Traffic Sign Recognition

Writeup

You can use this file as a template for your writeup if you want to submit it as a markdown file, but feel free to use some other method and submit a pdf if you prefer.

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

Rubric Points

Here I will consider the rubric points individually and describe how I addressed each point in my implementation.

Writeup / README

1. Provide a Writeup / README that includes all the rubric points and how you addressed each one. You can submit your writeup as markdown or pdf. You can use this template as a guide for writing the report. The submission includes the project code.

You're reading it! and the project code is submitted as zip.

Data Set Summary & Exploration

1. Provide a basic summary of the data set. In the code, the analysis should be done using python, numpy and/or pandas methods rather than hardcoding results manually.

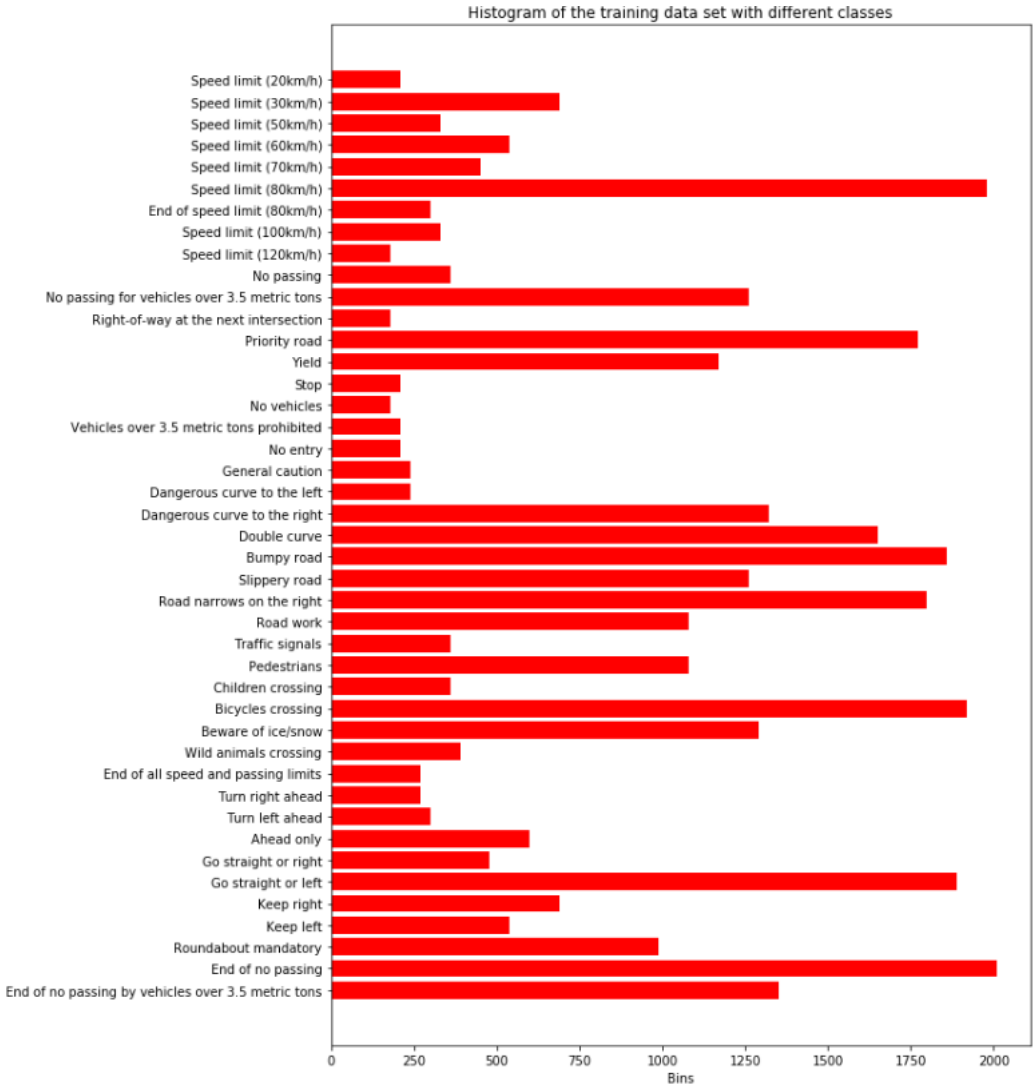
I used the pandas library to calculate summary statistics of the traffic signs data set:

- The size of training set is 34799
- The size of the validation set is 4410
- The size of test set is 12630

- The shape of a traffic sign image is (32,32,3)
- The number of unique classes/labels in the data set is 43

2. Include an exploratory visualization of the dataset.

Here is an exploratory visualization of the data set. It is a bar chart showing how the data ...



