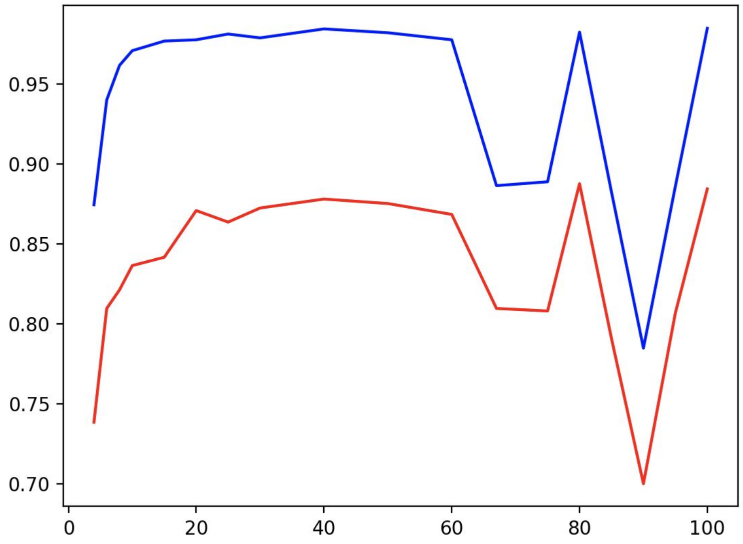
Assignment 3

1.2

1) The dimension of input layer for tinyTOY dataset is 2. The dimension of hidden layer for tinyTOY dataset is 30. The dimension of output layer for tinyTOY dataset is 2. (2,30,2) is the dimension for tinyTOY dataset.

   The dimension of input layer for tinyMNIST dataset is 196. The dimension of hidden layer for tinyMNIST dataset is 30.The dimension of output layer for tinyMNIST dataset is 10. .
   (196,30,10) is the dimension for tinyTOY dataset.

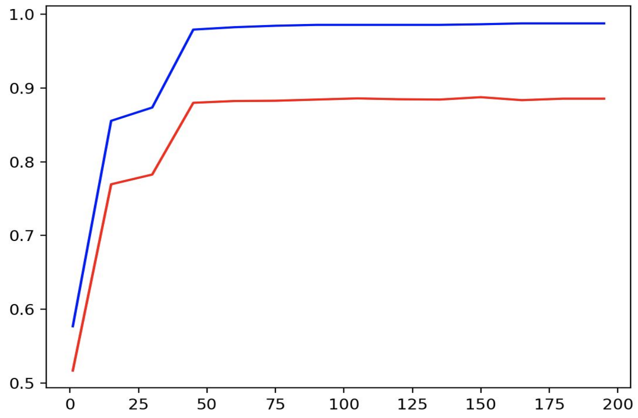
2)



Y-axis:
Red - testing accuracy
Blue - training accuracy
X-axis : size of hidden layer

The role of size of hidden layer is same on both training and testing accuracy is similar. The training and testing accuracy both increase initially with size of hidden layer and then decreases with size of hidden layer. Then increases and dips with size of hidden Layer. Finally increases with size of hidden layer till 100(I captured till 100 size).

3)



X -Axis - epochs
Y - Axis : Red -Testing Accuracy , Blue - Training Accuracy.
The role of number of epochs is same on both training and testing accuracy is similar. Both training and testing accuracy increase initially with increase in number of epochs. After certain number of epochs , the increase becomes very less and the curve becomes almost flat.(tends to gradient zero)

2.2)

1) The layers I used in my model are : Conv2D, MaxPool2D, Dropout , Flatten , Dense.

Conv2D layer : This layer used uses filter and it slides the filter across width and height image. This layer is used to identify some visual features (ex edge) which will be used in subsequent layers.

MaxPool2D layer : The basic use of this layer is to reduce number of parameters and hence the computation reduces. The input of conv2D is fed to the MaxPool2D layer. The max pool uses a filter and passes the filter through the image matrix and downsizes the matrix size by selecting the max value in the place of the matrix where the filter is placed.This helps to capture the prominent part in the filter. This reduces the noise and prevents overfitting.

Dropout Layer : It is the regularization technique. It prevents the model from overfitting.

