#Traffic Sign Recognition

##Writeup Template

###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 numpy library to calculate summary statistics of the traffic signs data set:

- The size of training set is 34799

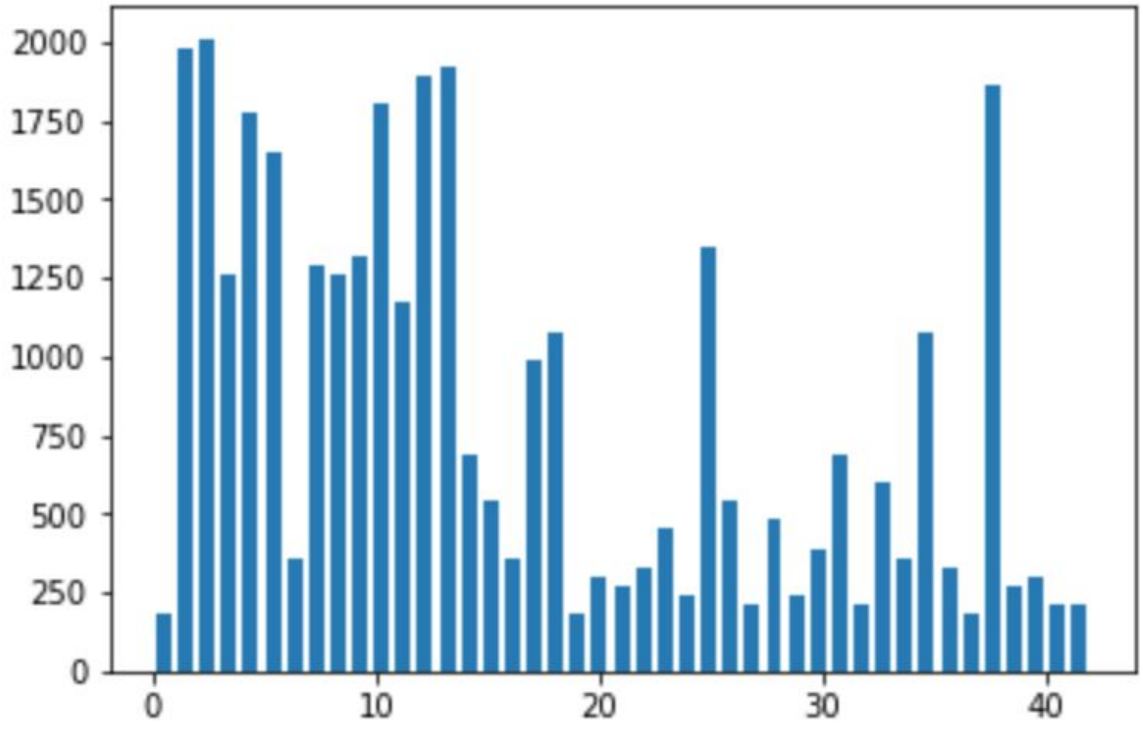
- The size of the validation set is 4410
- The size of test set is 12630
- The shape of a traffic sign image is (32,32,3)
- The number of unique classes/labels in the data set is 43

####2. Include an exploratory visualization of the dataset.

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

The number of labels in the data set is 43. There number of samples per label is varies from 200 to 2000. Balancing the number of samples in each label should be considered.

But in this project, I did not balanced it.



###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

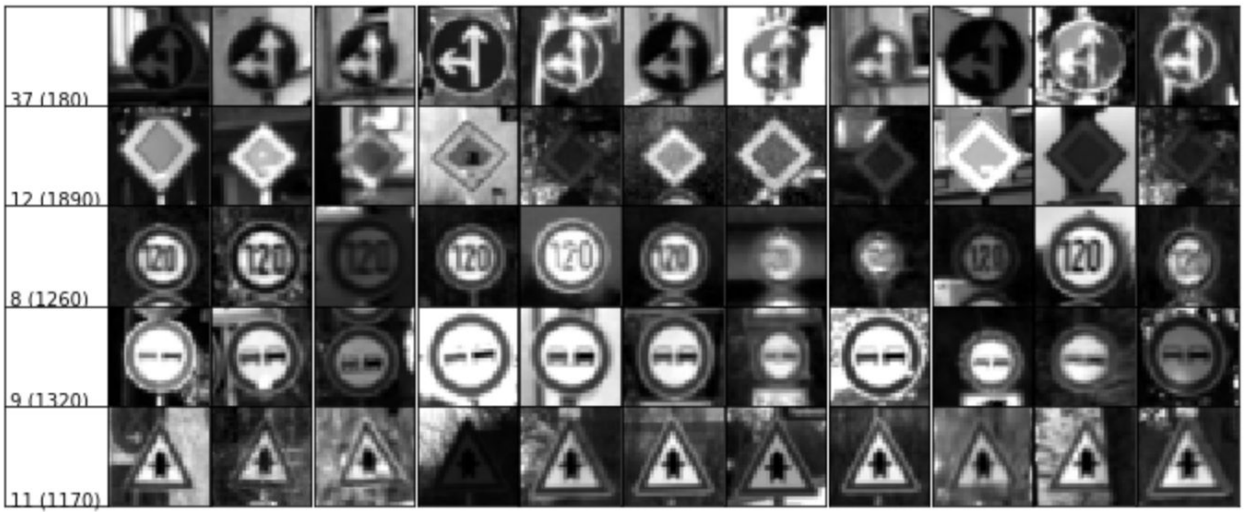
why you decided to generate additional data, how you generated the data, and provide example images of the additional data. Then describe the characteristics of the augmented training set like number of images in the set, number of images for each class, etc.)

