

Report Project 2
Deep learning by PyTorch

Name: *Jashanraj Singh Gosain*

Student Number: *301435386*

Late Days: *None used for this assignment.*

Team Members: *Archit Verma and Hitesh Juneja*

Question 1:

Question one is submitted as a group in Kaggle competition. The name of group and group members (including me) are as follows:

Group Name: GOATs

Group Members:

1) Jashanraj Singh Gosain

2) Archit Verma

3) Hitesh Juneja

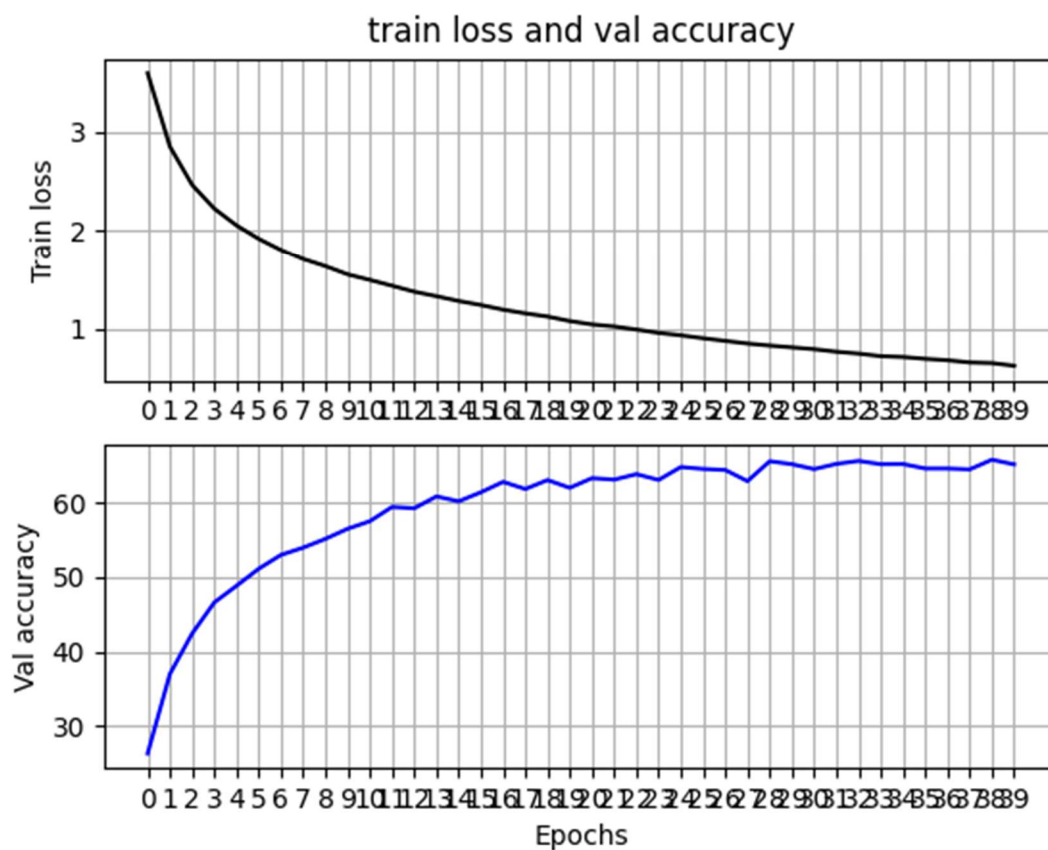
The architecture of the Convolution Neural network in this assignment is as follows:

Layer No.	Layer Type	Kernel Size (for conv layers)	Input Output dimension	Input Output Channels (for conv layers)
1	conv2d	3	32 32	3 64
2	conv2d	3	32 32	64 64
3	BatchNorm2d	-	32 32	-
4	MaxPool	2	32 16	64 64
5	conv2d	3	16 16	64 128
6	conv2d	3	16 16	128 128
7	BatchNorm2d	-	16 16	-
8	MaxPool	2	16 8	128 128
9	conv2d	3	8 8	128 256
10	conv2d	3	8 8	256 256
11	BatchNorm2d	-	8 8	-
12	MaxPool	2	8 4	256 256
13	conv2d	3	4 4	256 512
14	conv2d	3	4 4	512 512
15	BatchNorm2d	-	4 4	-
16	MaxPool	2	4 2	512 512

This architecture achieves a val accuracy of roughly 65 %. There are 8 convolution layers in this network. In each layer first we apply convolution to increase the output channels. After that we use batchNormalization and Relu layers to improve the performace. In each layer, we apply convolution with same input and output channels. This prevents the spatial dimensions of the input feature maps after applying the convolution.

After 50 Epochs, the model achieves an accuracy of 65%.

Plot of the model accuracy and training loss



The overall score on Kaggle is 0.649

Ablation study

- Initially I followed a basic network consisting of variety of convolutional layers:
Conv 6 => Conv 16 => Conv 32 => Conv 64 => Conv 128 => Conv 256 => Conv 512 => Conv 1024 along with batchNormalization layers in between. Followed by fc_layers in the following order: linear, batchNormalization, Relu, linear.
This above architecture was giving me almost zero accuracy.
I figured once we reach the 1x1 size. There was too much information lost in the next convolutional layers. Reducing helped increase the performance.
- In data normalization, I used the RandomResizedCrop instead of Random crop which helped me further improve the performance of the model. Also using the scale (0.5,1.0) highly improved the performance of the model. This makes sure atleast half the image size is represented in the random crops. It improved the performance from 60 upto 65 % val accuracy.
- Using conv layer with kernel 5 would not allow to apply more than 2-3 pool layers as the input dimensions will be smaller than output dimensions. Applying kernel 3 helped with the need of more pool layers. Adding more pooling layers after each iteration helped improve the performance from 40% to 50%
- Adding convolution layers without having any change in channel length helped with improving the performance. If convolution layers increased output channels without having a layer with same channel length in between it resulted in loss of information.
- Weight decay helped improve the accuracy by 2-3 percent. It helps mitigating overfitting as model would be stuck around 60%.

