CS7015: Deep Learning
Assignment 3

Divya K Raman, EE15B085

28 October 2018

1 Introduction

In this assignment, we explore the effect that pre trained models and data augmentation strategies have on the accuracy of Resnet18 model being trained on the CIFAR 10 dataset.

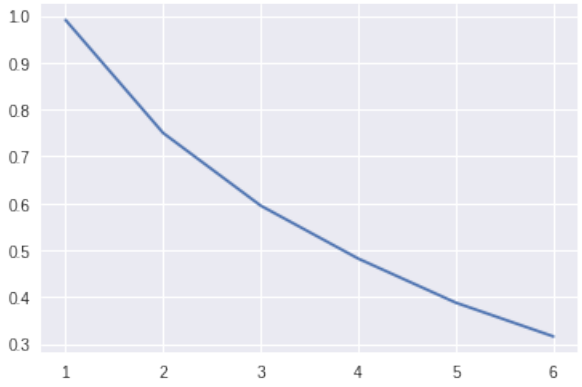
2 Specifications

Model : ResNet-18 Image input size: 224*224(we re-size CIFAR 10 32*32 images to this size) CIFAR-10 dataset : 60000 3232 colour images in 10 classes, with 6000 images per class divided in 5:1 train test split ratio. Tiny-CIFAR-10 : 500 images per class from CIFAR-10 for training. Same 10000 images for testing as per CIFAR-10 dataset.

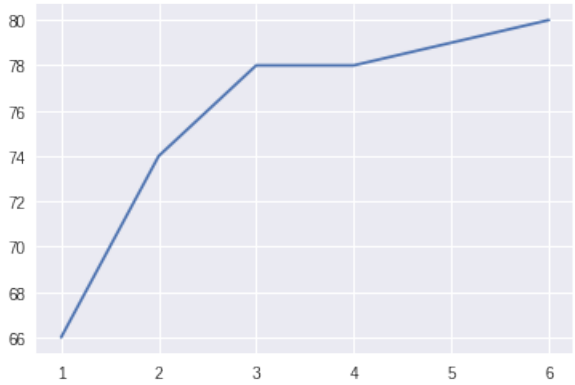
3 Experiment 1

In the first part of the experiment, we use the entire cifar 10 dataset. In the first part, we train the network from scratch using resnet-18 architecture. We modify the last layer of the architecture to give an output of 10 classes for cifar 10 instead of 1000 classes for imagenet. 6 epochs and a learning rate of 0.001 give us a test accuracy of 80 percent. The training loss is 0.316
Accuracy of plane : 90
Accuracy of car : 91
Accuracy of bird : 64
Accuracy of cat : 78
Accuracy of deer : 82
Accuracy of dog : 56
Accuracy of frog : 76
Accuracy of horse : 87
Accuracy of ship : 91
Accuracy of truck : 84

Training error plot

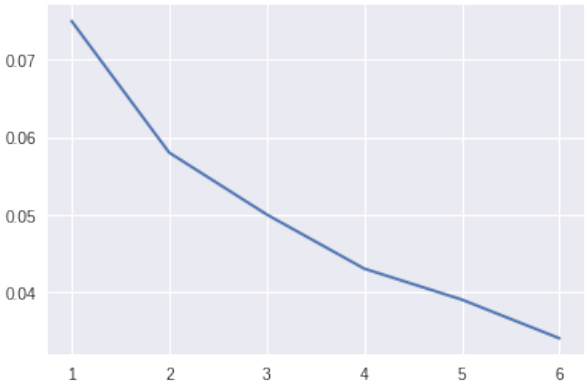


Test accuracy

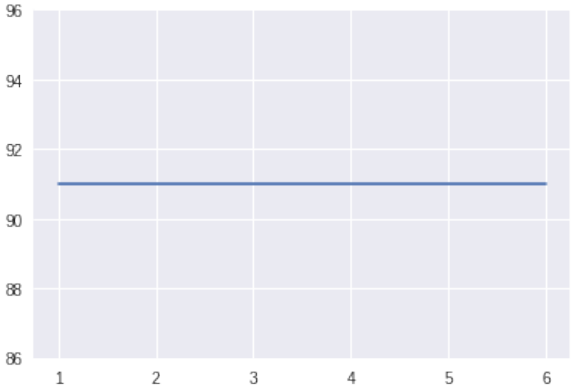


In the second part of the experiment, we use pre trained imagenet weights for all layers except the last layer which outputs only 10 classes instead of 1000 and apply transfer learning.

