Indian Institute of Technology, Madras
Deep Learning For Imaging (EE6132), Autumn 2017

Programming Assignment 3
Recurrent Neural Networks

Akshit Kumar
EE14B127

17th October 2017

Contents

1 MNIST Classification Using RNN

1.1 Goal

In this experiment, we will do classification on MNIST digits dataset using a Recurrent Neural Network and its variants.

1.2 Approach

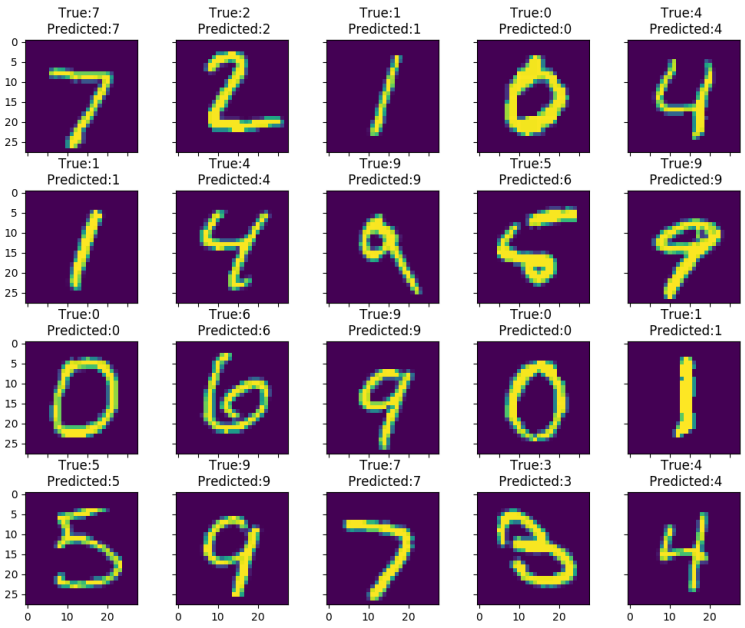
The approach will be deliberately simple i.e. reading the image row by row and treating that as one long sequence rather than a single entity.Input to the network is a digit image (28×28) which will be converted to a sequence in order to enable usage of RNN , which is a many to one network.

1.3 Using a network with vanilla RNN Cells

1.3.1 Network Details

Learning Rate	10^{-4}
Optimizer	AdamOptimizer
No. of units in RNN cell	128
Regularization	L-2 Regularization
Loss Function	Cross Entropy Loss

1.3.2 Prediction on 20 Test Images

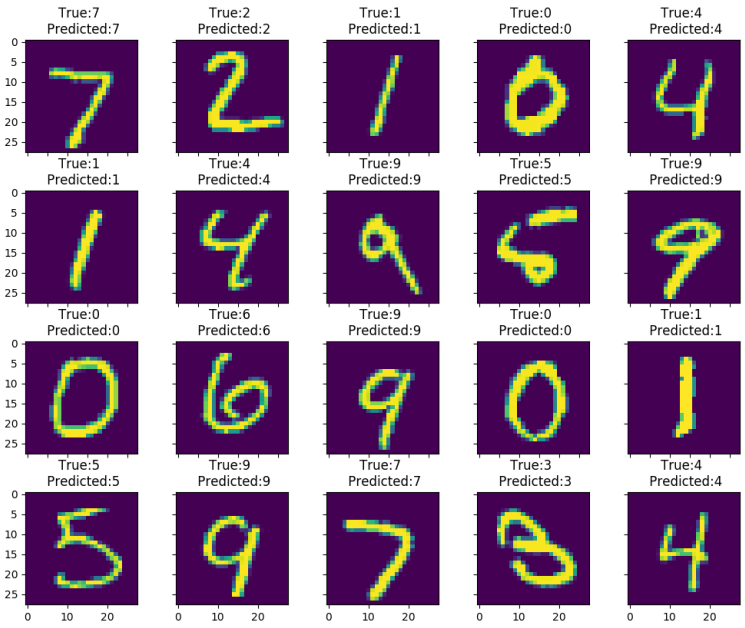


1.4 Using a network with LSTM/GRU cells

1.4.1 Network Details

Learning Rate	10^{-4}
Optimizer	AdamOptimizer
No. of units in RNN cell	128
Regularization	L-2 Regularization
Loss	Cross Entropy Loss

