Machine Learning Engineer Nanodegree

Capstone Proposal

Vijay Souri Maddila

December 8th, 2000

Proposal

For my capstone project I am trying to find a solution for the kaggle challenge "**Dogs vs. Cats Redux: Kernels Edition**". The objective of this task is to classify whether a given image is of a cat or a dog.

Domain Background

In 2013, Kaggle hosted one of the favorite for-fun competitions: Dogs vs. Cats. Much has since changed in the machine learning landscape, particularly in deep learning and image analysis. Back then, a tensor flow was the diffusion of the creamer in a bored mathematician's cup of coffee. Now, even the cucumber farmers are neural netting their way to a bounty.

Dogs vs Cats has been one of the standards of Image analysis for quite some time. Kaggle is excited to bring back the infamous Dogs vs. Cats classification problem as a playground competition with kernels enabled. Although modern techniques may make light of this once-difficult problem, it is through practice of new techniques on old datasets that we will make light of machine learning's future challenges.

Problem Statement

The goal of this project is to predict whether a given image is of a dog or a cat. This is a classic classification problem. In this kaggle competition we need to output a prediction based on the image as [0] for the image being a cat and [1] for the image being a dog.

Machine Learning Engineer Nanodegree

Datasets and Inputs

This project will use the data set provided by Kaggle on their Kaggle Playground Prediction Competition. The data provided by kaggle has a testing and a training data set. The train folder contains 25,000 images of dogs and cats. Each image in this folder has the label as part of the filename. The test folder contains 12,500 images, named according to a numeric id.

Solution Statement

The solution for this problem can be achieved in many ways but I plan on using tensorflow's tflearn deep learning library. Using the tflearn library I will try to construct a Convolution Neural Network with a relu activation function and a maxpool layer after every convolution layer.

Benchmark Model

The benchmark model can be chosen from the leaderboards of this Kaggle competition. These models can be found here.

Evaluation Metrics

Submissions are scored on the log loss:

$$\text{LogLoss} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right],$$

Machine Learning Engineer Nanodegree

where

- n is the number of images in the test set
- $\hat{y}_i$ is the predicted probability of the image being a dog
- y_i is 1 if the image is a dog, 0 if cat
- $log()$ is the natural (base e) logarithm

Project Design

Programming Language: Python 3

Libraries: tensorflow , tflearn , cv2(opencv) , matplotlib , numpy , os , random .

The project would follow the below workflow:

- Preprocess the given data: Extract the label of image from it ID and create a proper testing and training data sets.
- Split the training data into sub training and testing data sets for measuring accuracy and other measures.
- Creating Convolution Neural Nets with tflearn.
- Fitting the model with appropriate data sets and measuring the performance.
- Add or subtract Convnet layers depending on the performance of the model.
