CIFAR10

I initially looked at TResNet, as it achieved a 99% accuracy on CIFAR10, but I soon decided not to use it because I didn't completely understand the Squeeze and Excitation Blocks (meaning I would essentially copy/paste code) and because the time required to train TResNet-M is almost 24 hours (according to the paper), which I considered infeasible.

I used a combination of ResNet50 and Full Pre-Activation (from the paper assigned for reading this week). I tried using the regular ResNet50, but I found adding Full Pre-Activation gave better results. It converged faster, and slightly higher than the regular ResNet50.

I used L2 Regularization. On my initial attempts, I used the default value of 0.01 for my penalization parameter. This proved to be too large and caused my network to converge to around 65% accuracy. On subsequent attempts, I reduced it to 0.001.

I used Data Augmentation on the images, which I found helped with reducing overfitting. I also added a decay in the learning rate, which gave more stable learning near the end and reduced the chance that I would get a random drop in accuracy on the last iteration.

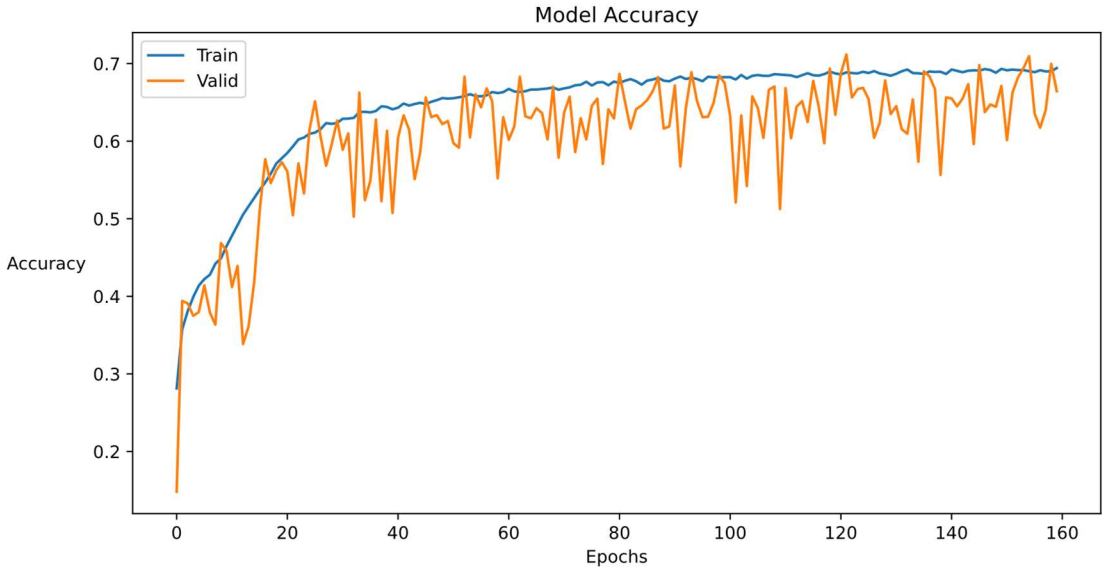

Figure 1: Example plot for the training and validation accuracy on CIFAR10 using ResNet50 with Full Pre-Activation during training with L2 Parameter = 0.01