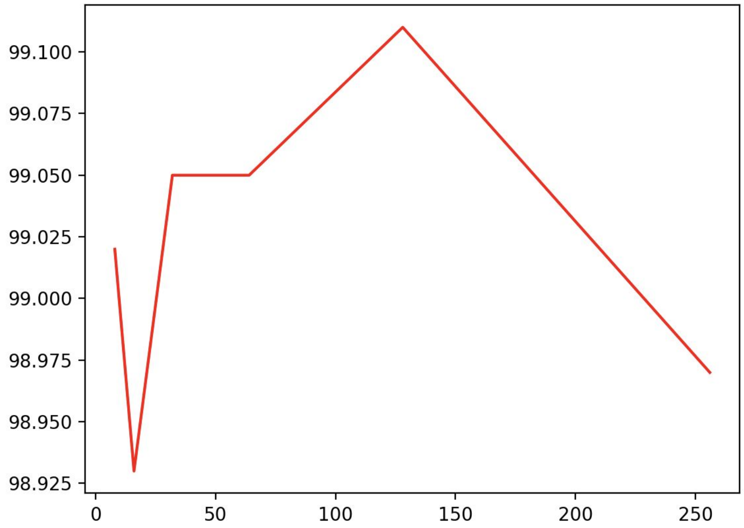
Flatten Layer : It just flattens the input obtained from previous layer.(ex : (24,24,32) to (18432))

Dense Layer : It is fully connected from every activation in previous layer(usually flatten). It is usually used to reduce the dimension of previous layer and make the layer dense(reduce sparsity). I used it as final layer to reduce the previous layer(i used flatten layer) to number of classes possible for data set.

2) Initially I used conv2D layer with 20 filters of dimension (4,4). This layer uses relu activation. This layer is followed by Dropout layer which is followed by flatten and dense layer. Finally, I used softmax activation to compute the class label of the input instance. I used the sgd(stochastic gradient descent) optimizer for fitting the weights of the different layers of neural network.

Later , I used another same conv2D layer (same dimensions as first) which follows the first conv2D layer. I also used MaxPool2D layer with filter (2,2) which helped to minimize the noise(only selects prominent value in the part of input where filter is placed) and reduce number of parameters. This layer follows the second conv2D layer. I used adam optimizer instead of sgd. This things helped me to improve the accuracy of the model.

3)



This is plot of batch size vs accuracy for (epoch - 15, hidden layer dimension - 32,batchSize - 128)

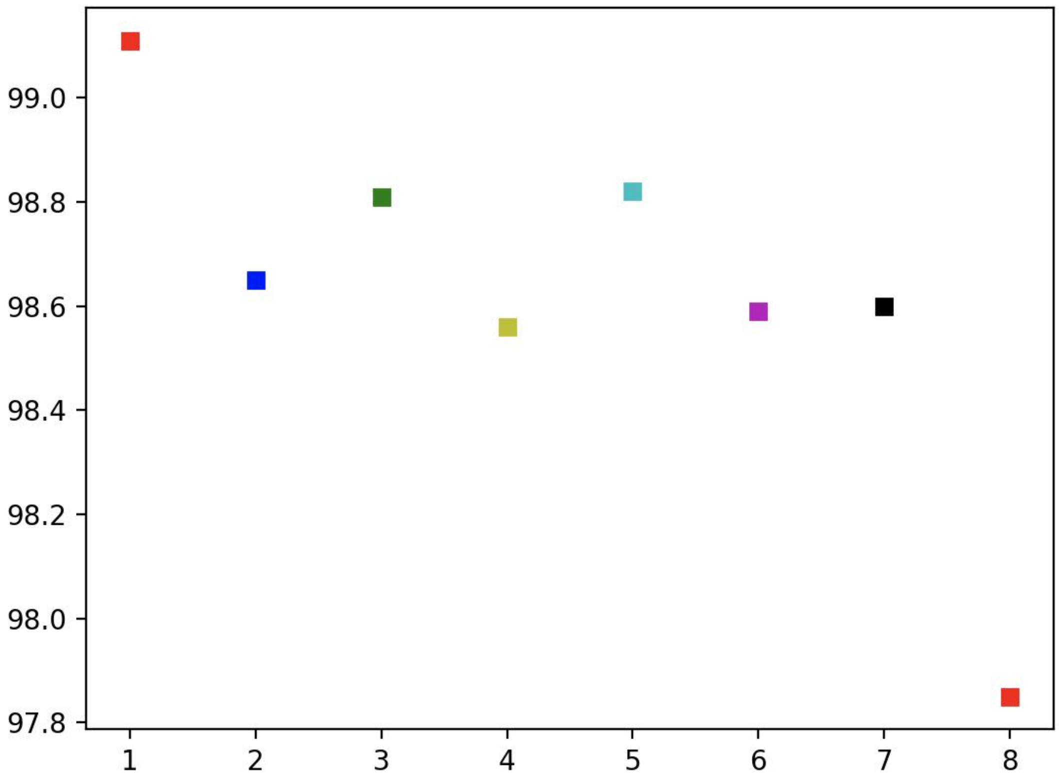
BatchSize	Accuracy
8	99.02
16	98.93
32	99.05
64	99.05
128	99.11
256	98.97

