657 Assignment 3

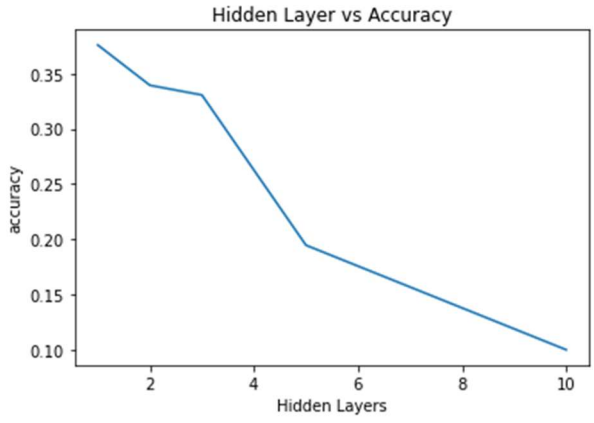
Problem 1: CNN for Image Classification

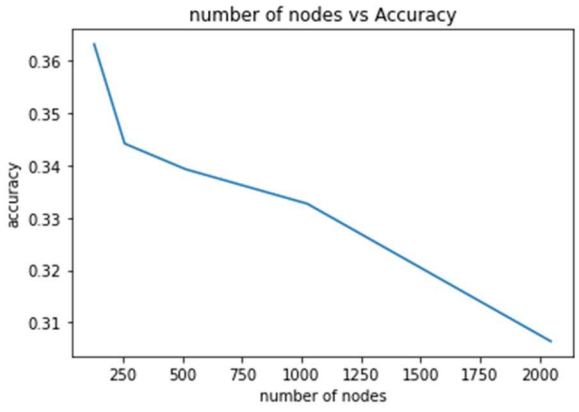
Answers:

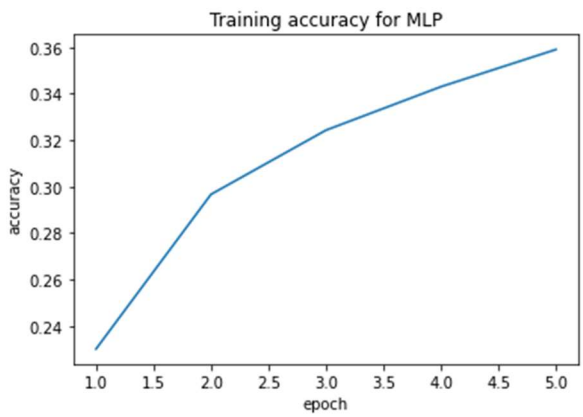
1. The dataset is downloaded for keras.datasets. This dataset contains 50,000 training images and 10,000 testing images by default. The training dataset is reduced to 10,000 values by using train_test_split method. The downloaded test dataset is used for validation. The 40,000 leftover data from the training set is used for testing. Thus we got 10,000 training data, 10,000 validation data and 40,000 testing data. The input dataset is normalized using min max normalization. Thus, the resulting value lies between 0-1. Batches of 32 are created on the train, validation and test. The default batch size for a model is 32.

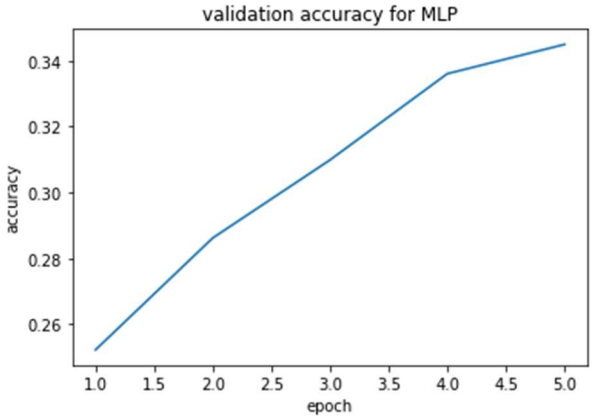
2. The output layer consists of 10 nodes since there are 10 categorical outputs. The activation function used here is softmax which predicts the probability of the categorical output. This setting remains same for all the 3 networks since the target output is the same. The loss function used here is 'sparse_categorical_crossentropy' since the output values integers ranging from 1-10 and have probabilistic outputs.

3. Changing the hidden layers and nodes will impact the accuracy of the model. For number of hidden layers ranging from 2-10, the accuracy is decreased as the hidden layer increases. This can also be seen in the hidden layer vs accuracy plot of MLP. Now for the best value of hidden layer i.e 2 the nodes are varied from 128 - 2048 (discrete values are selected). The accuracy decreases after a cetain increase in number of nodes and attains best value when number of nodes are 512. The nodes vs accuracy graph is plotted for better visualization. The training and validation curves are plotted and the model attains accuracy of 35.95 for training and 34.56 for validation. The test accuracy is 34.35.