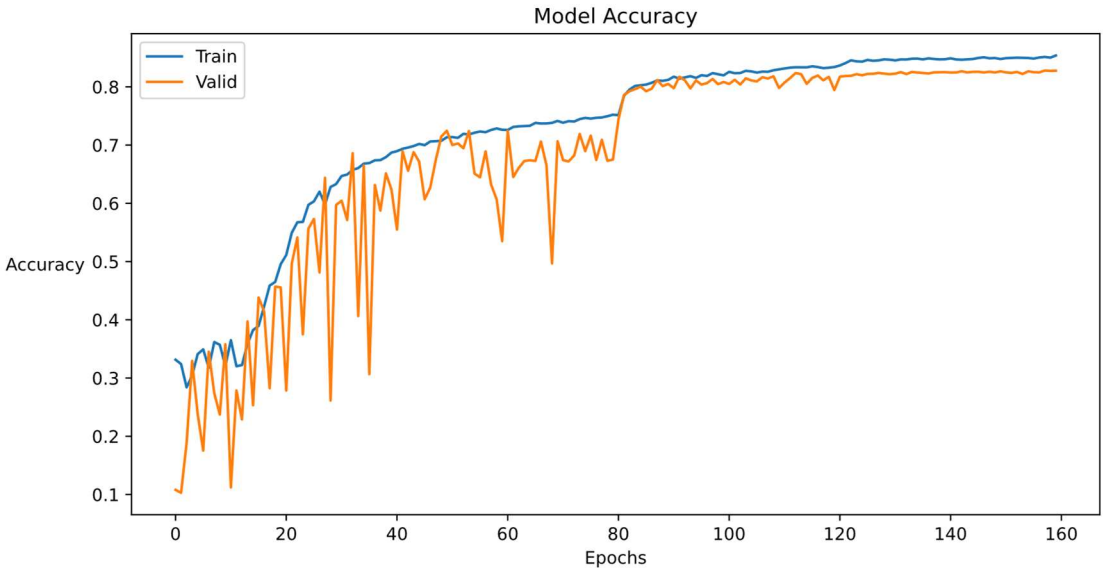

Figure 2: Example plot for the training and validation accuracy on CIFAR10 using ResNet50 during training with L2 Parameter = 0.001

313/313 [==============================] - 4s 11ms/step - loss: 0.6549 - accuracy: 0.8170

Figure 3: Final accuracy on the CIFAR10 test set using ResNet50 with L2 Parameter = 0.001

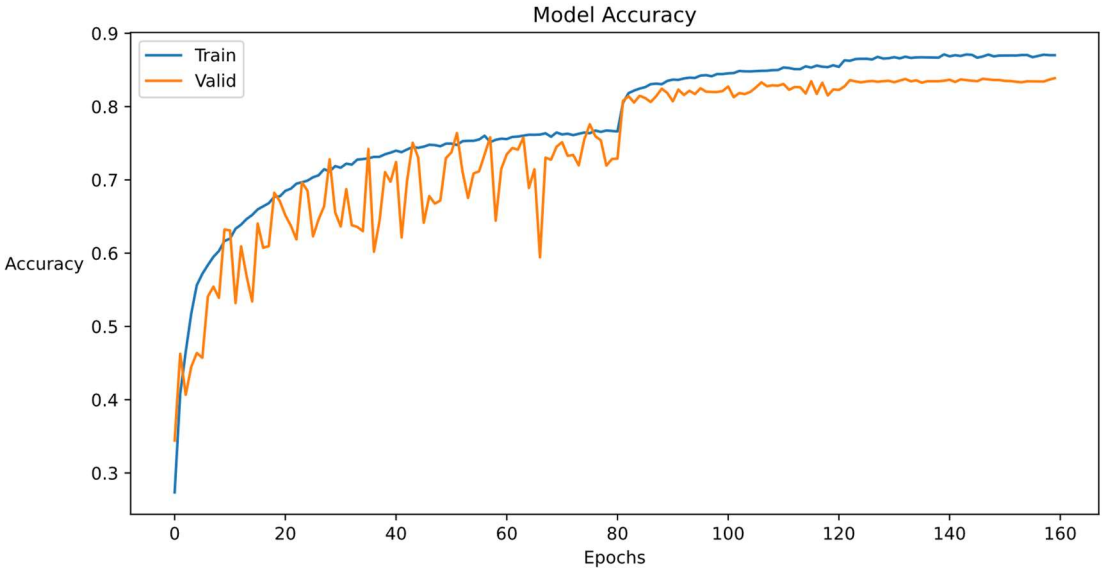

Figure 4: Example plot for the training and validation accuracy on CIFAR10 using ResNet50 with Full Pre-Activation during training with L2 Parameter = 0.001

313/313 [==============================] - 4s 12ms/step - loss: 0.6280 - accuracy: 0.8340

Figure 5: Final accuracy on the CIFAR10 test set using ResNet50 with Full Pre-Activation with L2 Parameter = 0.001

CIFAR100

I used the same program for CIFAR100. On my first attempt, I had an almost 10% discrepancy between my training and validation accuracies. My test accuracy was 79%, barely under the target. On my next attempt, I added a 20% dropout layer right before the output layer. This reduced the discrepancy to about 7%. I barely got over 80%, so I decided to try another modification. I added a 15% dropout layer after Stage 2 and a 20% dropout layer after Stage 3 to further counter overfitting. The discrepancy dropped to around 5% and gave me a slightly better validation accuracy.

