

Traffic Sign Recognition

Alejandra Vargas - 11.11.17

Project 2

Data Set Summary & Exploration

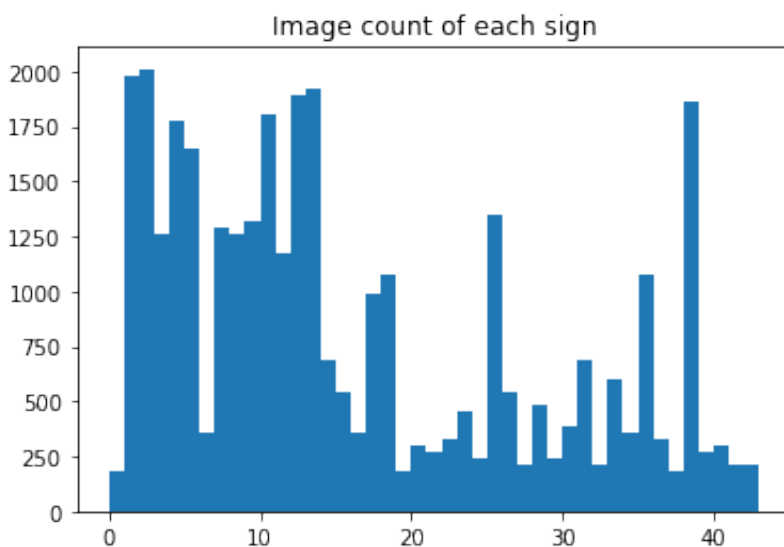
1. Provide a basic summary of the data set.

I used the numpy 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:
- 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. Exploratory visualization of the dataset.

Here is an exploratory visualization of the data set. It is a histogram showing the amount of images for each sign.



Design and Test a Model Architecture

1. Preprocessing the image data.

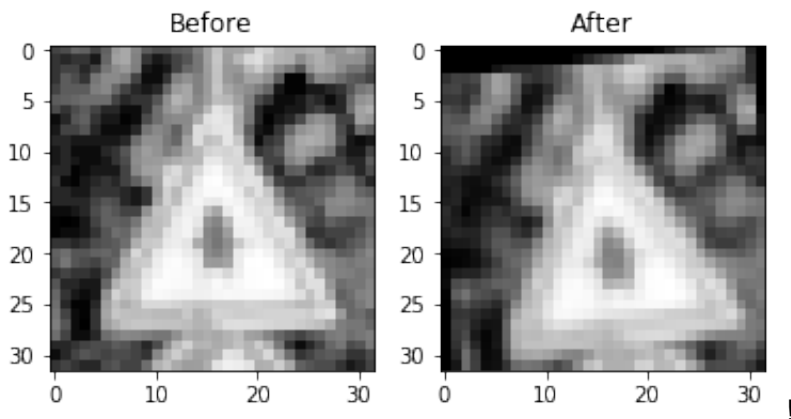
As a first step, I decided to convert the images to grayscale because this way the number of parameters in the model is significantly reduced. Then I decided to apply histogram equalization to improve the contrast of the images, in order to stretch out the intensity range.

As a last step, I normalized the image data to get from the grayscale image: [0 255] to the normalized image: [-1 1].

I decided to generate additional data by applying random similarity transformation to each of the images and then returning n transformed images. I found that increasing the training data size had a direct effect in the accuracy for the testing. While training the model I found that generating additional data times 5, gave me the best result.

To add more data to the the data set, I applied a random angle rotation between -15 and 15, varied the scale image by .3 and applied a random translation in x and y . This with the objective of generating more data slightly different to better train the model.

Here is an example of an original image and an augmented image:



2. Final model Architecture.

My final model consisted of the following layers:

Layer	Description
Input	32x32x3 RGB image
Convolution 3x3	1x1 stride, same padding, outputs 28x28x6
RELU	
Max pooling	2x2 stride, outputs 14x14x6
Convolution 3x3	1x1 stride, outputs 10x10x16
RELU	
Max pooling	2x2 stride, outputs 5x5x16
Flatten	output 400
Fully connected	output 120
RELU	
Dropout	
Fully connected	output 43
Dropout	

3. Training the model.

To train the model, I used Adam Optimizer, the hyperparameters are: * Batch size: 160 * EPOCHS: 25 * Learning rate: .001

4. Approach to finding the solution

My final model results were: * training set accuracy of 0.974 * validation set accuracy of 0.935 * test set accuracy of 0.60

I worked over the LeNet from the CNN lesson and modified it so that it would take grayscale images, that helped reducing the amount of parameters. Generating the extra training data to decrease the shadow network, I noticed that the lower the data, the lower the performance. After that I worked mainly in trial and error experimenting with different batch sizes, number of epochs and learning rates to try and avoid overfitting or underfitting.

Test a Model on New Images

1. Five German traffic signs found on the web.

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



2. Model's predictions on new traffic signs

Here are the results of the prediction:

Image	Prediction
Children crossing	Double curve
Road work	Beware of ice/snow
General caution	General caution
Priority road	Priority road
Speed limit (30km/h)	Speed limit (30km/h)

The model was able to correctly guess 3 of the 5 traffic signs, which gives an accuracy of 60%. I selected 3 images with high contrast and good resolution, 2 with not so great contrast and somehow pixelated. This result was expected.

3. Top 5 softmax probabilities for each image along with the sign type of each probability.

For the first image, the model is positively sure that this is a Double curve sign (probability of 0.9), but the image contains a Children crossing sign. I thought there were high chances that the prediction for this image was going to be wrong since the image inside the triangle is a bit blurry and doesn't contrast much with the background of the sign. It also appears that the dark background influenced the prediction.

The top five softmax probabilities were:

Probability	Prediction
.90	Double curve
.08	Right-of-way at the next intersection
.01	Children crossing
.01	Beware of ice/snow
.0	Slippery road

For the second image, the model is sure that this is a Beware of ice/snow sign (probability of 1), but the image does contain a Road work sign. I think this prediction was wrong since the image it's really pixelated, it doesn't seem to be able to recognize what's kind of shape is inside the triangle. The top five soft max probabilities were:

Probability	Prediction
1	Beware of ice/snow
.0	Road narrows on the right
.0	Road work
.0	Bumpy Road
.0	Slippery Road

For the third image, the model is sure that this is a General caution sign (probability of 1), and the image does contain a General caution sign. The Contrast of the image is clear so it was clear for the prediction. The top five soft max probabilities were:

Probability	Prediction
1	General caution
.0	Right-of-way at the next intersection
.0	Pedestrians
.0	Traffic signals
.0	Slippery Road

For the fourth image, the model is sure that this is a Priority road sign (probability of 1), and the image does

contain a Priority road sign. This sign it's really different from all the other in the data. so the prediction shouldn't have been a problem. The top five soft max probabilities were:

Probability	Prediction
1	Priority road
.0	Turn left ahead
.0	Roundabout mandatory
.0	No entry
.0	No vehicles

For the fifth image, the model is sure that this is a Speed limit (30km/h) sign (probability of 1), and the image does contain a Speed limit (30km/h) sign. The contrast is really good for this image, the model had no issue to classify it correctly. The top five soft max probabilities were:

Probability	Prediction
1	Speed limit (30km/h)
.0	Speed limit (80km/h)
.0	Speed limit (50km/h)
.0	Speed limit (70km/h)
.0	Speed limit (100km/h)