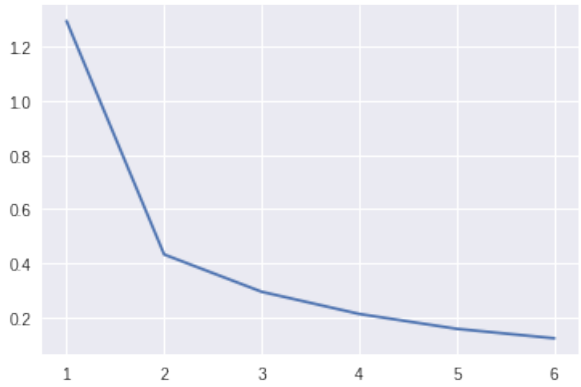
Method 1: Load pre trained weights and train the entire model with a very small learning rate. We use 6 epochs and a learning rate of 0.00005.
Final training loss: 0.034 Accuracy of the network on the 10000 test images: 91
Accuracy of plane : 94
Accuracy of car : 95
Accuracy of bird : 89
Accuracy of cat : 82
Accuracy of deer : 92
Accuracy of dog : 88
Accuracy of frog : 94
Accuracy of horse : 92
Accuracy of ship : 93
Accuracy of truck : 93
Training error plot



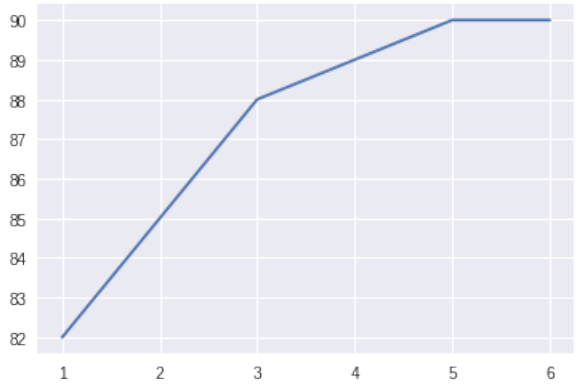
Test accuracy



Method 1: Load pre trained weights; freeze all the initial layers and train only the final layer which was not loaded with pre trained weights; use a very small learning rate. We use 6 epochs and a learning rate of 0.0001.
Final training loss: 0.122
Accuracy of the network on the 10000 test images: 90
Accuracy of plane : 90
Accuracy of car : 94
Accuracy of bird : 85
Accuracy of cat : 84
Accuracy of deer : 91
Accuracy of dog : 85
Accuracy of frog : 92
Accuracy of horse : 90
Accuracy of ship : 93
Accuracy of truck : 95
Training error plot



Test accuracy



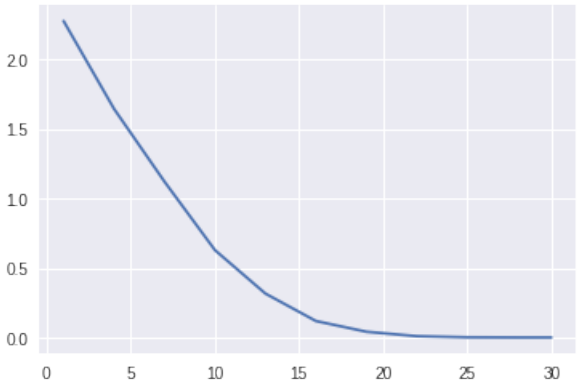
Conclusion:

The experiment with pre trained weights converges much faster than the experiment done by training the network from scratch. Also, the accuracies are much better. Also, method 1 of using the pre trained weights gives us much better results than method 2 of using pre trained weights. This is because we are fine tuning and training the entire network in method 1. However, in method 2, we train only the final layer and don't finetune the weights of the entire model.

