

Diego Andrade

CS 460

Dr. Harrison

Homework 5

Corgi Classification

A standard problem in the world of machine learning is image recognition and classification. A standard problem in the world of dog-aficionados is recognition and classification of dog breeds, specifically between Pembroke and Cardigan Corgis. In this project I implemented a solution for the intersection between these two topics by using TFLearn, NumPy, and the provided dataset.py program.

My program is currently classifying pictures of these dogs with an accuracy of about 94%. I achieved this using a convolutional neural network, the structure for this network consists of 4 convolutional layers, using relu as an activation function, followed by a pooling layer after each layer. After the convolutional layers there is a fully connected layer also using a relu, followed by regularization by using dropout. Finally, the fully connected output layer, followed by a regression with an ADAM optimizer and a categorical cross-entropy loss function. The best learning rate I found for this setup was one of 0.0001, and I arbitrarily chose 50 pixels squared as the size for the images.

When running the program, as it is right now, the training doesn't occur, instead it loads in the previously trained model and makes a prediction for the test set. In order to continue training, line 65 in classifier.py can be uncommented in order to train for another 20 epochs. To entirely recreate results, one can delete the following files in the directory:

- `corgiClass-0.0001--4_conv_layers_final.data-00000-of-00001`
- `corgiClass-0.0001--4_conv_layers_final.index`
- `corgiClass-0.0001--4_conv_layers_final.meta`

and after uncommenting line 65 rerunning the program will start to train for 20 epochs. This did not prove to be enough training, and as a result I ran the program multiple times in order to reach my final accuracy on my validation set of about 95%. Since I load in the previous model each time, the training adds to itself each time the code is run.