CS7015: Deep Learning
Assignment 1A

Divya K Raman, EE15B085

22 August 2018

1 Introduction

In this assignment, we train different networks for image classification on the CIFAR 10 dataset. We present a study of how the performance of the network varies as the network parameters, namely, fully connected layers, convolutional layers, stride, number of channels, max pooling, dropout, number of epochs or training time are varied. We further also analyze the images which are misclassified and try to come up with a probable explanation for the same.

2 Platform

Code uses pytorch library in python, google colab's gpu used to run the code. Dataset: CIFAR 10(https://www.cs.toronto.edu/ kriz/cifar.html) jupyter notebook files containing all the models with different hyper parameters and their results have been uploaded to moodle in a zip file. We present a brief summary of the same below.

3 Experimentation

Learning rate = 0.001 in all models

Only number of epochs, number of convolutional layers, number of fully connected layers and final accuracy is presented here. Stride in each layer, number of channels, zero padding, ReLU, Max pooling, etc can be got from the model descriptions written in pytorch in the jupyter notebooks. We have also used dropout and regularisation for a few models, the details which are provided below.

Part 1:
Model: 15 epochs, 2 convolutional layers, 3 fc, max accuracy: 59

Model: 1 epoch, 2 conv, 5 fc, max accuracy: 28
Model: 1 epoch, 3 conv, 3 fc, max accuracy: 41
Model: 10 epoch, 3 conv, 3 fc, max accuracy: 62
Model: 5 epoch, 3 conv, 3 fc, 1 dropout max accuracy: 58
Part 2:
Model: 10 epoch, 3 conv, 3 fc max accuracy: 57
Model: 5 epoch, 3 conv, 3 fc, 2 dropout max accuracy: 59
Model: 1 epoch, 4 conv, 3 fc, 3 dropout max accuracy: 34
Model: 1 epoch, 4 conv, 3 fc, 2 dropout max accuracy: 39
Model: 20 epoch, 4 conv, 3 fc, 2 dropout max accuracy: 60
Model: 5 epoch, 0 conv, 5 fc, 4 dropout max accuracy: 42
Model: 10 epoch, 3 conv, 2 fc max accuracy: 69
Part 3:
Model: 19 epoch, 3 conv, 2 fc max accuracy: 71
Part 4:
Model: 5 epoch, 4 conv, 2 fc max accuracy: 67
Model: 10 epoch, 3 conv, 2 fc max accuracy: 73
Part 5:
Model: 15 epoch, 4 conv, 3 fc max accuracy: 74
Part 6:
15 epoch, 5 conv, 3 fc max accuracy: 73
10 epoch, 4 conv, 2 fc max accuracy, regularisation=0.0001: 73
15 epoch, 4 conv, 2 fc max accuracy, regularisation=0.001: 73
Part 7:
15 epoch, 4 conv, 3 fc max accuracy: 76
Part 8:
15 epoch, 4 conv, 3 fc max accuracy, regularisation=0.001: 75
Part 9:
15 epoch, 4 conv, 3 fc max accuracy, regularisation=0.001: 75

A few more models were tried were the regularisation parameter, stride and kernel size were varied. The results of those are not reported above as those model descriptions and the results were not saved.

4 Results, Observations and Conclusions

The maximum accuracy obtained from the experimentation is 76 percent.

While too few layers give low accuracy on the test dataset, too many layers also decrease the the accuracy. The ideal number of layers seems to be 3-4 convolutional layers and 2-3 fully connected layers. A max pool layer after each or after every two convolutional layer yields good results. It is advisable to use a ReLU layer after each convolutional or fully connected layer. The best kernel sizes seem to be 3 or 5 for the convolutional layers. Very big kernels don't seem to yield good results. Most max pool layers that we used have a kernel size of

2 and stride 2. Large strides are also not preferable. Most convolutional layers that we have used have a stride of 1. As the number of epochs increases, training error decreases. Test set accuracy initially increases, reaches a threshold and then decreases or fluctuates around the same value. This is because increasing the number of epochs too much tends to overfit the model on the training data and hence decreases the test error. The width of the network(maximum number of channels) also plays a role. Similar to many other parameters, networks with too less or too high width tend to perform poorly. Adding dropout layers can increase or decrease accuracy depending on the probability parameter, also it depends on how well the rest of the network is fitting the model. Regularisation can help prevent overfitting the model but too much regularisation will underfit the model.

Class scores for one of my best models:

Accuracy of plane : 68 percent
Accuracy of car : 90 percent
Accuracy of bird : 68 percent
Accuracy of cat : 57 percent
Accuracy of deer : 76 percent
Accuracy of dog : 58 percent
Accuracy of frog : 88 percent
Accuracy of horse : 84 percent
Accuracy of ship : 92 percent
Accuracy of truck : 83 percent

The trend for the scores seems to be similar in almost all the models. Cat and dog have similar accuracy in almost all the models and the lowest too. The probable reason could be high similarity in the images corresponding to cat and dog which leads to a cat being mis-classified as dog and vice versa. This explains both their similar scores and low scores. Plane also has a low score, probably due to it being mis-classified as a ship. Car and ship have highest scores. The probable reason is their unique distinguishing features which helps them not being mis-classified.

5 References

1. https://adeshpande3.github.io/A-Beginner-27s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/

2. http://karpathy.github.io/2011/04/27/manually-classifying-cifar10/

3. https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a

4. https://touren.github.io/2016/05/31/Image-Classification-CIFAR10.html