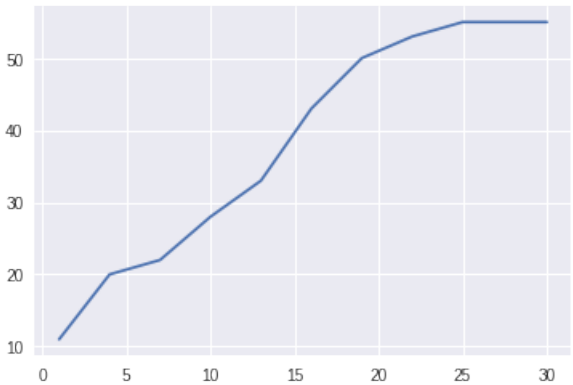
4 Experiment 2

In the first part of the experiment, we use the tiny cifar 10 dataset. In the first part, we train the network from scratch using resnet-18 architecture. 30 epochs, 0.005 learning rate.
epoch 30 training loss: 0.001
Accuracy of the network on the tiny cifar 10 1000 test images: 55
Accuracy of plane : 57
Accuracy of car : 75
Accuracy of bird : 39
Accuracy of cat : 36

Accuracy of deer : 41
Accuracy of dog : 43
Accuracy of frog : 63
Accuracy of horse : 65
Accuracy of ship : 67
Accuracy of truck : 71
Accuracy of the network on the 10000 test images(entire cifar 10 test set): 56
Accuracy of plane : 60
Accuracy of car : 74
Accuracy of bird : 39
Accuracy of cat : 35
Accuracy of deer : 44
Accuracy of dog : 44
Accuracy of frog : 63
Accuracy of horse : 62
Accuracy of ship : 70
Accuracy of truck : 67
Training error

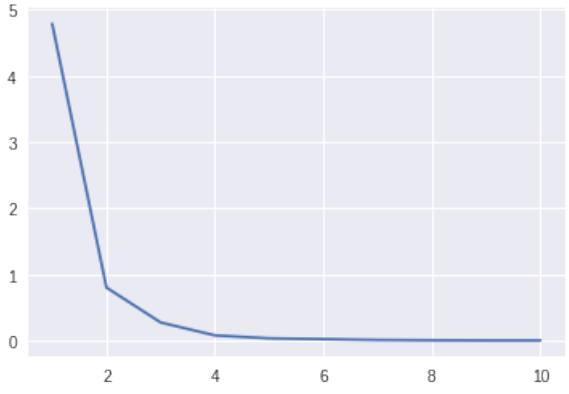


Test accuracy

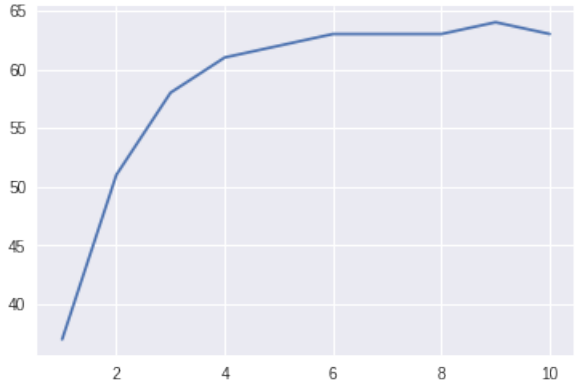


Pretrained weights method 1:

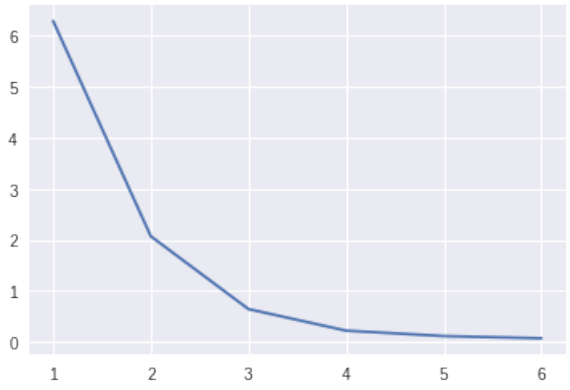
10 epochs, 0.0001 learning rate
epoch 10 training loss: 0.003
Accuracy of the network on the tiny cifar10 1000 test images: 63
Accuracy of plane : 62
Accuracy of car : 85
Accuracy of bird : 56
Accuracy of cat : 44
Accuracy of deer : 43
Accuracy of dog : 53
Accuracy of frog : 72
Accuracy of horse : 70
Accuracy of ship : 69
Accuracy of truck : 77
Accuracy of the network on the entire cifar 10 test set 10000 test images: 64
Accuracy of plane : 67
Accuracy of car : 79
Accuracy of bird : 57
Accuracy of cat : 44
Accuracy of deer : 55
Accuracy of dog : 56
Accuracy of frog : 68
Accuracy of horse : 70
Accuracy of ship : 70
Accuracy of truck : 75
Training error