1.4.2 Prediction on 20 Test Images

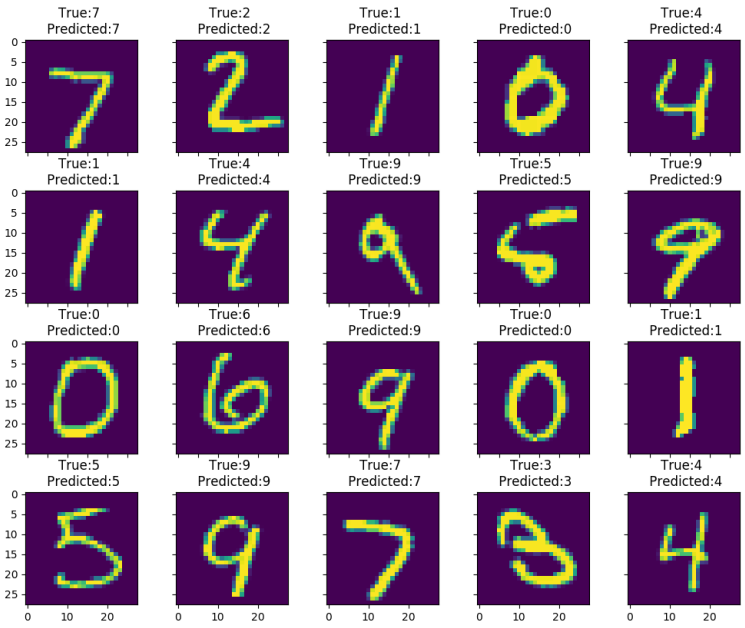


1.5 Using a bi-directional RNN/LSTM network

1.5.1 Network Details

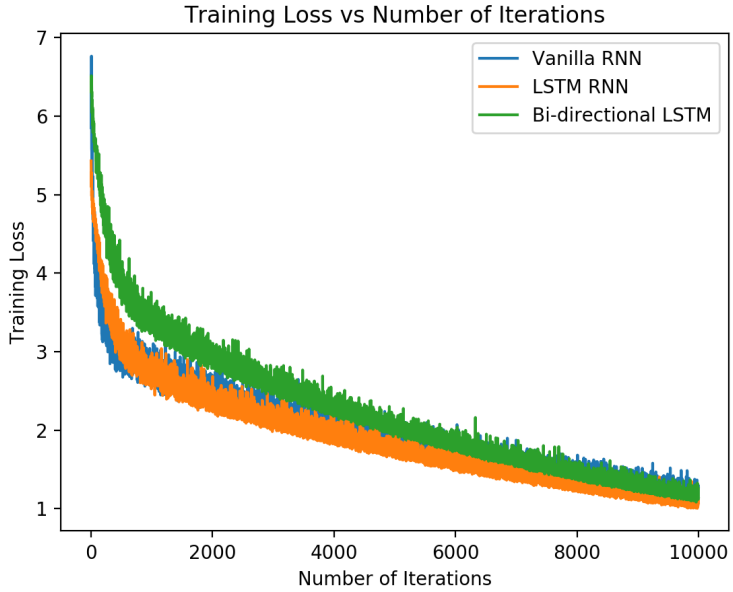
Learning Rate	10^{-4}
Optimizer	AdamOptimizer
No. of units in RNN cell	128
Regularization	L-2 Regularization
Loss	Cross Entropy Loss

