ML 2 Programming Assignment 2 Report

- Submitted by : Anurag Saraswat (M20CS066)

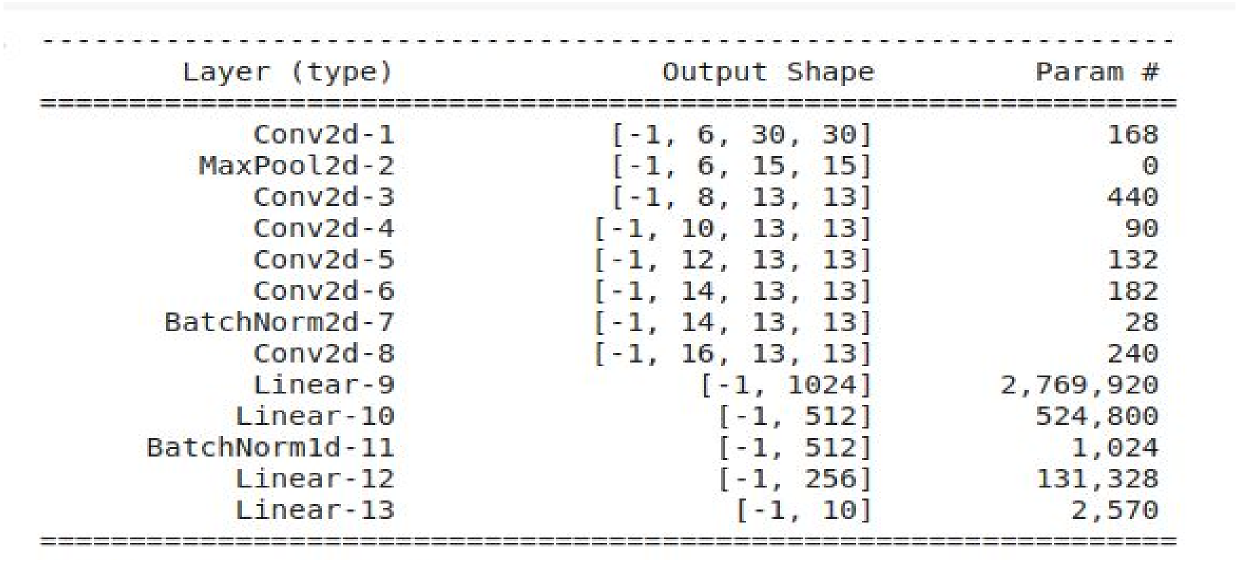
Q1. Use CIFAR dataset, Implement a 6-layer CNN network (Choose your own architecture) to perform classification.

CIFAR-10 dataset is used. It consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.
The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.
Class Present in the dataset is Air Plane, Automobile, Bird, Cat, Deer, Dog, Frog, Horse, Ship, Truck.

I. Accuracy on test data by varying the **Optimization Techniques**

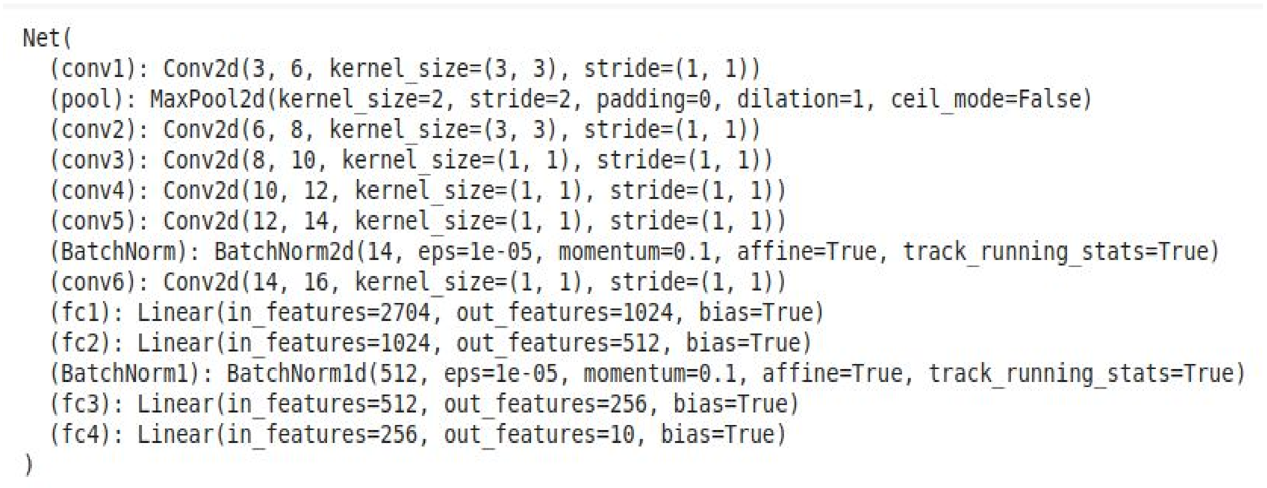
Model Information:

CNN Model consists of a total of 13 layers. 6 layers are Convolution Layer, 1 Max pool, 2 Batch Norm and 4 linear layers.
Arrangement of these layers are as follow:



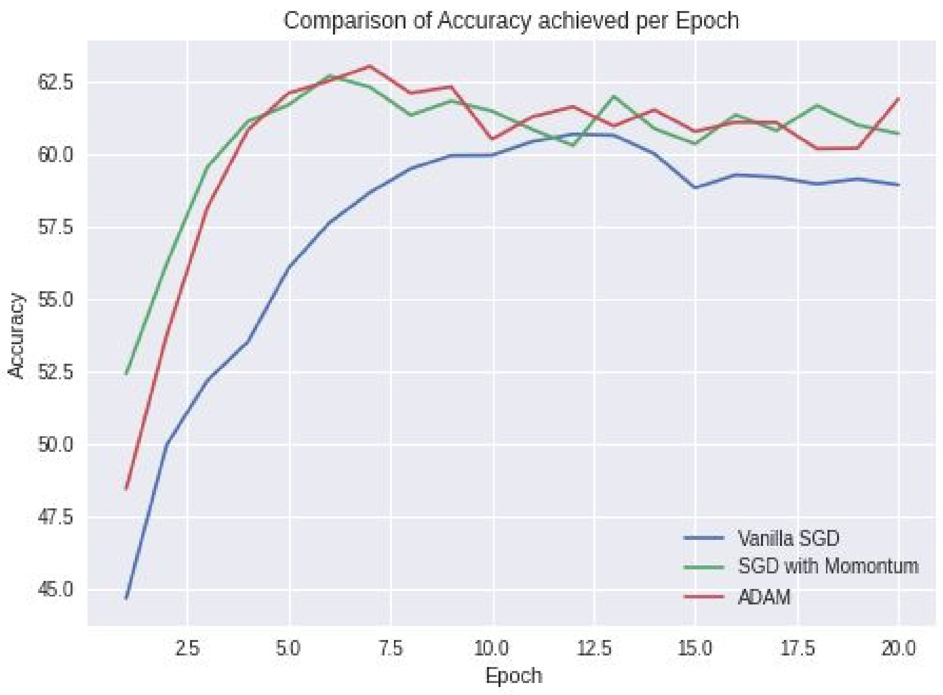
```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1            [-1, 6, 30, 30]             168
         MaxPool2d-2            [-1, 6, 15, 15]               0
            Conv2d-3            [-1, 8, 13, 13]             440
            Conv2d-4           [-1, 10, 13, 13]              90
            Conv2d-5           [-1, 12, 13, 13]             132
            Conv2d-6           [-1, 14, 13, 13]             182
       BatchNorm2d-7           [-1, 14, 13, 13]              28
            Conv2d-8           [-1, 16, 13, 13]             240
            Linear-9                 [-1, 1024]       2,769,920
           Linear-10                  [-1, 512]         524,800
      BatchNorm1d-11                  [-1, 512]           1,024
           Linear-12                  [-1, 256]         131,328
           Linear-13                   [-1, 10]           2,570
================================================================
```

