Deep Learning Project 2

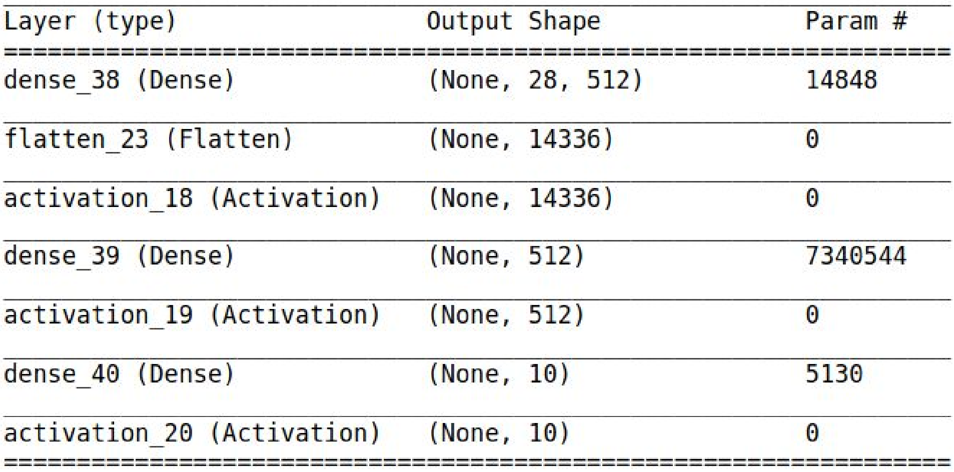
Some common specification for both the networks

1. How input was taken in both the networks :
   In fully connected I used a flatten layer to convert the input from(28 x 28) images to 784 dimensional inputand in convolutional neural network I converted the input in (28x 28 x1)
2. I used validattion_split=0.33 i.e I used 40199 samples to train and 19801 samples to validate
3. I used an adam optimizer. I tried other optimizers also like Stochastic gradient descent and RMS but with adam optimizer I was getting good accuracy.
4. I used the learning rate as 0.001

How I selected the model :

FULLY CONNECTED NETWORK

1. In this I tried different numbers of neurons in each layer but the accuracy was not changing considerably
2. Even by increasing the number of hidden layers, I was getting the same accuracy. So I continued using 1 hidden layer.
3. I used learning rate as 0.001
4. And I found that 20 epochs are sufficient to stabilize the validation accuracy so I used 20 epochs here



