**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
* Analyze the softmax probabilities of the new images
* Summarize the results with a written report

**Rubric Points**

**Here I will consider the** [**rubric points**](https://review.udacity.com/#!/rubrics/481/view) **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](https://github.com/udacity/CarND-Traffic-Sign-Classifier-Project/blob/master/Traffic_Sign_Classifier.ipynb)

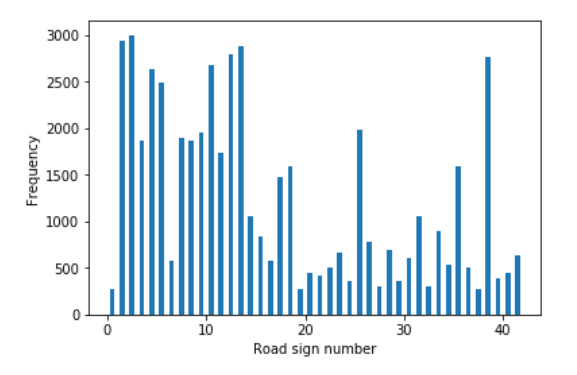
**Data Set Summary & Exploration**

**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 is distributed between different types of road signs,

**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.)**

To process the data I normalized it so that the data has mean zero and equal variance so as to ensure that none of the input data causes the learning to oscillate along different dimensions due to wide variance in the data.

Also, I have retained the colored channels so as to ensure none of the input features are lost.

**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 | 32x32x3 RGB image |
| Convolution 2x2x1 | Stride(1), valid padding, outputs 31x31x130 |
| Convolution 1x1x1 | Stride(1), valid padding, outputs 29x29x100 |
|  |  |
| RELU |  |
| Max pooling | 2x2 stride(2), outputs 14x14x100 |
| Convolution 5x5 | 5x5x1 Stride(1),valid padding, outputs 10x10x80 |
| RELU  Dropout(0.5)  Flatten (2000) |  |
| Fully connected | Input 2000 Output 240 |
| RELU  Fully connected | Input 240 Output 160 |
| RELU  Fully Connected | Input 160 Output 43 (equal to number of classes) |
|  |  |

**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 with one hot encoding and a learning rate of 0.00045.

The number of epochs I used was 11 since I could observe a drop in accuracy. I used a batch size of 32 to quickly increase the accuracy and I also used a logic to ensure that as soon s as the accuracy drops the batch size is changed to 500 which makes the training more diverse and then improves the accuracy.

**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:

* validation set accuracy of 95.4%
* test set accuracy of 93.9%

I used a batch size of 32 to quickly increase the accuracy and I also used a logic to ensure that as soon s as the accuracy drops the batch size is changed to 500 which makes the training more diverse and then improves the accuracy. This technique helped me to increase the overall accuracy and reduce the training time.

I also started with a higher learning rate and continuously reduced the learning rate. I used dropout to ensure that the model does not overfit and it worked since I got a good accuracy on the final test dataset. Initially I did not use dropout and hence the accuracy was lower in the firs few iterations but it increased gradually and I realized that my model was overfitting.

I also started with bigger filters at the beginning of size 5\*5 but then moved to 2\*2 and also 1\*1 to increase the number of trainable parameters. I also observed that reducing the number of fully connected layers was increasing the accuracy up to a certain point.

**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 images have been taken out of the German traffic sign library and are not from the real world. It will be difficult for the network to predict the signs since the network has not seen signboards taken out of context and virtualized. Also two of the signs in the above set are not as common according the histogram plotted and hence the model might face challenge while identifying them.

**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** |
| --- | --- |
| Stop Sign | Stop sign |
| Curve-Left | Curve Left |
| Wild-Animal | Wild-Animal |
| Caution | Caution |
| Curve-right | Curve-Right |

The model was able to correctly guess 5 of the 5 traffic signs, which gives an accuracy of 100%. This compares favorably to the accuracy on the test set of 94

**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)**

The top five soft max probabilities for the first image were

| **Probability** | **Prediction** |
| --- | --- |
| ~1 | Danger left-Curve |
| 7.8915644e-18 | Bicycle Crossing |
| 7.2352173e-18 | Children’s Crossing |
| 1.5088219e-19 | Slippery Road |
| 1.3931444e-20 | Speed Limit(30) |

For the second image ...

| **Probability** | **Prediction** |
| --- | --- |
| 9.9695337e-01 | Danger Right-Curve |
| 3.0465701e-03 | End of no crossing |
| 8.6810630e-09 | Slippery Road |
| 6.1244031e-11 | Children’s Crossing |
| 2.0274543e-13 | No Vehicles |

For the third image ...

| **Probability** | **Prediction** |
| --- | --- |
| ~1 | General caution |
| 4.4432764e-34 | Priority road |
| 3.8573950e-35 | Traffic Signal |
| 1.9818937e-35 | Pedestrians |
| 1.5515999e-35 | Right of Way at next intersection |
|  |  |

For the fourth image ...

| **Probability** | **Prediction** |
| --- | --- |
| 9.9999642e-01 | Stop |
| 3.1390944e-06 | Speed Limit(60) |
| 2.0243918e-07 | Yield |
| 1.4075691e-07 | Wild Animals Crossing |
| 3.2209240e-08 | Speed Limit(80) |

For the fifth image ...

| **Probability** | **Prediction** |
| --- | --- |
| 9.9929011e-01 | Wild Animals Crossing |
| 6.8787538e-04 | Slippery road |
| 2.1187383e-05 | Road Narrow on right |
| 7.7643818e-07 | Beware of Ice/snow |
| 1.3493771e-09 | Double Curve |