Assignment 7: Image Processing with CNN by Mimi Trinh

Section 1: Summary and Problem Definition

The project this week assumes that we are a consultant group providing advice to a website provider who is looking for tools to automatically label images provided by end users. The problem is similar to image recognition for tagging purpose on Facebook in which the machine can learn from past images to be able to identify the person or object of a particular image. In this project, the goal is to develop a machine learning system that can differentiate cat and dog images. The study is first developed as a Kaggle competition. The scope and methodology of this project is quite similar to the ones of assignment 6 last week. However, in this case, model accuracy is so critical for us that we are willing to sacrifice training time to develop the best model with the highest accuracy rate. Therefore, the focus of the project this week in assignment 7 is to achieve the highest accuracy rate possible instead of a mixture of high accuracy rate and reasonable training time like the scopes in assignment 6.

Section 2: Research Design, Measurement, and Statistical Methods

As mentioned earlier, the methodology for assignment 7 this week is similar to the one in assignment 6 last week. Specifically, we will employ at least a 2x2 completely crossed experimental design. We will again use a simple train and test regimen to partition the dataset into train and test sets. Unlike assignment 6 last week, for this week project, we will utilize convolutional neural networks (CNN) within Python TensorFlow and Scikit-Learn. Specifically, two blocks of code are developed with the following factors in the design.

- Code block #1: 300 nodes in layer 1, 100 nodes in layer 2, 50 epochs, 100 batch size
- Code block #2: 150 nodes in layer 1, 50 nodes in layer 2, 25 epochs, 50 batch size

In addition, the original dataset contains 25,000 images of cats and dogs. To ensure that the problem may be run on typical personal computers with limited memory and no graphical processing units, we work with only 2,000 images, including 1,000 dog images and 1,000 cat images.

Section 3: Programming Work

The images are first converted into numpy array format provided in a zipped folder. So, we won't have to repeat the process to prepare the data to convert images into pixel. We just simply read these files into Python. Then we create a TensorFlow session within Python to start building the machine learning system. It's best practice to always scale the data, so we use MinMaxScaler() in Scikit-Learn to standardize the dataset so that the minimum pixel is 0 (all white) and the maximum pixel is 255 (all black). Then we use train_test_split in Scikit-Learn to partition the dataset into train and test sets. Finally, we develop the model and determine its accuracy in TensorFlow.

Section 4: Results and Recommendations

The first code block gives 50 outputs with accuracy for train and test sets. Among these, epochs #45 and 50 have the highest accuracy in train set of 0.9 whereas epoch # 26 has the highest accuracy in test set of 0.605. The second code block gives 25 outputs. Among these, the last four outputs have the highest accuracy of 0.84 on the train set whereas epoch #24 has the highest accuracy of 0.5975 on the test set. Since the accuracy on the train set is significantly higher than the test set, perhaps we overfit the train set. Though code block #1 takes longer time, it's still a better option with higher accuracy, which is the focus of this project.

As we look across the factors in this study, CNN is the machine learning model that works best using the type of network with 300 nodes in layer 1, 100 nodes in layer 2, 50 epochs, 100 batch size. Regarding the type of images, we recommend using 64x64 numpy array files that are already processed in the zipped folder. These initial images are the types of images that work best in this project. We use them as input data for the classification task.