Traffic Sign Recognition

Build a Traffic Sign Recognition Project

The goals / steps of this project are the following:

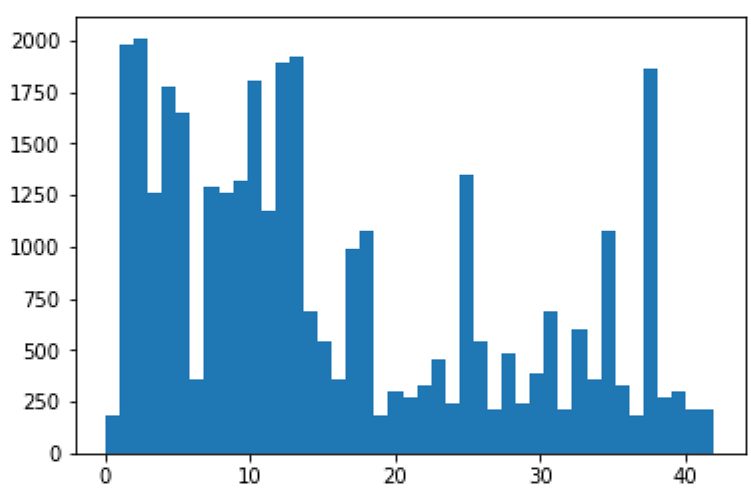
- Load the data set (see below for links to the project data set)
- Explore, summarize and visualize the data set
- Design, train and test a model architecture
- Use the model to make predictions on new images
- Analyze the softmax probabilities of the new images
- Summarize the results with a written report

Data Set Summary & Exploration

I used the python 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

Histogram of the DataSet:

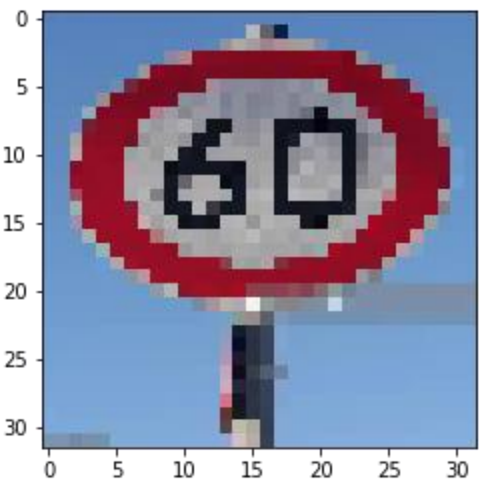


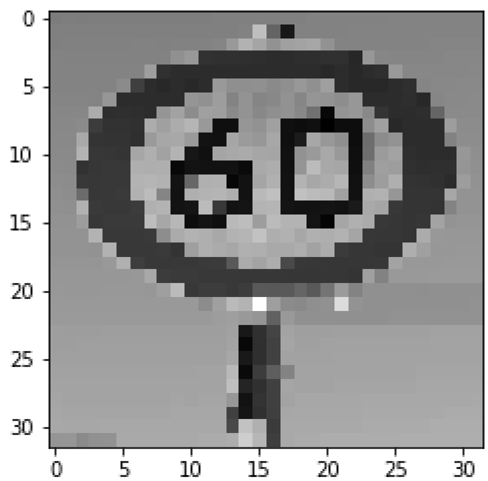
Design and Test a Model Architecture

Q: Describe how you preprocessed the image data. What techniques were chosen and why did you choose these techniques?

As a first step, I decided to convert the images to grayscale because it has been observed that color channel in an image are not of great use, so converting it to grayscale will reduce the complexity of the processing.

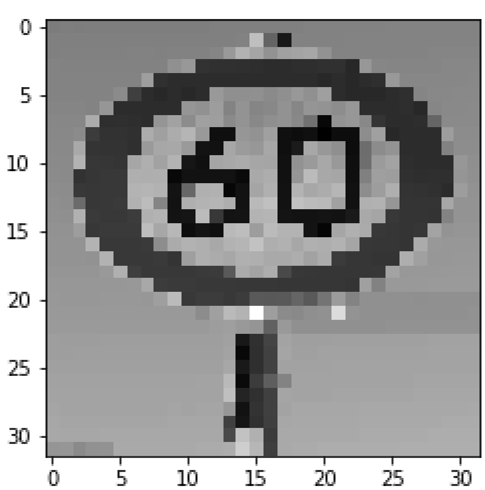
Here is an example of a traffic sign image before and after grayscaling.

