Steven Lapp, 204785240

Abstract:

The following is a brief description of my model and how I chose this model. The first layer (before conv and maxpooling) was Gaussian noise in order to regularize. Fallowed by four repetitions of Convolution, max pooling, and Dropout (rate 0.25). Finally fully connected layer of 128 nodes afterwards another Dropout (rate 0.5) and another fully connected of 10 for softmax support. The filters of each Convolutional Layer were as fallows [64, 128, 256, 512]. Batch size of learning was 50 and initialized a learning rate of 0.001 to adam optimizer which was built in to Keras. During training used built in keras data augmentation to generate new images. Shift range up and down left and right was by 0.2. Rotation was up to 15 degrees. Used model checkpoint callback to save the best model (lowest val loss) and early stopping callback to strop training if after 10 epochs not improved to stop.

The way I got to this layout was through a lot of trial and error. I initially was under fitting by quite a lot and expected that when the number of epochs will increase it will achieve desired accuracies and realized was not ever reaching goal and consistently was at an accuracy between 99.80 -99.88 on validation1.txt. Then I prefer to be in a state of overfitting and after words gradually reintroduce noise. I went from filters of [16, 32, 64, 128] to [64, 128, 256, 512]. Noticed immediate improvement and pretty quickly outperformed previous results. Then I began reintroducing dropout noticed less overfitting and then introduced Gaussian noise with 0.5 which caused huge under fitting then changed noise to 0.01 to strike more of a balance. For train added val2 to train. On final test added val1 to train as well. Accuracy achieved approximately 99.5