Traffic Sign Classifier

Introduction:

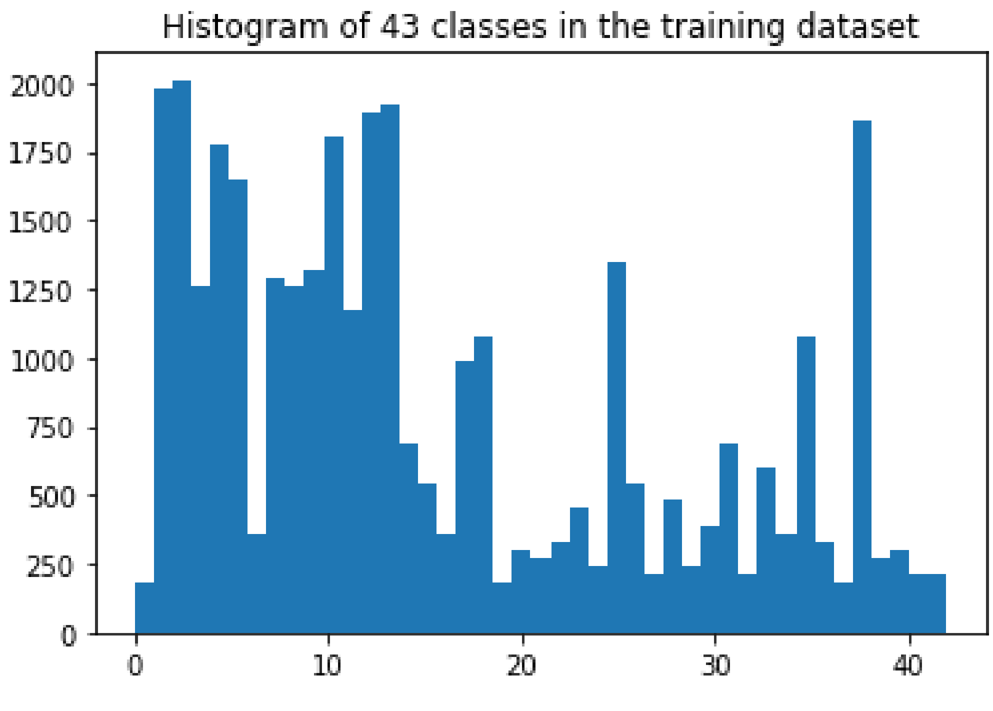
In this document I wrote the instruction for the 3rd project.
The goals / steps of this project are the following:

- 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 and exploration:

- The size of training set is 34799
- The size of the validation set is 12630
- The size of test set is 4410
- The shape of a traffic sign image is 32x32x3
- The number of unique classes/labels in the data set is 43

Traning / Validation classes histogram



The classes are not equally distributed but we can use the training to teach our model.

Design and Test a Model Architecture

In the traffic signal the shapes are more important than the colors, for this reason I convert the images to grayscale with cv2.

Model Architecture

Layer	Description	Output
Input	32x32x1 gray-scale	
Convolution 5x5	1x1 stride, valid padding	28x28x20
RELU		
Max pooling	2x2 stride	14x14x20
Convolution 3x3	1x1 stride	12x12x40
RELU		
Convolution 3x3	1x1 stride	10x10x80
RELU		
Max pooling	2x2 stride	5x5x80
Flatten		2000
Fully Connected 84x43		43
Softmax		

Model training

Epochs = 20
Batch Size = 128
Optimization= Adam

The model is so, 3 convolutions, 3 ReLU activation functions and 2 dropouts.
I used Adam optimizer to train, with learning rate as 0.01, batch size= 128 and 20 epochs.
For the hyper-parameter, I used a mean of 0 and standard deviation of 0,1.
Adam is an adaptive learning rate optimization algorithm that's been designed specifically for training deep neural networks. I used Adam because it works very good for me and because Adam works well in practice and compares favorably to other stochastic optimization methods.