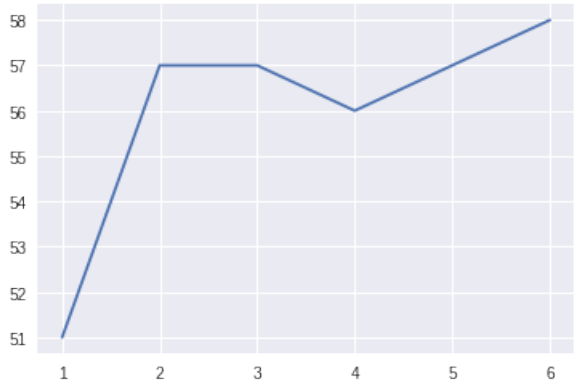
Test accuracy



Pre trained weights method 2:
6 epochs, 0.00001 learning rate
epoch 6 training loss: 0.070
Accuracy of the network on the tiny cifar10 test set 1000 test images: 58
Accuracy of plane : 53
Accuracy of car : 71
Accuracy of bird : 49
Accuracy of cat : 48
Accuracy of deer : 50
Accuracy of dog : 60
Accuracy of frog : 63
Accuracy of horse : 62
Accuracy of ship : 59
Accuracy of truck : 71
Accuracy of the network on the entire cifar10 test set 10000 test images: 60
Accuracy of plane : 64
Accuracy of car : 74
Accuracy of bird : 49
Accuracy of cat : 43
Accuracy of deer : 53
Accuracy of dog : 58
Accuracy of frog : 65
Accuracy of horse : 64
Accuracy of ship : 67
Accuracy of truck : 68
Training error



Test accuracy

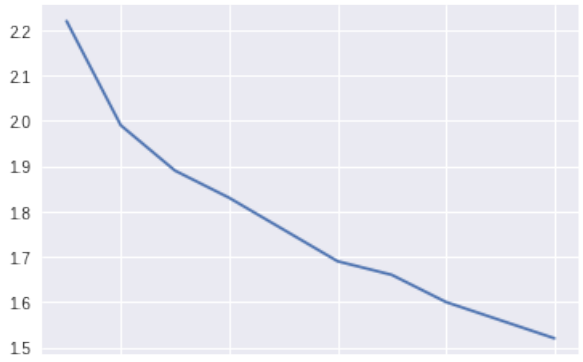


Conclusion:
Less data gives worser results. Test accuracy on entire cifar10 test set is slightly better than test accuracy on tinycifar10 test set.

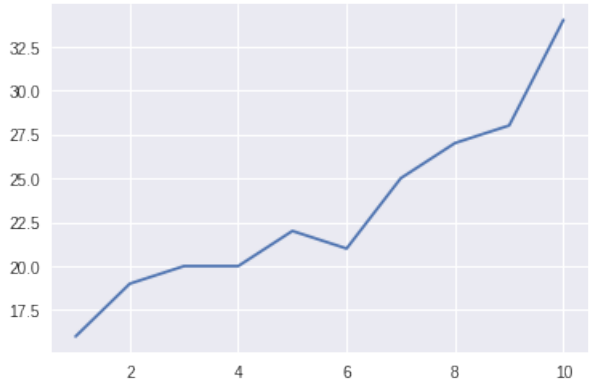
5 Experiment 3

In this experiment, we use data augmentation strategies to increase the size of the training set of the tiny cifar10 dataset.
Case 1: Training from scratch; each image of the tiny cifar 10 training set is subjected to 1 random rotation, 1 horizontal flip and 2 random affine transformations. The size of the training set after all these data augmentation strategies goes from 500 images per class to 2500 images per class. 10 epochs, learning rate 0.01
epoch 10 training loss: 1.5221
Accuracy of the network on the 1000 test images: 34
Accuracy of plane : 48
Accuracy of car : 66
Accuracy of bird : 2
Accuracy of cat : 11