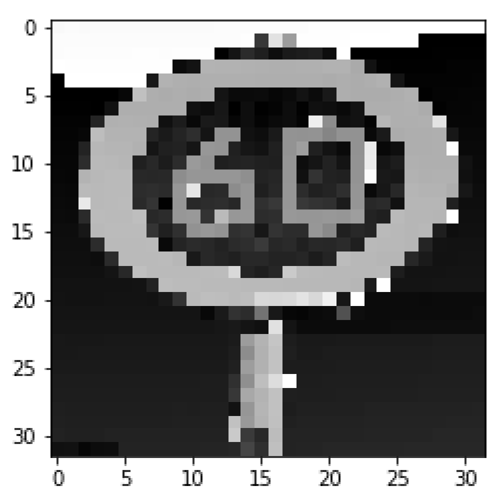




As a last step, I normalized the image data because in the process of training our network, we're going to be multiplying (weights) and adding to (biases) these initial inputs in order to cause activations that we then back propagate with the gradients to train the model. We'd like in this process for each feature to have a similar range so that our gradients don't go out of control (and that we only need one global learning rate multiplier).

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





Q. 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	1x1 stride, 'VALID' padding, outputs 28x28x6
RELU	
Max pooling	2x2 stride, 'VALID' padding ,outputs 14x14x6
Convolution 5x5	1x1 stride, 'VALID' padding, outputs 10x10x16
RELU	
Max pooling	2x2 stride, 'VALID' padding ,outputs 5x5x16
Flatten	Input = 5x5x16. Output = 400.
Fully Connected	Input = 400, Output = 120
RELU	
Fully Connected	Input = 120. Output = 84
RELU	
Fully Connected	Input = 84. Output = 43.

Q: 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 LeNet architecture and for optimizing I used ADAM Optimizer. The batch size was 128 along with the number of epochs as 30 with the learning rate of 0.0009.

Q: 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:

validation set accuracy of 0.95

test set accuracy of 0.936

LeNet Architecture was chosen.

Q: Why did you believe it would be relevant to the traffic sign application?

The LeNet architecture is an excellent "first architecture" for Convolutional Neural Networks. It is small and easy to understand — yet large enough to provide interesting results.It was also discussed and implemented in Udacity classroom videos,so it provided me a better option to move on.

Q: How does the final model's accuracy on the training, validation and test set provide evidence that the model is working well?

With the validation accuracy of 96.5% and testing validation of 93.6% it is proved that the model is working well.

Test a model on new Images

Q: 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 images might be difficult to classify because it was not of size 32*32 and it was taken from web and are from real world images/videos so they are not clean.

Q: 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
30 km/h	30 km/h
60 km/h	No entry
Road Work	Road Work
Stop	Stop
Children crossing	Right-of-way at the next intersection

The model was able to correctly guess 3 of the 5 traffic signs, which gives an accuracy of 60%.

Q: 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, 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 soft max probabilities were

image 1

```
Speed limit (30km/h)                      0.923017
Speed limit (50km/h)                      0.074243
Wild animals crossing                     0.00273986
Speed limit (60km/h)                      1.44386e-07
Right-of-way at the next intersection     6.60928e-08
```

image 2

```
No entry                                  0.993648
No passing                                0.00449681
Roundabout mandatory                      0.00184728
Ahead only                                5.57253e-06
Go straight or left                       2.30837e-06
```

image 3

```
Road work                                 1.0
Beware of ice/snow                        7.28722e-10
Traffic signals                           2.11157e-11
Bumpy road                                2.4872e-14
Turn right ahead                          1.15276e-14
```

image 4

```
Stop                                      0.996109
Speed limit (30km/h)                      0.00131432
Keep right                                0.000979941
Yield                                     0.000769689
Turn left ahead                           0.000361594
```

image 5

```
Right-of-way at the next intersection     0.999873
Pedestrians                               0.00010406
Dangerous curve to the right              2.16564e-05
General caution                           1.00799e-06
Slippery road                             7.58725e-08
```