

Convolutional Neural Networks

I have 10,000 images of cats and dogs. The goal is to train a neural network to recognize the difference between cat and dog images. Because the files are images they are too large to upload to Github. I have uploaded them to Dropbox where you can download them (<https://www.dropbox.com/s/iw1uhqwr10oeqo/CNN%20Dataset.zip?dl=0>).

Data Preprocessing

The 10,000 images are kept in their respective folder. They have been split up as follows

- 4000 images of dogs in the training set.
- 4000 images of cats in the training set.
- 1000 images of dogs in the training set.
- 1000 images of cats in the training set.

80% of the images are in the training set while 20% are in the test set.

Convolution & Pooling & Flattening

I have created 32 feature detectors of 3×3 denominations making our convolutional layer composed of 32 feature maps. We also added a second convolutional layer in order to obtain more accurate results.

I chose to use a 64 bit color input in order to get the results quickly. This could be increased to 128 or 256 if using a GPU or if you don't mind waiting a number of hours for the algorithm to complete.

I chose a pool size is 2 by 2, then flattened the classifier.

Fitting CNN to the images.

I then took and ran the algorithm. I used 25 epochs with a total of 8000 steps per epoch (because there are 8000 images). The following epoch appeared as such (To see the full epoch view *Epoch (CNN).txt*).

2680s - loss: 0.0108 - acc: 0.9966 - val_loss: 1.5131 - val_acc: 0.8144

From this we can see the training set has an accuracy of 99.66% while the test set had an accuracy of 81.44%. This is quite a large difference between the training set and the test set. This could be due to overfitting. Ideally we would want the test set to be closer to 99% as well.