Jake Foglia

EE551 Final Project Report

Problem:

Train a neural network to differentiate between pictures of cats and dogs.

Solution:

TensorFlow for python allows for the easy creation and training of machine learning models. I used the GPU version which has a CUDA dependency. Essentially, it used my Nvidia GPU in the backend to accelerate the process of training the model. I found a [dataset](https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/data) online with roughly 25,000 labeled images of cats and dogs. Essentially the python program creates two parallel arrays for the training set. One array holds the pixel information of each image (images were rescaled to fixed dimensions for consistency of input into the model). The other array holds Booleans of whether the image is a dog or cat. These Boolean values were determined based on the label contained in the image file names.

The model generation function has a few parameters. The first two parameters are lists containing the sizes of any number of convolutional layers, followed by the sizes of any numbers of dense layers. I found in my testing that [256, 128, 64] and [512] were relatively good values in terms of yielding accurate results. The last parameter is the training data which I mentioned in the previous paragraph. Inside of this function the model is generated and trained using a variety of TensorFlow functions, which I relied on the YouTube [series](https://www.youtube.com/watch?v=gT4F3HGYXf4) by sentdex for reference. The model is then saved as an .h5 file and stored in the logs directory. The model the gets returned to the main function and then the mail while loop is entered.

In the main while loop, the user is repeatedly asked to input a number corresponding to an image in the test directory, that they’d like the model to predict whether it’s a cat or dog.