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). Apply a distortion correction to raw images.
- 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. Here is a template writeup for this project you can use as a guide and a starting point.

You're reading it!

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
I used panda to create the below table showing 43 unique traffic signs that I am going to classify. The class ID for each sign name is shown in this table.

```
       ClassId                          SignName

0          0                               Speed limit (20km/h)
1          1                               Speed limit (30km/h)
2          2                               Speed limit (50km/h)
3          3                               Speed limit (60km/h)
4          4                               Speed limit (70km/h)
5          5                               Speed limit (80km/h)
6          6                        End of speed limit (80km/h)
7          7                              Speed limit (100km/h)
8          8                              Speed limit (120km/h)
9          9                                         No passing
10        10       No passing for vehicles over 3.5 metric tons
11        11              Right-of-way at the next intersection
12        12                                      Priority road
13        13                                              Yield
14        14                                               Stop
15        15                                        No vehicles
16        16           Vehicles over 3.5 metric tons prohibited
17        17                                           No entry
18        18                                    General caution
19        19                        Dangerous curve to the left
20        20                       Dangerous curve to the right
21        21                                       Double curve
22        22                                         Bumpy road
23        23                                      Slippery road
24        24                          Road narrows on the right
25        25                                          Road work
26        26                                    Traffic signals
27        27                                        Pedestrians
28        28                                  Children crossing
29        29                                  Bicycles crossing
30        30                                 Beware of ice/snow
31        31                              Wild animals crossing
32        32                End of all speed and passing limits
33        33                                   Turn right ahead
34        34                                    Turn left ahead
35        35                                         Ahead only
36        36                               Go straight or right
37        37                                Go straight or left
38        38                                         Keep right
39        39                                          Keep left
40        40                               Roundabout mandatory
41        41                                  End of no passing
42        42  End of no passing by vehicles over 3.5 metric ...
```

The blow picture also shows some samples of the traffic signs.

0-Speed limit (20km/h)

1-Speed limit (30km/h)

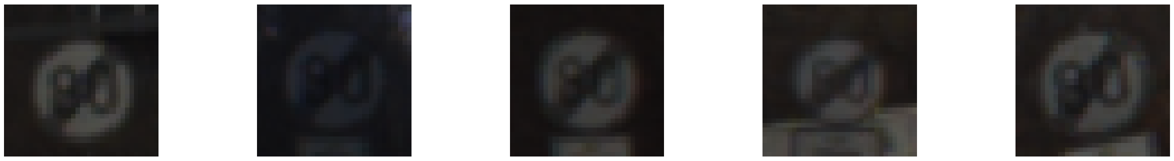
2-Speed limit (50km/h)

3-Speed limit (60km/h)

4-Speed limit (70km/h)

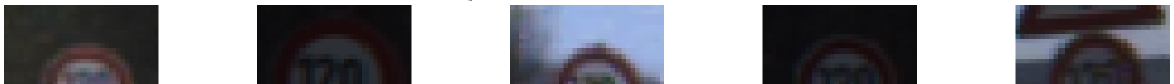
5-Speed limit (80km/h)

6-End of speed limit (80km/h)

