Traffic Sign Recognition

Udacity SDC Term 1, P2

By David Rose

2017-05-11

Convolutional Neural Network in Base TensorFlow

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**
    - You're reading it! and here is a link to my code: github.com/cipher982/Traffic-Sign-Conv-Net-Recognition

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.**

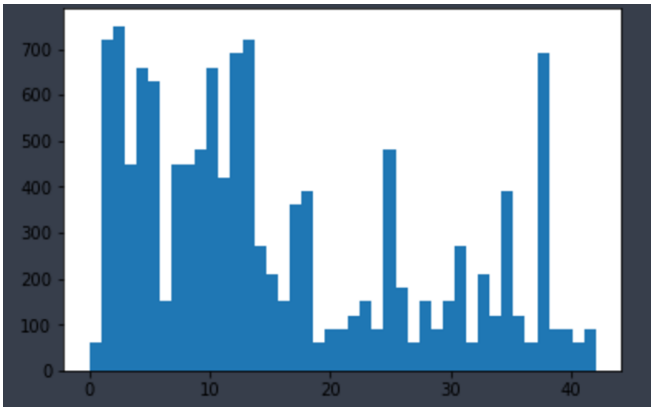
    I used the pandas library to calculate summary statistics of the traffic signs data set:

    - The size of training set is:
        - Before processing/augmentation: **34799**
    - The size of the validation set is --
    - The size of test set is **12630**
    - The shape of a traffic sign image is **32x32** pixels, with **3** levels of color depth

- The number of unique classes/labels in the data set is **43**

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

   Here is a bar chart showing how the data

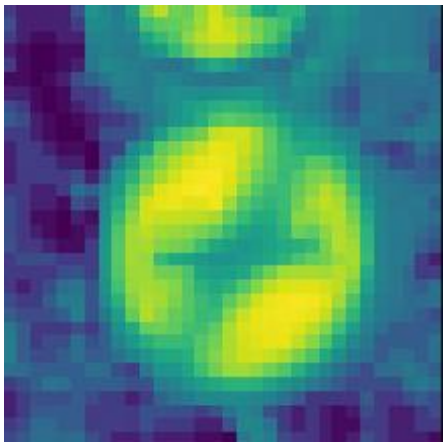


Design and Test a Model Architecture

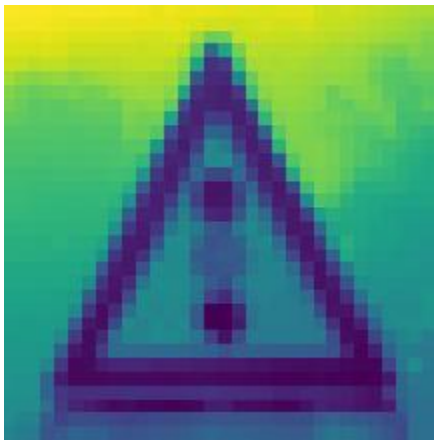
1. Describe how you preprocessed the image data.

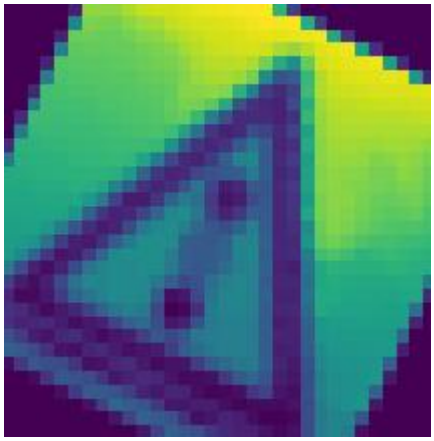
I first converted to **one color channel** and **equalized the pixel brightness** average

- I did not want differences in lighting/exposure/brightness to bias or skew the training



Then I created additional training data from the original, as it is possible some signs may not be seen straight on. I did this by randomly rotating the images between **-30 and +30** degrees from original. The dataset ended up **3 times larger** than the original, at **104397**





2. **Describe what your final model architecture looks like**

   My final model consisted of the following layers:

Layer
Input, 32 input
Convolution
Convolution
Convolution
Drop Layer, keep 70%
Convolution
Flatten
Fully Connected
Fully Connected
Fully Connected, 43 output
Dropout, keep 70%

3. **Describe how you trained your model.**
   - I used minibatches of size 150, with 7 total epochs. Trained over my Nvidia 960M GPU in laptop with tensorflow. Learning rate set at 0.001, sigma at 0.1, mu at 0.
4. **Describe the approach taken for finding a solution and getting the validation set accuracy to be at least 0.93.**
   - My final results were the work of lots of random tweaking. Modifying parameters up/down until something started to work better. Switching from CPU to GPU allowed this process to go along much faster.
   - I started out adding two extra fully connected layers, but this did not seem to help much at all. Then I read some others had better luck adding convolutional layers instead.
   - I also came across a function for installing new layers, which took out a lot of the manual work of explicitly typing out layer by layer in TensorFlow.

training set accuracy	.965
validation set accuracy	.944
test set accuracy	.921

~~Dropout was also added, but I did not use as much as others do, only a single layer keeping 90%.~~

Dropout was changed to 70%, and added another dropout after the last layer. Also, per recommendation I have used a placeholder for the dropout value, so that I can set it to no dropout when evaluating.

Test a Model on New Images

1. **Choose five German traffic signs found on the web and provide them in the report.**

I tried to find relatively simple, straight on images. They vary a bit on backgrounds, but otherwise looks quite similar.

2. **Discuss the model's predictions on these new traffic signs and compare the results to predicting on the test set.**

Here are the results of the prediction (using probabilities from task below:

Image	Prediction
Yield	Double Curve
Double Turn	Double Curve
Workers Ahead	Double Curve
70 km/h	No passing for vehicles over 3.5 metric tons
Slippery Road	Double Curve

The model was only able to guess a single correct sign, I imagine there's some sort of error here as it just wants to guess the same thing over and over. (Except when I just run the evaluate code on the new data, then it says 60%, but I don't know how to find the predictions from that)

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.

Sign	Predictions
Yield	1 : Double curve 0.931545 2 : Speed limit (80km/h) 0.0643561 3 : No passing for vehicles over 3.5 metric tons 0.00408106 4 : Wild animals crossing 1.4956e-05 5 : End of all speed and passing limits 9.77684e-07
Double Turn	1 : Double curve 0.999779 2 : No passing for vehicles over 3.5 metric tons 0.000211025 3 : Speed limit (80km/h) 9.87119e-06 4 : Dangerous curve to the right 8.32696e-09 5 : Ahead only 4.39636e-09
Workers Ahead	1 : Double curve 0.999988 2 : No passing for vehicles over 3.5 metric tons 1.12904e-05 3 : Go straight or left 2.01945e-07 4 : Speed limit (80km/h) 7.71859e-09 5 : End of no passing 7.01678e-09
70 km/h	1 : No passing for vehicles over 3.5 metric tons 0.734856 2 : Double curve 0.264369 3 : End of no passing 0.000436071 4 : Speed limit (80km/h) 0.000308796 5 : Wild animals crossing 1.3751e-05
Slippery Road	1 : Double curve 0.996778 2 : No passing for vehicles over 3.5 metric tons 0.00311182 3 : No entry 9.51355e-05 4 : Go straight or left 5.567e-06 5 : Speed limit (80km/h) 3.93932e-06

It is very confident in the completely wrong signs, with the exception of one of them.