Homework 5
MSCS 550 | CMPT 404

Pablo Rivas

Assigned: Nov/3/16; Due: Nov/15/16; Points: 60

1 Instructions

This assignment could be written in LaTeX, just as the last homework assignment. Write in understandable, easy to follow English. Make sure you provide good illustrations and figures; **however, do not just throw in figures withouth a proper explanation and discussion. Figures are meant to prove a point you are making verbally; figures are a resource and not the main point.** Remember to include your Python programs in your assignment (**in GitHub only please**).

Your assignment should be submitted in two ways: through GitHub, and in hardcopy (in class). Use the **same** repository you have been using and submit your work in a folder named "`lastname-xx`", where lastname is your last name xx is the number of the assignment.

2 Problem Set

The following is a list of problems you will work on. When providing your solutions (hopefully using LaTeX), do not simply give the final answer, show how you arrived to the solution, justify your assumptions, and explain your results clearly.

1. **Experiment with k-means for color quantization**.

   Using sklearn's implementation of k-means, find the best color clustering for an image of your choice. *A good portrait of yourself could be fun (just sayin').* Color quantization is the science behind compression of images. The idea is to represent an image with fewer colors than the original. The experiment consists of the following steps:

   (a) Download to your computer the file `http://reev.us/cmpt404f16/hw/hw5.kmeans.img.py`

   (b) In the same folder where you downloaded the program, save a copy of your picture for experimentation.

   (c) Go to line 23 and set `n_colors` with your choice of a number of colors between 2 and 64. This number is actually the number of clusters (or k) we are searching for in an unsupervised manner using k-means.

   (d) Then go to line 25 and replace the image file name with the name of **your** image file. This is where your image is read into a numpy array.

   (e) Run the program. Observe the result. If the result does not look funny to you, repeat from 1.(c) until it does. Then, report your result image along with your answers to the following questions:

      i. Explain... what happens when you increase or decrease the value of `n_colors`?

      ii. Explain... in what other possible applications do you think this can be useful?

      iii. Why do you think the resulting picture was funny at the end?

2. **Neural networks: The MLP**

   (a) Download the python program `http://reev.us/cmpt404f16/hw/hw5.MLP.sol.py` which implements a 10-fold cross-validation approach to find the best number of neurons and the best learning parameter η (eta) in an MLP for regression. For learning purposes you could download `http://reev.us/cmpt404f16/hw/hw5.MLP.py` first, which is a simple implementation of the MLP.

   (b) Download the python program `http://reev.us/cmpt404f16/hw/hw5gendata.py` which generates random data.

   (c) Run the program in 2.(a) for 1,000 samples, and then take note of the best number of neurons and η. Go here `https://goo.gl/forms/QFmaNWYaFLcPWdim2` and report your results. You can do it as many times as you want, but at least one is required.

   (d) **Explain** your results. What do you think is happening? What is your interpretation of the number of neurons with respect to the performance of the network?

   (e) (**graduate students**) Repeat 2.(c)-(d) but for 10,000 samples.