COL341 : Machine Learning
Assignment 2.2

Gaurav Jain (2019CS10349)

1 Part (a) : CNN for Devanagari Dataset

- **Observations**
  - The accuracy on the public test data is approximately 0.98 after 8 epochs and across multiple runs of the network.
  - The cross entropy loss on the training data is approximately 0.049 after 8 epochs and across multiple runs of the network.
  - Stochastic behavior is observed in test accuracy and training loss due to random initialization of the weights.
- **Plot of Training Loss v/s Number of Epochs** The following plot is observed for the training loss (Y axis) and number of epochs (X axis):

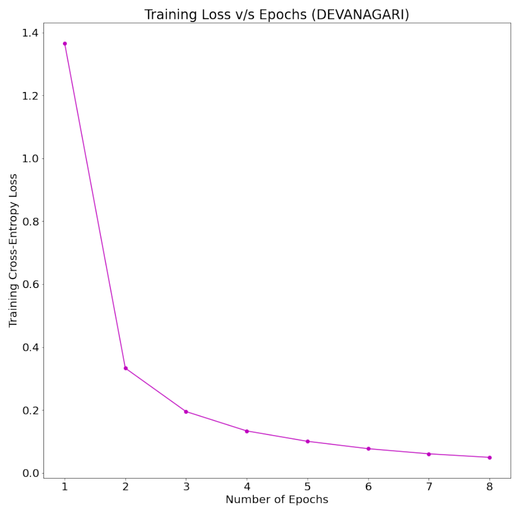

Figure 1: Training Loss v/s Number of Epochs (DEVANAGARI)

- **Plot of Test Accuracy v/s Number of Epochs** The following plot is observed for the accuracy on the public test data (Y axis) and number of epochs (X axis):

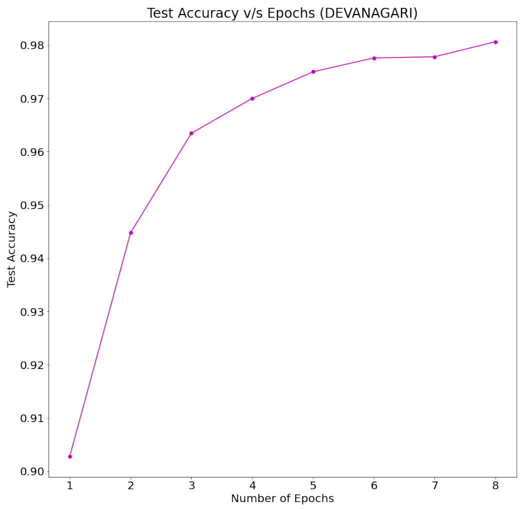

Figure 2: Test Accuracy v/s Number of Epochs (DEVANAGARI)

- **Comparison with Traditional Neural Network**

  The accuracy on the public test data obtained using the traditional neural network consisting of only linear layers is approximately 0.95. The accuracy obtained using CNN is approximately 0.98.
  The timeout limit in first case is 15 minutes whereas the time taken in CNN is less than 10 minutes.

2 Part (b) : CNN for CIFAR10 Dataset

- **Observations**
  - The accuracy on the public test data is approximately 0.70 after 5 epochs and across multiple runs of the network.
  - The cross entropy loss on the training data is approximately 0.64 after 5 epochs and across multiple runs of the network.
  - Stochastic behavior is observed in test accuracy and training loss due to random initialization of the weights.
- **Plot of Training Loss v/s Number of Epochs**

  The following plot is observed for the training loss (Y axis) and number of epochs (X axis):

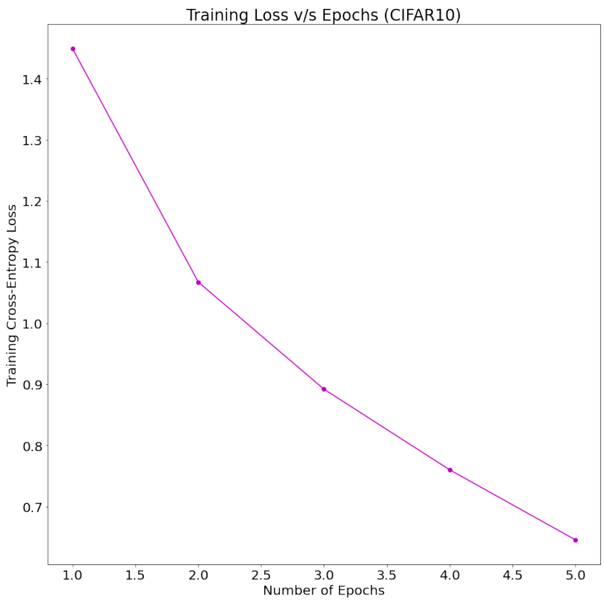

Figure 3: Training Loss v/s Number of Epochs (CIFAR10)

- **Plot of Test Accuracy v/s Number of Epochs**

  The following plot is observed for the accuracy on the public test data (Y axis) and number of epochs (X axis):