Details of each layer are as follows:

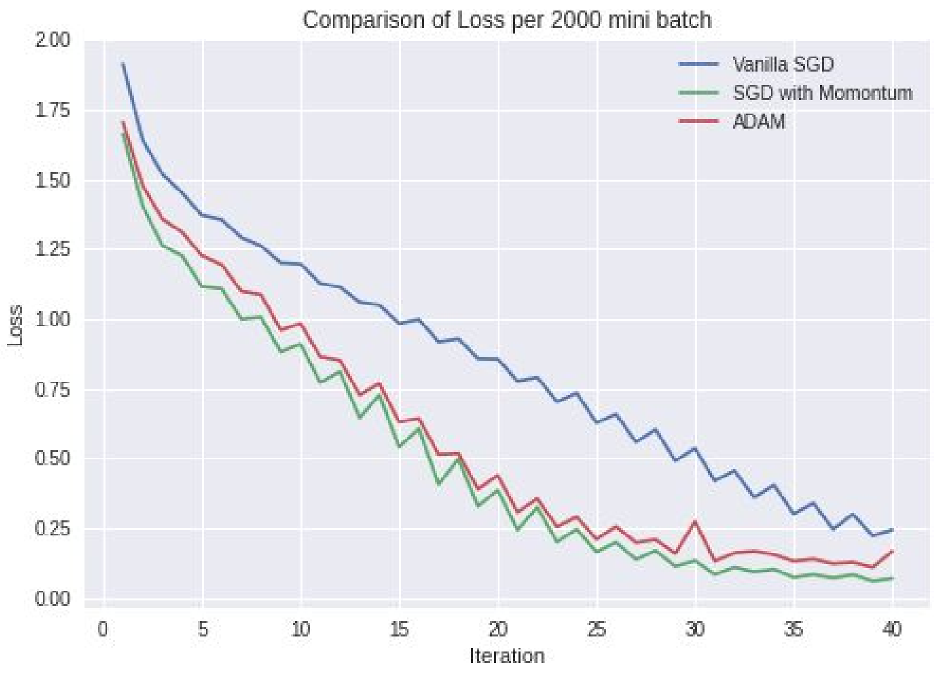


```
Net(
  (conv1): Conv2d(3, 6, kernel_size=(3, 3), stride=(1, 1))
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv2): Conv2d(6, 8, kernel_size=(3, 3), stride=(1, 1))
  (conv3): Conv2d(8, 10, kernel_size=(1, 1), stride=(1, 1))
  (conv4): Conv2d(10, 12, kernel_size=(1, 1), stride=(1, 1))
  (conv5): Conv2d(12, 14, kernel_size=(1, 1), stride=(1, 1))
  (BatchNorm): BatchNorm2d(14, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (conv6): Conv2d(14, 16, kernel_size=(1, 1), stride=(1, 1))
  (fc1): Linear(in_features=2704, out_features=1024, bias=True)
  (fc2): Linear(in_features=1024, out_features=512, bias=True)
  (BatchNorm1): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (fc3): Linear(in_features=512, out_features=256, bias=True)
  (fc4): Linear(in_features=256, out_features=10, bias=True)
)
```

Model is trained for 20 epoch by varying optimization technique i.e. Vannila SGD, SGD with Momentum and ADAM.
Model is analysed at each epoch against test data and the following curve is obtained between variation in accuracy after each epoch using test data for different optimizing techniques:



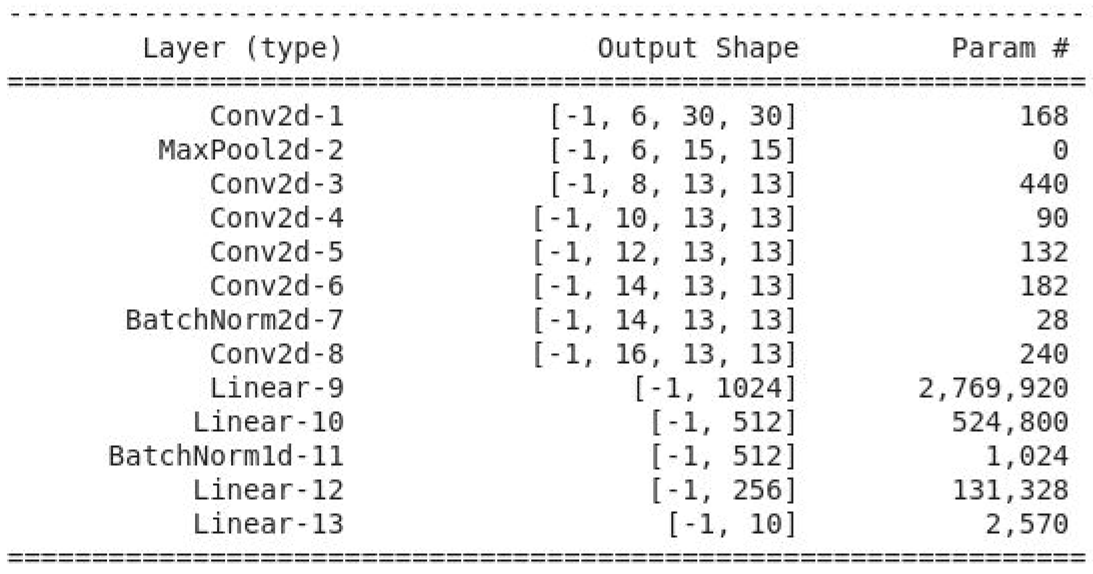
The curve of variation of loss during training are :



