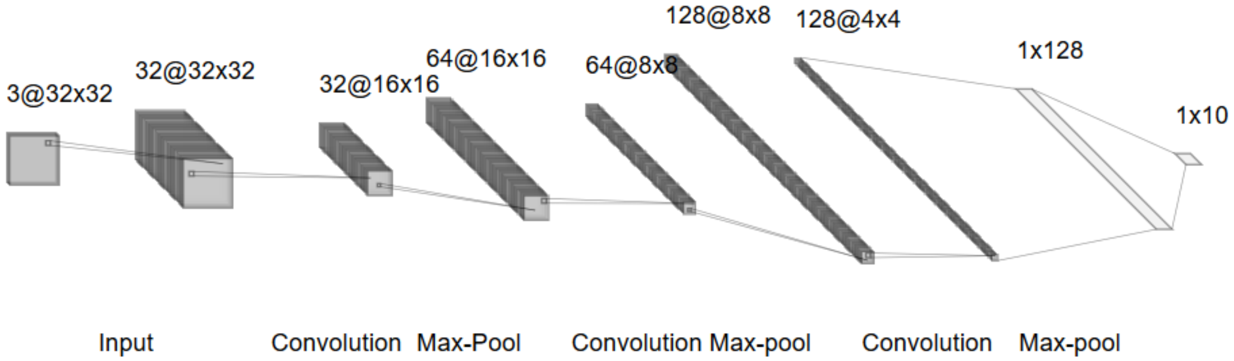
Name: Shreya Kate

Email: shreyak@usc.edu

USC ID: 2334973997

Homework 5 (P2)

Convolutional Neural Network:



The input image is a 32x32 dimension color image from CIFAR10 dataset. Therefore, it has 3 channels (RGB).

1. I have increased the depth of the CNN as compared with HW5 P1.

2. This network has 3 convolutional layers. The previous LeNet5 has 2 convolutional layers.

3. This network has 32 filters in the first convolutional layer, 64 filters in the second and 128 filters in the third. Then, I flatten this to vectors of dimension 128 and then reduce the dimension to 10 to get the probability vector. The filter size is 3x3. The number of filters used in this are much higher than HW5 P1, so the accuracy is increased.

4. I used dropout in this network, which I hadn't used in LeNet5, which also led to improving the performance.

Because of these, this network performs better and gives high accuracy as compared to LeNet5 architecture in HW5 P1.

There is a max pooling layer after each convolutional layer with size 2x2.

The activation function used is ReLU. The last layer has softmax.

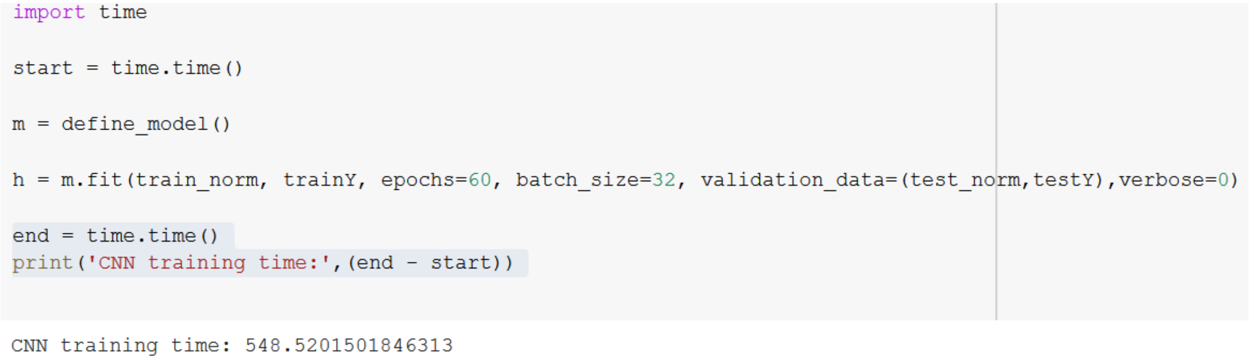
I have used Sequential Gradient Descent. The learning rate is 0.001 and the momentum is 0.9. I have done 60 epochs to get the best result. The batch size is 32.

I have used Keras and Google colab to impement this network.

I tried to augment the dataset, but I noticed that I was getting better accuracy without augmenting the dataset, so I dropped it.

