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

[//]: # (Image References)

[image1]: ./examples/visualization.jpg "Visualization"
[image2]: ./examples/grayscale.jpg "Grayscaling"
[image3]: ./examples/random_noise.jpg "Random Noise"
[image4]: ./examples/placeholder.png "Traffic Sign 1"
[image5]: ./examples/placeholder.png "Traffic Sign 2"
[image6]: ./examples/placeholder.png "Traffic Sign 3"
[image7]: ./examples/placeholder.png "Traffic Sign 4"
[image8]: ./examples/placeholder.png "Traffic Sign 5"

Rubric Points
Here I will consider the [rubric points](https://review.udacity.com/#!/rubrics/481/view) individually and describe how I addressed each point in my implementation.

Writeup / README

1. 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! and here is a link to my [project code](https://github.com/udacity/CarND-Traffic-Sign-Classifier-Project/blob/master/Traffic_Sign_Classifier.ipynb)

Data Set Summary & Exploration

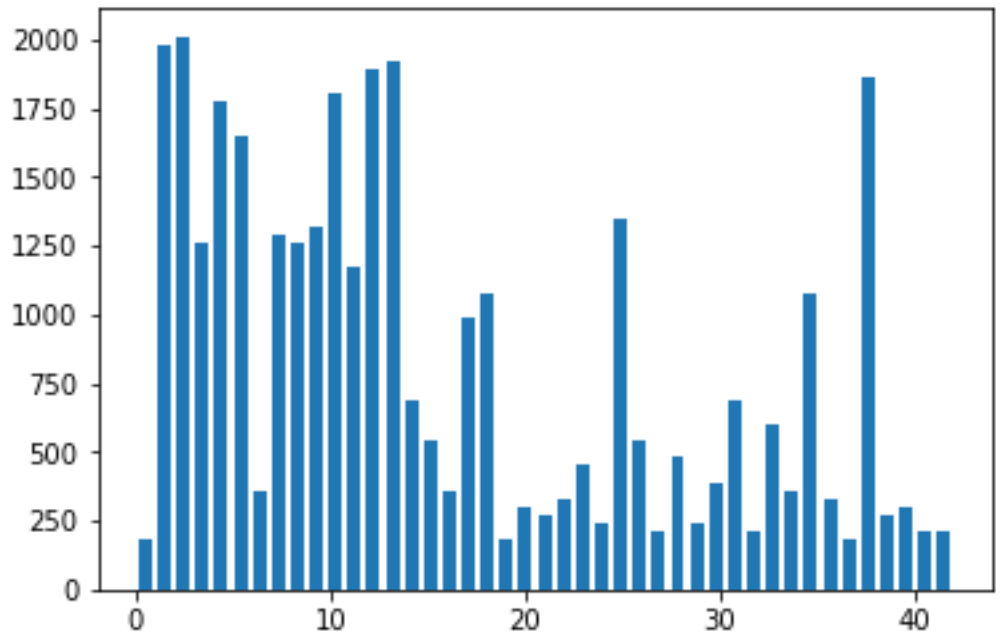
1. Provide a basic summary of the data set. In the code, the analysis should be done using python, numpy and/or pandas methods rather than hardcoding results manually.

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

Number of training examples = 34799
Number of testing examples = 12630
Image data shape = (32, 32, 3)
Number of classes = 43

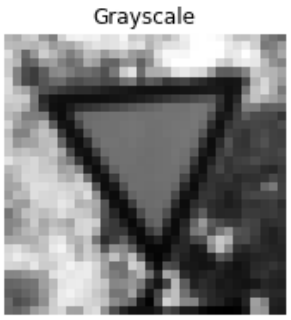
2. Include an exploratory visualization of the dataset.

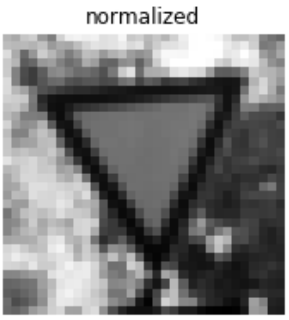
Here is an exploratory visualization of the data set. It is a bar chart showing how the data ...