II. Accuracy on test data by varying the **Normalization Techniques.**

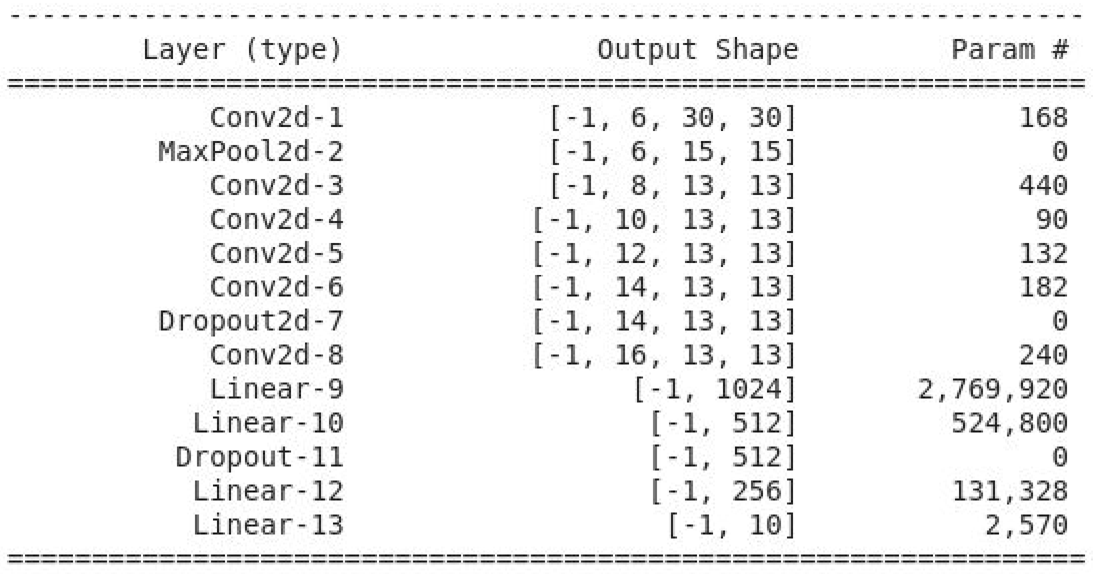
Model Information:

For **Batch Normalization**, CNN Model consists of a total of 13 layers. 6 layers are Convolution Layer, 1 Max pool, 2 Batch Norm and 4 linear layers. 2-D Batch Normalization layer is added after **layer 6** and 1-D Batch Normalisation layer is added after **layer 10**.
Arrangement of these layers are as follow:



```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1            [-1, 6, 30, 30]             168
         MaxPool2d-2            [-1, 6, 15, 15]               0
            Conv2d-3            [-1, 8, 13, 13]             440
            Conv2d-4           [-1, 10, 13, 13]              90
            Conv2d-5           [-1, 12, 13, 13]             132
            Conv2d-6           [-1, 14, 13, 13]             182
       BatchNorm2d-7           [-1, 14, 13, 13]              28
            Conv2d-8           [-1, 16, 13, 13]             240
            Linear-9                 [-1, 1024]       2,769,920
           Linear-10                  [-1, 512]         524,800
      BatchNorm1d-11                  [-1, 512]           1,024
           Linear-12                  [-1, 256]         131,328
           Linear-13                   [-1, 10]           2,570
================================================================
```

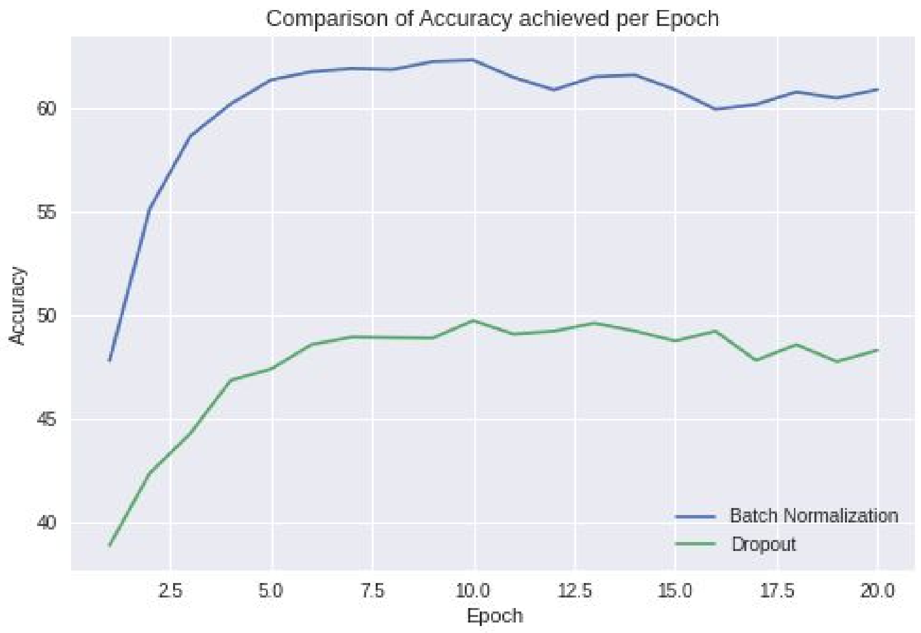
For **Dropout**, CNN Model consists of a total of 13 layers. 6 layers are Convolution Layer, 1 Max pool, 2 Dropout and 4 linear layers. Dropout layer is added after **layer 7 and 10**. Arrangement of these layers are as follow:



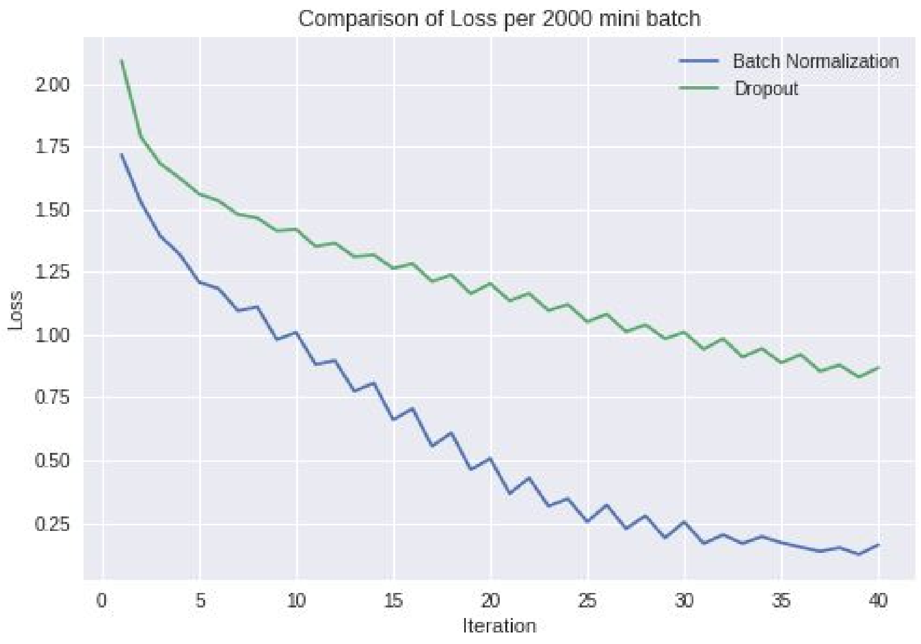
```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1            [-1, 6, 30, 30]             168
         MaxPool2d-2            [-1, 6, 15, 15]               0
            Conv2d-3            [-1, 8, 13, 13]             440
            Conv2d-4           [-1, 10, 13, 13]              90
            Conv2d-5           [-1, 12, 13, 13]             132
            Conv2d-6           [-1, 14, 13, 13]             182
         Dropout2d-7           [-1, 14, 13, 13]               0
            Conv2d-8           [-1, 16, 13, 13]             240
            Linear-9                 [-1, 1024]       2,769,920
           Linear-10                  [-1, 512]         524,800
          Dropout-11                  [-1, 512]               0
           Linear-12                  [-1, 256]         131,328
           Linear-13                   [-1, 10]           2,570
================================================================
```

Model is trained for 20 epoch by varying Normalization technique i.e. Dropout and Batch Normalization.

