Traffic Sign Classifier (TSC) Project Submission Writeup Report

For the Udacity Self-Driving Car Program – Term 1

Submitted by:

Stewart Teaze

May 26, 2017

Project Goals

The goals / steps of this project are:

- Load the provided German Traffic Signs 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

Rubric Points

The following sections of this document relate to each of the project's rubric points (project requirements), and provide details of how each requirement was addressed in the project implementation.

Writeup

1. This file is the Writeup itself, and is provided in .pdf format. The project code can be found at: https://github.com/blunderbuss9/CarND-TSC-StewartTeaze.html

Data Set Summary & Exploration

1. **Following are basic summary statistics of the provided German Traffic Signs data set:**

   - The size of (original) training set is 34799
   - The size of the validation set is 8699 (1/4 * 34799)
   - The size of test set is 26100 (3/4 * 34799)
   - The shape of a traffic sign image is (32,32,3)
   - The number of unique classes/labels in the data set is 43

2. **Exploratory visualization of the dataset.**

Table 1 below provides an exploratory visualization of the provided German Traffic Signs data set, and includes additional summary statistics. In the project code itself, this analysis and visualization was performed and rendered using python, with the numpy and matplotlib libraries. The table provides one representative image from the training set file, corresponding to each Traffic Sign type that was found in the training set file.

The title text above each image is X:Y, where X is the Traffic Sign Class label/type, & Y is the number of occurrences of that label/type that were encountered in the training set file:

TABLE 1

Design and Test of the Model Architecture

1. Image Data Preprocessing.

The images that were provided in the German Traffic Sign dataset file train.p (a subset of the German Traffic Sign Recognition Benchmark dataset of more than 50,000 traffic sign images [Stallkamp][AnsBach]), are read in, and then split into two parts: 75% training data, and 25% validation data; by doing this, validation set accuracy measurements of between 0.944 and 0.961 were achieved, during various test runs against the created validation set - all of these measurements were above the 0.93 minimum requirement for a successful project submission, and a significant increase over the roughly .88 accuracy achieved during initial runs made with the provided validation dataset (whose size was only roughly 15% of the training set).

The training data is shuffled randomly for each epoch processed during each run – it is important to randomly shuffle the training data, otherwise the order of the data could have a huge effect on how well the network model trains.

1. Model Architecture Description.

The machine learning network model architecture implementation is a multi-layer (Deep Learning) Convolutional Neural Network (CNN), that is based on and adapted from the LeNet-5 5-Layer architecture [LeCun98]. The following table provides the details for the stages (steps/techniques) implemented, within each layer of this project's machine learning network model's architecture implementation:

	STAGES / LAYER			
LAYER 1	<u>Convolutional</u> Stride: 1x1 Padding: VALID Input: 32x32x3(RGB) Output: 28x28x6	<u>Activation</u> Type: ReLU	<u>Max Pooling</u> Stride: 2x2 Padding: VALID Input: 28x28x6 Output: 14x14x6	
LAYER 2	<u>Convolutional</u> Stride: 1x1 Padding: VALID Input: 14x14x6 Output: 10x10x16	<u>Activation</u> Type: ReLU	<u>Max Pooling</u> Stride: 2x2 Padding: VALID Input: 10x10x16 Output: 5x5x16	<u>Flatten</u> Input: 5x5x16 Output: 400
LAYER 3	<u>Fully Connected</u> Input: 400 Output: 120	<u>Activation</u> Type: ReLU		
LAYER 4	<u>Fully Connected</u> Input: 120 Output: 84	<u>Activation</u> Type: ReLU		
LAYER 5	<u>Fully Connected</u> Input: 84 Output: 43 (Logits)			

TABLE 2

2. Model Training Description

The last output (logits), described in the table above, is then softmax cross-entropy one-hot encoded, for later use by the AdamOptimizer loss minimization functional step in each epoch pass in the training pipeline.

A batch size of 128 per epoch was used; and the training pipeline used 10 epochs (passes). The learning rate hyperparameter was set to .001. Mean and standard deviation parameters within all layers of the model were set to 0 and .1, respectively.

3. Solution Approach used for Achieving Validation Accuracy > 0.93

The well-known LeNet-5 5-Layer CNN architecture was used, as it has been shown to yield good results on image recognition type problems, and has been well-studied, with numerous support resources readily available on The Internet; for example, the LeNet and CNN approaches are the core of the Stanford Computer Vision course CS231n, which has generated huge interest throughout the Artificial Intelligence machine learning community, and whose resources are accessible via The Internet to those interested **[CS231n17].**

The main approach used, to achieve validation accuracy was to split the provided German Traffic Sign dataset file train.p, into two parts: 75% training data, and 25% validation data; by doing this, validation set accuracy measurements of between 0.943 and 0.961 were achieved, during numerous test runs against this increased-size validation set. These measurements were a significant improvement over the roughly .88 accuracy achieved during initial runs I had made with the provided validation dataset file valid.p (whose size was only roughly 15% of the training set). This approach was used, because numerous research results have shown the validation dataset should be a "substantial fraction" of the training dataset size, to achieve optimal results… "In most cases a single validation set of respectable size substantially simplifies the code base, without the need for cross-validation with multiple folds. You'll hear people say they "cross-validated" a parameter, but many times it is assumed that they still only used a single validation set" **[CS231n17].**

My final model results were:

- validation set accuracies of between .0943-.0961
- test set accuracy of .866

Model Test Results using Web-based Images

Following are five sample German traffic signs that I found on the web, and used for the test evaluation run.