1.5.2 Prediction on 20 Test Images

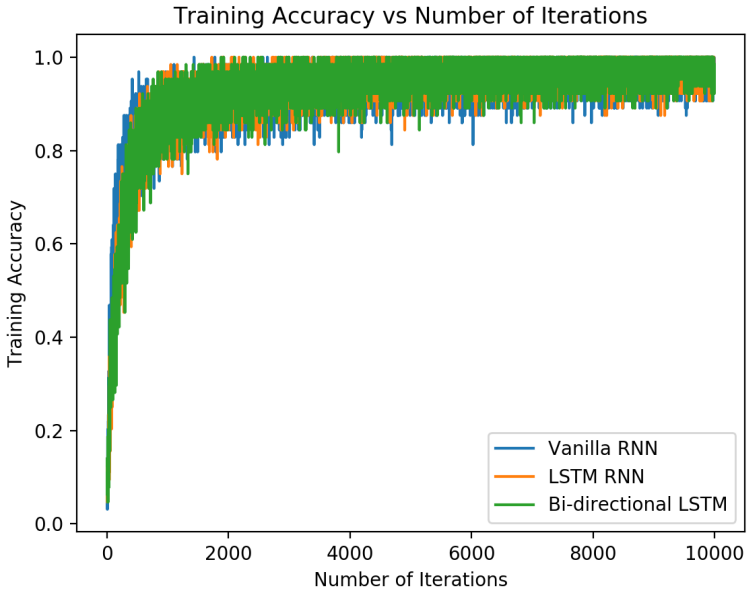


1.6 Comparison of 3 Models

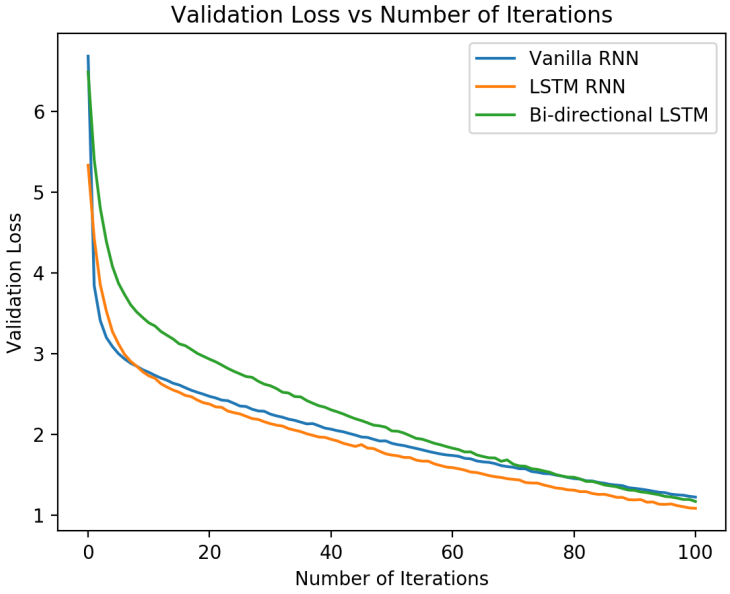
1.6.1 Training Loss for the 3 Models



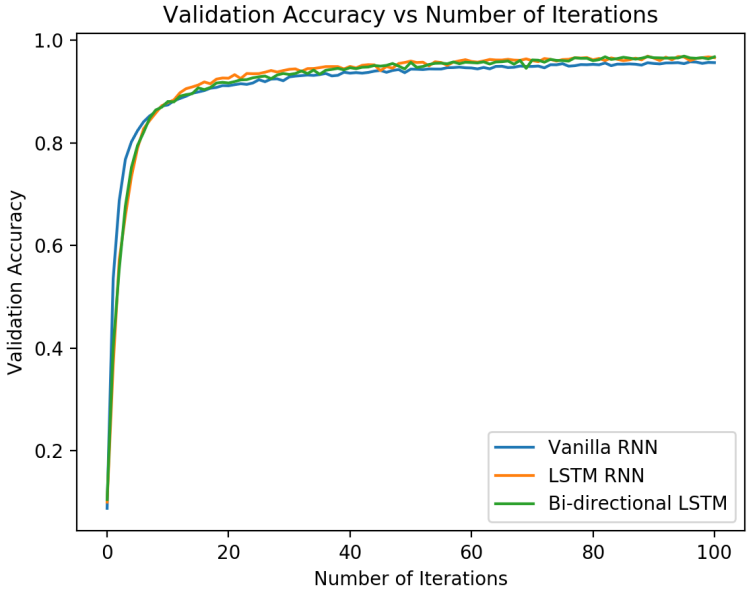
1.6.2 Training Accuracy for the 3 Models



