

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. Writeup / README includes all the rubric points

You're reading it! and here is a link to my [project code](#)

Data Set Summary & Exploration

1. Provide a basic summary of the data set.

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 10 random pics from training set with its class number.



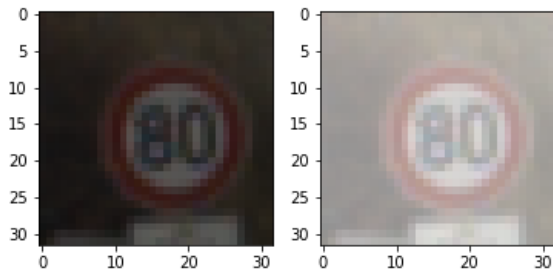
Design and Test a Model Architecture

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 so that the data has mean zero and equal variance. By doing $(data/255 - 0.5)$

Here is an example of a traffic sign image before and after normalization.



As a last step, I convert the images to grayscale because I want to speed up the training process, and further more the color is unlikely to give any performance but the shape of signs are more important

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	32x32x1 gray image
Convolution 5x5	1x1 stride, valid padding, outputs 28x28x12
RELU	
Max pooling	2x2 stride, outputs 14x14x12
Convolution 5x5	1x1 stride, valid padding, outputs 10x10x36
RELU	
Max pooling	2x2 stride, outputs 5x5x36
Flatten	
Fully Connected	Input = 900. Output = 320.
RELU	
Dropout	
Fully Connected	Input = 320. Output = 144.
RELU	
Fully Connected	Input = 144. 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 the follow global parameters:

- Number of epochs = 15
- Batch size =128
- Learning rate = 0.001
- Dropout = 0.5

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 0.99747
- validation set accuracy of 0.945
- test set accuracy of 0.931

This solution based on modified LeNet-5 architecture. With the original LeNet-5 architecture, I've got a validation set accuracy of about 0.87.

Architecture adjustments:

- perform preprocessing (grayscale and normalization). Results for training and validation sets were 0.99 and 0.86 that's overfitting.
- Add a dropout layer with keep_prob 0.5. Results for training and validation sets are 0.99796 and 0.955

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 (32x32x3):








The sign #1 might be difficult to classify because it has noisy background.

The sign #3 might be difficult to classify because it's rotated.

The sign #4 might be difficult to classify because it's dark.

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:

	SignName: Right-of-way at the next intersection Predicted: Right-of-way at the next intersection
	SignName: Road work Predicted: Road work
	SignName: Stop Predicted: Keep right
	SignName: Priority road Predicted: Priority road
	SignName: Speed limit (60km/h) Predicted: Speed limit (30km/h)

The model was able to correctly guess 3 of the 5 traffic signs, which gives an accuracy of 60%. This compares favorably to the accuracy on the test set of 93.1%

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 26th cell of the Ipython notebook.

The top five softmax probabilities are:

SignName	
0.97178089618682861328%	- Right-of-way at the next intersection
0.02821105904877185822%	- Right-of-way at the next intersection
0.00000579641573494882%	- Priority road
0.00000201223474505241%	- Beware of ice/snow
0.0000007892993636460%	- Children crossing
	- Roundabout mandatory

SignName	
0.99999952316284179688%	- Road work
0.00000041525356664351%	- Road work
0.00000004666575748047%	- Dangerous curve to the right
0.00000003664667858061%	- Turn right ahead
0.00000001693380546897%	- Wild animals crossing
	- Ahead only

SignName	
0.39681142568588256836%	- Stop
0.17977146804332733154%	- Keep right
0.14190417528152465820%	- Yield
0.13524436950683593750%	- Speed limit (60km/h)
0.05636766552925109863%	- Stop
	- Speed limit (50km/h)

SignName	
0.99999892711639404297%	- Priority road
0.00000111901010768634%	- Priority road
0.00000000713661574281%	- Roundabout mandatory
0.00000000165021574183%	- Keep right
0.0000000013054050896%	- Stop
	- End of all speed and passing limits

SignName	
0.93602138757705688477%	- Speed limit (60km/h)
0.06295760720968246460%	- Speed limit (30km/h)
0.00035545934224501252%	- Speed limit (60km/h)
0.00034560993663035333%	- Keep right
0.00025409553200006485%	- Speed limit (50km/h)
	- Speed limit (80km/h)

(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?