Accuracy

1- relu - 99.11%

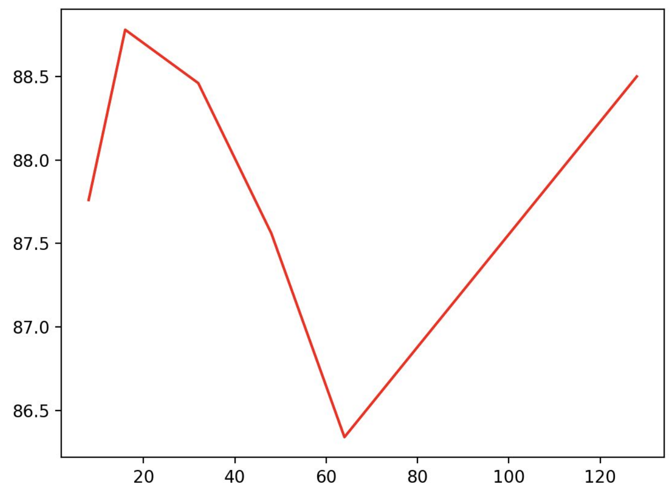
2-sigmoid - 98.65%

3-elu - 98.81%
4-selu - 98.56%
5-softplus - 98.82 %
6-softsign - 98.59%
7-tanh - 98.60%
8-hardsigmoid -97.85%



3.2)

1) Word Embeddings provide a dense representation of words. So embedding layer converts the words to embeddings which are denser (compared to one hot encoding of words which are sparse). The words similar in meaning have least distance(euclidean) between their embeddings.

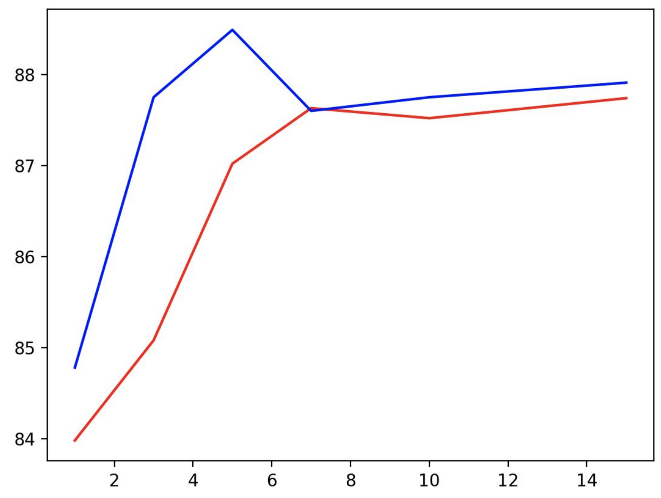


X- axis - dimension of hidden layer size

Y - axis - testing accuracy (I considered epochs = 5 for this plot)

The testing accuracy increases with the increase in the hidden dimension size. Later it decreases with increase in hidden dimension size. Finally , It increases with increase with increase in hidden dimension size.

3)



X-axis - number of epochs . Y-axis : blue - LSTM , red - GRU.

The LSTM has testing accuracy greater than or equal to GRU for epochs (1,3,5,7,10,15).