Machine Learning **S2020**

Problem Set 4

Handed Out: Apr 13^{th}, 2020 *Due: April* 26^{th}, 2020

In this assignment we will be solving probably the most important problem that faces humanity today: Corgi prediction. I've gathered a subset of images from the ImageNet corpus depicting two different Corgi breeds: Pembroke and Cardigan. In this assignment, you will train a neural network to classify Corgi pictures as containing either a Pembroke or a Cardigan corgi. While we did do image recognition earlier in the semester, here we're going to use convolutional neural networks (CNNs) instead of feed forward neural networks.

Included with this assignment are two folders containing training and test data (*training_data* and *testing_data respectively*). Inside each of these folders are two subfolders containing images of Pembroke corgis and Cardigan corgis. To help you work with these images, I have also provided some code to help you out! I've also included a file, dataset.py, which contains functions to read in and resize the images. It should work for Python 3, but just let me know if anyone has any issues using the file. Also, it relies on OpenCV, a common set of libraries used in image processing, as well as glob and sklearn. Note, you do not need to use this file! You are free to set things up the way that you like for this assignment.

You will submit a writeup in Word or PDF that summarizes your results and all code as a zip file. Submit the writeup (with attached source code) to the Canvas submission locker before 11:59pm on the due date.

Corgi Classification using Convolutional Neural Networks (80 points)

For this assignment, implement a convolutional neural network to identify Corgi breeds. In your writeup, you need only report the overall accuracy of your technique as well as a description of your network's architecture. In other words, just report the percentage of the test set that your neural network is able to accurately predict as well as information on the layers of your network, the filters used, the choice of optimizer, libraries used, etc.

For this part of the assignment, I am giving you free reign to use whatever libraries and whatever functions you feel like using. I highly suggest using a library such as Tensorflow or PyTorch to simplify the process of designing and implementing your network.

Details:

- I am letting you loose. Use any libraries that you want for this assignment and any functions that you want for this assignment. Just put in your writeup what libraries are necessary to run your code!
- All code should be written in Python 3.
- While you are, ultimately, in control of your network architecture, your network MUST have at least one convolutional layer. You can choose whether or not you want pooling layers, fully connected layers, etc.; however, you must at the very least have one convolutional layer.
- You can also use whatever activation, optimizer or loss function you'd like for your network as long as you specify it in the final report. A common optimizer used for these types of problems is the ADAM optimizer, and a common loss function used is cross-entropy.
- Explain any implementation choices you had to make in the final report.
- Include the accuracy obtained by your network on the provided dataset in your writeup.

Presentation (20 bonus points)

Your report must be complete and clear. A few key points to remember:

- Complete: the report does not need to be long, but should include everything that was requested.
- Clear: your grammar should be correct and it should be easy to follow what you're saying.
- Concise: I sometimes print out reports to ease grading, don't make figures larger than they need to be. Graphics and text should be large enough to get the point across, but not much larger.

- Credit (partial): if you are not able to get something working, or unable to generate a particular figure, explain why in your report. If you don't explain, I can't give partial credit.