**CISC 452 Pet Adoption Prediction  
Description**

**Garett MacGowan (10197107)  
Areege Chaudhary (10197607)**

**Overview**

The problem presented is a classification problem. The goal is to take a large dataset of features including pet attributes, sentiment analysis on pet descriptions, metadata on pet images, and pet images, and predict how quickly a pet will be adopted. There are 5 classes for adoption speed. 0 means the pet was adopted on the same day it was listed. 1 means the pet was adopted between 1 and 7 days after being listed. 2 means the pet was adopted between 8 and 30 days after being listed. 3 means between 31 and 90 days after being listed. 4 means no adoption after 100 days of being listed.

**Contributions**

Both members participated in a pair programming style for the development of the data pre-processing component of the prediction algorithm. Areege Chaudhary worked on version one of the model. Her approach was to create an ordinal regression model with rectified linear unit dense layers with dropout to prevent overfitting, and a single linear activation dense output node with mean squared error as the loss function. Results from her predictor are binned to fit the ordinal labels. Garett MacGowan worked on version two of the model. His approach was to create a classification model with sigmoid dense layers with dropout to prevent overfitting, and a five node Softmax activated output layer with sparse categorical cross entropy as the loss function.

Note that contributions are pointed out in the in-code comments.

**Code Requirements**

* Python
  + Version 3.6 (TensorFlow compatible)
* TensorFlow
  + pip install tensorflow in command prompt
  + Can install tensorflow GPU with
    - pip install tensorflow-gpu
  + Used as the back-end for Keras
* NumPy
  + pip install NumPy
  + Used mainly for data preprocessing
* Pandas
  + pip install pandas
  + For reading csv files. It handles csv files with commas in content
* Scikit-learn
  + pip install scikit-learn
  + Used for metric functions, normalizing, splitting data into equal distribution training and testing sets
* csv
  + Comes with python installation
  + Used to read csv files that do not contain commas in content
* json
  + Comes with python installation
  + Used to read json files
* glob
  + Comes with python installation
  + Used for building an iterable of file names given wildcard directory
* random
  + Comes with python installation
  + Used for checking random prediction accuracy

**Running the Code**

* Ensure that all required libraries are installed
* Run the python file in idle or another editor
* You can change the running parameters by modifying the call to main() at the bottom of the python script. In its current state, the parameters are
  + String dataDirectory
    - The directory that points to the data folder
  + Boolean preprocessed
    - Whether or not the training data and testing data has already been pre-processed. This is used to speed up computation time. It only needs to be set to False if the pre-processing strategy has changed, or if the model is being deployed. (new data needs to be pre-processed).
  + Boolean retrain
    - Whether or not to retrain the network.

**System Specs**

* The model was created, trained, and tested on the system specification below.
* Intel i7 4770k @ 4.2Ghz
* 2x GTX 780 3gb
* 16gb RAM @ 1866Mhz
* Windows 10

Note that only one GPU is utilized because the training procedure is not parallelized.