Question 2

The model with fine tuning just the last layer: **RESNET_LAST_ONLY = True**

For tuning just the last layer the training accuracy comes out to be 0.8260 after 30 Epoch.

However, the test accuracy dips down to 0.4652 for the learning rate 0.001

 `test(model, criterion)`

Test Loss: 0.2684 Test Accuracy 0.4652

```
TRAINING Epoch 1/30 Loss 0.6543 Accuracy 0.0200
TRAINING Epoch 2/30 Loss 0.5673 Accuracy 0.0993
TRAINING Epoch 3/30 Loss 0.4971 Accuracy 0.2260
TRAINING Epoch 4/30 Loss 0.4405 Accuracy 0.3317
TRAINING Epoch 5/30 Loss 0.3973 Accuracy 0.4140
TRAINING Epoch 6/30 Loss 0.3574 Accuracy 0.4763
TRAINING Epoch 7/30 Loss 0.3249 Accuracy 0.5410
TRAINING Epoch 8/30 Loss 0.2994 Accuracy 0.5637
TRAINING Epoch 9/30 Loss 0.2760 Accuracy 0.6103
TRAINING Epoch 10/30 Loss 0.2615 Accuracy 0.6227
TRAINING Epoch 11/30 Loss 0.2413 Accuracy 0.6580
TRAINING Epoch 12/30 Loss 0.2284 Accuracy 0.6807
TRAINING Epoch 13/30 Loss 0.2149 Accuracy 0.7010
TRAINING Epoch 14/30 Loss 0.2027 Accuracy 0.7143
TRAINING Epoch 15/30 Loss 0.1960 Accuracy 0.7140
TRAINING Epoch 16/30 Loss 0.1858 Accuracy 0.7377
TRAINING Epoch 17/30 Loss 0.1735 Accuracy 0.7560
TRAINING Epoch 18/30 Loss 0.1687 Accuracy 0.7670
TRAINING Epoch 19/30 Loss 0.1650 Accuracy 0.7543
TRAINING Epoch 20/30 Loss 0.1551 Accuracy 0.7823
TRAINING Epoch 21/30 Loss 0.1507 Accuracy 0.8013
TRAINING Epoch 22/30 Loss 0.1492 Accuracy 0.7803
TRAINING Epoch 23/30 Loss 0.1424 Accuracy 0.7993
TRAINING Epoch 24/30 Loss 0.1391 Accuracy 0.7950
TRAINING Epoch 25/30 Loss 0.1355 Accuracy 0.8090
TRAINING Epoch 26/30 Loss 0.1295 Accuracy 0.8193
TRAINING Epoch 27/30 Loss 0.1282 Accuracy 0.8200
TRAINING Epoch 28/30 Loss 0.1253 Accuracy 0.8183
TRAINING Epoch 29/30 Loss 0.1218 Accuracy 0.8217
TRAINING Epoch 30/30 Loss 0.1206 Accuracy 0.8260
Finished Training
-----
```

The model with fine tuning the entire network: RESNET_LAST_ONLY = False

It achieves a training accuracy of 0.9907 in 20 Epoch with learning rate= 0.001.

The test Accuracy when fine tuning just the last layer is 0.5948.

Higher learning rate would have high loss and Accuracy low. Accuracy starts to fluctuate and sometimes dip down the previous accuracy.

To low training reduced the accuracy achievements for this model. Requiring more and more Epoch.

```
▶ test(model, criterion)
```

```
📁 Test Loss: 0.1965 Test Accuracy 0.5948
```

```
TRAINING Epoch 1/20 Loss 0.6179 Accuracy 0.0480
TRAINING Epoch 2/20 Loss 0.4459 Accuracy 0.2587
TRAINING Epoch 3/20 Loss 0.3313 Accuracy 0.4440
TRAINING Epoch 4/20 Loss 0.2539 Accuracy 0.6017
TRAINING Epoch 5/20 Loss 0.1938 Accuracy 0.7077
TRAINING Epoch 6/20 Loss 0.1500 Accuracy 0.7947
TRAINING Epoch 7/20 Loss 0.1165 Accuracy 0.8540
TRAINING Epoch 8/20 Loss 0.0934 Accuracy 0.8953
TRAINING Epoch 9/20 Loss 0.0742 Accuracy 0.9270
TRAINING Epoch 10/20 Loss 0.0612 Accuracy 0.9460
TRAINING Epoch 11/20 Loss 0.0491 Accuracy 0.9577
TRAINING Epoch 12/20 Loss 0.0410 Accuracy 0.9677
TRAINING Epoch 13/20 Loss 0.0326 Accuracy 0.9787
TRAINING Epoch 14/20 Loss 0.0286 Accuracy 0.9800
TRAINING Epoch 15/20 Loss 0.0238 Accuracy 0.9880
TRAINING Epoch 16/20 Loss 0.0197 Accuracy 0.9903
TRAINING Epoch 17/20 Loss 0.0183 Accuracy 0.9887
TRAINING Epoch 18/20 Loss 0.0167 Accuracy 0.9893
TRAINING Epoch 19/20 Loss 0.0159 Accuracy 0.9897
TRAINING Epoch 20/20 Loss 0.0159 Accuracy 0.9907
Finished Training
-----
```