1.6.3 Validation Loss for the 3 Models



1.6.4 Validation Accuracy for the 3 Models



1.6.5 Test Loss for the 3 Models

Model	Test Accuracy
Vanilla RNN	95.94%
LSTM RNN	96.69%
Bi-Directional RNN	96.76%

1.7 Discussion

We can clearly see that Bi-directional RNN performs the best amongst the three models we experimented with. LSTM RNN and Bi-directional RNN seem to perform similar. As for the validation loss, we can see that the validation loss is coming down for all the three models hence no overfitting is taking place.

2 Adding two binary strings

2.1 Goal

In this experiment, we will explore the simple problem of teaching an RNN to add binary strings.

2.2 Approach

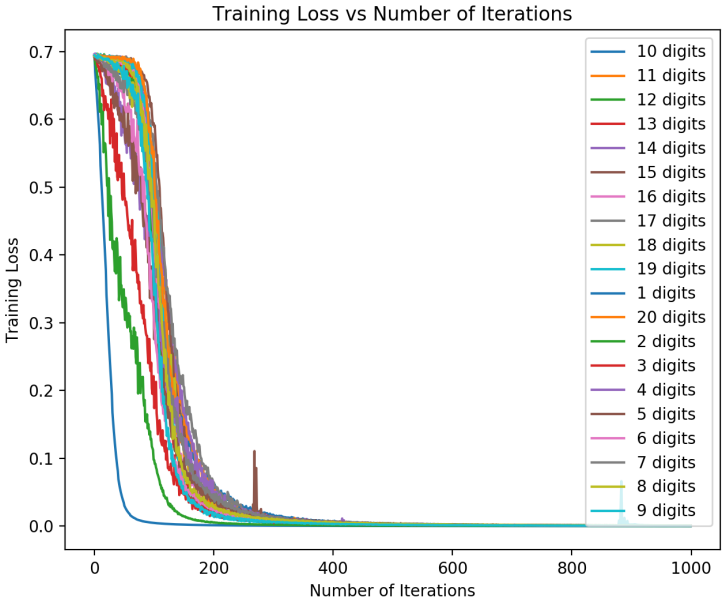
The RNN is fed two input bit-sequences and the target "sum" sequence. The sequence is ordered from LSB to MSB, i.e., time-step $1(t = 1)$ corresponds to LSB, and the last time-step is the MSB.

2.3 Hyper-Parameters Details

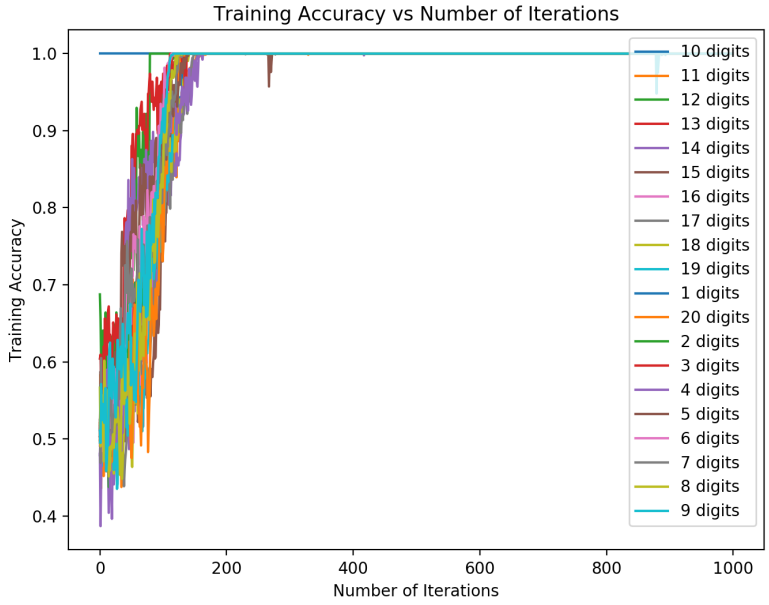
Learning Rate	10^{-4}
Training Steps	1000
Batch Size	64
Optimizer	AdamOptimizer
Loss Function	Cross Entropy Loss & MSE
No. of units in RNN Cell	20