Design and Test a Model Architecture

1. Describe how you preprocessed the image data. What techniques were chosen and why did you choose these techniques? Consider including images showing the output of each preprocessing technique. Pre-processing refers to techniques such as converting to grayscale, normalization, etc. (OPTIONAL: As described in the "Stand Out Suggestions" part of the rubric, if you generated additional data for training, describe why you decided to generate additional data, how you generated the data, and provide example images of the additional data. Then describe the characteristics of the augmented training set like number of images in the set, number of images for each class, etc.)

As a first step, I decided to normalize the images as learnt in the lesson to increase the accuracy and it was indeed reflected in the results !!

2. Describe what your final model architecture looks like including model type, layers, layer sizes, connectivity, etc.) Consider including a diagram and/or table describing the final model.

My final model consisted of the following layers:

Layer	Description
Input	32x32x3 RGB image
Convolution 1	1x1 stride, VALID padding, output = 28x28x6
ReLU	
Max pooling	2x2 stride, VALID padding, output = 14x14x6
Convolution 2	1x1 stride, VALID padding, output = 10x10x16
ReLU	
Max pooling	2x2 stride, VALID padding, output = 5x5x16
Flatten	Output = 400
Fully connected 1	Input = 400 Output = 120
ReLU	
Dropout	Prob = 0.6
Fully connected 2	Input = 120 Output = 84
ReLU	
Dropout	Prob = 0.6
Fully connected 3	Input = 84 Output = 43

3. Describe how you trained your model. The discussion can include the type of optimizer, the batch size, number of epochs and any hyperparameters such as learning rate.

To train the model, I used an epoch size = 25 and batch size = 128 with learning rate = 0.001. Initially I did not use dropout which resulted approximately 0.95 and after using the dropout, it significantly improved the accuracy !

4. Describe the approach taken for finding a solution and getting the validation set accuracy to be at least 0.93. Include in the discussion the results on the training, validation and test sets and where in the code these were calculated. Your approach may have been an iterative process, in which case, outline the steps you took to get to the final solution and why you chose those steps. Perhaps your solution involved an already well known implementation or architecture. In this case, discuss why you think the architecture is suitable for the current problem.

My final model results were:

* training set accuracy of = 99.7

* validation set accuracy of = 97.7

* test set accuracy of = 93.72

If a well known architecture was chosen: * What architecture was chosen? * Why did you believe it would be relevant to the traffic sign application? * How does the final model's accuracy on the training, validation and test set provide evidence that the model is working well?

I chose LeNet architecture itself as it is well suited for small sized images like 32x32. Though I was skeptical initially, but surprised to observe above 90% accuracy in test/validation test sets. Also, I did not have to try augmenting the dataset !

Test a Model on New Images

1. Choose five German traffic signs found on the web and provide them in the report. For each image, discuss what quality or qualities might be difficult to classify.

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

The first image might be difficult to classify because ...

2. Discuss the model's predictions on these new traffic signs and compare the results to predicting on the test set. At a minimum, discuss what the predictions were, the accuracy on these new predictions, and compare the accuracy to the accuracy on the test set (OPTIONAL: Discuss the results in more detail as described in the "Stand Out Suggestions" part of the rubric).

Here are the results of the prediction:

Image	Prediction
Keep Right	Keep right
No Entry	No entry

Stop	Stop
Watch for children	Speed limit (20km/h)
Yield	Yield

The model was able to correctly guess 4 of the 5 traffic signs, which gives an accuracy of 80%.

3. Describe how certain the model is when predicting on each of the five new images by looking at the softmax probabilities for each prediction. Provide the top 5 softmax probabilities for each image along with the sign type of each probability. (OPTIONAL: as described in the "Stand Out Suggestions" part of the rubric, visualizations can also be provided such as bar charts)

The code for making predictions on my final model is located in the 14th cell of the Ipython notebook.

The top five soft max probabilities were

```
1_keep_right.jpg:
Keep right: 100.00%
Roundabout mandatory: 0.00%
Turn left ahead: 0.00%
Speed limit (20km/h): 0.00%
Speed limit (30km/h): 0.00%

2_No_entry.jpg:
No entry: 100.00%
Stop: 0.00%
Bicycles crossing: 0.00%
Children crossing: 0.00%
Dangerous curve to the right: 0.00%

3_Stop_Sign.jpg:
Stop: 99.96%
Yield: 0.03%
Speed limit (60km/h): 0.01%
No vehicles: 0.00%
Road work: 0.00%

4_Watch_for_Children.jpg:
Speed limit (20km/h): 98.89%
Speed limit (30km/h): 0.96%
General caution: 0.14%
No entry: 0.00%
Dangerous curve to the right: 0.00%

5_yield.jpg:
Yield: 100.00%
No vehicles: 0.00%
No passing: 0.00%
Turn left ahead: 0.00%
Road work: 0.00%
```

(Optional) Visualizing the Neural Network (See Step 4 of the Ipython notebook for more details)

1. Discuss the visual output of your trained network's feature maps. What characteristics did the neural network use to make classifications?