

Traffic Sign Recognition

Building a Traffic Sign Recognition Project

The goals / steps of this project are the following:

- Load the 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

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.

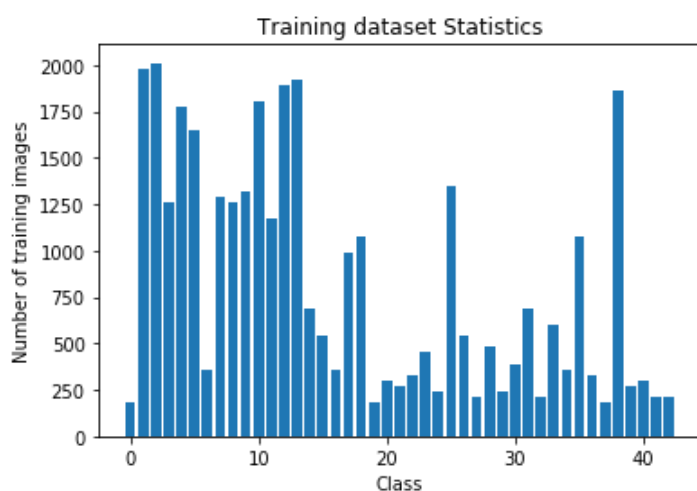
I used the pandas library to calculate summary statistics of the traffic signs data set:

The data set can be found here: <https://s3-us-west-1.amazonaws.com/udacity-selfdrivingcar/traffic-signs-data.zip>

- 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 ...



It can be noticed that some classes have low number of training data of approximately 250 images such as, class 0, 41 and 42, while other classes have nearly 20,000 images.

This could result in the model capable of predicting some classes much better than others.

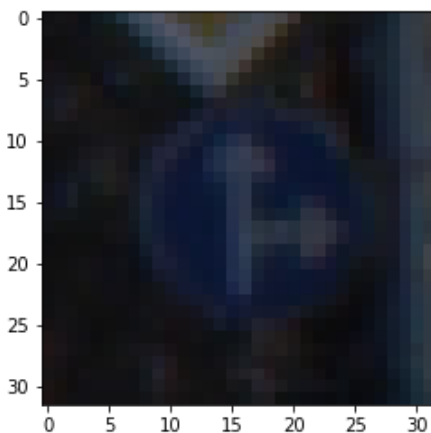
Augmenting the training data of the classes lacking training image could be a solution for this problem.

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.)

Below is a random traffic sign image taken from the training dataset.

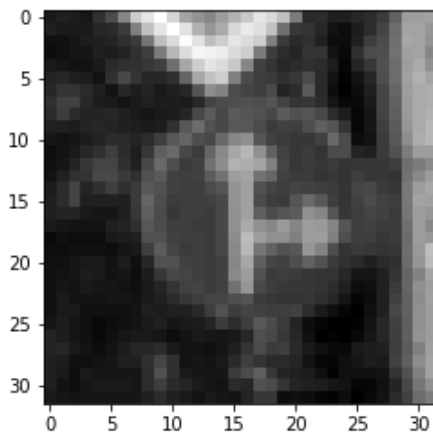
The image is a colored one (i.e. 3-dimensional image)



As a first step in preprocessing the training images, I decided to convert the images to grayscale. This will make the images of only one dimension, decreasing the image depth will in turn also decrease the number of parameters that have to be adjusted by the CNN, also color is not a significant factor in specifying the traffic signs.

The second step in preprocessing the images is done by normalizing them to take values between 0 and 1 that corresponds to the grayscale range 0 to 255. This normalization will help decrease the variance of the pixels intensities and will in turn help the neural network to tune its parameters faster.

Here is the traffic sign image after preprocessing.



