**A Convolutional Neural Network for Classification of Traffic Signs**

David Chrzanowski

**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**

**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 numpy size and length methods to output the dataset summary below:

Number of training examples = 34799

Number of validation examples = 4410

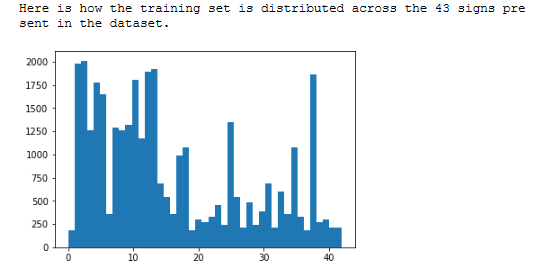
Number of testing examples = 12630

Image data shape = (32, 32, 3)

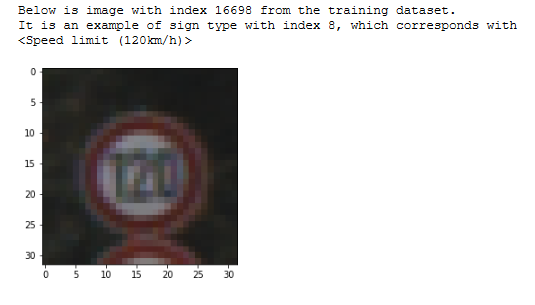
Number of classes = 43

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

I used the matplotlib histogram function to output a plot of how many of each kind of image is present in the training set – see below:



I also randomly selected an image in the training set, and displayed it and the index value for its assigned label. I then used csv.reader to reference this index in the csv file and display as a part of the image caption. This helped me build confidence that I had loaded the data correctly – see below:



**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?***

I chose to normalize each color channel between -1 and +1 via subtracting 128 from the pixel value for that channel, and then dividing by 128. This normalization, with a mean around 0 creates more numerical stability when training the model.

I considered also making the images grayscale before normalization, however examination of the test images led me to believe that color might be a valuable feature in classification. Specifically, blue backgrounds, red borders, and the presence or absence of color within the body of the sign all could serve to identify signs in ways that would not be possible with grayscale images.

I was able to achieve the required accuracy against the validation set and never made the images grayscale.

***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.***

|  |  |  |
| --- | --- | --- |
| Layer 1: Convolutional | Convolution | Input = 32x32x3. Output = 28x28x6. strides = [1,1,1,1], padding = VALID |
| Activation | RELU |
| Pooling | Ksize=[1,2,2,1] strides=[1,2,2,1] |
| Layer 2: Convolutional | Convolution | Input = 14x14x6. Output = 10x10x16. strides = [1,1,1,1], padding = VALID |
| Activation | RELU |
| Pooling | Ksize=[1,2,2,1] strides=[1,2,2,1] |
| Flatten | | Input = 5X5X16. Output = 400. |
| Layer 3: Fully Connected | Combination | Input = 400. Output = 120. |
| Activation | RELU |
| Dropout | Keep prob = .5 (training only) |
| Layer 4: Fully Connected | Combination | Input = 120. Output = 84. |
| Activation | RELU |
| Dropout | Keep prob = .5 (training only) |
| Layer 5: Fully Connected | Combination | 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.***

The final training of the model was done using TensorFlow’s AdamOptimizer, set to minimize the cross entropy between the softmax values of the models output logits and the one-hot encoded labels.

The training was done with a batch size of 128 over 25 EPOCHs, with a learning rate of .001 (initializing parameter of Adam Optimizer, which ultimately uses a dynamic rate).

***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 99.7%

Validation set accuracy of 95.5%

Test set accuracy of 93.9%

This is calculated by restoring the saved model parameters from training and using the “evaluate” function on each of the 3 datasets. This is done in the code cell immediately following the “#CODE TO TRAIN” cell below the accuracy outputs of the 25 training EPOCHs.

***What architecture was chosen?***

Having been previously unacquainted with neural networks, I began with the LeNet-5 architecture as was recommended in the lessons. I experimented with different numbers of EPOCHs, learning rate, batch size, and inclusion of dropout in various layers. Ultimately dropout instituted across the first 2 fully connected layers and 25 EPOCHs was sufficient to exceed the required 93% accuracy on the Validation Set. I kept the individual color channels present in the images to allow for use of this additional classifying feature.

***Why did you believe it would be relevant to the traffic sign application?***

I knew that this architecture is effective for symbol/icon detection as demonstrated on the MNIST database. Traffic signs are not an altogether different problem – with slight variations in different instances of the same sign likely less severe than the different handwriting present in the MNIST dataset.