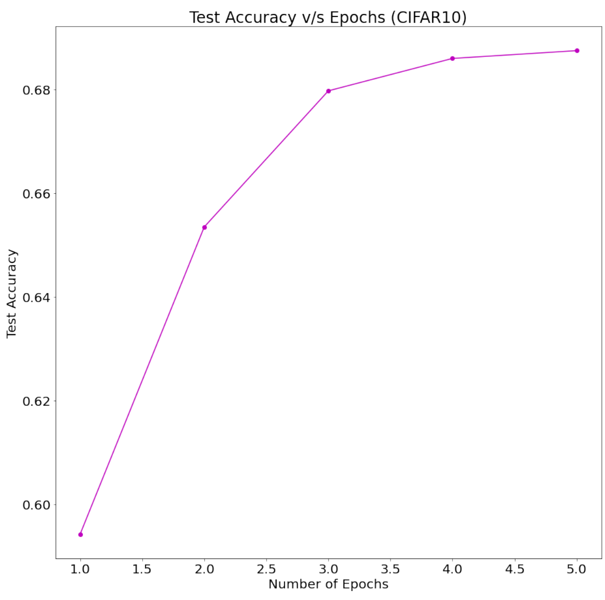

Figure 4: Test Accuracy v/s Number of Epochs (CIFAR10)

3 Part (c) : Best Architecture of CNN for CIFAR10 Dataset

The current architecture gives an accuracy of at most 70% which is not a good score for a CNN. The total number of parameters using this architecture is 2.7 million. So, it is necessary to improve the accuracy by changing parameters like learning rate, learning rate method, number of convolution and linear layers and so on.
The following parameters are changed to improve the accuracy on CIFAR10 dataset:

- **Learning rate of Adam optimizer**

  Learning rates of different orders are compared by using them to compute test accuracy on public test data. The following plot is obtained:

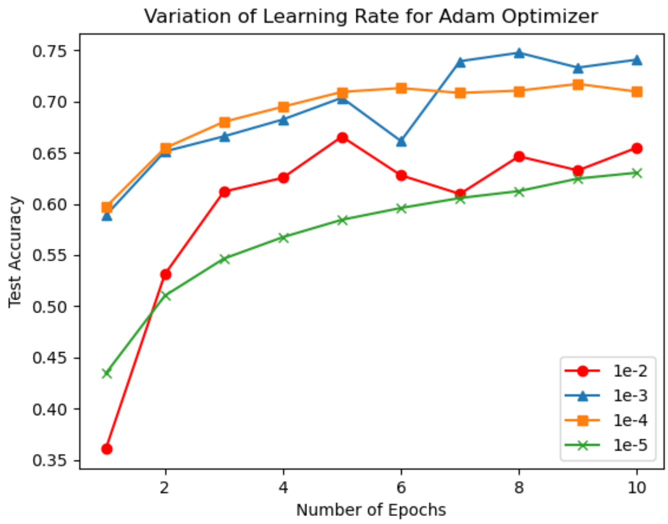

Figure 5: Training Loss v/s Number of Epochs with different learning rates

  As visible in the above graph, using 1e-3 as learning rate improves the accuracy by 4% after 10 epochs compared to using 1e-4 as learning rate.

- **Optimizers**

  Optimizers like SGD, SGD with momentum, Nesterov SGD, Adam are compared to find the best optimizer among these optimizers. Default values of the hyperparamters are used for this experiment. The following plot is obtained as a result of this experiment:

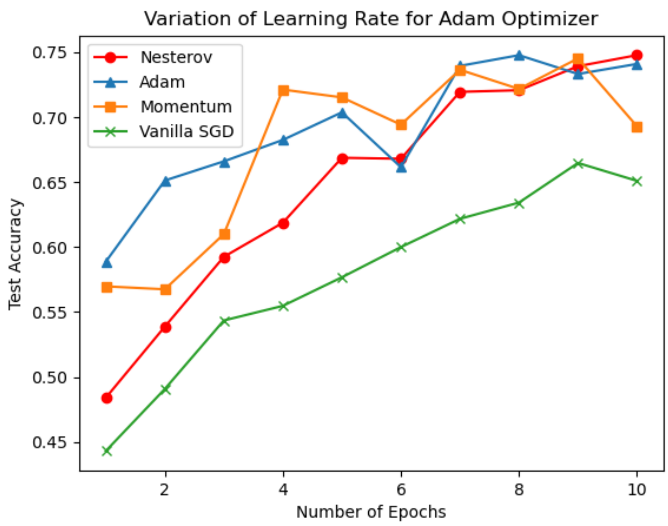

Figure 6: Training Loss v/s Number of Epochs with different Optimizers

As visible in the above graph, Nesterov gives the best accuracy after 10 epochs. The accuracy score is now 75% which means a 5% improvement! This optimizer is used for further experimentation.

- **Data Augmentation**

  Training data usually contains images of similar nature. A common strategy to improve accuracy is to normalize pixels of both train and test data. While training, images are also transformed by cropping them randomly and flipping the images horizontally at random. These steps allows us to overcome the problem of over-fitting.