2.4 Plots for Cross Entropy Loss

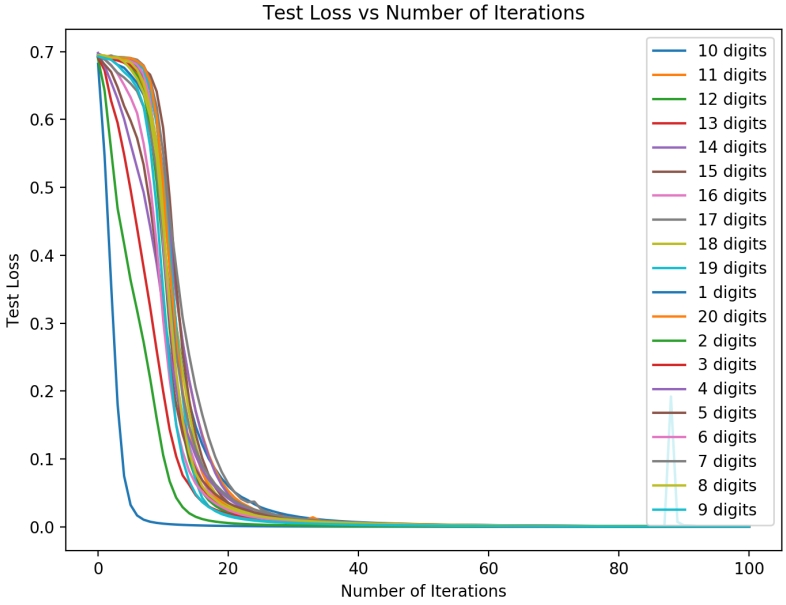
2.4.1 Training Loss for 20 Digits



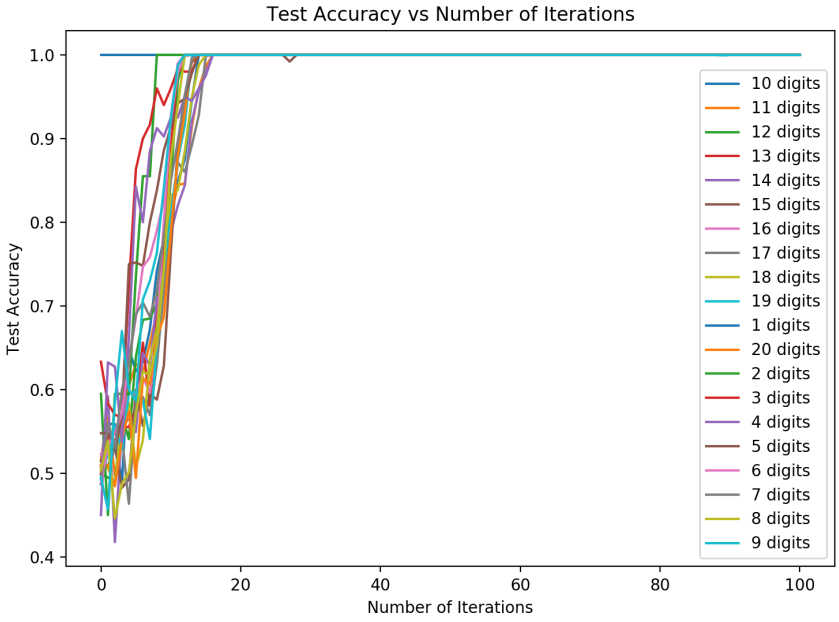
2.4.2 Training Accuracy for 20 Digits



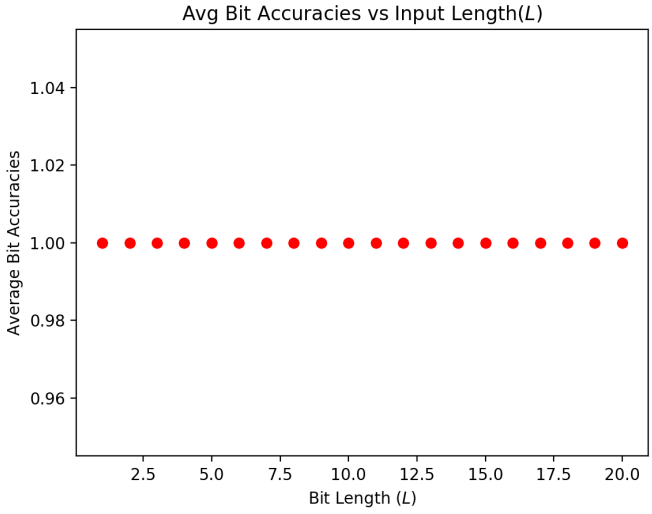
2.4.3 Test Loss for 20 Digits



2.4.4 Test Accuracy for 20 Digits

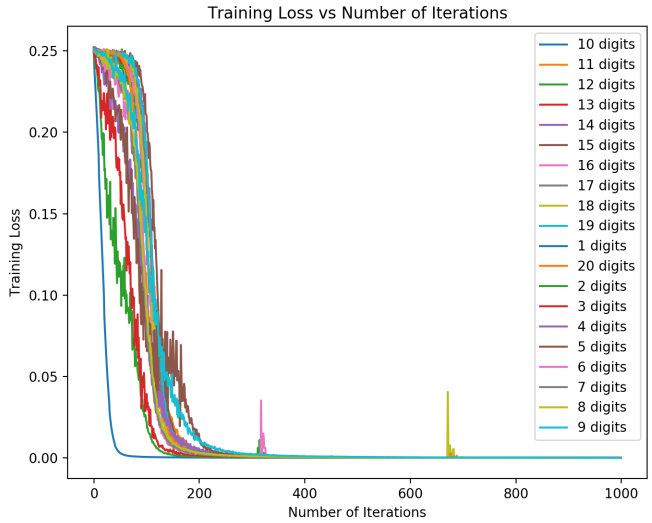


2.4.5 Bit-Accuracy Plots For Cross Entropy Loss Function