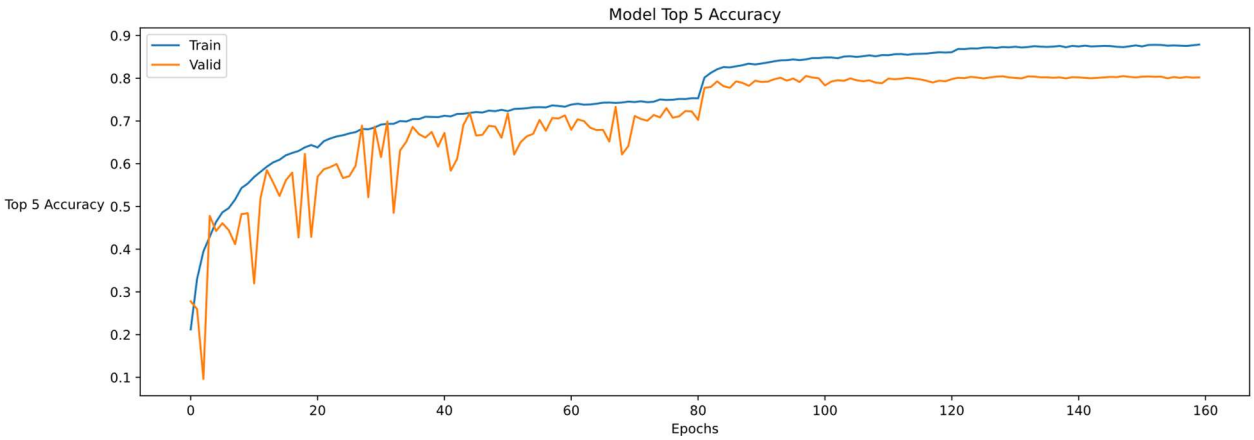

Figure 6: Example plot for the training and validation accuracy on CIFAR100 using ResNet50 with Full Pre-Activation with L2 Parameter = 0.001 and Dropout = 0.2

313/313 [==============================] - 4s 11ms/step - loss: 2.1422 - sparse_top_k_categorical_accuracy: 0.8004

Figure 7: Final accuracy on the CIFAR100 test set using ResNet50 with Full Pre-Activation with L2 Parameter = 0.001 and Dropout = 0.2

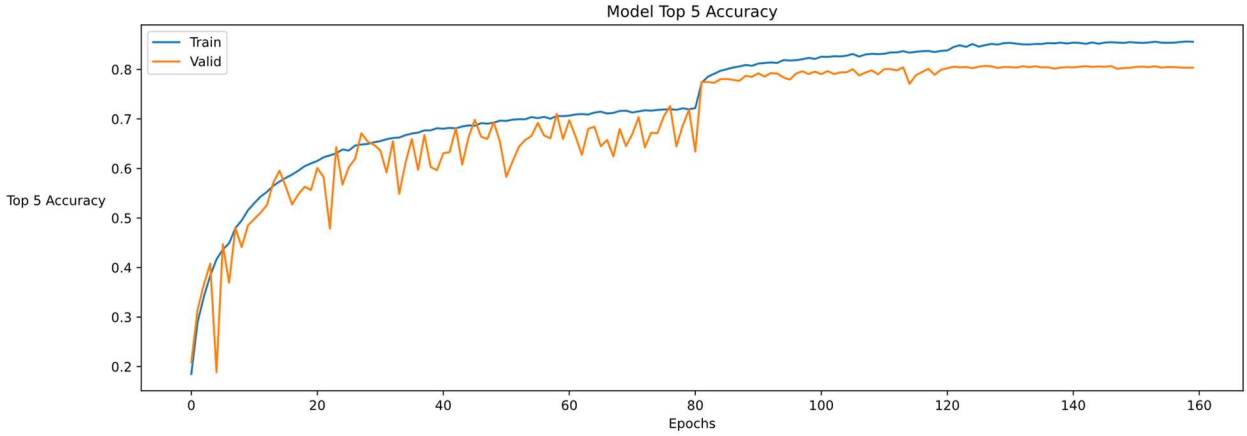

Figure 8: Example plot for the training and validation accuracy on CIFAR100 using ResNet50 with Full Pre-Activation with L2 Parameter = 0.001 and Dropout = 0.15, 0.2, 0.2

313/313 [==============================] - 4s 13ms/step - loss: 2.0996 - sparse_top_k_categorical_accuracy: 0.8026

Figure 9: Final accuracy on the CIFAR100 test set using ResNet50 with Full Pre-Activation with L2 Parameter = 0.001 and Dropout = 0.15, 0.2, 0.2