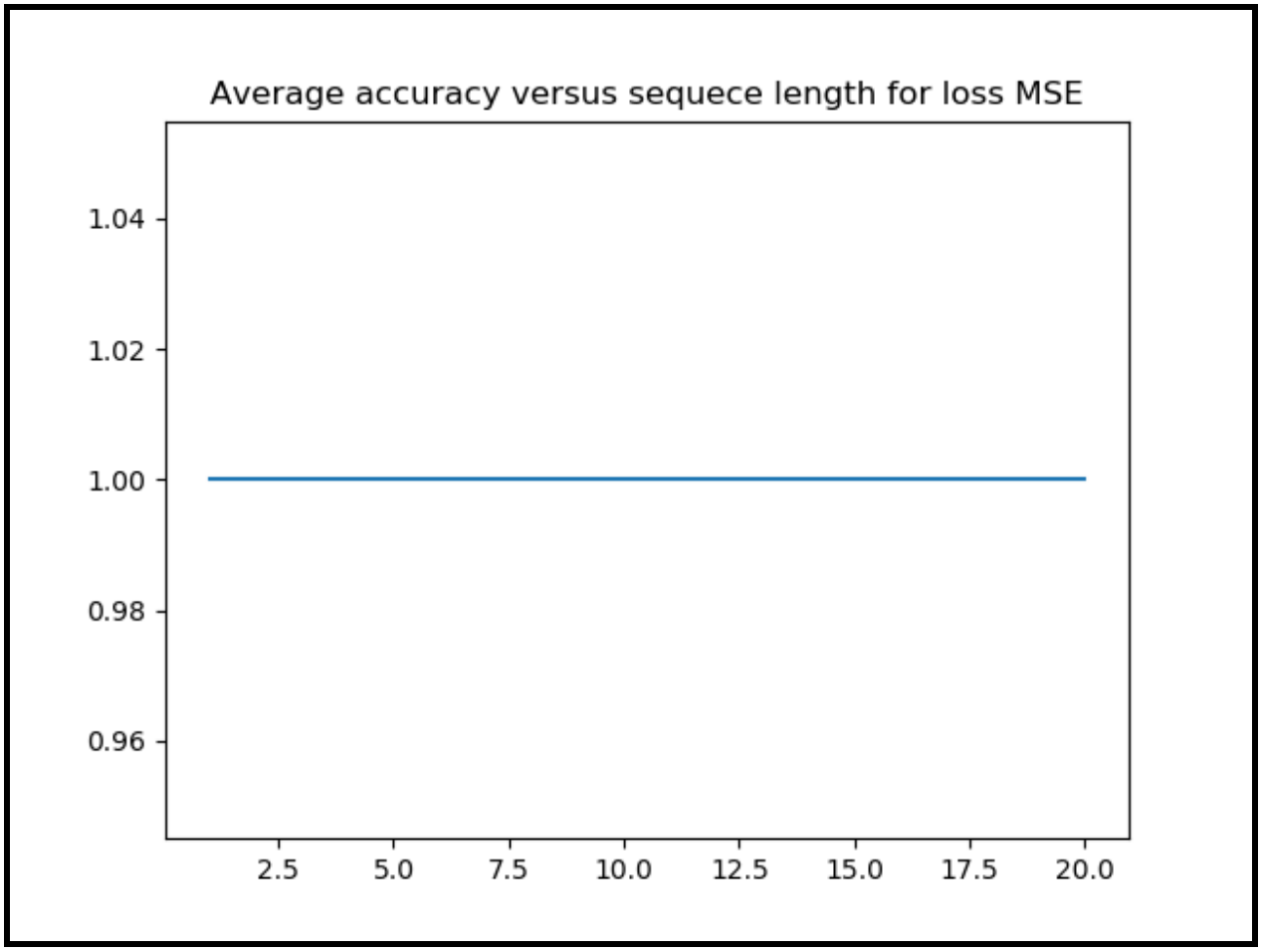

Accuracy-versus-Epochs-10-MSE.png
0.62
0.60
0.58
0.56
0.54
0.52
0.50
0.48
0
2
4
6
8

L=10:MSE


Average accuracy versus sequece length for loss CE
0.4
0.3
0.2
0.1
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Average accuracy versus sequece length for loss MSE
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Average accuracy versus sequece length for loss MSE
1.0000
0.9975
0.9950
0.9925
0.9900
0.9875
0.9850
0.9825
0.9800
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0



Conclusion:

1. L=3,5,10 were used on the training data and corresponding graphs were provided.
2. Bit accuracy was also implemented.
3. MSE performed better than CE.