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 convert the images to grayscale because it reduces time as the image is monotonous rather than having 3 layers which consumes time.

As a last step, I normalized the image data because wider distribution of data makes it more difficult to train by single learning rate. Different features have different ranges weight can be diverged by single learning rate

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:

Input	32x32x1 grayscale image
Convolution 5x5	2x2 stride, valid padding, outputs 28x28x6
RELU	
Max pooling	2x2 stride, outputs 14x14x6
Convolution 5x5	2x2 stride, valid padding, outputs 10x10x16
RELU	

Max pooling	2x2 stride, outputs 5x5x16
Convolution 1x1	2x2 stride, valid padding, outputs 1x1x412
RELU	
Fully connected	input 412, output 122
RELU	
Dropout	50% keep
Fully connected	input 122, output 84
RELU	
Dropout	50% keep
Fully connected	input 84, 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 an

I used The LeNet Architecture and modified it according to my requirements.

Learning rate it kept 0.009 with AdamOptimizer.

Epochs = 30

Batch size = 156

I played around with epochs and batch size and learning rate. Before deciding to keep it this way. My test accuracy is around 93. But the architecture was able to predict all the real world image quite well

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 99.9%
- validation set accuracy of 99.0%
- test set accuracy of 93.6%

If an iterative approach was chosen:

- What was the first architecture that was tried and why was it chosen?

  It would be similar architecture to the paper offered us by the mentors. It has given me such a good score

- What were some problems with the initial architecture?

  Lack of data and lack of knowledge of other parameters. Adding a convolution layer help me reaching higher accuracy

- How was the architecture adjusted and why was it adjusted?

  It was adjusted by adding a convolution layers and aiding couple dropouts with a 50% probability as the previous architecture was not able to achieve it

- Which parameters were tuned? How were they adjusted and why?
  - Epoch: I played around with this and set it to 30 as I was able to achieve good probability.
  - The batch size I increased it as learned we can increase the data size if memory allows it .
  - The learning rate I think could have been left at .001 as it was default good rate by I tried changing it as it mattered a little
  - The dropout probability mattered a lot early on, but after a while I set it to 50% and just left it. The biggest thing that effected my accuracy was the data images generated with random modifications.
- What are some of the important design choices and why were they chosen?

  It have provide me with such learning opportunity that I could take weeks more to learn. I think this is a good question and I could still learn more about that. I think the most important thing I learned was having a more uniform dataset along with enough convolutions to capture features will greatly improve speed of training and 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.

These were semi easy images to classify. I didn’t find difficulty

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 8 of the 8 traffic signs, which gives an accuracy of 100%. This compares favorably to the accuracy on the test set of 93.6%

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 probabilities are given above the image


input
top guess: 18 (100%)
2nd guess: 26 (0%)
3rd guess: 0 (0%)
3rd guess: 1 (0%)
3rd guess: 2 (0%)
input
top guess: 25 (96%)
2nd guess: 20 (3%)
3rd guess: 23 (0%)
3rd guess: 11 (0%)
3rd guess: 30 (0%)
input
top guess: 3 (100%)
2nd guess: 25 (0%)
3rd guess: 23 (0%)
3rd guess: 36 (0%)
3rd guess: 20 (0%)
input
top guess: 38 (100%)
2nd guess: 0 (0%)
3rd guess: 1 (0%)
3rd guess: 2 (0%)
3rd guess: 3 (0%)
input
top guess: 11 (100%)
2nd guess: 30 (0%)
3rd guess: 27 (0%)
3rd guess: 21 (0%)
3rd guess: 16 (0%)
input
top guess: 34 (100%)
2nd guess: 38 (0%)
3rd guess: 36 (0%)
3rd guess: 3 (0%)
3rd guess: 11 (0%)
input
top guess: 1 (100%)
2nd guess: 2 (0%)
3rd guess: 4 (0%)
3rd guess: 0 (0%)
3rd guess: 5 (0%)
input
top guess: 12 (100%)
2nd guess: 40 (0%)
3rd guess: 7 (0%)
3rd guess: 13 (0%)
3rd guess: 42 (0%)