Accuracy of deer : 4
Accuracy of dog : 54
Accuracy of frog : 12
Accuracy of horse : 66
Accuracy of ship : 40
Accuracy of truck : 37
Training error

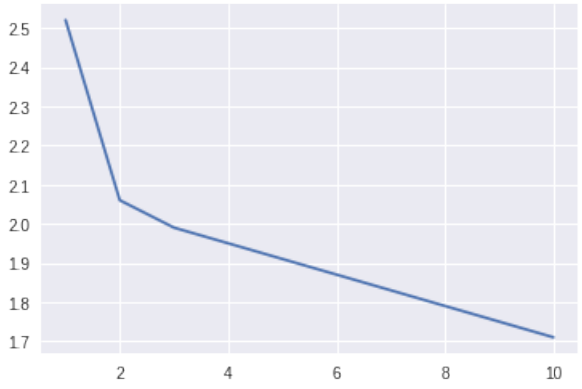


Test accuracy

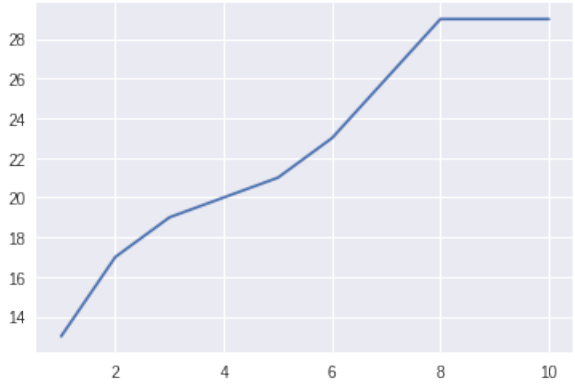


Case 2: Training from scratch; each image of the tiny cifar 10 training set is subjected to 4 random rotations. The size of the training set after all these data augmentation strategies goes from 500 images per class to 2500 images per class. 10 epochs, 0.07 learning rate.
epoch 10 training loss: 1.718
Accuracy of the network on the 1000 test images: 29
Accuracy of plane : 21
Accuracy of car : 88
Accuracy of bird : 0
Accuracy of cat : 8
Accuracy of deer : 9
Accuracy of dog : 58
Accuracy of frog : 17

Accuracy of horse : 60
Accuracy of ship : 24
Accuracy of truck : 11
Training error