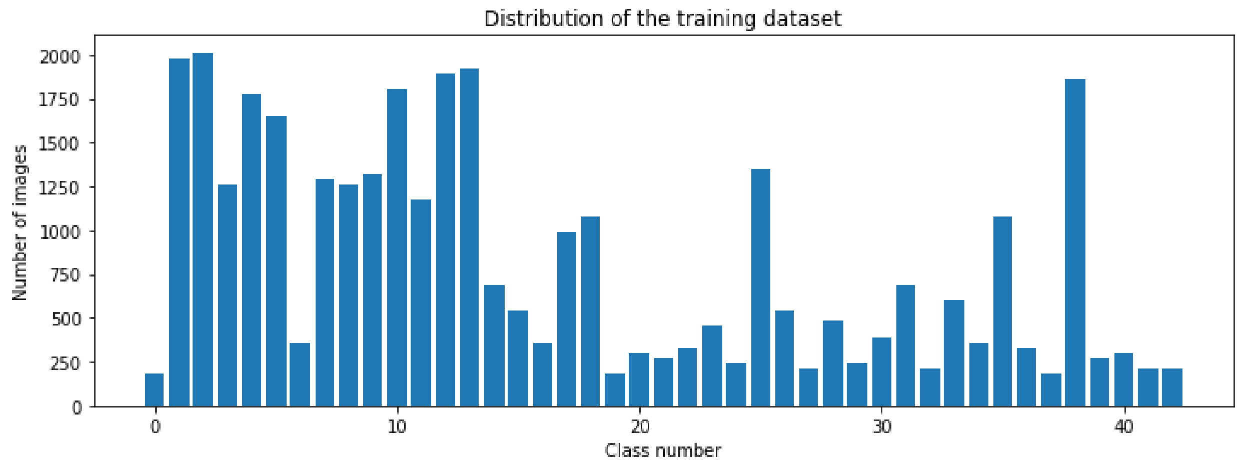


7-Speed limit (100km/h)

8-Speed limit (120km/h)



I also created a diagram showing how the training dataset is distributed. We can see that the distribution of the number of images for each case number is not equal, with the lowest of 180 for the class number 0 and the higheste of 2010 for the class number 2.



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.)

Preprocessing the images was one of the most important step in making sure that model will train sucessufully. The original images are in color format and I did not think the color information will be necessary to feed to the model. To simplify the image inputs and decreasing the channels from 3 to 1, I converted the images to grayscale format. The image below shows the original versus grayscale format for one of the images.

I also perfomed normalization on images by subtracting the image pixle values from the mean value and dividing by the standard deviation. This way the mean value will be zero. Normalizing is a well known technique in deep learning and would help to improve the training process and accuracy.

img = (img - img.mean())/np.std(img)

These steps changed the shape of the input images to (32,32,1). I also decided to shuffle all the data using shuffle from sklearn.utils library. This will make sure that the inputs to the model are in random order.

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.

I choose LeNet architecture for classification initially and then modified it by adding dropouts and L2 regulization to improve the validation accuracy. 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 14x14x6
Convolution 5x5	1x1 stride, valid padding, outputs 10x10x16
RELU	
Max pooling	2x2 stride, outputs 5x5x16

Flatten	400 outputs
Dropout	
Fully connected	120 Outputs
RELU	
Dropout	
Fully connected	84 Outputs
RELU	
Dropout	
Fully connected	43 Outputs
Softmax	
L2 Regulization	

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.

The final parameters after several trial and error are:
Epoch: 100
Batch Size: 200
Learning rate: 0.001
Drop out: 0.75
Beta = 0.01 (for L2 regulization)
Optimizer: AdamOptimizer

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 (the results are on cell 34 of my ipython notebook) results were:
* training set accuracy of 99.99%
* validation set accuracy of 96.64%
* test set accuracy of 95.53%

I initially chose LeNet architecture for classification and only by using this model, I was able to achieve the validation accuracy of 93.4%. I used these parameters: epoch 50, batch size100 and learning rate of 0.001.

To improve the validation accuracy and also avoid overfitting, I decided to use both dropout and L2 regulization techniques. Both these techniques have been explained in the course content. As you can see from my final model above, I added dropout to fully connected layers and L2 reg term to the end. For the dropout, I tried many values for the "keep_prob" parameter ranging from 0.5 to 0.8, but the best value I chose was 0.75 which could give the validation accuracy of 96.64%. The beta value for the L2 regularization was set at 0.01 and I did not change it.

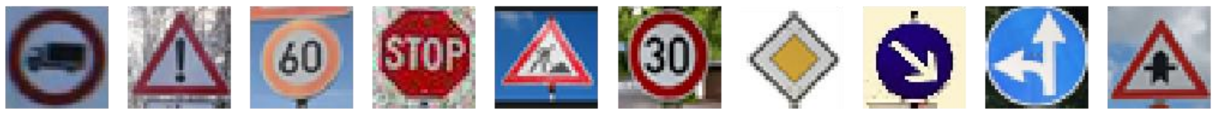
The epoch value I reported here is 100, but I noticed that the same validation accuracy could be achieved with the epoch of 50 as well. The best batch number I found after several trial and errors was the batch number of 200.