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_38 (Dense)             (None, 28, 512)           14848
_________________________________________________________________
flatten_23 (Flatten)         (None, 14336)             0
_________________________________________________________________
activation_18 (Activation)   (None, 14336)             0
_________________________________________________________________
dense_39 (Dense)             (None, 512)               7340544
_________________________________________________________________
activation_19 (Activation)   (None, 512)               0
_________________________________________________________________
dense_40 (Dense)             (None, 10)                5130
_________________________________________________________________
activation_20 (Activation)   (None, 10)                0
=================================================================
```

Fig1 : This is the model summary for Fully connected Network

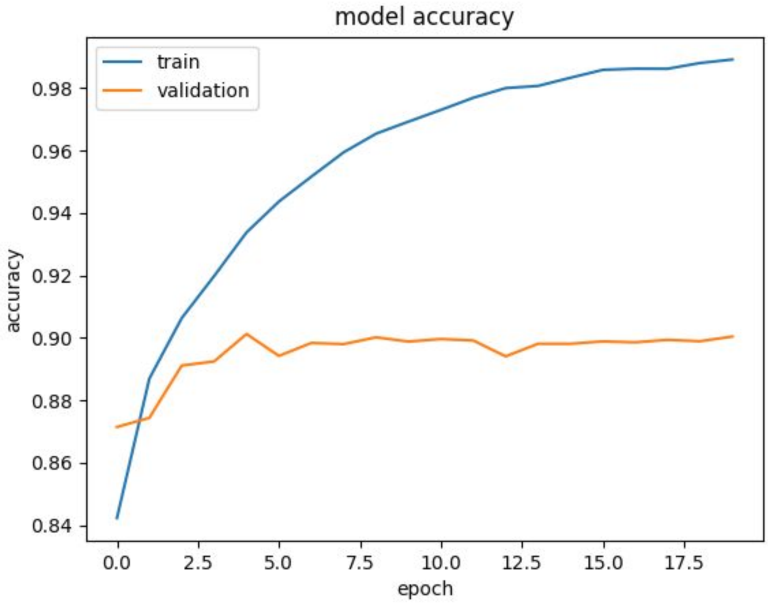


Fig 2: This is graph showing train and validation accuracy for fully connected model

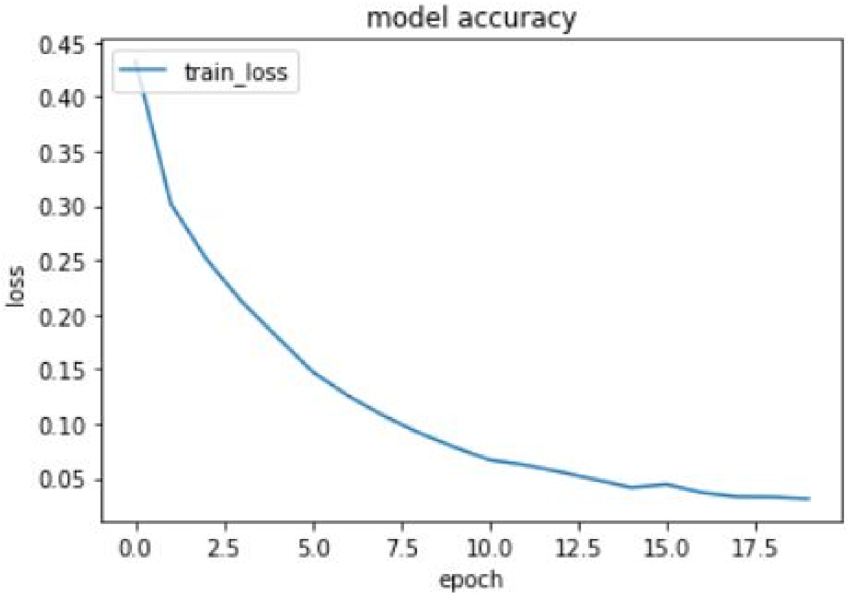
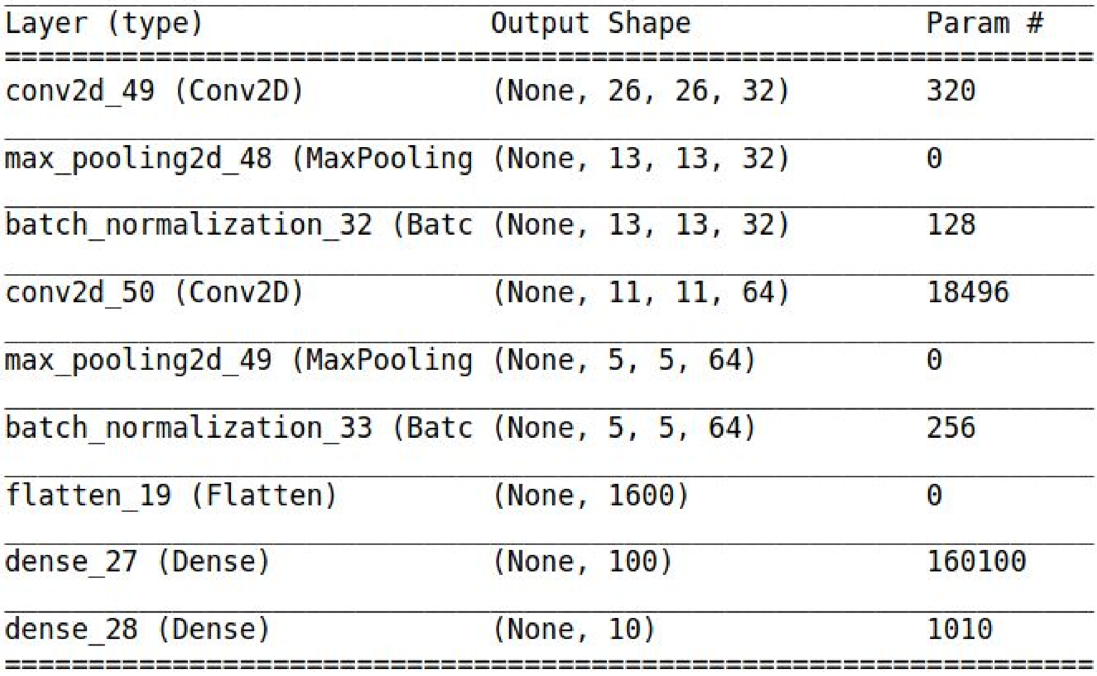


Fig 3: This is the graph showing training loss vs epoch for the fully connected model

CONVOLUTIONAL NEURAL NETWORK

1. Initially in this with using 3 hidden layers with 32, 64 and 64 neurons, I was getting near to 83% accuracy in test data
2. Then I tried different normalization techniques
    - With dropout, the test accuracy was near to 82%. I also tried the different value for dropout percentage but with all of them I was getting less accuracy.
    - By using Batch normalization, I was getting near to 84% accuracy in test data
3. I also tried different weight initialization techniques

- By using h2_uniform weight initialization in keras I was getting 87% accuracy. h2_unifrom takes variance as `sqrt(6 / fan_in)`
- By using both batch normalization and h2_uniform weight initialization I got accuracy near to 89.9%



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_49 (Conv2D)           (None, 26, 26, 32)        320
_________________________________________________________________
max_pooling2d_48 (MaxPooling (None, 13, 13, 32)        0
_________________________________________________________________
batch_normalization_32 (Batc (None, 13, 13, 32)        128
_________________________________________________________________
conv2d_50 (Conv2D)           (None, 11, 11, 64)        18496
_________________________________________________________________
max_pooling2d_49 (MaxPooling (None, 5, 5, 64)          0
_________________________________________________________________
batch_normalization_33 (Batc (None, 5, 5, 64)          256
_________________________________________________________________
flatten_19 (Flatten)         (None, 1600)              0
_________________________________________________________________
dense_27 (Dense)             (None, 100)               160100
_________________________________________________________________
dense_28 (Dense)             (None, 10)                1010
=================================================================
```