Model is analysed at each epoch against test data and the following curve is obtained between variation in accuracy after each epoch using test data for different optimizing techniques:



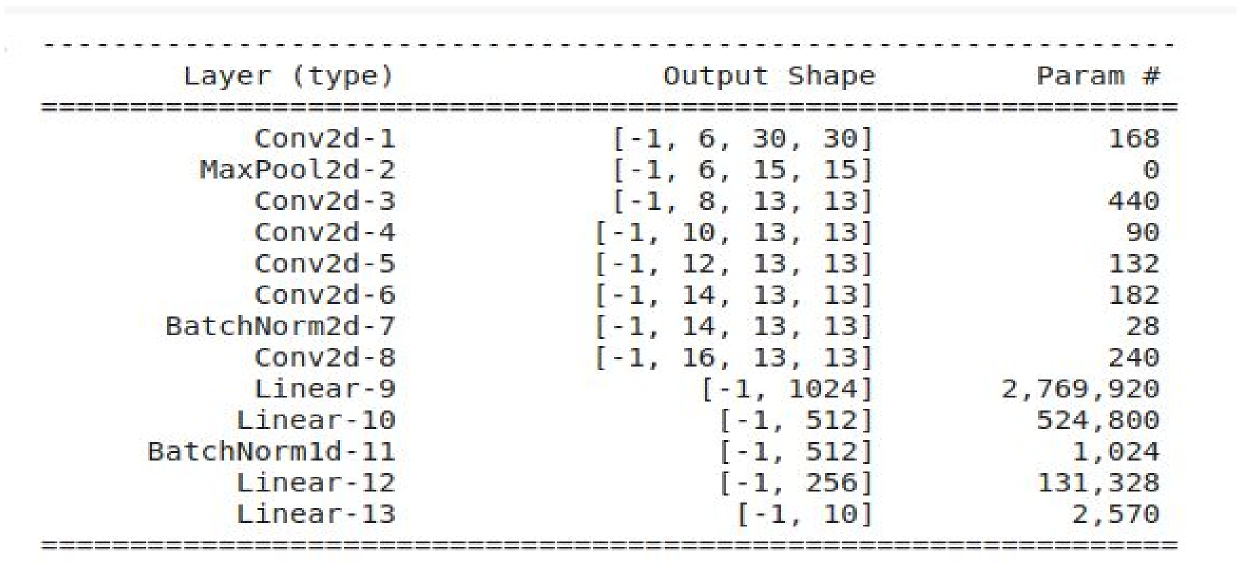
The curve of variation of loss during training are :



III. Accuracy on test data by varying activation function **in between CNN layers**

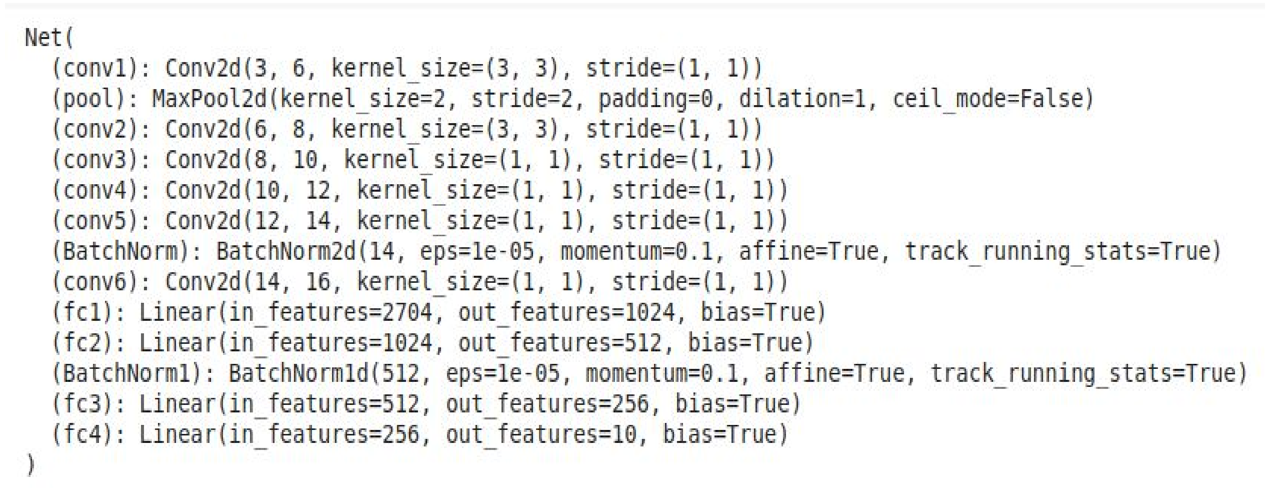
Model Information:

CNN Model consists of a total of 13 layers. 6 layers are Convolution Layer, 1 Max pool, 2 Batch Norm and 4 linear layers.
Arrangement of these layers are as follow:



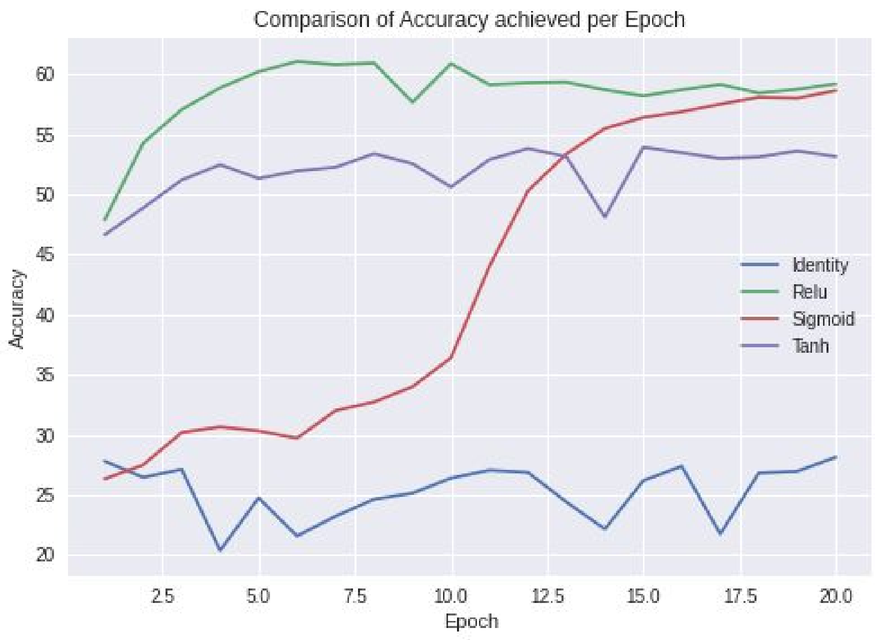
```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1             [-1, 6, 30, 30]             168
         MaxPool2d-2             [-1, 6, 15, 15]               0
            Conv2d-3             [-1, 8, 13, 13]             440
            Conv2d-4            [-1, 10, 13, 13]              90
            Conv2d-5            [-1, 12, 13, 13]             132
            Conv2d-6            [-1, 14, 13, 13]             182
       BatchNorm2d-7            [-1, 14, 13, 13]              28
            Conv2d-8            [-1, 16, 13, 13]             240
            Linear-9                  [-1, 1024]       2,769,920
           Linear-10                   [-1, 512]         524,800
      BatchNorm1d-11                   [-1, 512]           1,024
           Linear-12                   [-1, 256]         131,328
           Linear-13                    [-1, 10]           2,570
================================================================
```