1. **Sample Traffic Sign Images**

    a. This image, of an 100 km/h speed limit sign, found at http://media.istockphoto.com/vectors/german-road-sign-speed-limit-100-kmh-vector-id472360055 was cropped to 512x512 pixels using MS Paint; while it has some watermark artifacts, it can be seen that the effects on classification should be minimal, once the image is re-sized to 32x32 pixels in the project code during the evaluation run (NOTE: for comparison purposes, the 32x32 re-sized images, produced and output during the project code evaluation section run, are included next to each of the original-sized images presented here).

b. This image, of a Traffic Signal Ahead sign, found at

https://thumb10.shutterstock.com/display_pic_with_logo/2535709/456268447/stock-vector-germany-traffic-signals-sign-456268447.jpg

was originally cropped to 492x488 pixels; while there are some watermark overlay artifacts, it can be seen that the effects on classification should be minimal, once the image is re-sized to 32x32 pixels in the project code during the evaluation run (see 2nd image below-right).

c. This image, of a No Passing by Trucks sign, found at

http://blog.skytee.com/2013/01/autobahn-a-guide-for-my-foreign-friends/

was originally cropped at 389x383 pixels; the image re-sized to 32x32 pixels in the project code during the evaluation run is shown in the 2nd image below-right.

d. This image, of a German Yield Sign, found at

http://www.gettyimages.com/detail/illustration/yield-german-road-sign-royalty-free-illustration/176430515

was originally cropped to 392x335 pixels; while not square(leading to some aspect ratio issues when re-sized to 32x32), and while there are some watermark artifacts, it can be seen that the effects of these issues on classification should be minimal, once the image is re-sized to 32x32 pixels in the project code during the evaluation run (see 2nd image below).

e. This image, of a German Warning Sign, found at

http://www.gettyimages.com/detail/illustration/dangerous-area-risk-of-ice-german-warning-royalty-free-illustration/472360059

was originally cropped at 570x491 pixels; there are some watermark artifacts, but it can be seen that the effects on classification of these artifacts should be minimal, once the image is re-sized to 32x32 pixels in the project code during the evaluation run(see 2nd image below-right).


2. Trained Model's Prediction Accuracy on New Dataset.

Here are the results of the predictions on the 5 web sample images:

Image	Prediction
100 km/h	100 km/h
Signal Ahead	Signal Ahead
No Truck Passing	No Truck Passing
Yield	60km/h
Danger	Danger

The trained model was able to correctly guess 4 of the 5 traffic signs, which gives an accuracy of 80%. This compares closely to the accuracy of 86% against the test set of images.

3. Top 5 Softmax Probabilities for Each of the 5 Sample Images.

The code for making predictions for each of the 5 sample images, with the final version of this project submission's model, is located in the 12th(last) code cell of the submitted Jupyter notebook.

For the first image, the model is slightly sure that this is an 100 km/h sign (probability of 0.0654); and indeed the actual image is a 100 km/h sign.

The top five soft max probabilities were:

Probability	Prediction
.0654	100 km/h
.0653	120 km/h
.062	80 km/h
.057	70 km/h
.054	No Truck Passing

For the second image, the model is somewhat sure that this is a Traffic Signal Ahead sign (probability of 0.086); and indeed the actual image is a Traffic Signal Ahead sign.

The top five soft max probabilities were:

Probability	Prediction
.086	Traffic Signal Ahead
.070	Danger
.059	Deer Warning
.050	Single Curve Warning
.049	Freeze Warning

For the third image, the model is relatively sure that this is a No Truck Passing sign (probability of 0.118); and indeed the actual image is a No Truck Passing sign. The top five soft max probabilities were:

Probability	Prediction
.118	No Passing for Trucks
.067	No Passing
.064	80 km/h
.052	100 km/h
.041	60 km/h

For the fourth image, the model is slightly sure that this is an 60 km/h sign (probability of 0.61); however, the actual image is a Yield sign, which the model chose as its second prediction (probability of .059). The top five soft max probabilities were:

Probability	Prediction
.061	60 km/h
.059	Yield
.068	No Passing
.048	Single Curve Warning
.047	No Passing for Trucks

For the fifth image, the model is slightly sure that this is a Danger sign (probability of 0.061); and indeed the actual image is a Danger sign. The top five soft max probabilities were:

Probability	Prediction
.061	Danger
.053	Single Curve Warning
.052	Traffic Signal Ahead
.047	120 km/h
.044	Deer Warning

REFERENCES

[Stallkamp] Stallkamp, I., Schlipsing, M., Salmen, J., Igel, C. (2011). The German Traffic Sign Recognition Benchmark: A multi-class classification competition; The 2011 International Joint Conference on Neural Networks (IJCNN). http://ieeexplore.ieee.org/document/6033395/

[Ansbach] US Army Garrison Ansbach; "FOR SAFETY You must KNOW German Traffic Signs" Pamphlet. USAREUR POSTER 190-34-12, Jan, 2001; http://www.ansbach.army.mil/documents/EuropeanRoadSignsPamphlet.pdf

[LeCun98] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998d). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.

[CS231n17] Karpathy, A., Li, F. (2017) http://cs231n.stanford.edu/ Convolutional Neural Networks for Visual Recognition; Stanford University, Spring 2017.