

CS632

Assignment #2

Due: October 24th.

## Cats vs Dogs (and a hidden test set)

**About:** In this assignment, you will write a ConvNet to classify images of cats or dogs using [Keras](#) or [TensorFlow](#). You have been given a training dataset extracted from [Image-Net](#) containing 8,000 images. The images are 32x32x3 - small enough to fit into memory - and you should be able to train a model to reasonable accuracy in 5 to 20 minutes on a laptop without a GPU. Much like in reality - you will not see the labels for the testing data until *after* you submit your assignment.

Your goal is to write an image classifier that performs reasonably well on the training data - and more importantly - accurately on the unseen test set.

**Download your starter code** and program to generate the dataset from this link:

<https://goo.gl/NjoCVT>

**Grading:** This assignment is worth 10% of your grade. Don't sweat the details, and have fun with it. There are two parts, weighted equally.

**Submission instructions:** As before, please upload your code to a private repository in your GitHub account. See assignment #1 for a link to create one free of charge. When you are ready to turn in your assignment, submit a link to this your repo via [BlackBoard](#) before the due date. Please structure your repository as follows.

1. From assignment one, you should already have a repository called "cs632". In this repo, create a folder called "hw2". Your folder should contain these files:

cs632/

- hw2/
  - README.md (your write-up, in [Markdown](#) format)
  - train.py (code from part 1 of your assignment)
  - predict.py (code from part 2 of your assignment)
  - predictions.txt (output of your classifier on the testing data)
  - ... and any other files you use.

Remember to share your repository with the instructor, by adding them as a collaborator (if you haven't already). Instructions are in assignment #1.

### **Part 1) Train and save a model**

- Your first goal is to write, train, and evaluate a neural network that can classify the training set with better than 80% accuracy. Write your code in a file called “train.py”. Running this file should read the training dataset (you can take a path to it as a command line argument), train a model, and evaluate its accuracy.
- Once your model has been trained, your program will save a checkpoint to disk. You will load this checkpoint in part 2, to make predictions on the test set.

### **Part 2)**

- In this part of the assignment, you will write a file called “predict.py”. Running this file should load your saved model from part 1, read the testing dataset, then make predictions for every image.
- You should save these predictions to a file called “predictions.txt”, and commit it to your repo. Notice that although you have testing images, you do not have testing labels. After you submit your assignment, your predictions will be compared against the true labels.
- In terms of grading on this part of the assignment, the most important thing is that your code successfully loads the model from disk, and makes a prediction for each testing image.

### **Tips**

- Run extract\_cats\_dogs.py to create your training and validation data
- Try train\_starter.py to see how to read and display the images created
- Try predict\_starter.py to see the correct format for your output file
- There is a good chance a reasonable model to create might look something like the Keras [example](#) for cifar10...