Amaan Rahman

ECE 472: Deep Learning

Professor Curro

Assignment 4: CIFAR Image Classification

CIFAR - 10:

Various experiments were performed with the CIFAR-10 image classification task. Initially a VGG-19 architecture was implemented with some modifications in the depth of the model to be suitable for CIFAR-10. What was interesting was how the loss spiked and the accuracy plummeted.

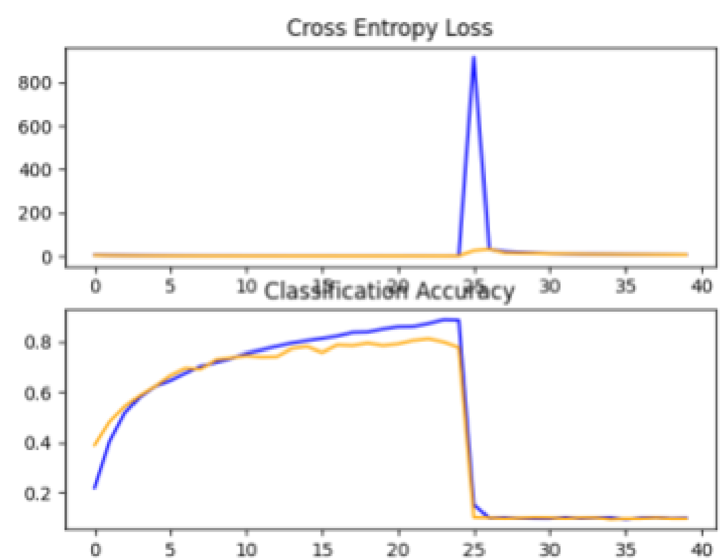

Figure 1: VGG-19-Like Model presenting vanishing gradient problem

Blue: train, **Yellow:** validation

One way I mitigated this issue was through introducing a **learning rate scheduler** to shave off the learning rate after a fixed number of epochs so the vanishing gradient problem could be mitigated.

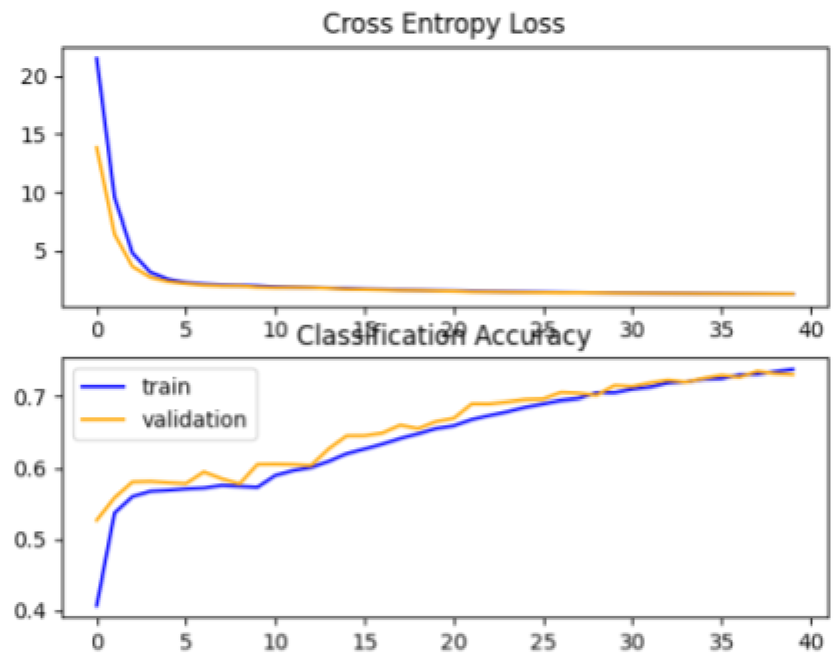

Figure 2: VGG-19-Like model with Learning Rate Scheduler

After a while tampering with the VGG like model, my test accuracy kept converging to as high as 70%. I decided to move on to a **ResNet** architecture [1], [2].

