CSCI3230 Fundamentals of Artificial Intelligence
Neural Network Project
The Street View House Numbers Prediction
Due date: 23:59:59 (GMT +08:00), 1st January, 2018

1. Project Specification

Deep neural network have become a hot research topic in machine learning in recently years. Compare to other methods, deep learning have showed its advantages in handling large amount of data and achieving better results. Based on the structure of the network and the neurons, there are many types of deep neural networks. Among them, the deep Convolutional neural network is extremely popular in the area of image processing.

In this project, you are asked to develop a deep neural network to classify the images of dogs using TensorFlow. TensorFlow is an open source software library for numerical computation using data flow graphs. It has many convenient APIs for implementing the deep neural network, which makes the implementing convenient. The information of the dogs' images is as follows:

We have the images of 25 classes of dogs, and for each class, we have around 200 images. In total, we have 4978 images. Those images have RGB colors and are various sizes.

2. Dataset format

The 25 classes of different dogs are labeled with digits 0 to 24.

The total 4978 images are divided into a training set with 4728 images, and a testing set with 250 images (10 images for each class).

For the training data, tow folders of images are provided. The first folder is the original images with various size. The second folder contain the reshaped images, where all images are reshaped to 100*100 pixels. All images in the first and second folder are using the *.jpg* format.

The is a file named "train_label.txt", which contains the image name in the train data and its label.

The testing data, images are stored in a *.npy* format file, which is a python dictionary if you load it using the following code:

Dictionary = numpy.load("testing.npy").item()

There are three key-values pairs inside the dictionary:

1) "original": a list of ndarray with shape [height, width, 3], where height, weight is the height and weight of the testing images, and 3 is the number of RGB color in the image
2) "shapes": a list of ndarray with shape [2], where the first number is the height of the corresponding image in "original", and the second number is the weight.
3) "reshaped": a list of ndarray with shape [10000, 3]. We have compressed the testing image from [100, 100, 3] into [10000, 3] in a row first order.

Since you cannot access the testing data, to help you get familiar with the *.npy* format, we have randomly selected 250 images from the training data to create another *.npy* format file called validation.npy. The structure of validation.npy is almost same with testing.npy, except that there are two more key-value pairs in validation.npy :

1) "file_name": a list of names of the images in training data
2) "label": the label of the images.

Please keep in mind that those two key-value pairs only exist in validation.npy, not testing.npy

3. Submission requirements

1. The only supported language for this project is python2.7 (we don’t accept python3.x program.)
2. Submission list
    1. TensorFlow generated files which stores the value of your variables
    2. Source file for recovering your network model, Name it as “test.py”
    3. Any other files that helps your programs to work, such like preprocessing files, format converting files, etc
3. Submission Package:

Put everything in the submission list above into a folder, name the folder as your student id, zip the folder WITHOUT encryption. Submit the zip file into our submission system.

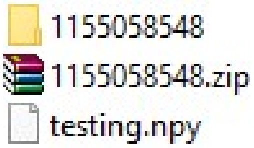
Do not add any images in your package, because the testing dataset will be prepared for you on the server as follows:

4. Testing dataset arrangement

The submission system has prepared *.npy* file of the testing dataset, which is a dictionary after you load it with numpy.

After your submission, we will unzip your package, put your sid named folder alongside the *.npy* file.

For example, there is a student whose sid is 1155058548, after his submission, the files in the submission system is organized like this:



1155058548
1155058548.zip
testing.npy

5. Grading policy

We will run your “test.py” file, which is expected to read the “testing.npy” file alongside your submission folder. You should try to predict the labels for the 250 testing images, and output your predictions into a file named “labels.txt” inside your folder. The “labels.txt” file is arranged as follows: it contains 250 lines, and each line is a digit (from 0 to 24) representing the label for the corresponding image. The accuracy score is the percentage of correct prediction that you have made. Your accuracy points is your relative performance compared to other students.

Given the time and memory constrain below, actually you cannot build a model too complex, thus it is not easy to get a high accuracy. 40% is already considered a good model. The detailed marking table will be announced after the TA run different settings.

.

6. Important Points

To make this project fair and meaningful, there are some other points you **MUST** follows:

1. The time limits to run your “test.py” is 100s
2. The maximum size of your model is 200MB.
3. Plagiarism will be SERIOUSLY punished (ZERO mark plus reporting to department)
4. Late submission will NOT be entertained according to our submission system settings

7. Late Submission

No. of Days Late	**Marks Deduction**
1	10%
2	30%
3	60%
4 or above	100%