As a first step, I decided to convert the images to grayscale because I read Lecun's Traffic sign classifer paper. Lecun says the gray scale image is better.

I think gray scale image is better because various luminance under colorful lights.

So, it's better to ignore the color.

Here is an example of an original image and an gray image:



As the second step, I normalized the image data because Lecun says It's better to enhance contrast. Lecun uses local contrast enhancement. But I just global enhancement.

####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:

1. Input 32x32x1 Gray Image
2. conv2d(5x5) & max pooling to 14x14x30 image
3. conv2d(3x3) & max polling to 6x6x60 image
4. conv2d(3x3) to 4x4x100 images
5. conv2d(3x3) from 2 step , to 12x12x30 (Multi-resolution from Lecun's paper)
6. fully connected from step 4 and 5 to 5920.
7. hidden layer : fully connected 5920 to 100
8. output layer: 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 AdamOptimizer, batch size is 100, number of epochs is 20, and learning rate is 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:

- training set accuracy of 1.0
- validation set accuracy of 0.96
- test set accuracy of 0.95

If an iterative approach was chosen:

- What was the first architecture that was tried and why was it chosen?
  Lecun Net was choosen,
- What were some problems with the initial architecture?
  Too low accuracy.,
- 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.
  First, I have changed it to normalize and gray scale.
  Second I have read Lecun's Traffic Sign paper. I appiled Multiresolution Convolution. It works better.
- Which parameters were tuned? How were they adjusted and why?
  I tried number of feature map from 30-60 to 100-100 for 1st and 2nd Conv-Layer. If I have appied image augmentation 100-100 would be better. But I think 30-60 will be sufficient.
- 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?

I think dropout does not affect this project, because My layer is not deep.

If a well known architecture was chosen:

- What architecture was chosen? From Lecun's Traffic Sign Paper
- Why did you believe it would be relevant to the traffic sign application?
  Multi resolution would be better. Traffic Sign Has Both of Large Shape(circle, triangle) and detailed Fonts(number)
- How does the final model's accuracy on the training, validation and test set provide evidence that the model is working well?
  The training accuracy is 1.0,. validation set shows 0.96, and test set shows 0.95.
  I think 0.95 is sufficient because I have not used much features.

###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:

00013/00058_00016

00007/00045_00001

00008/00027_00026

00015/00011_00023

00005/00041_00000

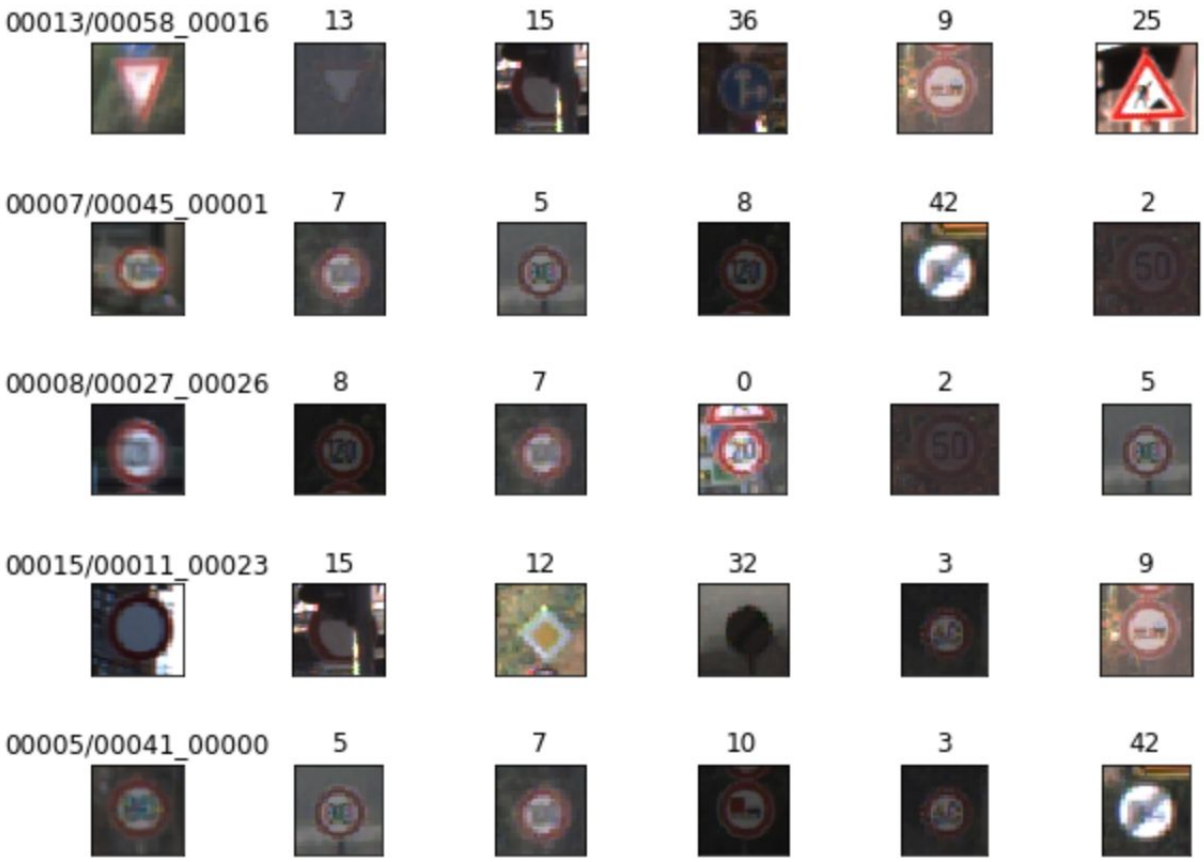
All the images are hard to recognize by myself.

####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:

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

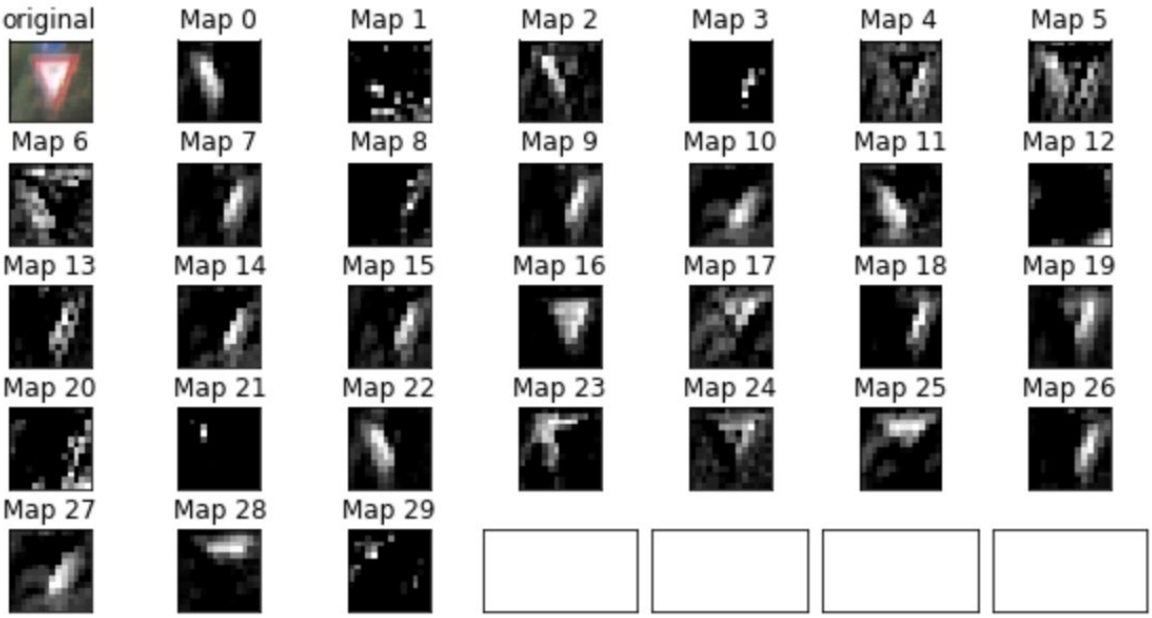
####3. Describe how certain the model is when predicting on each of the five new images by looking at the softmax probabilities for each prediction. Provide the top 5 softmax probabilities for each image along with the sign type of each probability. (OPTIONAL: as described in the "Stand Out Suggestions" part of the rubric, visualizations can also be provided such as bar charts)



In each row of above images, first column is original image, 2~6 column is predicted image from training set. 2 column is very similar to the original images.

(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?



Feature map captures edges at each direction.
And it also capture some shape like triangle area.
It also captures background image.
It means that, when recognizing traffic sign, machine will consider the context(environment).
I don't know it's good or bad.