Project 2 Writeup - Traffic Sign Recognition

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

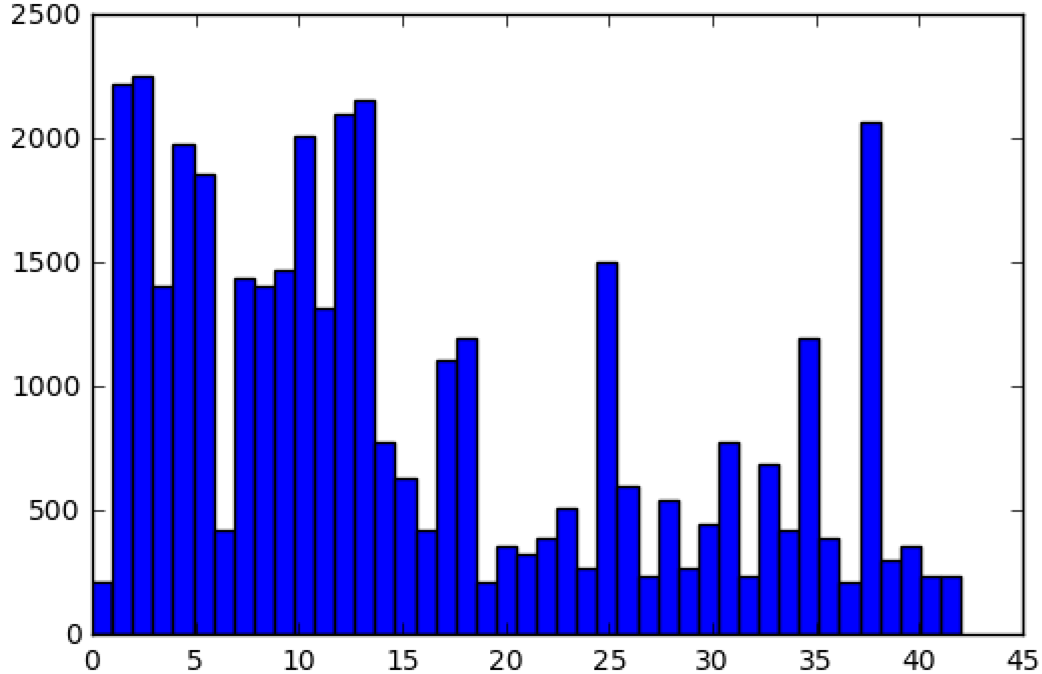
1. Provide a basic summary of the data set. In the code, the analysis should be done using python, numpy and/or pandas methods rather than hardcoding results manually.

I used the 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 32x32x3
- 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 the amount of pictures for each label within the dataset.



<u>Design and Test a Model Architecture</u>

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.)

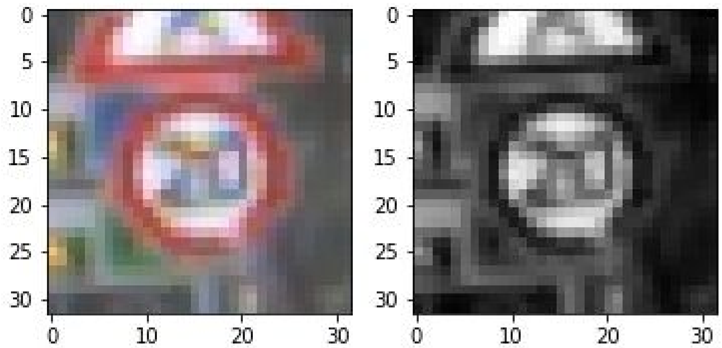
Step 1: Applied a simple shuffle on the training data. This ensured that the model didn’t overfit the data. This is very important to do whenever training a model.

Step 2: Convert the RGB images in training data(3-Channel) to Grayscale (1- Channel). By converting the data to grayscale, the model was forced to classify the images only on one color channel. When I applied this preprocessing technique, I noticed that my validation accuracy greatly increased.

I originally also applied normalization to my preprocessing step, but it reduced my final validation accuracy. For times sake, I decided to move on, but I plan to return to this issue and correct it at a later point.

I also plan to incorporate translation and rotations to the data set which would allow for better generalization when training the model. But as I said before, I was able to get a high validation accuracy by just using a shuffle and grayscaling.

Here is an example of a traffic sign image before and after grayscaling.



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.

The architecture, sizes and connectivity are based on the LeNet lab with the introduction of dropouts before the first and the second fully connected layers.

I chose to perform drop out before the fully connected layers based off of the lessons. The introduction of Dropout significantly increased the overall training accuracy.

