Ve581 Convolutional Neural Networks for Visual Recognition

Summer 2019

Homework #2 (Due 7/23, Tuesday)

Note:

- In Homework #2 mini-project assignment, you will build and train a convolutional neural networks (classifier) for traffic sign recognition based on German Traffic Sign Dataset, with an accuracy on the validation set of 95% or greater.

Submission files (to Canvas):

- Jupyter notebook with code or python file with code
- PDF of the code
- A writeup PDF report:
    - The report would describe the main steps that you used to
        - explore the dataset
        - design and test your classifier
        - use your classifier to make predictions on the new images
    - Please include the necessary code or figures

(Total: 100 points)

1. Import your data:
    a. We've provided the dataset through canvas which includes the training, validation and test set.
    b. Please use the following code to import your data:

```
import pickle

training_file = 'train.p'
validation_file= 'valid.p'
testing_file = 'test.p'

with open(training_file, mode='rb') as f:
    train = pickle.load(f)
with open(validation_file, mode='rb') as f:
    valid = pickle.load(f)
with open(testing_file, mode='rb') as f:
    test = pickle.load(f)

X_train, y_train = train['features'], train['labels']
X_validation, y_validation = valid['features'], valid['labels']
X_test, y_test = test['features'], test['labels']
```

2. (10 points) Data Exploration:
    a. (6 points) Dataset Summary
        i. Number of samples in each set
        ii. Shape of the traffic image ie. (x, x, x)
        iii. Number of classes/labels

    b. (4 points) Exploratory Visualization
        i. For each class/label, plot a sample image
3. (70 points) Design and Test a Classifier (or model architecture)
    a. (10 points) Preprocess the dataset using techniques such as normalization, colors converting, and explain why you choose those techniques in report
    b. (15 points) Design your model architecture with at least 5 layers, and show the architecture in your report
    c. (10 points) Compile your model: choose the loss function, optimizer, and metrics, batch size, number of epochs, and other relevant values of hyperparameters
    d. (10 points) Train your model, and plot your training history
    e. (15 points) Tune your model or change your model if your validation accuracy is less than 95%, and save your final model which has a validation accuracy higher than 95%
    f. (10 points) Evaluate your final model's performance, and make predictions on 10 randomly selected test image
4. (20 points) Test Your Classifier on New Images
    a. (10 points) Download 10 pictures of the traffic signs from the internet and use your model to predict the traffic sign type. You might need to preprocess the pictures.
    b. (10 points) Output the top 5 softmax probabilities for each above picture
5. (optional bonus: 5 points) Visualize selected the convolutional neural networks of the test images to help you understand what features your model extracted from the images.