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 grayscale image
Convolution 5x5	60 feautre maps, 1x1 stride, valid padding, outputs 28x28x60
RELU	
Max pooling	2x2 stride, outputs 14x14x60
Convolution 3x3	30 feautre maps, 1x1 stride, valid padding, outputs 12x12x30
RELU	
Max pooling	2x2 stride, outputs 6x6x30
Fully connected	Input = 1080, Output = 500
Fully connected	Input = 500, Output = 120
Softmax	Input = 120, 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 following parameters:

- EPOCHS of 15
- BATCH_SIZE of 128
- learning rate of 0.001

To minimize the cross entropy I used Adam optimizer, that is an extension to stochastic gradient descent to update the network weights based on the training data.

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 99.6 %
- validation set accuracy of 95.5 %
- test set accuracy of 93.4 %

What were some problems with the initial architecture?

The first architecture I tried was the typical LeNet architecture.

The initial architecture of LeNet didn't have dropout layers. So I added one dropout layer to prevent the network from overfitting depending on the training set, as this will result in a low validation accuracy.

How was the architecture adjusted and why was it adjusted?

- In order to increase the accuracy, I also added another Fully connected layer to increase the number of the model parameters.
- Also I increased the feature maps in the convolution layers to 60 and 30 for the first and the second convolution layers respectively.
- The number of epochs was also increased to 15 in order to compensate for having a low learning rate and to reach in the end of the training the required accuracy.

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 seven German traffic signs that I found on the web:



The Correct Prediction percentage of these 7 images was 71.43 %

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	Correct Prediction?
Roundabout	Roundabout	yes

Image	Prediction	Correct Prediction?
Forward or right	Forward or right	yes
60 km/h limit	No entry	No
Go right	Go right	yes
Road work	Speed limit (80km/h)	No
Yield	Yield	yes
Intersection	Intersection	yes

The model was able to correctly guess 5 of the 7 traffic signs, which gives an accuracy of 71.43 %.

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)

For the first image (Roundabout) was correctly predicted, the model is relatively sure that this is a stop sign (probability of 0.6), and the image does contain a stop sign. The top five softmax probabilities were

Probability	Prediction
0.997	40 Roundabout
2.8e-3	12 Priority road
3.1e-4	7 speed limit (100km/h)
6.04e-6	8 Speed limit (120km/h)
1.27e-6	12 Priority road

For the second image (Forward or right): correctly predicted

Probability	Prediction
0.995	36 Go straight or right
4.5e-3	35 Ahead only
2.9e-5	34 Turn left ahead
8.3e-6	25 Road work

Probability	Prediction
1.9e-6	32 End of all speed and passing limits

For the third image (60 km/h limit): not correctly predicted

I believe the reason this image was not correctly classified is due to the fact that the image contain other details like the pole where the sign is mounted plus some text

Probability	Prediction
0.48	17 No entry
0.33	34 Turn left ahead
0.14	40 Roundabout
0.046	35 Ahead only
0.003	23 Slippery road

For the fourth image (Go right): correctly predicted

Probability	Prediction
0.72	33 Turn right ahead
0.17	40 Roundabout
0.04	21 Double curve
0.038	39 Keep left
0.011	25 Road work

For the fifth image (Road work): not correctly predicted

I believe the reason this image was not correctly classified is due to the fact that the image contain other details like trees and also text

Probability	Prediction
0.88	5 Speed limit (80km/h)
0.102	31 Wild animals crossing
0.009	29 Bicycles crossing
0.004	25 Road work
0.001	10 No passing for vehicles over 3.5 metric tons

For the sixth image (Yield): correctly predicted

Probability	Prediction
~ 1	13 Yield
~ 0	38 Keep right
~ 0	14 Stop
~ 0	12 Priority road
~ 0	9 No passing

For the seventh image(intersection): correctly predicted

Probability	Prediction
0.99	11 Right-of-way at the next intersection
7.7e-7	30 Beware of ice/snow
~ 0	21 Double curve
~ 0	19 Dangerous curve to the left
~ 0	40 Roundabout