Details of each layer are as follows:

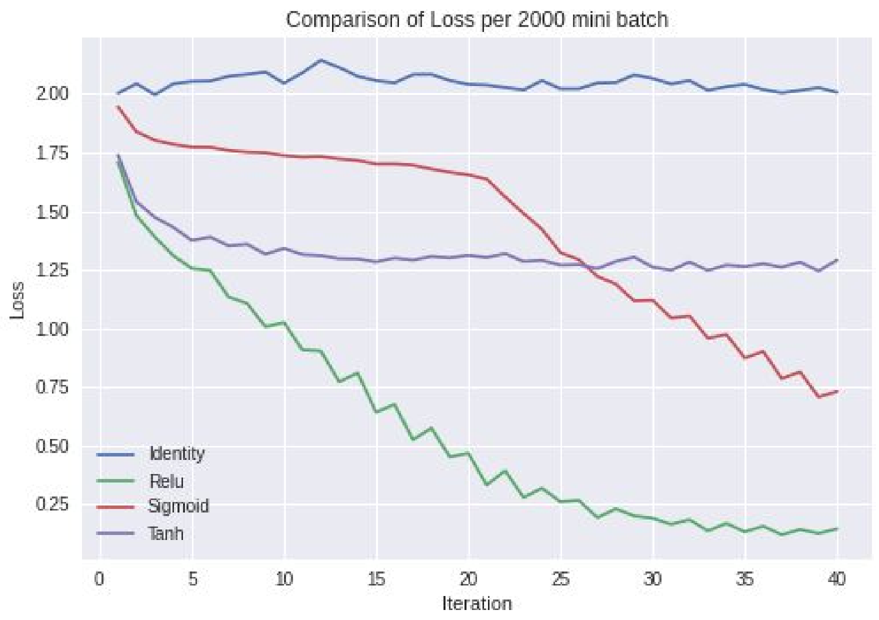


```
Net(
  (conv1): Conv2d(3, 6, kernel_size=(3, 3), stride=(1, 1))
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv2): Conv2d(6, 8, kernel_size=(3, 3), stride=(1, 1))
  (conv3): Conv2d(8, 10, kernel_size=(1, 1), stride=(1, 1))
  (conv4): Conv2d(10, 12, kernel_size=(1, 1), stride=(1, 1))
  (conv5): Conv2d(12, 14, kernel_size=(1, 1), stride=(1, 1))
  (BatchNorm): BatchNorm2d(14, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (conv6): Conv2d(14, 16, kernel_size=(1, 1), stride=(1, 1))
  (fc1): Linear(in_features=2704, out_features=1024, bias=True)
  (fc2): Linear(in_features=1024, out_features=512, bias=True)
  (BatchNorm1): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (fc3): Linear(in_features=512, out_features=256, bias=True)
  (fc4): Linear(in_features=256, out_features=10, bias=True)
)
```

Model is trained for 20 epoch by varying activation function between CNN Layers that are Identity, Relu, Tanh, Sigmoid. Model is analysed at each epoch against test data and the following curve is obtained between variation in accuracy after each epoch using test data for different optimizing techniques:



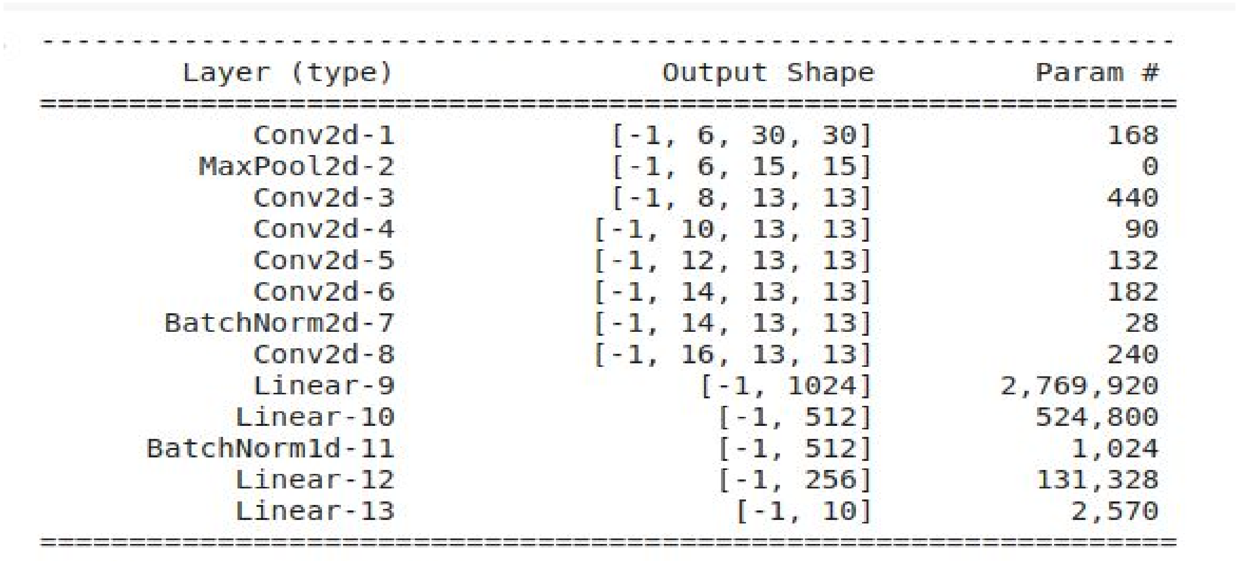
The curve of variation of loss during training are :



IV. Accuracy on test data by varying the **loss function**

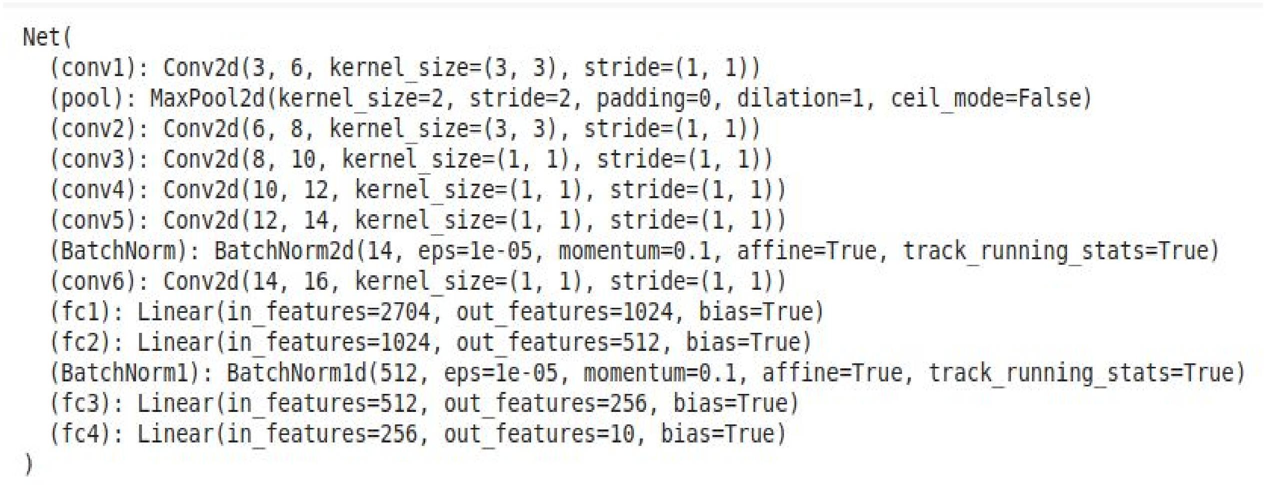
Model Information:

