Traffic Sign Recognition

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 plot of 10 randomly selected traffic sign images

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.)

I decided to normalize the images to zero mean and standard deviation of 1. I did this so that images with different brightness and contrast would be treated in the same manner. I decided to keep the images as color images because I thought color information may be important in distinguishing the different traffic signs.

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 is a slight tweak of the standard LeNet model used in class. Since color images are used, the depth of the convolutional layers were slightly increased and the number of neurons in the fully connected layers increased slightly.

Layer	Description
Input	32x32x3 RGB image
Convolution 5x5	1x1 stride, valid padding, outputs 28x28x6
ReLU	
Max pool	2x2 stride, valid padding, outputs 14x14x6
Convolution 5x5	1x1 stride, valid padding, outputs 10x10x16

Layer	Description
Relu	
Max pool	2x2 stride, valid padding, outputs 5x5x16
flatten	
Fully connected layer	outputs: 120
ReLU	
Dropout	
Fully connected layer	outputs: 84
Relu	
Dropout	
Fully connected layer	outputs: 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 calculate the loss function, I first used the softmax cross entropy function for the logits output and one-hot encoded training output. I then calculate the loss by summing the mean of the cross enropy and the produt of sum of the weights times a weight_cost(regularization). To train the model, I used an 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 results were:

- validation set accuracy of 0.937
- test set accuracy of 0.936

I first started out with the default parameters carried over from the LeNet model lab. The LeNet model had performed well on the number classification problem so it was reasonable to believe that it may perform well in the traffic sign classification. The plan was to see how this architecture preformed and to develop further plans from there. This resulted in validation accuracy of 0.910.

I observed oscillation in the validation accuracy values between epochs. Based on this, I thought that perhaps that model is overiftting the training data. I introduced dropout and regularization in an effort to with a keep rate of 0.7 and and weight_cost of 0.002, the validation accuracy increased to 0.935, with upward trending accuracy in all epochs but one.

While some hyperparemeter tuning here and increasing the depth/size of the neural network could potentially increase the accuracy, it was seen as unecessary due to a high accuracy rate. High (and similar) accuracy level on the validation and traing show that the model is working well.

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 the image is not centered and is a bit crooked. The third one may be hard to classify since it is also crooked and occupies a fairly small portion of the image. The fifth one may be difficult to classify since the sign also occupies a fairly small portion of the entire frame. I did not augment the data with zoom or rotation so some of these things may be hard to detect.

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**
Right-of-way at the next intersection	Right-of-way at the next intersection
General caution	General caution
Turn right ahead	Turn right ahead
Road work	Road work
Stop sign	Priority road

The model was able to correctly guess 4 of the 5 traffic signs, which gives an accuracy of 80%.

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 all five images the model is very certain about its predictions, even when it got the answer wrong. As a matter of fact, the lowest softmax probability is 91.7%. On the fifth image where the prediction is wrong, the probability of it being a stop sign (correct answer) is not one of the top five softmax probabilities.

Image 1:

Probability	Prediction
.99991	Right-of-way at the next intersection
7.4E-6	Beware of ice/snow
1.7E-6	Pedestrians
3.5E-8	Double curve
1.3E-8	General caution

Image 2:

Probability	Prediction
.9999996	General caution
3.7E-7	Pedestrians
3.0E-8	Traffic signals
2.2E-11	Right-of-way at the next intersection
7.6E-12	Road narrows on the right

Image 3:

Probability	Prediction
0.92	Turn right ahead
0.04	Turn left ahead
0.04	Roundabout mandatory
2.4E-3	Ahead only
9.4E-4	Go straight or left

Image 4:

Probability	Prediction
0.9999999	Road work
1.0E-7	Dangerous curve to the right
1.3E-8	Right-of-way at the next intersection
2.4E-9	Beware of ice/snow
1.6E-10	Slippery road

Image 5:

Probability	Prediction
1	Priority road

Probability	Prediction
3.8E-8	Right-of-way at the next intersection
1.5E-8	No passing for vehicles over 3.5 metric tons
6.7E-10	Double curve
3.1E-10	Wild animals crossing