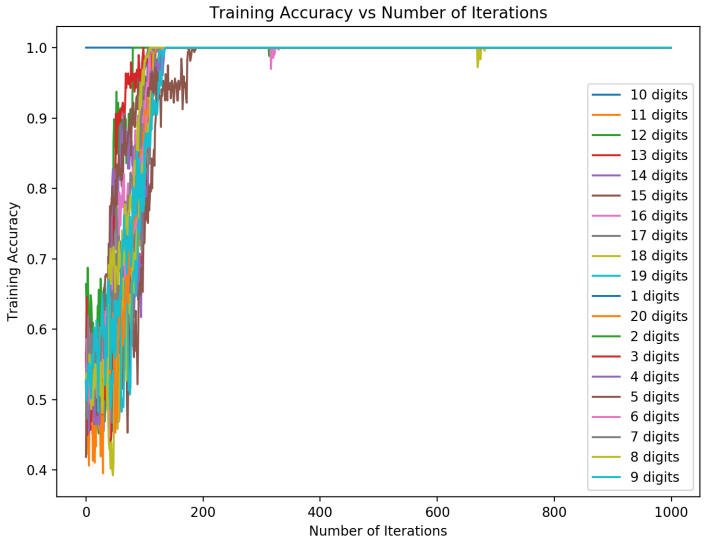


2.5 Plots for MSE Loss

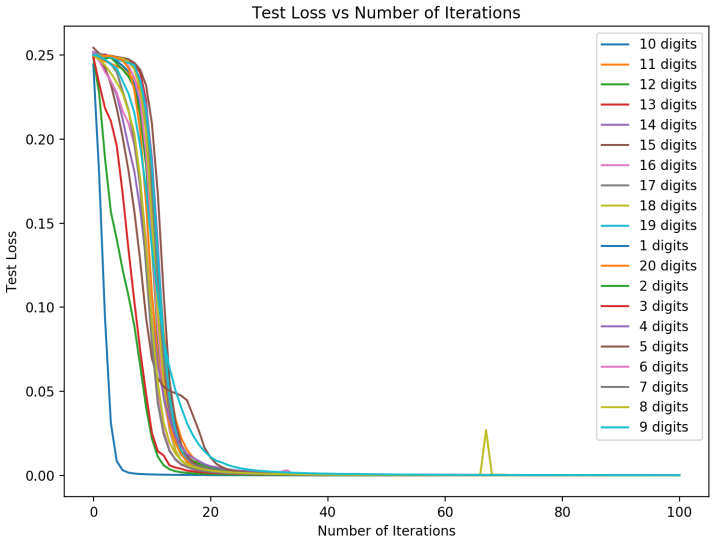
2.5.1 Training Loss for 20 Digits



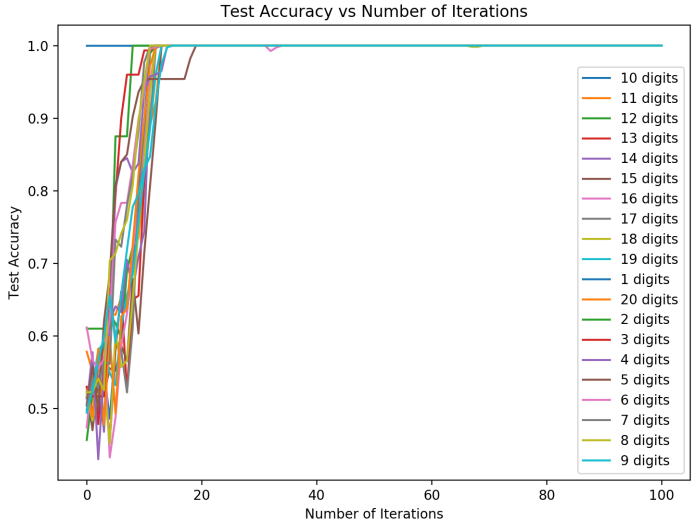
2.5.2 Training Accuracy for 20 Digits



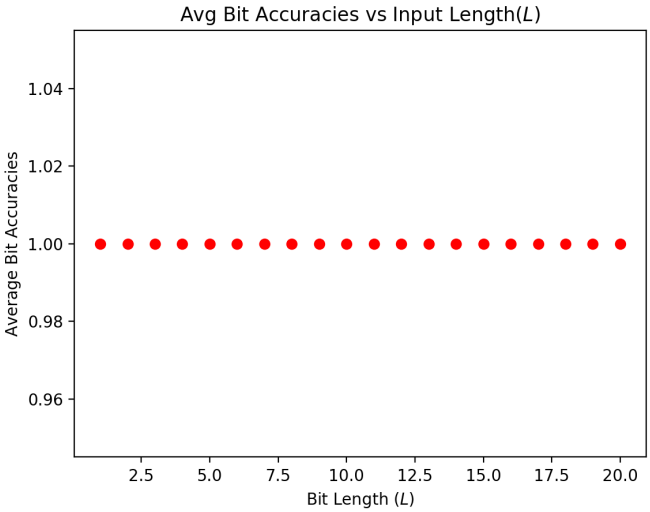
2.5.3 Test Loss for 20 Digits



2.5.4 Test Accuracy for 20 Digits



2.5.5 Bit-Accuracy Plots For Cross Entropy Loss Function



2.6 Discussion

We can clearly see that after training, the network learns how to add binary digits for all the length sizes considered in the experiments. However, the learning rate for the network differs. From the training and test loss plots, we can clearly see that smaller the bit length, quicker the network learns to perform - for both the loss functions (cross entropy loss function and mean squared error loss function)