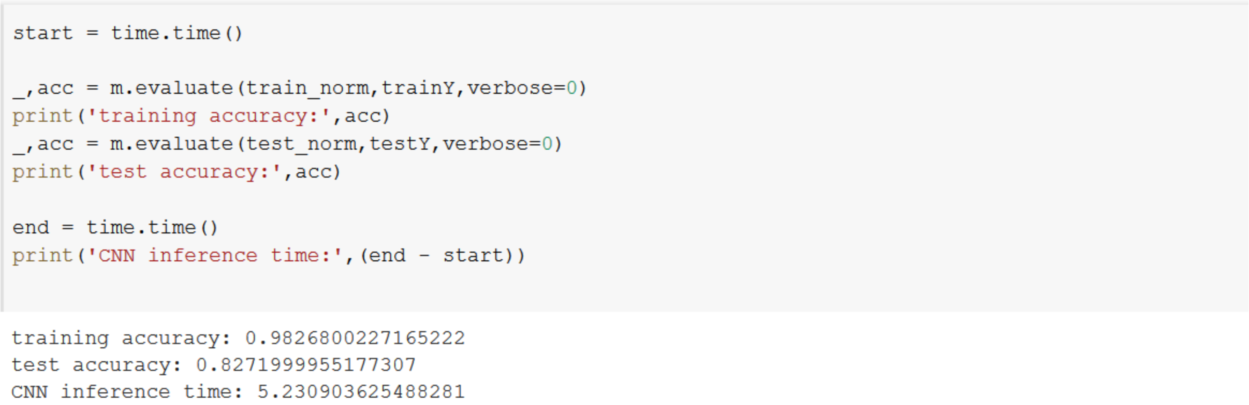
Accuracy:

The best accuracy I achieved is shown below:



```
import time

start = time.time()

m = define_model()

h = m.fit(train_norm, trainY, epochs=60, batch_size=32, validation_data=(test_norm,testY),verbose=0)

end = time.time()
print('CNN training time:',(end - start))
```

```
CNN training time: 548.5201501846313
```



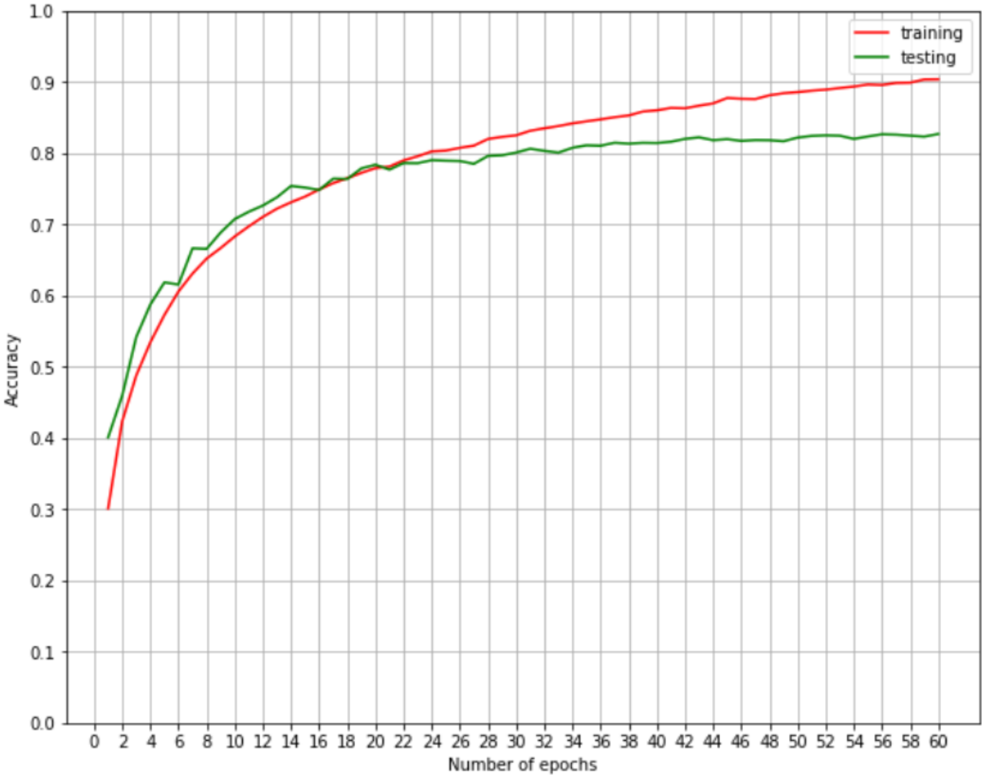
```
start = time.time()

_,acc = m.evaluate(train_norm,trainY,verbose=0)
print('training accuracy:',acc)
_,acc = m.evaluate(test_norm,testY,verbose=0)
print('test accuracy:',acc)

end = time.time()
print('CNN inference time:',(end - start))
```

```
training accuracy: 0.9826800227165222
test accuracy: 0.8271999955177307
CNN inference time: 5.230903625488281
```

This is for 50000 training images.

