

Project 2: Deep learning by PyTorch

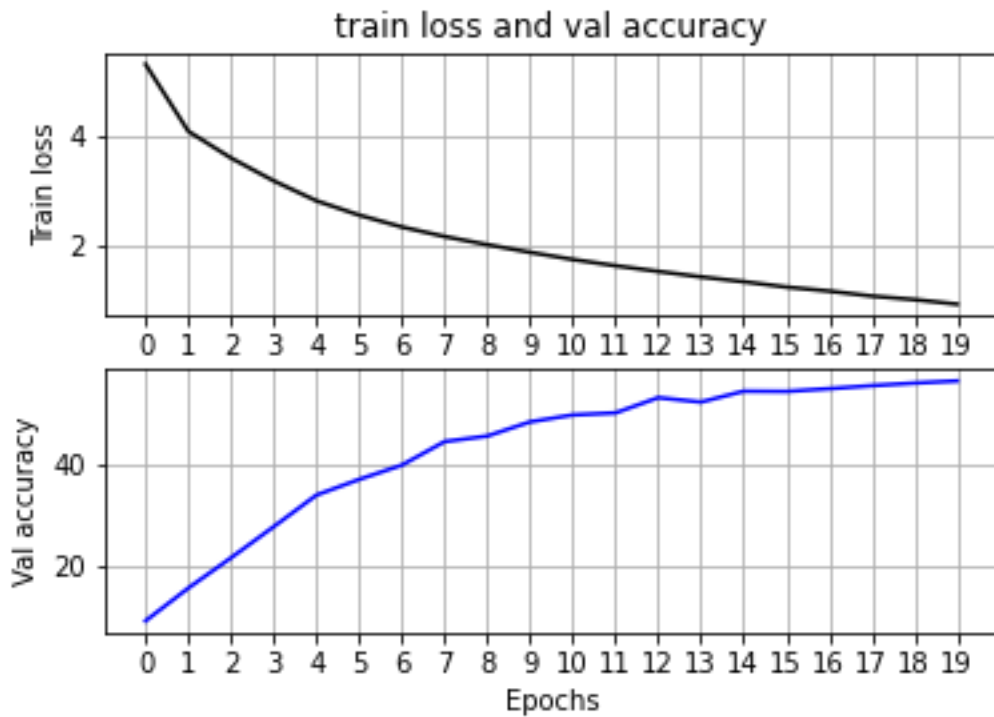
Part 1: Improving BaseNet on CIFAR100

For improving the basenet, I was inspired by VGG-19 and RESNET18 model architecture so I incorporated them into the base net. Firstly, I added various convolution layer to the model, starting from 64 channels to 1024 channels. Then based on resnet architecture, I created sequential convolution blocks (containing 2 same channel convolution layers) after each increase in no of channels for convolution layers, and implemented residual addition after each block. While, VGG model requires convolution filter to be 3x3, I chose to keep the filter 5x5 to convolve more data. I also reduced testing and training batchsize to 16 and 32 respectively, to decrease the amount of data being generalized during convolution. I augmented the training data, by random cropping input image to size 26 (80% of original size), and flipping images horizontally on random.

MyNet Architecture:

Layer No.	Layer Type	Kernel size	Input Output dimensions	Input Output Channels
1	conv2d	5	32 32	3 64
2	relu	-	32 32	-
3	Maxpool2d	2	32 16	-
4	Conv2d	5	16 16	64 128
5	relu	-	16 16	-
6	Conv2d x2 relu	5	16 16	128 128
7	Conv2d	5	16 16	128 256
8	relu	-	16 16	-
9	Conv2d x2 relu	5	16 16	256 256
10	Maxpool2d	2	16 8	-
11	Conv2d	5	8 8	256 512
12	relu	-	8 8	-
13	Conv2d x3 relu	5	8 8	512 512
14	Conv2d	5	8 8	512 1024
15	relu	-	8 8	-
16	Conv2d x3 relu	5	8 8	1024 1024
17	Maxpool2d	2	8 4	-
18	linear	-	16384 100	-

The model was trained for 20 Epochs, with learning rate of 0.001 and momentum of 0.9. This enabled the model to reach training accuracy of 54%.



The plot for train loss and val accuracy over 20 epochs give us a clear understanding of our neural network's performance with peak validation accuracy of 56%.

Part 2: Transfer Learning

When training the ResNet as a fixed feature extractor,

The model reaches the max accuracy of 50%, with least loss around 0.5896 at Epoch 29

```
TRAINING Epoch 1/30 Loss 1.3461 Accuracy 0.0073
TRAINING Epoch 2/30 Loss 1.2567 Accuracy 0.0357
TRAINING Epoch 3/30 Loss 1.1827 Accuracy 0.0693
TRAINING Epoch 4/30 Loss 1.1193 Accuracy 0.1067
TRAINING Epoch 5/30 Loss 1.0618 Accuracy 0.1433
TRAINING Epoch 6/30 Loss 1.0141 Accuracy 0.1937
TRAINING Epoch 7/30 Loss 0.9748 Accuracy 0.2150
TRAINING Epoch 8/30 Loss 0.9400 Accuracy 0.2507
TRAINING Epoch 9/30 Loss 0.8988 Accuracy 0.2927
TRAINING Epoch 10/30 Loss 0.8716 Accuracy 0.3053
TRAINING Epoch 11/30 Loss 0.8356 Accuracy 0.3210
TRAINING Epoch 12/30 Loss 0.8134 Accuracy 0.3443
TRAINING Epoch 13/30 Loss 0.7887 Accuracy 0.3603
TRAINING Epoch 14/30 Loss 0.7781 Accuracy 0.3750
TRAINING Epoch 15/30 Loss 0.7576 Accuracy 0.3870
TRAINING Epoch 16/30 Loss 0.7319 Accuracy 0.3980
TRAINING Epoch 17/30 Loss 0.7118 Accuracy 0.4217
TRAINING Epoch 18/30 Loss 0.6972 Accuracy 0.4303
TRAINING Epoch 19/30 Loss 0.6869 Accuracy 0.4337
TRAINING Epoch 20/30 Loss 0.6790 Accuracy 0.4520
TRAINING Epoch 21/30 Loss 0.6604 Accuracy 0.4627
TRAINING Epoch 22/30 Loss 0.6563 Accuracy 0.4500
TRAINING Epoch 23/30 Loss 0.6429 Accuracy 0.4593
TRAINING Epoch 24/30 Loss 0.6217 Accuracy 0.4823
TRAINING Epoch 25/30 Loss 0.6238 Accuracy 0.4893
TRAINING Epoch 26/30 Loss 0.6124 Accuracy 0.4947
TRAINING Epoch 27/30 Loss 0.6068 Accuracy 0.4860
TRAINING Epoch 28/30 Loss 0.5904 Accuracy 0.5060
TRAINING Epoch 29/30 Loss 0.5896 Accuracy 0.5060
TRAINING Epoch 30/30 Loss 0.5959 Accuracy 0.4950
Finished Training
```