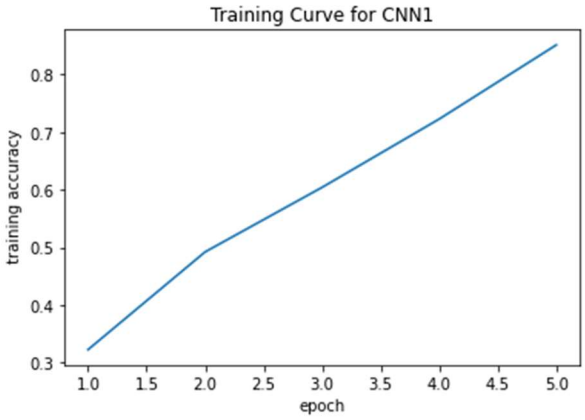


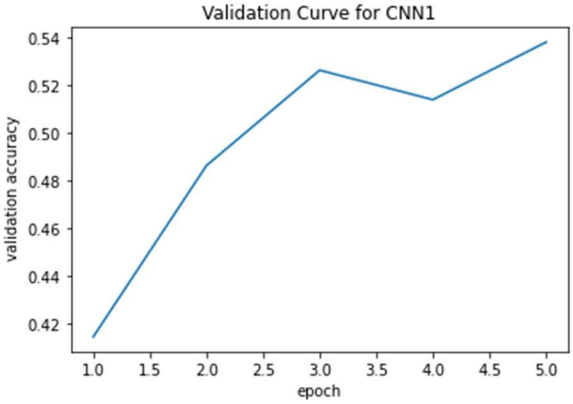


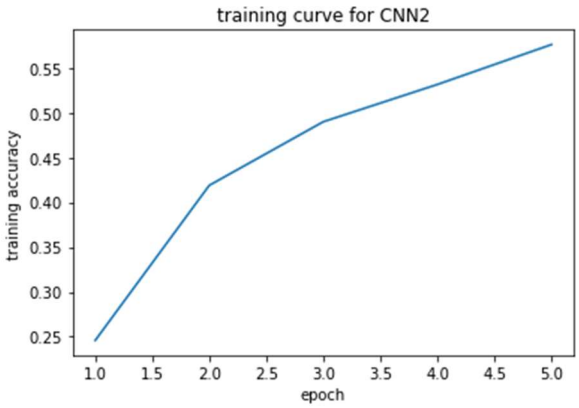


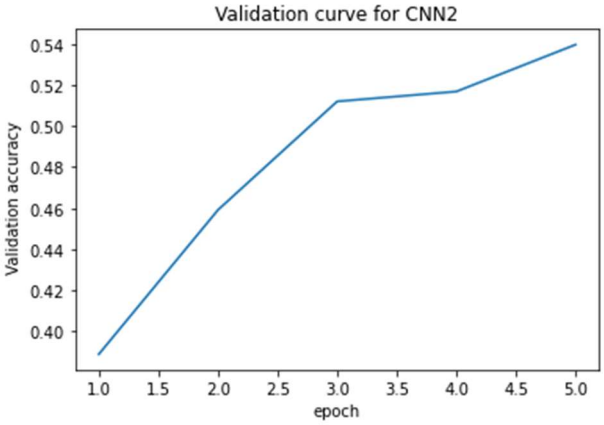
4. The training and testing accuracy is lowest for MLP. It attains an accuracy of around 36% for training dataset. The CNN1 attains an accuracy of 90% for training set and 53% for testing set. The CNN2 has an accuracy of 56% for training and 54% for testing dataset. There is a large difference in the training accuracy and testing accuracy of CNN1 which might be because of overfitting. This is removed in CNN2 by using Drop(0.2). Due to this, there is no much difference between training accuracy and test accuracy of CNN2. Thus the CNN2 performs the best amongst the 3 models.

5. The training and validation curves are plotted for CNN1 and CNN2 using the hyper parameters stated in the question.









The training time CNN1 is high when compared to that of CNN2. It takes almost 750 seconds to run CNN1 whereas it takes almost 150 seconds for CNN2. Thus CNN2 is almost 5 times faster than CNN1. This is because:

CNN1 has 25,997,130 parameters to train whereas CNN2 has only 1,486,666 parameters. These values can be seen in the summary of the model. Thus the training parameters of CNN1 are almost 17 times more than CNN2 due to which the training time increases. The training parameters are calculated based on the architecture of the model. However, though the training time and parameters are large for CNN1, they produce better accuracy than CNN2. Increasing the number of epoch's will increase the accuracy till some extent and then it will almost remain constant no matter how much we increase the epoch values.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 30, 30, 64)        1792
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 28, 28, 64)        36928
_________________________________________________________________
flatten (Flatten)            (None, 50176)             0
_________________________________________________________________
dense (Dense)                (None, 512)               25690624
_________________________________________________________________
dense_1 (Dense)              (None, 512)               262656
_________________________________________________________________
dense_2 (Dense)              (None, 10)                5130
=================================================================
Total params: 25,997,130
Trainable params: 25,997,130
Non-trainable params: 0
_________________________________________________________________
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 30, 30, 64)        1792
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 15, 15, 64)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 13, 13, 64)        36928
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 6, 6, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 2304)              0
_________________________________________________________________
dense (Dense)                (None, 512)               1180160
_________________________________________________________________
dropout (Dropout)            (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               262656
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 10)                5130
=================================================================
Total params: 1,486,666
Trainable params: 1,486,666
Non-trainable params: 0
_________________________________________________________________
```

6. From the changing accuracies of MLP for hidden layers and nodes, we concluded that hidden layer = 2 and hidden nodes = 512 gives max accuracy. This is what is specified in the problem statement as well. However, the sigmoid activation function reduces the accuracy. Thus, if we use Relu activation function for this dataset, the accuracy can be improved . When the CNN1 model was run on 40,000 training datasets, the average accuracy was 96% which shows that the given parameters are a perfect fit. It also proves the fact that 2 hidden layers provides a better accuracy for this dataset. Since the training dataset is reduced to 10,000 the accuracy is dropped to approx 90%. Thus, by changing the number of training samples accuracy of CNN1 can be improved. The case of overfitting can be removed by dropping some values by using the Drop() function. Similarly, accuracy the CNN2 model can be improved by removing the max pooling layer before the flatten command and remove droop(0.2) before output layer. By doing this, the accuracy almost improved by 15%. It can further be improved by increasing epochs and training samples