

The previous two pages are a chronological spreadsheet of model training, with each row representing a separate trained model.

The Model column lists the changes made to the model from the previous attempt. The following five columns list the validation and test performance of each trained model. For those models trained to completion, I provide a plot of the training vs. validation loss across the epochs of training. The final Notes column contains my assessment of what I think might be happening.

Summary

- 1) For the first five training runs, I play with the optimizer (SGD vs. Adam), learning rate (static vs. decreasing), batch size, and initial dropout. These runs mostly have exploding validation loss (that's what I'm calling it), though they also have the highest validation and test accuracy of any of the training runs. Sadly, in most of these cases, the best model is the initial model, so this is mostly due to luck. My highest overall test accuracy is 67.80%, with the model being identical with the default model except for an Adam optimizer rather than SGD.
- 2) For the following four training runs, I try tanh and sigmoid activation functions, add and remove more dropouts, and introduce L2 regularization as well as increasing stride size. I cut off three of these runs before they finished because the validation loss was increasing so dramatically with each epoch. The tanh activation function had the worst overall validation and test accuracy of any model I tried, at 50.89% and 50.90% respectively, which could be because tanh returns values between -1 and 1 instead of sigmoid and relu's 0-1 range.
- 3) For the final four training runs, I shrank the size of the model to two convolutional layers with more downsampling in the pooling layers. In these runs, the validation loss corresponded more directly with the training loss and test loss (and accuracy) throughout the training process, which I thought was a good sign. If I were to keep tweaking this model, I'd continue working with a model with fewer layers.

The final and longest training run had 25 epochs with a SGD optimizer, a static learning rate of 1e-3, L2 regularization, and a 54.93% accuracy, which isn't good and seemed to hold steady throughout the training process. The previous run had used exactly the same structure and seemed to have steadily decreasing losses. In the case of this assignment, I felt like I should pursue models where the training, validation, and test accuracy follow expected patterns and remain consistent with each other throughout the training process, since this might indicate a more stable training that would lead to more predictable results.