Traffic Sign Recognition

Writeup

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
- Analyse the SoftMax probabilities of the new images
- Summarize the results with a written report

Rubric Points

Submission Files

This project includes

- The notebook Traffic_Sign_Classifier.ipynb
- signames.csv is also uploaded.
- Traffic_Sign_Classifier.html, the exported HTML version of the python notebook
- A directory examples containing images found on the web
- Traffic_Sign_Classifier_Writeup.Pdf

Data Set Summary & Exploration

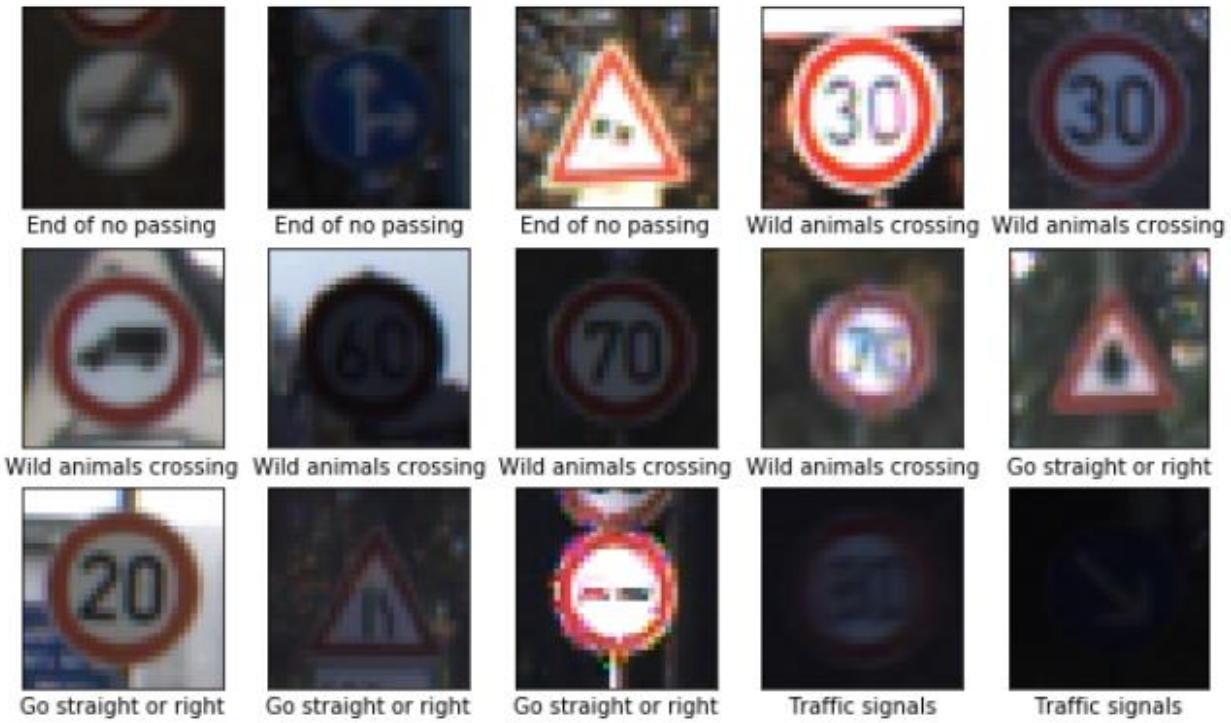
1. Provide a basic summary of the data set and identify where in your code the summary was done. In the code, the analysis should be done using python, numpy and/or pandas methods rather than hardcoding results manually.

- Number of training examples = 34799

- Number of valid examples= 4410
- Number of testing examples = 12630
- Image data shape = (32, 32, 3)
- Number of classes = 43

2. Include an exploratory visualization of the dataset and identify where the code is in your code file.

See the result in cell 9 inside notebook.



Design and Test a Model Architecture

1. Describe how, and identify where in your code, you preprocessed the image data. What tecniques 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.