  Accuracy obtained after transforming train and test data is listed below:

No of Epochs	Test Accuracy	Training Loss
1	0.492	1.487
2	0.547	1.245
3	0.603	1.142
4	0.638	1.056
5	0.681	0.964
6	0.695	0.903
7	0.736	0.832
8	0.744	0.788
9	0.759	0.747
10	0.762	0.712

  We can see an increase of 2% in the test accuracy after 10 epochs by using Adam optimizer with 1e-3 learning rate.

- **Adaptive Learning Rate**

  It is generally observed that adaptive learning rate has better accuracy compared to fixed learning rate strategy. Thus we should experiment on various adaptive learning rate strategies available in PyTorch and select the one with best accuracy for designing the best CNN architecture.

  Adaptive learning rate strategies present in *torch.optim.lr_scheduler* like *ExponentialR, CyclicLR* are used in this experiment. The following table summaries the outcomes:

No of Epochs	Test Accuracy (Exponential LR)	Test Accuracy (Cyclic LR)
1	0.516	0.539
2	0.558	0.572
3	0.612	0.620
4	0.644	0.658
5	0.683	0.697
6	0.701	0.715
7	0.732	0.741
8	0.741	0.756
9	0.754	0.763
10	0.768	0.782

  CyclicLR performs much better compared to ExponentialLR after 10 epochs and thus it is selected for best architecture implementation.

- **Convolution Layers**

  It is not possible to add a new layer in the given architecture as this results in total number of parameters to go above the limit of 3 million. Thus to improve accuracy, we need to change the in and out channels of different layers.

  Tweaking channels of convolution layers only changes accuracy by 1-2%. To see a big change in the accuracy, we need to consider some ideas based on SOTA implementations on CIFAR10 dataset. One of the most famous implementations is ResNet. The main concept used in ResNet is residual blocks with same input and output channel sizes and short circuited by the block input.

  The accuracy on CIFAR10 can shoot upto 94% using the original implementation but this implementation has 18 million parameters. To limit the number of total parameters below 3 million, we need to use residual blocks of small sizes. We can only add at max 2 such blocks in our implementations.

Thus, the main experimentation in this part contains changing number of channels in residual block like 32,64 and 256. Number of blocks are also varied to obtain the best architecture.

The following table containing training loss and test accuracy for 25 epochs summaries the overall experimentation process with selected optimizer, scheduler and data augmentation and 2 ResNet blocks of size 32 and 256:

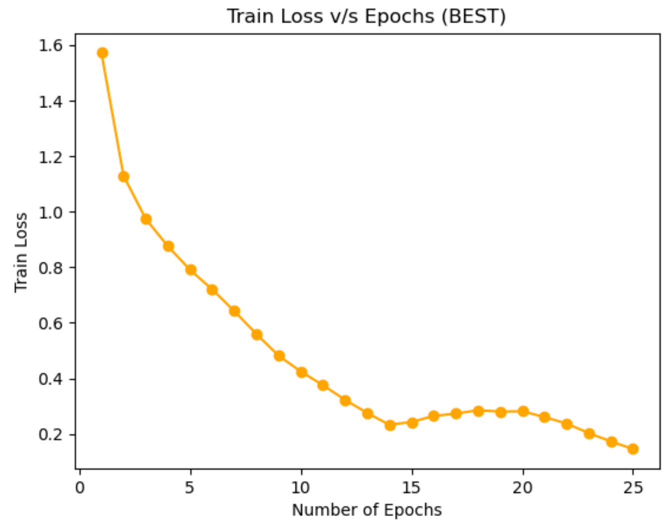

Figure 7: Training Loss v/s Number of Epochs with Best Architecture

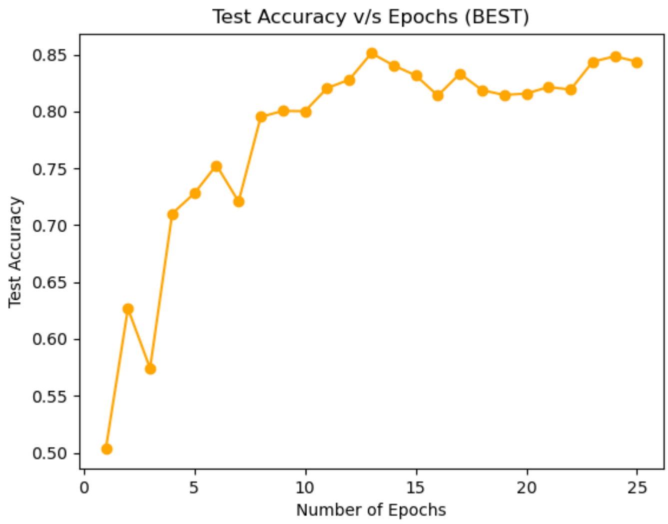

Figure 8: Test Accuracy v/s Number of Epochs with Best Architecture