Considerable improvements were observable: fewer parameters, faster training time, and larger test accuracy. I also introduced preprocessing techniques such as random cropping and flipping the input image to model. At first my initial implementation was a **ResNet-50** model and that yielded 75%, however it was unoptimized and poorly implemented. I then tested out a **ResNet-20** model, which yielded test accuracy of about 80%. A nuisance that I noticed was that the validation accuracy would converge faster than the training accuracy (same with the loss). Of course this is due to the model overfitting with the training data. I implemented a strong dropout procedure with my **ResNet-18** model; I applied dropouts with rates of 0.5 for the residual blocks and 0.3 for the fully connected classification layer. I also implemented the built-in tensorflow function: ReducedLROnPlateau to shave off the learning rate by 1/10th after observing no noticeable change in the validation loss over 10 trials. This yielded my best results thus far of a test accuracy **86.75%**.

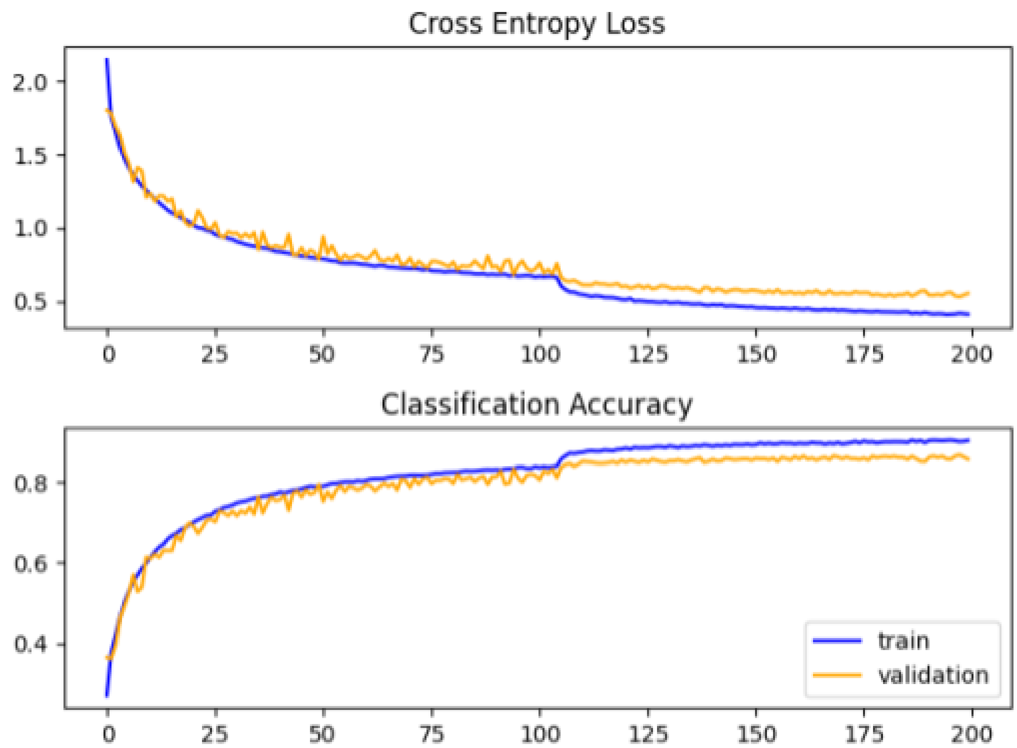

Figure 3: ResNet-18 with strong dropout; learning rate reduction effect can be seen around 105th epoch

Other miscellaneous experiments I conducted were experimenting with a cyclical learning rate scheduler to replicate the super convergence paper [3], [4]; this didn't work out so well because I wasn't optimizing it well with this specific use case. Another experiment I conducted was using a sparse gate block within a residual block [5]. This was an interesting experiment because the introduction of this block would discard residuals, analogous to a dropout function but for an entire residual block. I couldn't properly take advantage of this technique because I was experimenting on not-so-deep residual networks. This also motivated me to take a look at highway networks, however I didn't get the chance to implement anything of that sort.