CNN Model consists of a total of 13 layers. 6 layers are Convolution Layer, 1 Max pool, 2 Batch Norm and 4 linear layers.
Arrangement of these layers are as follow:



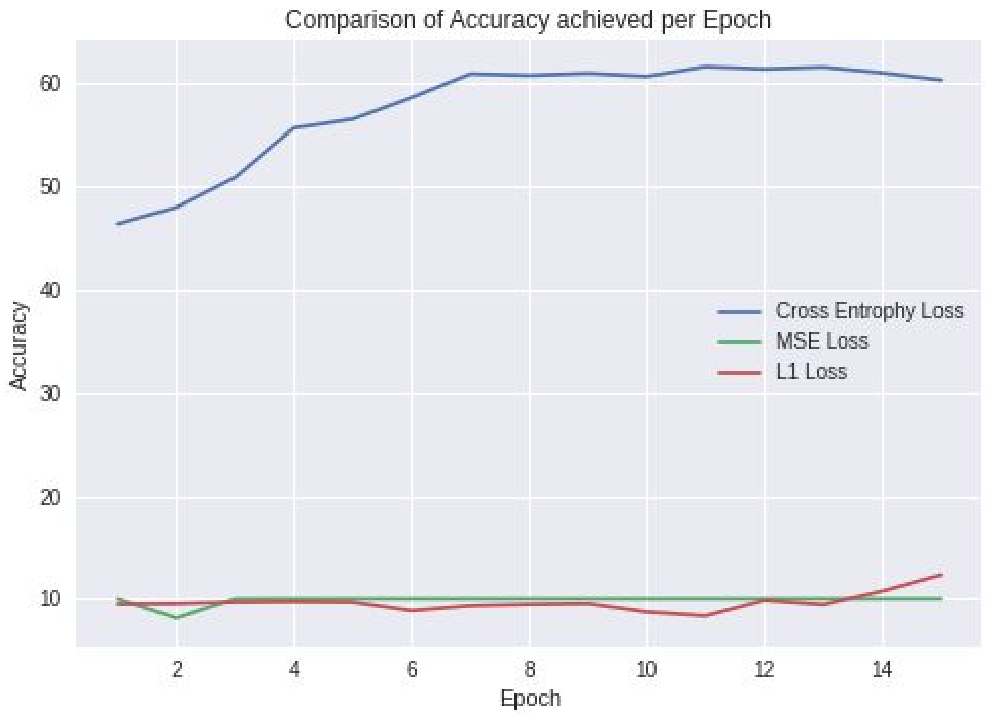
```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1            [-1, 6, 30, 30]             168
         MaxPool2d-2            [-1, 6, 15, 15]               0
            Conv2d-3            [-1, 8, 13, 13]             440
            Conv2d-4           [-1, 10, 13, 13]              90
            Conv2d-5           [-1, 12, 13, 13]             132
            Conv2d-6           [-1, 14, 13, 13]             182
       BatchNorm2d-7           [-1, 14, 13, 13]              28
            Conv2d-8           [-1, 16, 13, 13]             240
            Linear-9                 [-1, 1024]       2,769,920
           Linear-10                  [-1, 512]         524,800
      BatchNorm1d-11                  [-1, 512]           1,024
           Linear-12                  [-1, 256]         131,328
           Linear-13                   [-1, 10]           2,570
================================================================
```

Details of each layer are as follows:



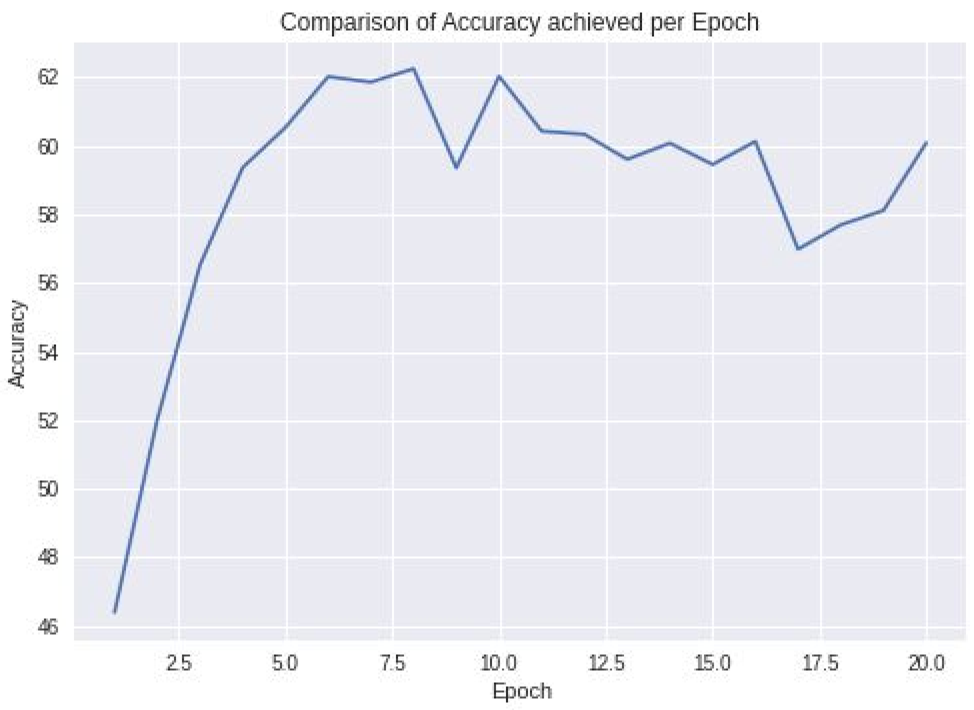
```
Net(
  (conv1): Conv2d(3, 6, kernel_size=(3, 3), stride=(1, 1))
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv2): Conv2d(6, 8, kernel_size=(3, 3), stride=(1, 1))
  (conv3): Conv2d(8, 10, kernel_size=(1, 1), stride=(1, 1))
  (conv4): Conv2d(10, 12, kernel_size=(1, 1), stride=(1, 1))
  (conv5): Conv2d(12, 14, kernel_size=(1, 1), stride=(1, 1))
  (BatchNorm): BatchNorm2d(14, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (conv6): Conv2d(14, 16, kernel_size=(1, 1), stride=(1, 1))
  (fc1): Linear(in_features=2704, out_features=1024, bias=True)
  (fc2): Linear(in_features=1024, out_features=512, bias=True)
  (BatchNorm1): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (fc3): Linear(in_features=512, out_features=256, bias=True)
  (fc4): Linear(in_features=256, out_features=10, bias=True)
)
```

Model is trained for 20 epoch by varying loss function during training that are Cross-Entropy loss, MSE loss, L1 loss. Model is analysed at each epoch against test data and the following curve is obtained between variation in accuracy after each epoch using test data for different optimizing techniques:



Observation:

By observing the graph of Accuracy achieved at each training epoch we came to conclude that the best optimization technique is **ADAM** optimizer, **Batch Normalization** works best on our data then using only Dropout. Loss function which converges best is the **Cross-Entropy Loss** and activation function which works best for our data is **Relu**. Using the best configuration for CNN Model the accuracy achieved at each epoch is obtained as follow:


Comparison of Accuracy achieved per Epoch
Accuracy
62
60
58
56
54
52
50
48
46
Epoch
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

Q2. Download a ResNet 50 trained on the ImageNet classification dataset and perform Transfer Learning.