- The code processing images is in cell 15.
- Converted Each image in each set to gray image version with the size (32,32), using the defined function rgb2gray.
- Normalized the Each gray image using normalize_grayscale_image function.
- Then reshaped each image into size (32,32,1), to make it compatible to the Network usage

2. Describe how, and identify where in your code, you set up training, validation and testing data. How much data was in each set? Explain what techniques were used to split the data into these sets. (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, identify where in your code, and provide example images of the additional data)

The train, valid and test data are loaded in the cell 1 from the respective **p files** available in Udacity Workspaces.

- Number of training examples = 34799
- Number of valid examples= 4410
- Number of testing examples = 12630

These data are preproceed using the above discussed processing methods from point no:2

3. Describe, and identify where in your code, 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.

The code is in function LeNet in code cell 27
I adapted LeNet architecture: Two convolutional layers followed by one flatten layer, and three fully connected linear layers.

1. convolution 1: 32x32x1 -> 28x28x32-> relu -> 14x14x32 (pooling)
2. convolution 2: 14x14x32 -> 10x10x64 -> relu -> 5x5x64 (pooling)
3. Flatten 5x5x64 → 1600
4. Linear 1600 → 1200
5. Linear 1200 → 1024
6. Linear 1024 → 43

4. Describe how, and identify where in your code, 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.

The code for training the model is in cell 36

I train the model in 20 iterations (epochs), and each iteration is trained with 128 batch size. Adam optimizer is used with learning rate 0.001.

5. Describe the approach taken for finding a solution. 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.

The code for calculating the accuracy of the model is in cell 31 and 35

My final model results were:

- `Validation Accuracy = 0.963`
- `Test Accuracy = 0.941`

The first model is adapted from LeNet architecture. Since LeNet architecture has a great performance, as suggested tried that it would also work on classifying traffic signs.

Initially I got accuracy problems due to wrong definitions of the pooling layers, later after some tweaks I got a stable accuracy which was more supported later by below hyperparameters to reach current accuracy rate.

- epoch 20
- batch_size 64
- learning rate 0.001

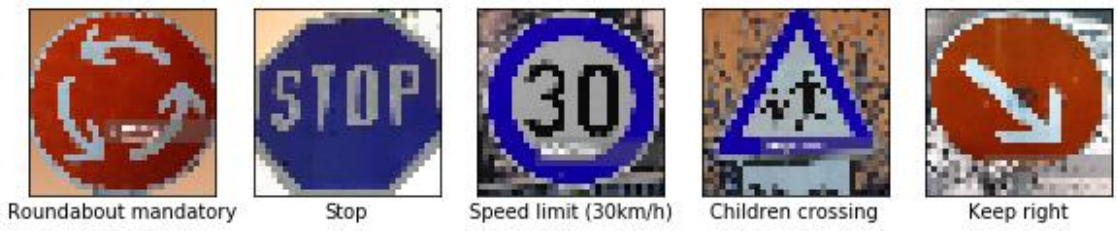
The final accuracy in validation set is around 0.95.

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.

The chosen signs are visualized in cell 36.

Downloaded some different data unseen from the web an loaded in examples folder present already in the workspaces.



I have used these images and preprocessed them to test them on the LeNet built.

I got an accuracy rate of 0.6 on these images.

Which means 2 out of 5 images are wrong which can be improved with more training and tuning.

2. Discuss the model's predictions on these new traffic signs and compare the results to predicting on the test set. Identify where in your code predictions were made. 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 code for making predictions on my final model is in cell 40.

The test accuracy is much better than the new unseen images,

This is valid as due to the test images are all set at one frequency related to the color formation ad information on them.

But the new data is varied much in the color contrast etc which LeNet trained model is not aware about.

This can be taken as a point of action to improvise and study the network further.

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 and identify where in your code softmax probabilities were outputted. 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)

In the submitted version, the model can correctly guess 3 out of 2 signs. The accuracy is 60%. However, it can be tuned further.