CIFAR-100:

I implemented various **ResNet** architectures such as **ResNet-110**, **ResNet-50**, and **ResNet-18**. The former 2 models yielded horrible convergence (35% or 50% convergence for validation set after 30 or 60 epochs out of 200 epochs), however **ResNet-18** fared well. I introduced **sparse dropout** after the shortcut addition within the identity residual block preventing overfitting and increasing generalization. Unfortunately, for this part of the project my accuracy graph is not based on the “top-5-accuracy” metric so there is no graph for it :(. The top-5-accuracy for CIFAR-100 with **ResNet-18** was **78.42%**.

```
loss: 2.2637 - top_k_categorical_accuracy: 0.7842 - accuracy: 0.4940
```

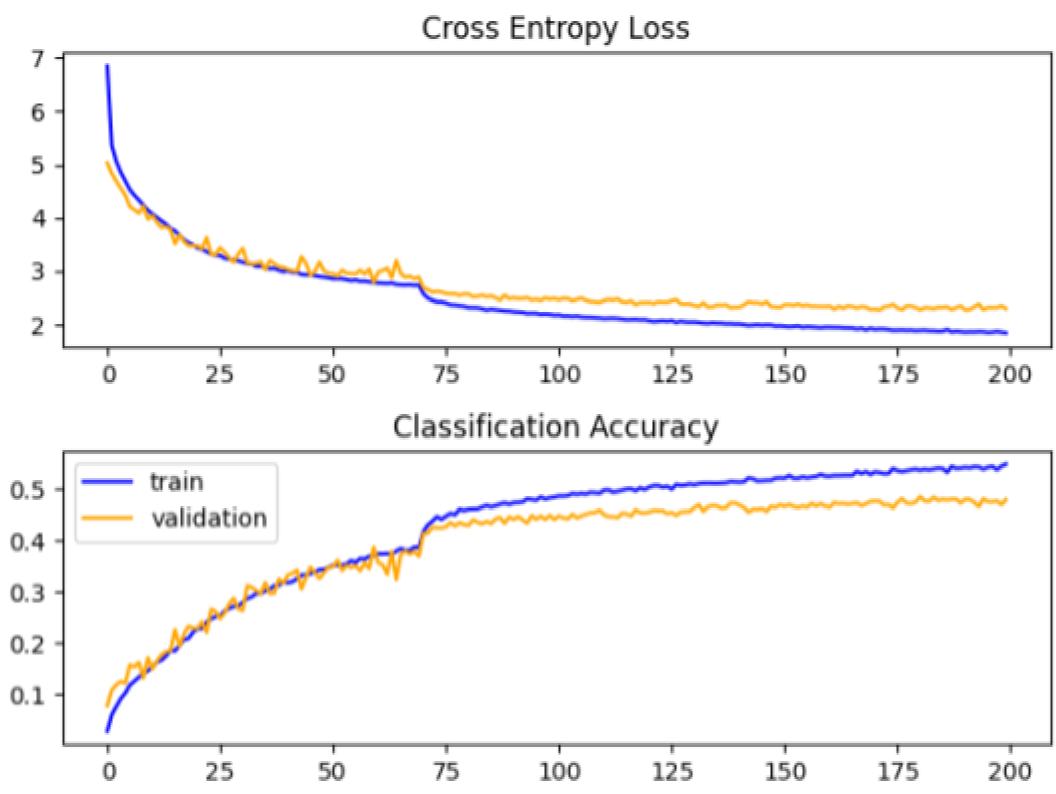

Figure 4: ResNet-18 model against **CIFAR-100** using strong sparse dropout rate of 0.5 and ReduceLROnPlateau

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[2] "How to code your ResNet from scratch in Tensorflow?," *Analytics Vidhya*, Aug. 26, 2021. https://www.analyticsvidhya.com/blog/2021/08/how-to-code-your-resnet-from-scratch-in-tensorflow/ (accessed Oct. 11, 2021).

[3] L. N. Smith, "Cyclical Learning Rates for Training Neural Networks," *arXiv:1506.01186 [cs]*, Apr. 2017, Accessed: Oct. 11, 2021. [Online]. Available: http://arxiv.org/abs/1506.01186

[4] C. T. Bs. H. MIAP, "Super Convergence with Cyclical Learning Rates in TensorFlow," *Medium*, Jan. 31, 2021. https://towardsdatascience.com/super-convergence-with-cyclical-learning-rates-in-tensorflow-c1932b858252 (accessed Oct. 11, 2021).

[5] X. Yu, Z. Yu, and S. Ramalingam, "Learning Strict Identity Mappings in Deep Residual Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, Jun. 2018, pp. 4432–4440. doi: 10.1109/CVPR.2018.00466.