My final model consisted of the following layers:

Layer	Description
Input	32x32x3 RGB image
Convolution layer	Input = 32x32x1. Output = 28x28x6.
RELU	Activation function
Pooling layer (2x2)	Input = 28x28x6. Output = 14x14x6
Convolution layer	Input = 14x14x1. Output = 10x10x16.
RELU	Activation function
Pooling layer (2x2)	Input = 10x10x16. Output = 5x5x16
Flatten Layer	Combine all layers
Dropout Layer	To help reduce overfitting

Fully Connected layer	Input = 400. Output = 120
RELU	Activation Function
Dropout Layer	To help reduce overfitting
Fully Connected layer	Input = 120. Output = 84
RELU	Activation function
Fully Connected layer	Input = 84. Output = 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.

Optimizer was again based on the LeNet Lab (Adam optimizer).

Here are the Hyper Parameters:

1. EPOCH Size: 55
2. Batch Size: 128
3. Learning Rate: 0.0008
4. Keep Prob: 0.5 (As suggested in the lessons)

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.

I spent a lot of time creating this final model. I tried many different parameters and architectures before I was able to obtain such a high validation accuracy. A lot of my progress came from making shifts in the architecture and running a set to see how it performed. I tried adding and removing dropout layers. I added and took away convolution layers. I added up to 10 different fully connected layers. All the guessing and checking gave me a much better idea of what changes increase the validation accuracy, and which ones made it fall. All the models were vaguely based off of the LeNet architecture, which proved to work very well almost as is.

The biggest impact came from converting the images to grayscale. I also attempted to normalize the data, but the validation accuracy fell fairly significantly. I lowered the learning rates and increased the epochs which helped me reach a good enough validation accuracy to complete the project.

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 first image might be difficult to classify because it could be confused with other triangle shaped signs.

The second image might be difficult to classify because it could be confused with other signs of the same shape.

The third image might be difficult to classify because it looks somewhat familiar to a stop sign.

The fourth image might be difficult to classify because you can't see the full shape of the sign. The top of the triangle is cut off.

The fifth image might be difficult to classify because there are several different signs that look familiar to the image drawn on this sign.

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 accuracy on the captured images is 60% while it was 94% on the testing set thus It seems the model is overfitting. I believe this difference also came from the difficulties of the images that I used for my uploaded images. That being said, the model was extremely confident when predicting 3 of the 5 images.

Here are the results of the prediction:

Image	Prediction
Right of Way	Right of Way
Priority Road	Priority Road
No Entry	No Entry
General Caution	Speed Limit 20km/h
Children/Bicycle Crossing	Traffic Signal

The model was able to correctly guess 3 of the 5 traffic signs, which gives an accuracy of 60%.

Ê **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 code for making predictions on my final model is located in the 11th cell of the Ipython notebook.

IMAGE 1: For the first image, the model is relatively sure that this is a right of way (probability of 0.6), and the image does contain a right of way sign. The top five soft max probabilities were

Probability	Prediction
.99	11 - Right of Way
>.01	27 - Pedestrian
>.01	30 - Beware of ice/snow

>.01	21 - Double Curve
>.01	12 - Priority road

IMAGE 2: For the second image, the model is relatively sure that this is a priority road sign (probability of 0.99), and the image does contain a priority road sign. The top five soft max probabilities were

Probability	Prediction
.99	12 - Priority road
>.01	40 - Roundabout mandatory
>.01	35 - ahead only
>.01	13 - yield
>.01	38 - keep right

IMAGE 3: For the third image, the model is relatively sure that this is a no entry sign (probability of 0.6), and the image does contain a no entry sign. The top five soft max probabilities were

Probability	Prediction
.99	17 - no entry
>.01	14 - stop
>.01	34 - turn left ahead
>.01	9 - no passing
>.01	0 - speed limit (20km/h)

IMAGE 4: For the fourth image, the model is relatively sure that this is a turn right ahead sign (probability of 0.47), but it is a general caution sign (prob .43). The top five soft max probabilities were

Probability	**Prediction**
.47	33 - Turn left ahead
.43	18 - general caution
.05	31 - wild animal crossing
.04	25 - road work
.01	39 - keep left

IMAGE 5: For the fifth image, the model is unsure if it's a traffic signal sign or a right of way sign. It is neither. It is a children/bicycle crossing sign. The top five soft max probabilities were

Probability	**Prediction**
.26	40 - roundabout mandatory
.25	11 - right of way
.08	18 - general caution
.07	25 - road work
.06	27 - children crossing

Thank you for checking out my project! I hope you learned something about classifying images using a convolutional neural network.