**CSE 450 Algorithm Design**

**Fall 2017**

**Extra Credit Individual Coding Assignment ( 200 points )**

Design and implement a Convolutional Neural Network (CNN) in Keras ( using a Tensorflow backend ) to distinguish between cats and dogs from the given dataset. The dataset given to you consists of a training set of 1000 images per class, a validation set of 100 images per class and a test set of 500 images of both classes combined. The width and height for each image in the dataset it 150 and 150.

You are to then train and test your CNN for the following number of epochs – 5,10,15,20 and 25. For each epoch value note down the accuracy and loss obtained during the training phase. Plot a graph of the training accuracy vs number of epochs and of the training loss vs number of epochs. Comment on the nature of the graph.

**Deliverables:**

1. Present a PDF report containing graph and comment on the results obtained.
2. Completed code in cnn.py

**Points:**

[75] graph of the training accuracy vs number of epochs

[75] graph of the training loss vs number of epochs.

[50] code submission

**Steps to install and run the CNN**

1. Setup Tensorflow for CPU ( you may use Tensorflow for GPU but it is not necessary for this project). Follow the tutorial given at the following website to help you set up Tensorflow for any given environment : https://www.tensorflow.org/install/
2. Download and install Keras on your machine. Use the following tutorial to do help you do so : <https://keras.io/#installation>
3. Once your environment is setup with Keras and Tensorflow, you can download the code given to you and modify it to accommodate a CNN which will perform the training and testing on the dataset given.
4. Repeat the training and testing phase for each epoch value given to you and note down the training loss and accuracy values for each value.

**Other helpful links**

Convolutional Neural Network :

1. <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>
2. <http://deeplearning.net/tutorial/lenet.html>
3. <https://keras.io/layers/convolutional/>
4. <https://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html>
5. <https://elitedatascience.com/keras-tutorial-deep-learning-in-python>
6. <https://machinelearningmastery.com/object-recognition-convolutional-neural-networks-keras-deep-learning-library/>
7. <https://chrisalbon.com/deep-learning-keras/convolutional_neural_network.html>