
Deep Image Classification

Garrison Neel

Department of Mechanical Engineering
Texas A&M University
College Station, TX 77840
gneel@tamu.edu

Abstract

This is a barebones report. I'm creating it because my internet is bad, and I might not be able to upload my submission closer to midnight as campus has issues and my internet has been unreliable of late. This report will include a link to my final model and code where they will be posted (timestamped) before midnight.

1 Introduction

Deep learning can do lots of things, classifying images is one of them.

2 Proposed Method

I used resnet with preactivation as a base, then tried improving from there. The paper "Improved Residual Networks for Image and Video Recognition" (<https://arxiv.org/pdf/2004.04989.pdf>) gives one way of improving resnets, and a second paper: "Don't decay the learning rate, increase the batch size" (<https://arxiv.org/pdf/1711.00489.pdf>) suggests training using increasing batchsize instead of a learning rate decay schedule.

3 Implementation Details

SGD was used over adam bc that's what the resnet people use, and it seems like in general that's what people are using now. This paper <https://arxiv.org/pdf/1705.08292.pdf> shows that SGD generalizes better than other optimization methods.

3.1 Image Augmentation

Multiple methods of image augmentation were evaluated. Random translation, random crop, random rotate, and random erasing were all tested. Image augmentation didn't necessarily improve the validation accuracy but did improve the validation loss, which suggests it is advantageous for generalizing the model. For this reason, it was used to train the final model.

3.2 Increasing Batch Size

Increasing batch size supposedly increases model generality because more samples are used in the update step. More samples per update step means the step is better averaged for the population rather than a small subset.

3.3 Improved Residual Blocks

These improved blocks come from <https://arxiv.org/pdf/2004.04989.pdf>, which claims they make the network easier to train. Implementing them was simple, and required only a slight restructuring of the bottleneck block.

4 Results

TBD. Most models stagnate in the high 80's accuracy-wise. I'm expecting 80+% on the private test dataset. I can achieve 91% validation error with a loss of 0.42.

5 Conclusion

The final version of this report, and the code and model will be uploaded to https://github.com/neelg1193/CSCE_636/tree/main/Project/submission