And the test accuracy and loss is:

```
Test Loss: 0.5770 Test Accuracy 0.4237
```

Whereas, when the resnet is trained on whole model network, We reach max accuracy of 75.23% and min loss of 0.28 at epoch 29

```
TRAINING Epoch 1/30 Loss 1.3096 Accuracy 0.0173
TRAINING Epoch 2/30 Loss 1.1388 Accuracy 0.0683
TRAINING Epoch 3/30 Loss 1.0039 Accuracy 0.1410
TRAINING Epoch 4/30 Loss 0.9005 Accuracy 0.2083
TRAINING Epoch 5/30 Loss 0.8202 Accuracy 0.2817
TRAINING Epoch 6/30 Loss 0.7562 Accuracy 0.3330
TRAINING Epoch 7/30 Loss 0.7130 Accuracy 0.3710
TRAINING Epoch 8/30 Loss 0.6508 Accuracy 0.4213
TRAINING Epoch 9/30 Loss 0.6100 Accuracy 0.4520
TRAINING Epoch 10/30 Loss 0.5826 Accuracy 0.4863
TRAINING Epoch 11/30 Loss 0.5473 Accuracy 0.5083
TRAINING Epoch 12/30 Loss 0.5152 Accuracy 0.5360
TRAINING Epoch 13/30 Loss 0.4876 Accuracy 0.5753
TRAINING Epoch 14/30 Loss 0.4735 Accuracy 0.5810
TRAINING Epoch 15/30 Loss 0.4447 Accuracy 0.6183
TRAINING Epoch 16/30 Loss 0.4349 Accuracy 0.6080
TRAINING Epoch 17/30 Loss 0.4168 Accuracy 0.6297
TRAINING Epoch 18/30 Loss 0.3923 Accuracy 0.6493
TRAINING Epoch 19/30 Loss 0.3924 Accuracy 0.6590
TRAINING Epoch 20/30 Loss 0.3611 Accuracy 0.6820
TRAINING Epoch 21/30 Loss 0.3619 Accuracy 0.6753
TRAINING Epoch 22/30 Loss 0.3480 Accuracy 0.6963
TRAINING Epoch 23/30 Loss 0.3403 Accuracy 0.7010
TRAINING Epoch 24/30 Loss 0.3212 Accuracy 0.7213
TRAINING Epoch 25/30 Loss 0.3242 Accuracy 0.7217
TRAINING Epoch 26/30 Loss 0.3043 Accuracy 0.7300
TRAINING Epoch 27/30 Loss 0.2909 Accuracy 0.7367
TRAINING Epoch 28/30 Loss 0.2897 Accuracy 0.7417
TRAINING Epoch 29/30 Loss 0.2802 Accuracy 0.7523
TRAINING Epoch 30/30 Loss 0.2924 Accuracy 0.7410
Finished Training
```

And the test accuracy /loss for entrie model is :

Test Loss: 0.4298 Test Accuracy 0.5908



Hyperparameter used:

NUM_EPOCHS = 30

LEARNING_RATE = 0.005

BATCH_SIZE = 8

RESNET_LAST_ONLY = False

Reference:

https://pytorch.org/tutorials/beginner/basics/optimization_tutorial.html

https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

<https://arxiv.org/abs/1502.03167>

<https://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html>

https://pytorch.org/tutorials/beginner/data_loading_tutorial.html#transforms

<https://python.plainenglish.io/a-gentle-introduction-to-cnns-in-pytorch-5cbcfb71df0d>

<https://androidkt.com/convolutional-neural-network-using-sequential-model-in-pytorch/>

<https://androidkt.com/convolutional-neural-network-using-sequential-model-in-pytorch/>

<https://medium.com/@dipti.rohan.pawar/improving-performance-of-convolutional-neural-network-2ecfe0207de7>

<https://arxiv.org/pdf/1512.03385.pdf>

<https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8>

<https://towardsdatascience.com/resnets-for-cifar-10-e63e900524e0>

<https://medium.com/@alitbk/image-classification-in-a-nutshell-5-different-modelling-approaches-in-pytorch-with-cifar100-8f690866b373>

<https://stackoverflow.com/questions/57161524/trying-to-understand-cross-entropy-loss-in-pytorch>

https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html#training-on-gpu

<https://stackoverflow.com/questions/54374935/how-to-fix-this-strange-error-runtimeerror-cuda-error-out-of-memory>

<https://machinelearningmastery.com/improve-deep-learning-performance/>