Traffic Sign Recognition Project

Submitted by:- Krishna Trivedi (kvtrivedi93@gmail.com)

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

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! Please find the ipynb file to my project in the submitted zip folder.

Data Set Summary & Exploration

1. Basic summary of the data set.

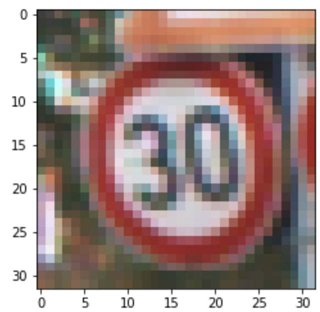
I used the numpy library to calculate summary statistics of the traffic
signs data set:
* The size of training set is 34799
* The size of the validation set is 4410
* The size of test set is 12630
* The shape of a traffic sign image is (32, 32, 3)
* The number of unique classes/labels in the data set is 43

2. An exploratory visualization of the dataset.

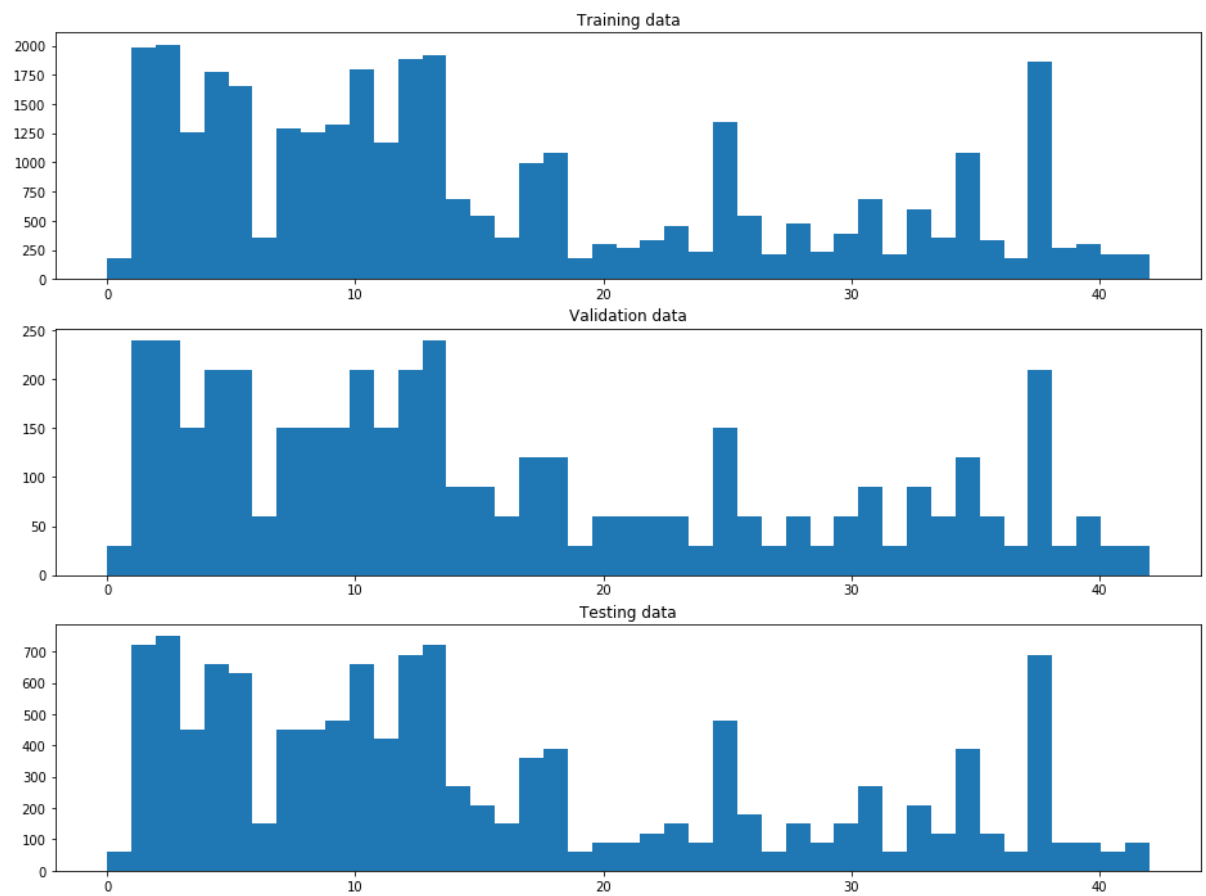
I mapped the sign names in the given csv file to any random image to visualize it correctness.

```
Speed limit (30km/h)

<matplotlib.image.AxesImage at 0x1e237c95e10>
```



Also, I have plotted bar graphs showing the distribution of the training, validation and testing data. This shows that the traffic signs are distributed similarly in all the three datasets.