I. Use the features extracted from the last fully-connected layer and train binary SVM classifier for a category [from CIFAR-10 categories] of your choice.

Model Information:

A ResNet-50 model trained on the ImageNet classification dataset is used. It is a convolutional neural network that is 50 layers deep. And at the end SVM classifier for binary classification is plugged.

The class selected for binary classification is **Bird** Class

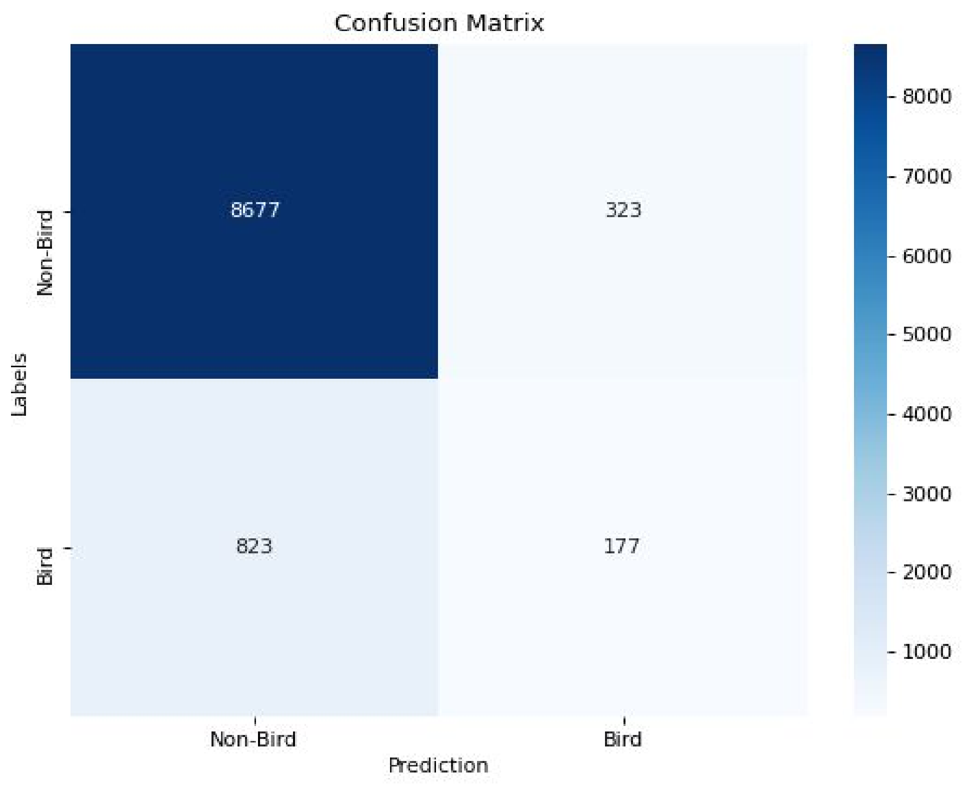
Approach:

The whole process can be summarized in the following steps:

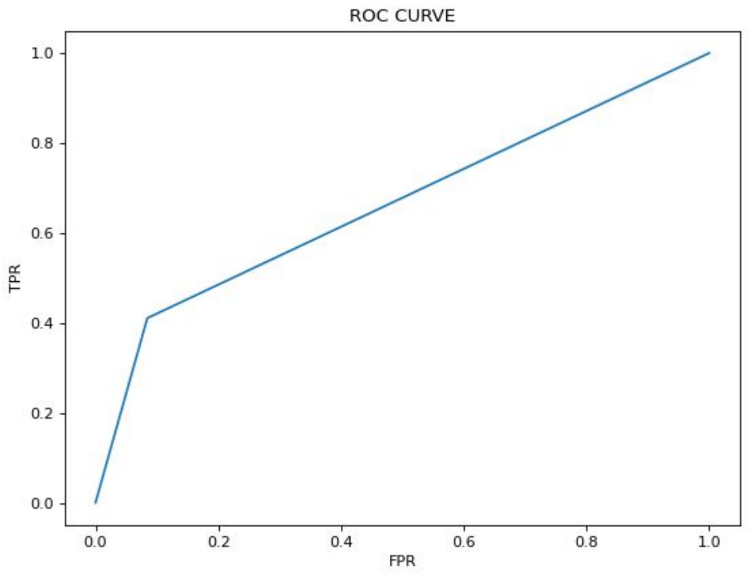
1. First, we import the pre-trained ResNet 50 model.
2. Now, pass data through this pre-trained model. This will Output data with 10000 features.
3. Now, choose one class which you want to classify. For this assignment bird class is chosen.
4. Now, fit this data to SVM. It will learn to classify data to bird and non- bird classes.
5. Now, test this model with SVM prediction.

Observation:

Model is tested for 10,000 data samples. Accuracy on test data is : **89.14**
Following confusion matrix is obtained :



Receiver Operating Characteristic Curve obtained from false positive rate and true positive rate are as follow:



II. Fine-tune the ResNet 50 model (you may choose what layers to fine-tune) for the CIFAR-10 dataset, and evaluate the classification performance on the test set before and after fine-tuning with respect to the following metrics.

Model Information:

A **ResNet-50** model trained on the ImageNet classification dataset is used. For finding performance before fine-tuning, a non pre-trained model of resnet is imported and the model is allowed to learn parameters for 10 epochs. And for fine tuning a pre-trained model is imported and then it is allowed to be fine tuned for 10 epochs.