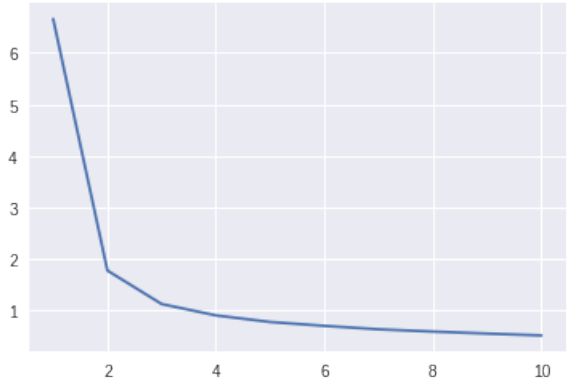


Test accuracy

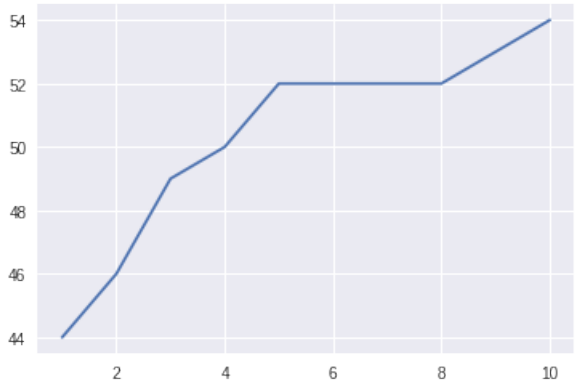


Pretrained weights method 1: 10 epochs, 0.0001 learning rate
epoch 10 training loss: 0.521
Accuracy of the network on the tiny cifar10 test set 1000 test images: 54
Accuracy of plane : 57
Accuracy of car : 87
Accuracy of bird : 51
Accuracy of cat : 47
Accuracy of deer : 41
Accuracy of dog : 56
Accuracy of frog : 56
Accuracy of horse : 49
Accuracy of ship : 46
Accuracy of truck : 51
Accuracy of the network on the entire cifar10 test set 10000 test images: 55
Accuracy of plane : 64

Accuracy of car : 77
Accuracy of bird : 54
Accuracy of cat : 44
Accuracy of deer : 46
Accuracy of dog : 53
Accuracy of frog : 58
Accuracy of horse : 55
Accuracy of ship : 47
Accuracy of truck : 55
Training error

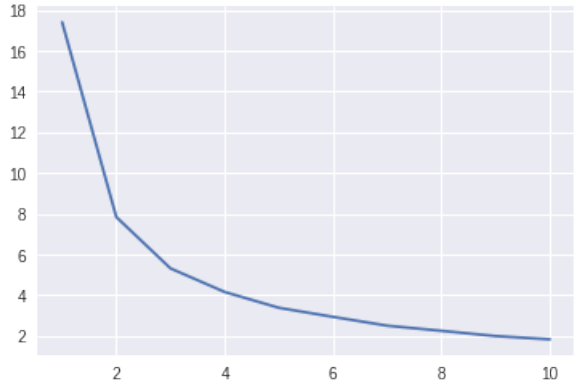


Test accuracy

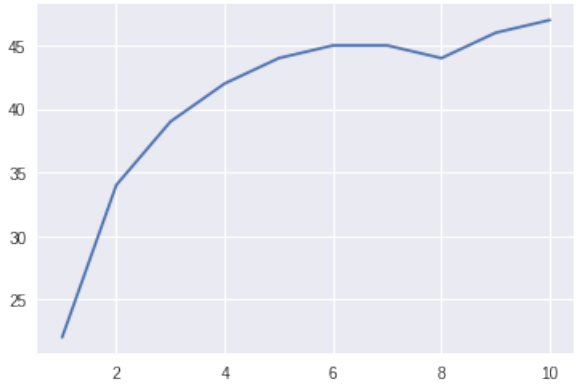


Pretrained weights method 2: 10 epochs, 0.00001 learning rate
epoch 10 training loss: 1.830
Accuracy of the network on the tiny cifar10 testset 1000 test images: 47
Accuracy of plane : 53
Accuracy of car : 46
Accuracy of bird : 61
Accuracy of cat : 27
Accuracy of deer : 27
Accuracy of dog : 54
Accuracy of frog : 50

Accuracy of horse : 53
Accuracy of ship : 45
Accuracy of truck : 54
Accuracy of the network on the entire cifar10 testset 10000 test images: 47
Accuracy of plane : 56
Accuracy of car : 45
Accuracy of bird : 60
Accuracy of cat : 21
Accuracy of deer : 27
Accuracy of dog : 56
Accuracy of frog : 54
Accuracy of horse : 47
Accuracy of ship : 49
Accuracy of truck : 54
Training error



Test accuracy



Conclusion:
Ideally, data augmentation is supposed to give us better results. Here, it does not. Data augmentation is mainly done to improve the networks' accuracy by preventing overfitting. Augmentors help to combat overfitting because they improve a network's ability to generalize. Here, that probably does not happen

because overgeneralization hurts performance because the network wastes its predictive capacity learning about irrelevant scenarios.

The networks take a lot of time to train. In a lot of cases, training for more number of epochs can give us much better performance.

6 References

1. https://pytorch.org/tutorials/beginner/transfer$_{l}earning_{t}utorial.html$
2.$https://towardsdatascience.com/when-conventional-wisdom-fails-revisiting-data-augmentation-for-self-driving-cars-4831998c5509$