Training accuracy: 98.26 %
Test accuracy: 82.71 %
The training time was 548.52 seconds = 9.142 minutes
The inference time was 5.23 seconds.

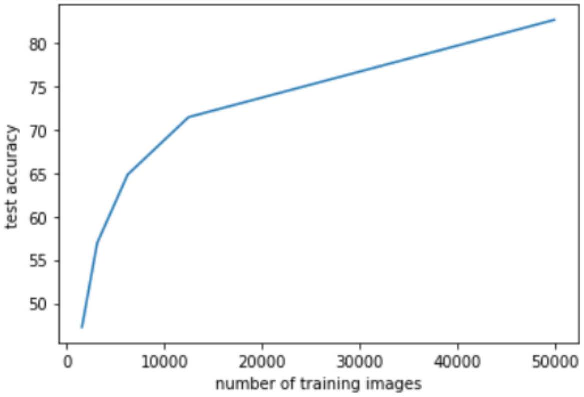
The train and test accuracy performance curve is shown here:



I dropped the number of training samples to see how the accuracy was affected:

Number of samples	Test accuracy (in %)
50000	82.71
12500	71.43
6250	64.81
3125	56.94
1562	47.26

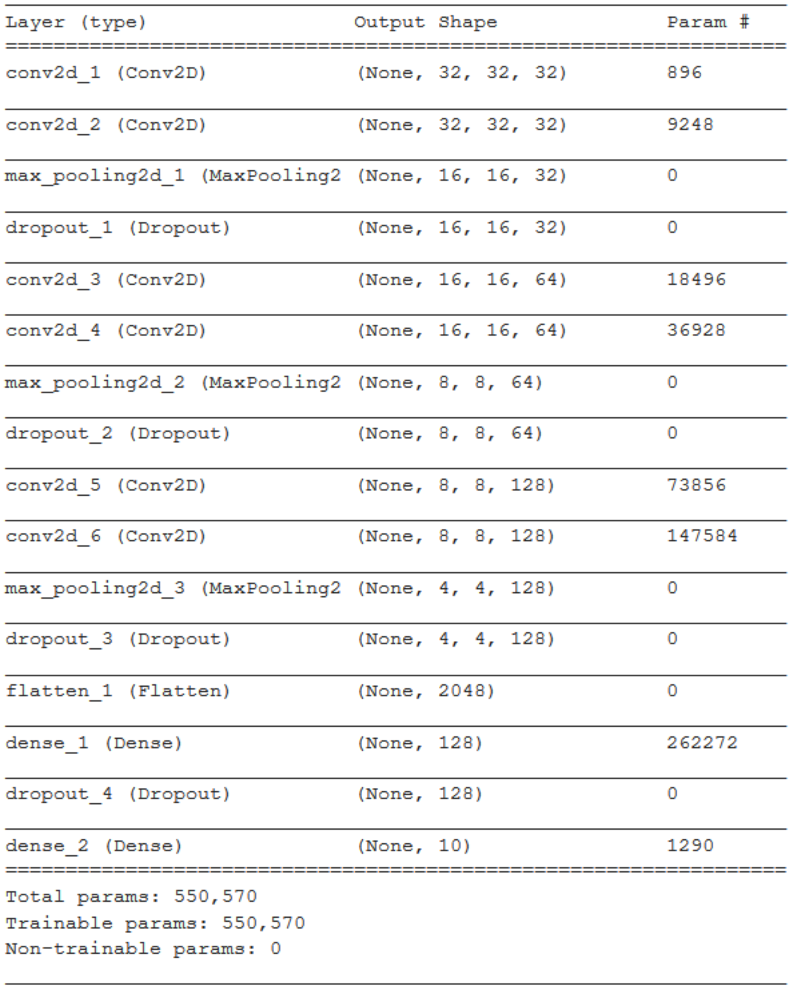
The test accuracy curve is shown below. The test accuracy decreases with the decrease in the number of samples given for training.



Model size:

The model size was calculated by using model.summary().

The total number of parameters were found to be 550,570 and all of these parameters are trainable.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 32, 32, 32)        896
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 32)        9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 16, 16, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 64)        18496
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 16, 16, 64)        36928
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 8, 8, 64)          0
_________________________________________________________________
dropout_2 (Dropout)          (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 8, 8, 128)         73856
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 8, 8, 128)         147584
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 4, 4, 128)         0
_________________________________________________________________
dropout_3 (Dropout)          (None, 4, 4, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 2048)              0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               262272
_________________________________________________________________
dropout_4 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 10)                1290
=================================================================
Total params: 550,570
Trainable params: 550,570
Non-trainable params: 0
_________________________________________________________________
```