Observation:

Before Fine Tuning (Training from scratch):

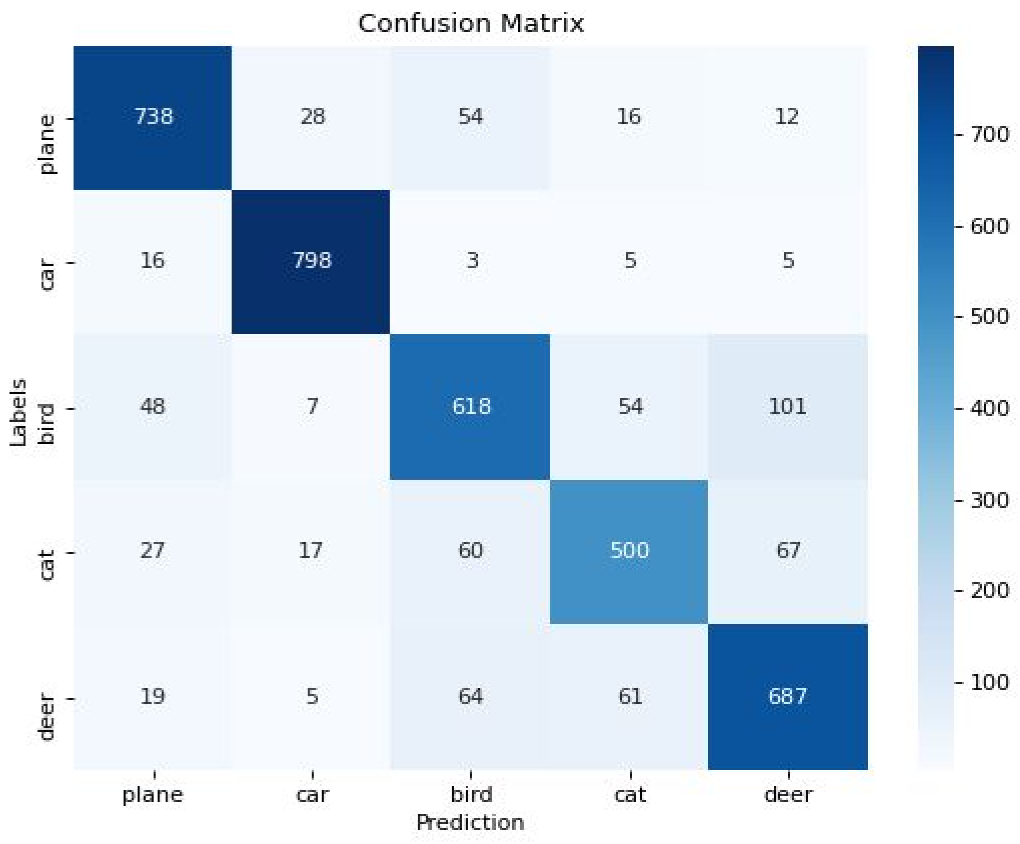
Accuracy of Classes obtained as follow:

```
Accuracy of class 0 i.e belongs to plane is 73.8
Accuracy of class 1 i.e belongs to car is 79.8
Accuracy of class 2 i.e belongs to bird is 61.8
Accuracy of class 3 i.e belongs to cat is 50.0
Accuracy of class 4 i.e belongs to deer is 68.7
Accuracy of class 5 i.e belongs to dog is 61.3
Accuracy of class 6 i.e belongs to frog is 83.0
Accuracy of class 7 i.e belongs to horse is 75.2
Accuracy of class 8 i.e belongs to ship is 83.7
Accuracy of class 9 i.e belongs to truck is 82.3
```

Confusion matrix for first five classes are as follow:

```
-------Confusion Matrix-------
     738   28   54   16   12
     16    798  3    5    5
     48    7    618  54   101
     27    17   60   500  67
     19    5    64   61   687
```

Heat map of Confusion Matrix is:



After Fine Tuning :

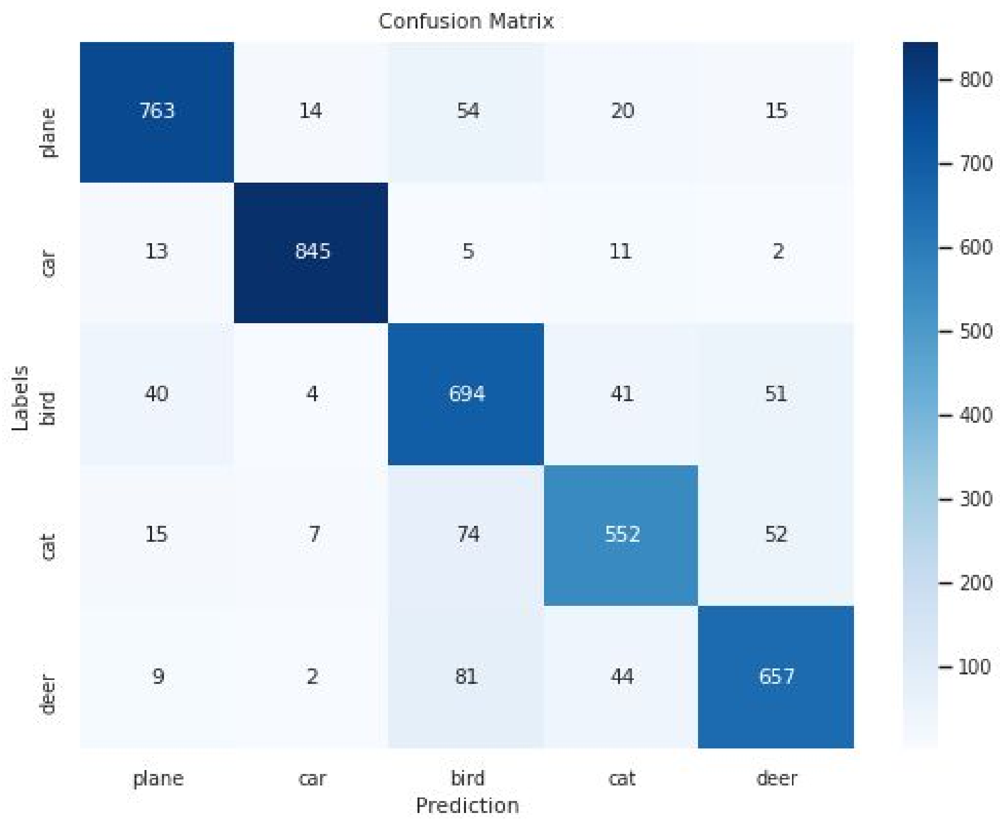
Accuracy of class is obtained as follow:

```
Accuracy of class 0 i.e belongs to plane is 81.3
Accuracy of class 1 i.e belongs to car is 84.4
Accuracy of class 2 i.e belongs to bird is 73.7
Accuracy of class 3 i.e belongs to cat is 70.1
Accuracy of class 4 i.e belongs to deer is 83.4
Accuracy of class 5 i.e belongs to dog is 73.4
Accuracy of class 6 i.e belongs to frog is 83.0
Accuracy of class 7 i.e belongs to horse is 84.5
Accuracy of class 8 i.e belongs to ship is 87.9
Accuracy of class 9 i.e belongs to truck is 87.9
```

Confusion matrix of first 5 classes are as follow:

```
-------Confusion Matrix-------
   813   11    44    30    16
   12    844   4     14    3
   30    1     737   62    70
   15    4     33    701   56
   9     2     47    31    834
```

Heat map of confusion matrix is:



References:

1. https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html
2. https://pytorch.org/tutorials/beginner/finetuning_torchvision_models_tutorial.html
3. https://cs231n.github.io/linear-classify/#interpret
4. https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826
5. https://www.analyticsvidhya.com/blog/2019/10/how-to-master-transfer-learning-using-pytorch/
6. https://towardsdatascience.com/transfer-learning-with-convolutional-neural-networks-in-pytorch-dd09190245ce
7. https://www.kaggle.com/pmigdal/transfer-learning-with-resnet-50-in-pytorch