Design and Test a Model Architecture

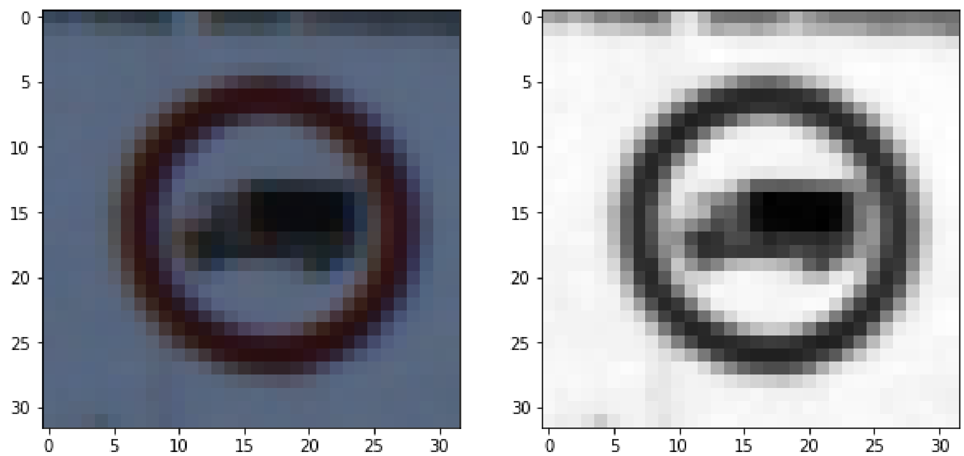
1. Describe how you preprocessed the image data. What techniques were chosen and why did you choose these techniques? Consider including images showing the output of each preprocessing

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.)

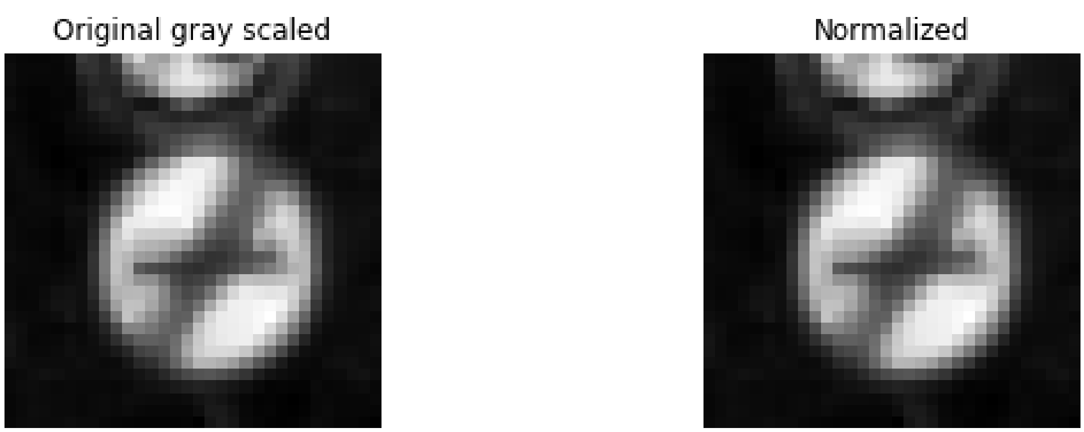
I have used the following methods to pre-process the data:

1) Coverting the image to grayscale
2) Normalizing the image
3) Shuffling the training set

I converted the data to gray scale because training the network was easier without the information of color. Also, even without the knowledge of color, the classification task remains unaffected since the grayscale data is sufficient for the task. Also, we can still achieve a very high accuracy.
Below is the result of gray-scaling an image,



I normalized the data so that its mean is close to zero. This helps because when the data has a wide distribution, training using a single learning rate becomes difficult. Below is the result of normalizing the gray scaled image.



Shuffling the training data helps avoid any bias being created in the model.

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 model was similar to the LeNet architecture.

My final model consisted of the following layers:

1. Input : 32x32x1
2. 5x5 Convolutional : 28x28x6
3. RELU
4. Pooling : 14x14x6
5. 5x5 Convolutional : 10x10x6
6. RELU
7. Pooling : 5x5x16
8. Fully Connected : 120
9. RELU
10. Dropout
11. Fully Connected : 84
12. RELU
13. Fully Connected/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.

Model was trained with the following characteristics:

1. Optimizer = Adam (similar to the LeNet lab)
2. Batch size = 100
3. Epochs = 60
4. Mu = 0
5. Sigma = 0.1
6. Learning rate = 0.0009
7. Keep probability for 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.

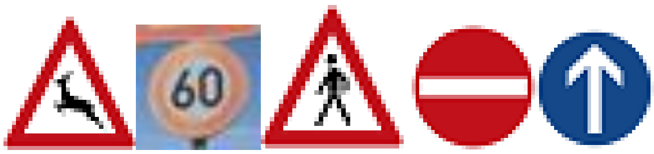
My final model results were:
Validation Accuracy = 99%
Test Accuracy = 93.4%

I chose to work on the LeNet Architecture with slight modifications in it. A high accuracy on the training set but low accuracy on the validation set implies overfitting. Hence, I applied a dropout to reduce over-fitting. With dropout being added and a number of trials of selecting an appropriate batch size, epochs and learning rate, my accuracy was improved from 90.2% to 93.4%. I believe that the accuracy can further be improved using other regularization techniques such as L2 regularization.

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:



These signs were also converted to gray scale and normalized and then tested.

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).

The model was able to correctly guess 5 of the 5 traffic signs, which gives an accuracy of 100%. This is better than the accuracy on the test set of 93.4%.

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)

Please find code for finding the softmax probabilities for each prediction in cell 44.