11-785 HW2 Part 2 Writeup Submission

Jae Kang

Jkang2

11-785 Introduction to Deep Learning

I tried three different architectures; one was MobileNetV2 and one was ResNet34 and ResNet50. I found that mostly the ResNet architectures trained at a much slower pace but at fewer epochs, the training accuracy and the validation accuracy jumped at a much quicker pace and were mostly the same. MobileNetV2, although much faster on both p3.2xlarge and p2.xlarge, the training accuracy jumped much further ahead the validation accuracy which was a sign of overfitting. For the mobilenetv2 architecture, I had cut off a lot of the final 4 or 5 bottlenecks described as in the original paper, and tried limiting the final channels to 32 and 64, which gave me a feature embedding vector of size 4096. The reason that mobilenetv2 might have been overfitting might have been due to the large size of my feature embedding, but I found that simplifying my mobilenetv2 architecture even further and reducing the channels made it perform even worse and converge to the optimum much more slowly. For both architectures, I had tested using a mix of center loss and crossentropy loss at first, but I found that training did not help the neural nets (both Mobilenet and ResNet) converge quicker and after much trials (and by testing out different weights for the center loss; 0.1 0.4 and even 0.8), I opted to use just Cross entropy in the end.

ResNet50 was the architecture I used in my submission and I maintained the architecture in the paper mostly, with the same number of blocks but with a stride of 1 across each layer. Kernel size was maintained at 3, my feature embedding vector size was 512, with a batch size of 32 and Adam optimizer with learning rate of 1e-3. Although training took very long (1 hour even on the p3.2x large instance), it was well worth it because after 5 – 6 epochs, the accuracy was well maintained at around 60 and I trained for a couple more epochs. No center loss was necessary again and I didn’t train much further after just training for the classification.