

6.869 Miniplaces 2

Nada Hussein

April 2021

Problem 1

Initially, I used a ResNet18 model with 10 epochs, batch size 128, and using SGD with a learning rate of 0.001 as my baseline. This gave a training top5 error of 0.392, and a validation top5 error of 0.423. I then compared this to other models by changing just one feature at a time: I chose learning rate, optimizer, and model as features to change and compare.

Learning Rate

First, I adjusted the learning rate. I tested 1, 0.1, 0.01, 0.001, and 0.001, keeping all other parts of the baseline model the same. Below is a graph of learning rates vs top5 errors:

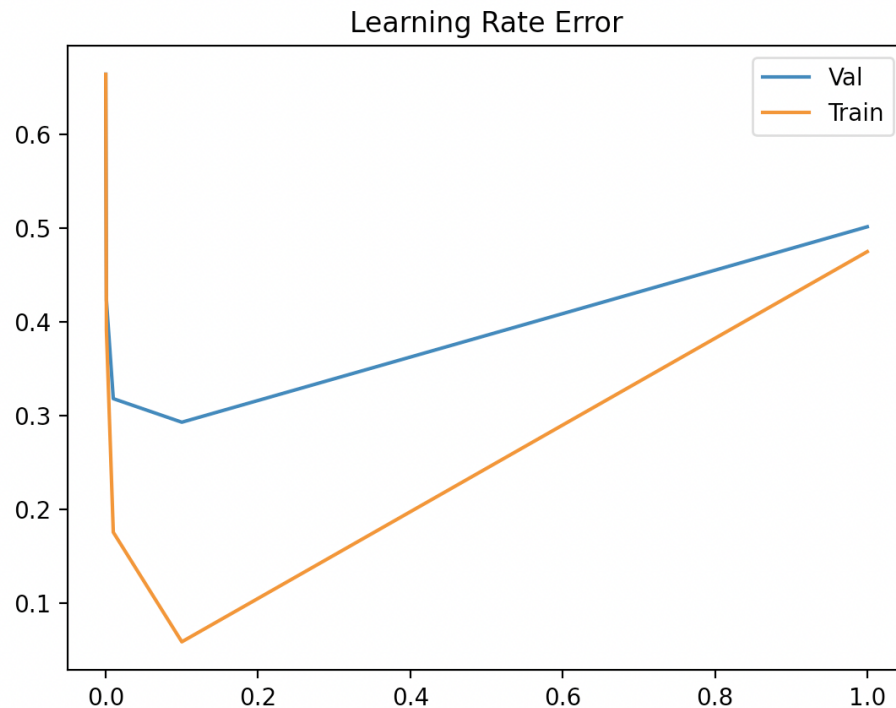


Figure 1: Learning Rate Error

Here you can see that the error is very high at small learning rates, goes down as learning rate increases, and then goes back up after approximately 0.1. This makes sense, as with very small learning rates, the model is not able to converge and so we end up with a large error. As we increase learning rate, it can

easily converge in less steps. However, after a certain point, increasing the learning rate too much results in overshooting the minimum error point, and then the error increases as our weights diverge.

Optimizer

Second, I adjusted the optimizer. I tested SGD, Adam, and Adagrad with a learning rate of 0.001 for each. Below is the result: Here we can see that SGD resulted in the most error, while Adam resulted in the