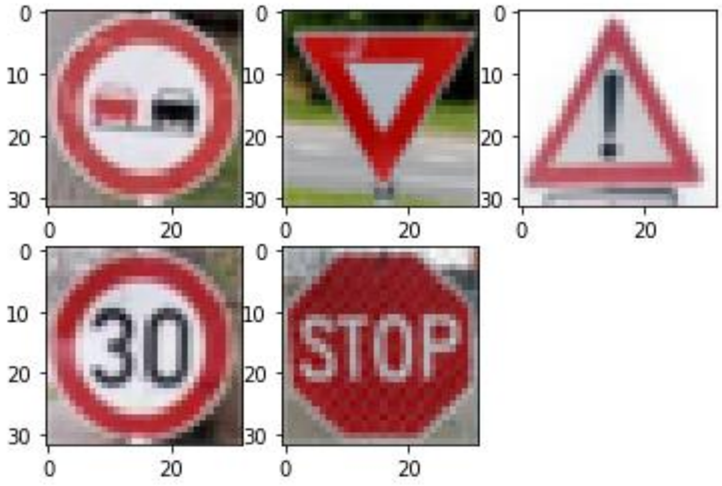
My final model results were:

- Validation set accuracy = 94,4
- Training set accuracy = 99,8
- Test set accuracy = 99,8

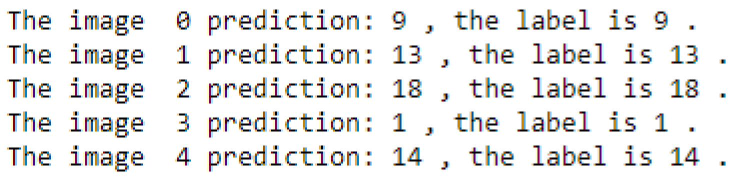
I chose learning rate of 0.001 because the loss function decreases when the learning rate is 0.001.

Test a Model on New Images

In order to test my model, I choice 5 images, German Traffic signs found on the web.



The prediction works well, though the images have the different background.

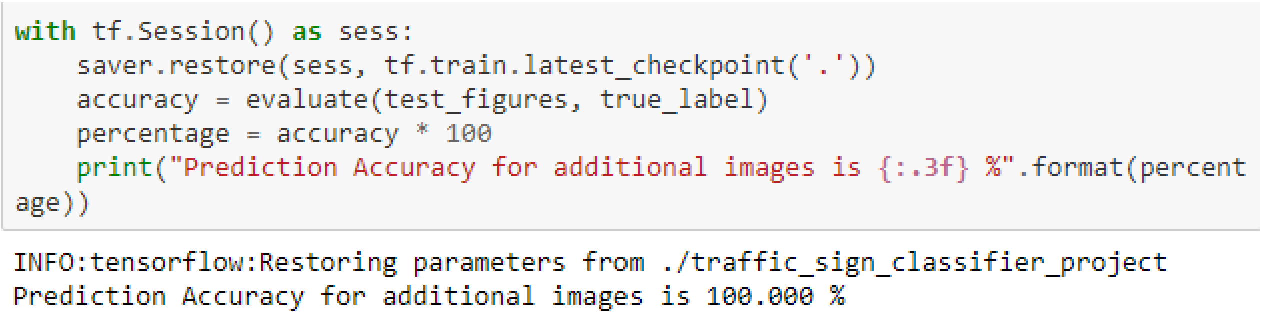


```
The image  0 prediction: 9 , the label is 9 .
The image  1 prediction: 13 , the label is 13 .
The image  2 prediction: 18 , the label is 18 .
The image  3 prediction: 1 , the label is 1 .
The image  4 prediction: 14 , the label is 14 .
```

Here are the result of the prediction:

Image	***Prediction***
NO Passing	NO Passing
Yield	Yield
Caution	Caution
Speed 30	Speed 30
Stop	Stop

The model was able to correctly guess 5 of the 5 traffic signs, which gives an accuracy of 100%



```
with tf.Session() as sess:
    saver.restore(sess, tf.train.latest_checkpoint('.'))
    accuracy = evaluate(test_figures, true_label)
    percentage = accuracy * 100
    print("Prediction Accuracy for additional images is {:.3f} %".format(percent
age))
```

```
INFO:tensorflow:Restoring parameters from ./traffic_sign_classifier_project
Prediction Accuracy for additional images is 100.000 %
```