

Project 3: (hot) Dog or Not?

November 7, 2017

Due Date: 11/22/2017

In this project you will be training two classifiers, one to classify whether an image is a dog or a cat, and one to classify which specific breed of dog or cat a given image is.

1 Part 1: Transfer Learning

In this part of the project you will be given a dataset of 6500 images of various kinds of dogs and cats. Using these images you will have to train a Convolutional Neural Network to classify whether the image is a dog or a cat, and what specific breed of dog or cat it is. Because training CNNs takes a long, long time on CPUs we will provide you with a CNN that is pretrained on a dataset called ImageNet. Using this CNN you can perform transfer learning to adapt the CNN to this task with minimal training time. One way to do transfer learning is to use the convolutional layers of the pretrained CNN to featurize your data and then train a fully connected network on the featurized data. In part 2, we will explore why this makes sense. The provided skeleton code will walk you through how to do this. By the end you should have two models, one capable of taking an image and determining whether its a cat or a dog, and one capable of taking an image and determining what breed of cat or dog it is.

Once you have trained this model, you will need to generate predictions for the test set. This means running the trained model on the test data provided and taking the output of the model as your predictions for whether those test images are dogs or cats, and what breed of dog or cat they are.

Once you have these predictions you will submit them to a site called Kaggle, where you will be able to see how good your predictions are relative to everyone else in the class. Instructions for submitting to Kaggle will be released at a later date.

2 Part 2: Visualizing Results

In the second part of this project, you will take the results from Part 1 and visualize them using PCA. For this part we want to visualize both the original images with PCA and the post-CNN featurization of the images with PCA. The skeleton code provided will walk you through how to do this. After completing this part, you should have two plots, one of the first two principal components of the original images, labeled by class, and one of the first two principal components of the post-CNN featurization, labeled by class. Once you have those plots, answer the following questions.

Question 1: Describe the differences you see between the two plots in terms of the clustering of the classes?

Question 2: Based on your answer to Question 1, describe why it is useful to featurize with a CNN before training a fully connected model to classify the images.

3 Deliverables

- IPython notebook with code for all of the above parts
- Kaggle submission of predictions for test data set.
- Two PCA plots from Part 2.
- Answers to the 2 questions from Part 2.