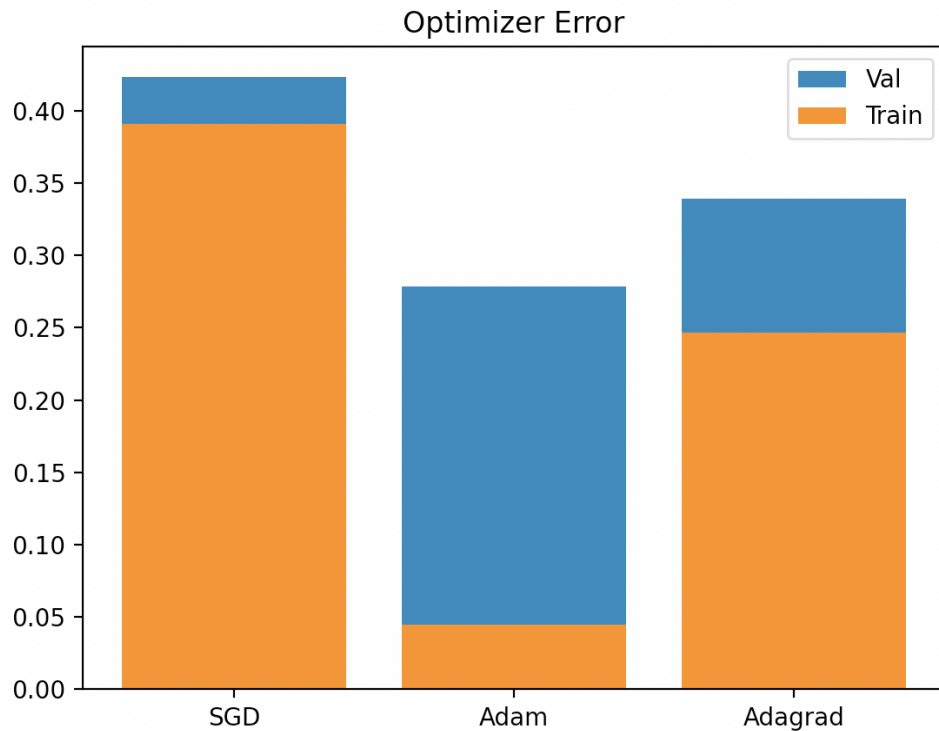


Figure 2: Optimizer Error

least error. However, the difference between train and val errors was the biggest with Adam, showing that Adam may have resulted in overfitting to our training data. SGD resulted in the smallest difference in error, showing it might have done the best at fitting to the correct features.

Model

Finally, I adjusted the model. I tested Resnet, VGG, Alexnet, and Densenet. Below are the results:

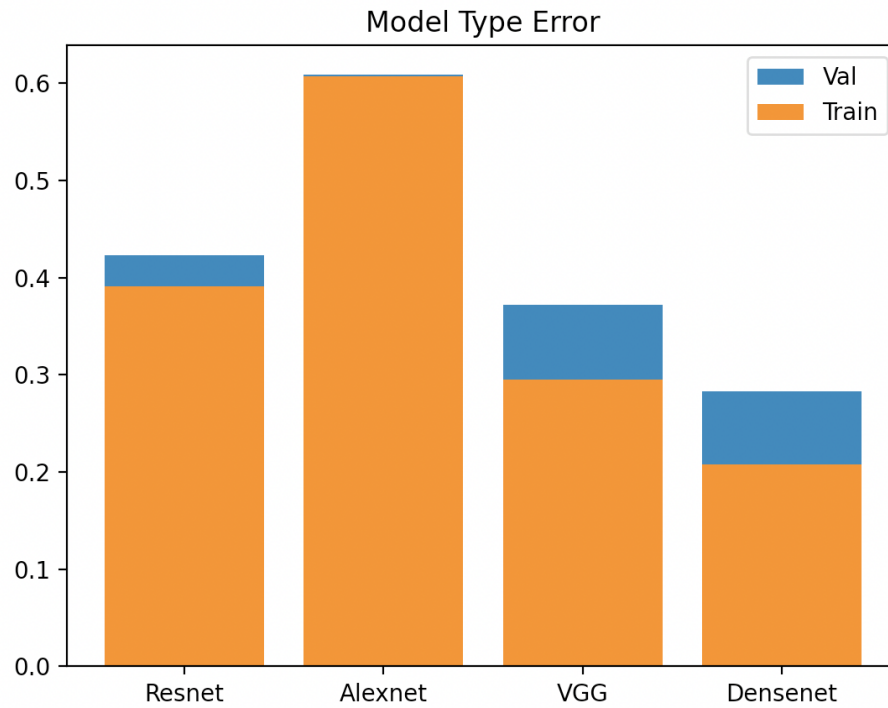


Figure 3: Model Error

Here we can see that Densenet resulted in the smallest amount of error, but also had the biggest delta between val and training data. This suggests it was overfitting to the training data. Alexnet had the largest error, but also had the closest errors between val and train, showing that it trained on the right features.

Problem 2

I ended up achieving a 77% top5 test accuracy with my model. I decided to use the Densenet architecture, running for 15 epochs and batch size of 16. I then used the Adam optimizer with learning rate 0.001.