Traffic Signs Recognition

Build a Traffic Sign Recognition Project

The goals / steps of this project are the following:

- Load the data set (see below for links to the project data set)
- Explore, summarize and visualize the data set
- Design, train and test a model architecture
- Use the model to make predictions on new images
- Analyze the softmax probabilities of the new images
- Summarize the results with a written report

Here is the link to my project code.

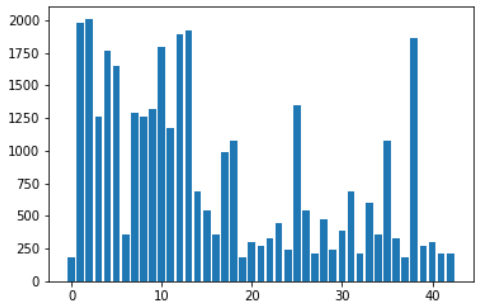
Data Set Summary & Exploration

1. Data Set Summary

First, I downloaded the data set from here and then extracted it. The data set includes three sets: **train.p** for training, **valid.p** for validation and **test.p** for test. The train, validation and test set contain 34799, 4410 and 12630 examples respectively. The size of each example is 32*32 in width and height. There are 43 unique classes for the entire sets.

2. Exploratory visualization of the dataset

Here is an exploratory visualization of the train data set. Each number in the horizontal axis represents a class and the vertical axis stand for the number of examples in corresponding class.



Design and Test a Model Architecture

1. Preprocess the data

At first, I augmented the train data set because, as can be seen from the above visualization picture, the train set is very unbalanced. Some classes such as the first class contain about only 200 examples while others may include around 2000 examples like the second class. If I train the model with this data set, the classifier may not unable to recognize some classes with small amount examples. So, I applied random translation, random rotation, random scaling and random brightness on the train set with OpenCV to make each class contain the same number of examples. Here are the examples of the traffic sign before and after each random processing.

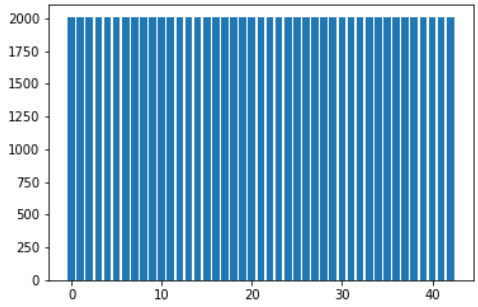
Random translation

Random rotation

Random scaling

Random brightness

After augmentation, the distribution of each class in the data set is shown below.



Then I processed the training set to make it suitable for training. The process includes grayscaling the images, enhancing the contrast and normalization. Shape is the essential for recognition instead of color. As a result, I applied the grayscaling to make the training process more efficient. Then, the quality of some examples is not satisfied, so I enhanced the contrast of the images by adaptive histogram equalization in OpenCV. Finally, I normalized the data set to make the optimization converge faster. The image after grayscaling and enhancing contrast is shown below.

2. Model structure

My final model consisted of the following layers:

Layer	Description
Input	32×32×1 gray image
Convolution1 5×5	1×1 stride, same padding, outputs 32×32×32
RELU	
Max pooling	2×2 stride, outputs 16×16×32
Convolution2 5×5	1×1 stride, same padding, outputs 16×16×64
RELU	
Max pooling	2×2 stride, outputs 8×8×64
Convolution3 5×5	1×1 stride, same padding, outputs 8×8×128
RELU	
Max pooling	2×2 stride, outputs 4×4×128
Fully connected	inputs conv1+conv2+con3, outputs 1024
Fully connected(output)	inputs 1024, outputs 43

3. Train model

Here are the hyperparameters I used for training this model:

Parameter	Value
Batch Size	128
Learning Rate	0.001
Optimizer	Adam Optimizer
Number of epochs	100
Weight Initializer	Xavier
Bias Initializer	Zeros
Dropout on conv1	20%
Dropout on conv2	30%
Dropout on conv3	40%
Dropout on fully connected layer	50%
L2 Regulation coefficient	0.001
Loss	Cross Entropy

4. Result

After 100 epochs, the model reached the loss of 0.058, the training set accuracy of 99.58%, the validation set accuracy of 98.55% and the test set accuracy of 96.32%.

Test a Model on New Images

1. Five new German traffic signs

Here are five German traffic signs that I found on the web:

First, I renamed these images with respective class number corresponding to the signname file so that I can find their classes easily. Then I resized these pictures to 32×32 to fit the input size of the classifier. Finally, I preprocessed them with the same method as the training examples.

2. Test result

Here are the results of the prediction:

Image	Prediction
Priority road	Priority road
Yield	Yield
Stop	Stop
Turn right ahead	Turn right ahead
Speed limit (70km/h)	Speed limit (70km/h)

The model performed really good on these five images, which gave an accuracy of 100%.

The top 5 softmax probabilities of these images were: