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

Rubric Points

Here I will consider the rubric points 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

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 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 histogram showing how the train, test and validation data sets distribute on the same scale.

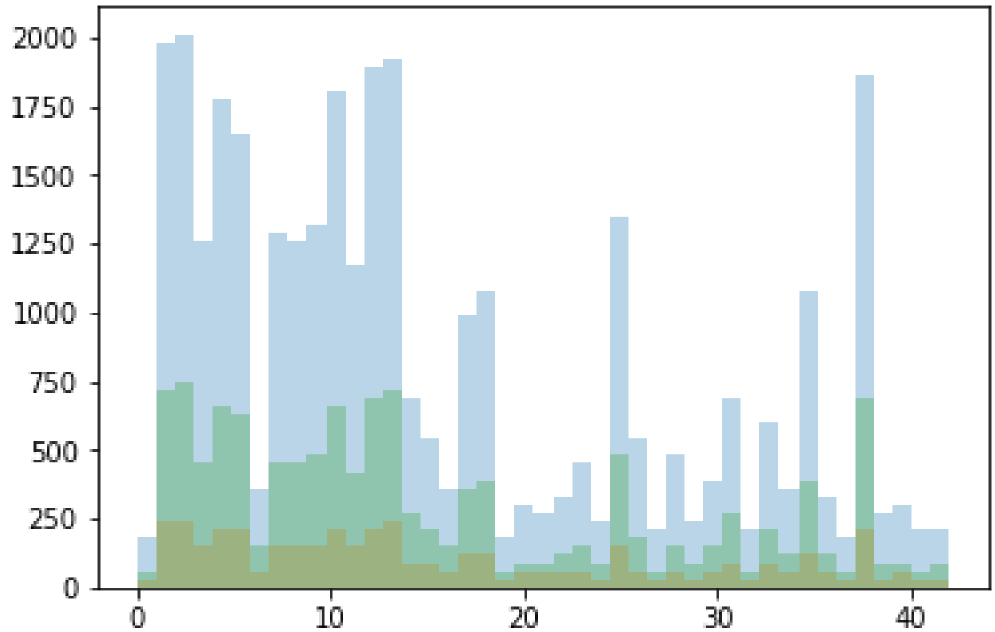

Figure 1: Histogram of train, test and validation data sets

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 for this project, grayscale images will provide as much information as the three channels color images for the task. And less parameters for the computing.

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

As a last step, I normalized the image data because neural network normally uses gradient descent for updating the weights and biases, and its denpending on the values of the data. By normalizing the data set, all the points will have a fair effect on the updating. Not to be radically changed by the extram values.

I decided to generate additional data because after ploting the histogram, some classes can be seem have less samples relative to other classes. These classes imblance will affect the training process to be more biased toward the data rich classes, in other words overfitting the classes which show up more often in the trainning process. This impede the model performance and general robust.

To add more data to the the data set, the classes with less than 300 samples were horizontally flipped.

Here is an example of an augmented image:

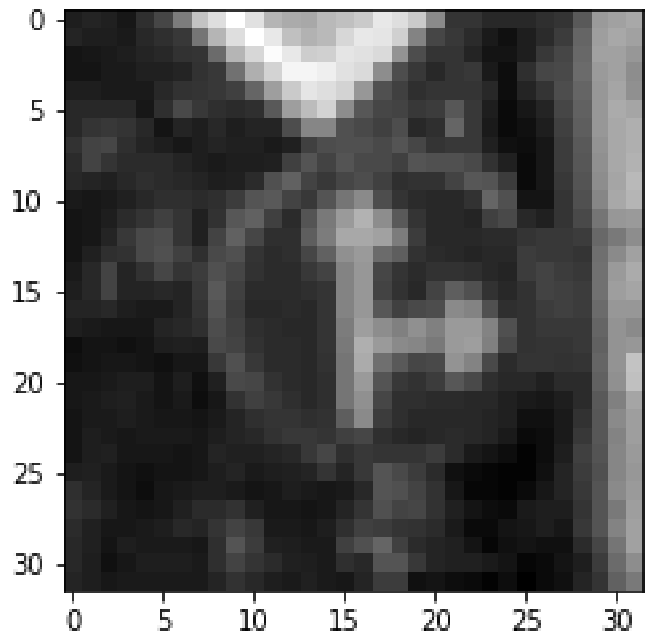

Figure 2: Grayscale converted image

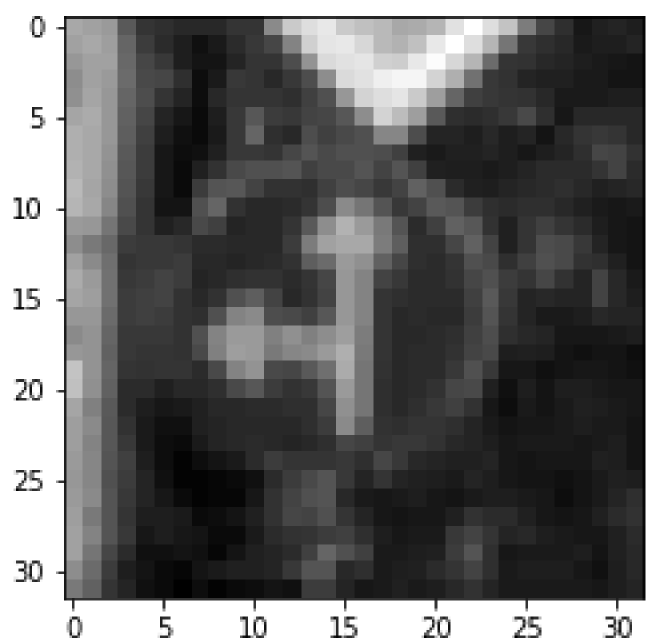

Figure 3: Grayscale horizontal flip

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	1x1 stride, VALID padding, outputs 28x28x6
RELU	
Max pooling	2x2 stride, outputs 16x16x6
Drop out	keep prob 0.9

Convolution 5x5 | 1x1 stride, VALID padding, outputs 10x10x16 |
RELU | |
Max pooling | 2x2 stride, outputs 5x5x16 |
Drop out | keep prob 0.9 |

Fully connected | input 400 output 240 |
RELU |
Drop out | keep prob 0.9 |

Fully connected | input 240 output 160 |
RELU |
Drop out | keep prob 0.9 |

Fully connected | input 160 output 43 |
Softmax | etc. |
| |
| |

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 Adam optimizer, bacth size 128, 15 epoches, learning rate 0.001.

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: * train set accuracy of 0.994 * validation set accuracy of 0.952 * test set accuracy of 0.926

If an iterative approach was chosen: * What was the first architecture that was tried and why was it chosen? The LeNet architecture was chose, easy to plug in and play. Many years experiences proved to be a good model to start for the classification task.

- What were some problems with the initial architecture? The initial architecture reached the validation limit at 0.89. It might be that the shallow and thin structure can not handle as many classes as 43 in the current task, instead of only 10 classes in the digits classification task, which it is working well.

- How was the architecture adjusted and why was it adjusted? Typical adjustments could include choosing a different model architecture, adding or taking away layers (pooling, dropout, convolution, etc), using an activation function or changing the activation function. One common justification for adjusting an architecture would be due to overfitting or underfitting. A high accuracy on the training set but low accuracy on the validation set indicates over fitting; a low accuracy on both sets indicates under fitting. The fully connected lays were added more layers to include more parameters. Drop out at 0.9 was added at the end of each layer.
- Which parameters were tuned? How were they adjusted and why? Different normalization, drop out rate, position of drop out layer. Normalization will change the scale of the input. Drop out will tune the robust.
- What are some of the important design choices and why were they chosen? For example, why might a convolution layer work well with this problem? How might a dropout layer help with creating a successful model? Add drop out in each layer, not to relay heavely in certain layers for the robust. Better performance with more dropout at a higher rate than a single dropout at 0.5.

If a well known architecture was chosen: * What architecture was chosen? LeNet for its performance and easy to plug in and play. * Why did you believe it would be relevant to the traffic sign application? A good start point for classification task. * How does the final model's accuracy on the training, validation and test set provide evidence that the model is working well? The train set accuray is higher than the validation accuracy, shown that the model is overfitting the train set. Overall, the model is working well on test set.

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:

The last image might be difficult to classify because after resizing the worker on the sign is deformed.

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
Speed limit (70km/h)	Speed limit (70km/h)
Priority road	Priority road
Stop	Stop
No entry	No entry
Road work	Road work

They are all correctly classified. The accuracy is 100%, which is comparable to the test set. We have only 5 images.

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

Figure 4: 4, Speed limit (70km/h)

"Stand Out Suggestions" part of the rubric, visualizations can also be provided such as bar charts)

The code for making predictions on my final model is located in the 164 cell of the Ipython notebook.

All the probabilities of prediction is higher than 99%. When its right, its good.. Should include some negative samples to see the prediction certainties.

Figure 5: 12, Priority road

Figure 6: 14, Stop

Figure 7: 17, No entry

Figure 8: 25, Road work