***How does the final model's accuracy on the training, validation and test set provide evidence that the model is working well?***

With the validation set altogether different images from the training set, and the test set a very large number of still different independent images, achieving higher than 93% on both sets indicates a fairly accurate model. The wide spread, from over 99% on the training set to under 96% on the validation set likely represents a slight overfitting to the test set.

**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.***

**C:\Users\chrzanowski.dm\Documents\carnd\p2\CarND-Traffic-Sign-Classifier-Project\Test Images\image_1_30kmhr.jpg C:\Users\chrzanowski.dm\Documents\carnd\p2\CarND-Traffic-Sign-Classifier-Project\Test Images\image_2_100kmhr.jpg C:\Users\chrzanowski.dm\Documents\carnd\p2\CarND-Traffic-Sign-Classifier-Project\Test Images\image_3_keepright.jpg C:\Users\chrzanowski.dm\Documents\carnd\p2\CarND-Traffic-Sign-Classifier-Project\Test Images\image_4_roadwork.jpg C:\Users\chrzanowski.dm\Documents\carnd\p2\CarND-Traffic-Sign-Classifier-Project\Test Images\image_5_nopass.jpg**

I chose the 5 images above to feed to my classifier for identification.

1. 30 km/hr Speed Limit

This image cold be difficult to classify because it is not facing the camera head-on, making its perceived shape less circular. It also has another opposite-facing sign behind it which also distracts from its circular shape.

1. 100 km/hr Speed Limit

I was curious to see if the classifier could accurately separate one speed limit sign from another. Exact placement of the sign in the image – along with almost all of them having 0’s (and this one having 2) both make correct identification of the exact speed limit a challenge.

1. Keep Right

This sign has the benefit of the unique sky-blue color – identifiable by my classifier since I did not utilize grayscale in pre-processing. However, resolution for this example is poor, making it possibly hard to distinguish from the highly similar “Keep Left” sign.

1. Road Work

In this image the sign is zoomed in particularly far. Additionally, the background of the image is both dark (high pixel values) and not-uniform, possibly distracting the classifier from what is and is not the sign.

1. No Passing

This sign is very similar to speed limit signs in that it is a red-bordered circle with two icons in it (like any 2-digit speed limit). However, color differentiation of said icons (since the left car is red) may help separate it. That said, this does NOT separate it from the EXTREMELY similar “No Passing for Vehicles over 3.5 TNE” sign, which will provide a significant challenge.

***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.***

The model gave an accuracy of .40 for the new images, or 2 out of 5. This was disappointing given the 93%+ accuracy of the model on the test set. See #3 below for the exact softmax probabilities of each of the predictions.

It is encouraging that for the 2 signs the model identified correctly (Keep Right and No Passing), it was VERY certain for them – with higher probabilities than for any other classifications. Further, for the 100 km/hr sign 4 of its top 5 estimates were speed limit signs. Unfortunately, none were for 100km/hr specifically. The model was highly unsure of the first sign, possibly due to its unclear shape (image somewhat angled). Added “jittered” images in the training set could help to rectify this. For the 4th sign, the model was fairly certain of its incorrect prediction, but it is worth noting that the guesses are all within the same sign family (red-bordered triangles).

**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)**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **30 km/hr** | | **100 km/hr** | | **Keep Right** | | **Road Work** | | **No Passing** | |
| **Prob** | **Prediction** | **Prob** | **Prediction** | **Prob** | **Prediction** | **Prob** | **Prediction** | **Prob** | **Prediction** |
| .414 | No Pass 3.5 TNE | .645 | 30km/hr | ~1.0 | Keep Right | .993 | Right of Way Next Intersect. | ~1.0 | No Passing |
| .307 | Turn Right Ahead | .101 | 60km/hr | ~0 | Turn Left Ahead | ~.007 | Beware of Ice/snow | ~0 | Vehicles over 3.5 TNE Prohib. |
| .120 | 100km/hr | .063 | Dangerous Curve Left | ~0 | No Entry | ~0 | Double Curve | ~0 | No passing for vehicles 3.5 TNE |
| .065 | Right of Way Next Intersect. | .053 | 50km/hr | ~0 | Turn Right Ahead | ~0 | Slippery Road | ~0 | Priority Road |
| .034 | Dangerous Curve Left | .044 | 80km/hr | ~0 | Roundabout Mandatory | ~0 | Pedestrians | ~0 | End of No Passing |