Fig 4: This the model summary for convolutional Neural network

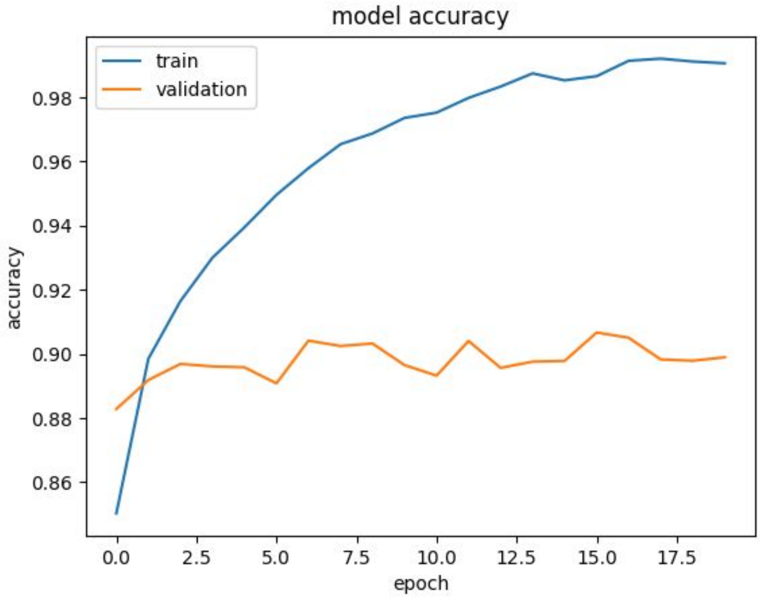


Fig 2: This is graph showing train and validation accuracy for convolutional neural network

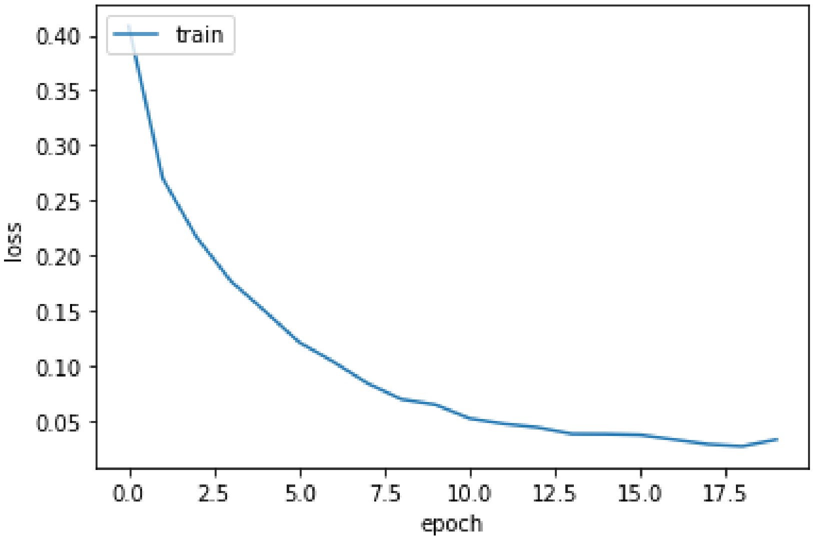


Fig 3: This is the graph showing training loss vs epoch for the convolutional neural network