I also tried couple of different learning rate values such as 0.0005, 0.0008 and 0.0001 and did not notice too much difference in each result. Therefore, I chose a learning rate of 0.0001.

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 ten German traffic signs that I found on the web:



The labels from left to right are : [16,18,3,14,25,1,12,38,37,11] which they translate to [Vehicles over 3.5 metric tons prohibited, General caution, Speed limit (60km/h), stop, road work, Speed limit (30km/h), Priority road, Keep right, Go straight or left, Right-of-way at the next intersection]

By referring to the diagram showing the number of images per each class, I noticed that classes of 16, 14 and 37 have the lower number of images available for training than the other classes and since I did not add any augmented image, I suspect the model might have difficulty classifying these than the other classes. I think using dropout and L2 reg has helped the model tremendously to avoid overfitting and helping the classes by the low number of images but we have to see the test results.

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 predictions for class of each new images are: **[16 18 3 1 25 1 12 38 37 11]**. Comparing this with the true class labels, the only misidentified class is class 14, which the model predicted as cl ass 1. In another words, instead of predicting "stop sign", the model predicted "speed limit (30km/h)" .

As I discussed in the previous section, I was expecting one of **16, 14 or 37** classes might be mis-class ified due to the limited image for training. Also notice that the class 14 (stop sign) has similarities with class 1 (speed limit 30 km/h), like O of the STOP and 0 of 30. Also, the red color around the text for both signs makes them to look similar. We might be able to increase the model accuracy, by addi ng more images using image augmentation techniques.

All other signs have been predicted correctly, which gives an accuracy of 90%. This compares favorab ly to the accuracy of the test set which was 95.53%.

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)

- 1st image: the model is relatively sure that this is a "Vehicles over 3.5 metric tons prohibited" sign (probability of 0.99), and the image does contain this. The top five soft max probabilities were [9.94557977e-01, 5.40072564e-03, 3.60843806e-05, 4.23837673e-06, 6.99515908e-07] which corresponds to array of class label [16, 9, 10, 7, 8]

- 2nd image : the model is 100% sure that sign is "General caution" (probability of 1.00), and the image does contain this sign. The top five soft max probabilities were [1.00000000e+00, 1.74257203e-16, 7.12649764e-18, 6.91479759e-20, 1.91671480e-21] which corresponds to array of class label [18, 26, 27, 11, 25]

- 3rd image : the model is 100% sure that sign is "speed limit 60 km/h" (probability of 1.0), and the image does contain this sign. The top five soft max probabilities were [1.00000000e+00, 4.83660278e-10, 2.74095555e-11,6.74744836e-15, 6.04897356e-15] which corresponds to array of class label [3, 6, 36, 28, 25].

- 4th image: the model is 41.9% sure that sign is "speed limit (30km/h)". (probability of 0.42), but the image does not contain this sign. The top five soft max probabilities were [4.19178426e-01, 3.83669108e-01, 1.60639793e-01,2.56505851e-02,7.20131164e-03] which corresponds to array of class label [1, 14, 5, 4, 3].

- 5th image : the model is 99.9% sure that this is a Road work sign (probability of 1.0), and the image does contain a Road work sign. The top five soft max probabilities were [9.99333441e-01, 5.05583128e-04,1.21859935e-04,1.60124964e-05,1.49630123e-05], which corresponds to array of class label [25, 21, 31, 30, 29].

For the rest of the images you can see the probabilities and the corresponding classes in the cell 40 of the code.

(Optional) Visualizing the Neural Network (See Step 4 of the Ipython notebook for more details)
1. Discuss the visual output of your trained network's feature maps. What characteristics did the neural network use to make classifications?

CREATE VISUALIZATIONS OF THE SOFTMAX PROBABILITIES
For each of the five new images, create a graphic visualization of the soft-max probabilities. Bar charts might work well.