Traffic Sign Classifier Project

Fan Zhou

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 pickle library to load all traffic signs data sets and then use ".shape" command to read out the size of each data set.

The size of training set is 34799.

The size of the validation set is 4410.

The size of test set is 12630.

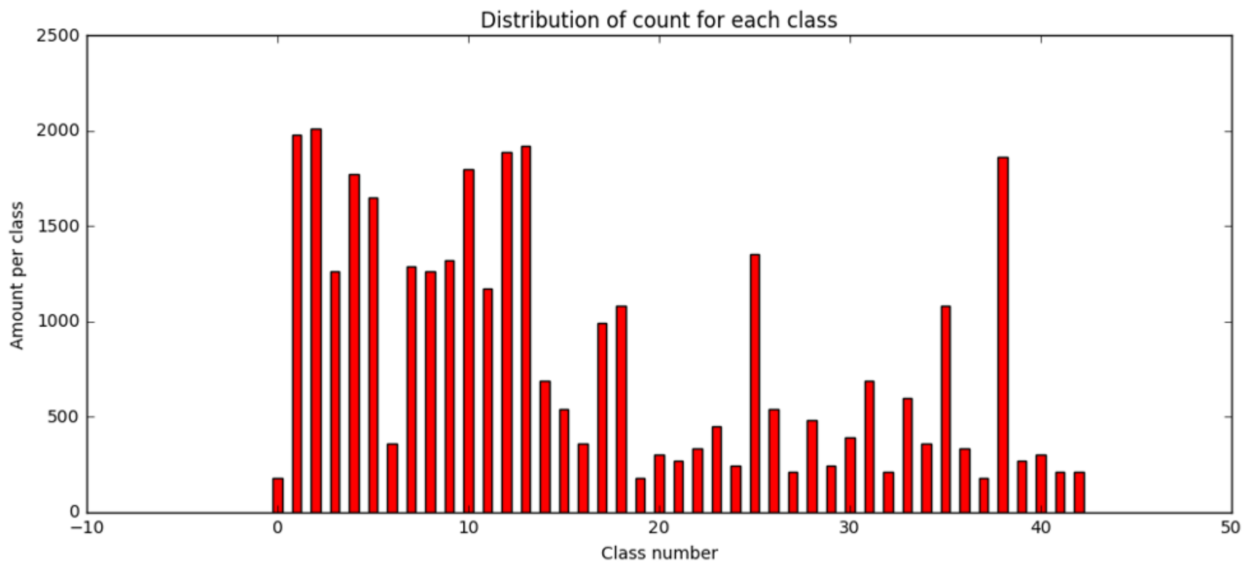
The shape of a traffic sign image is (32, 32, 3), the corresponding grayscale image is (32, 32, 1).

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

2. Include an exploratory visualization of the dataset.

First, I randomly pick one image from each classes.

Secondly, I visualize the distribution of amount per class in the training, validation and test data set. The following figure illustrates the distribution of the training set.



Design and Test a Model Architecture

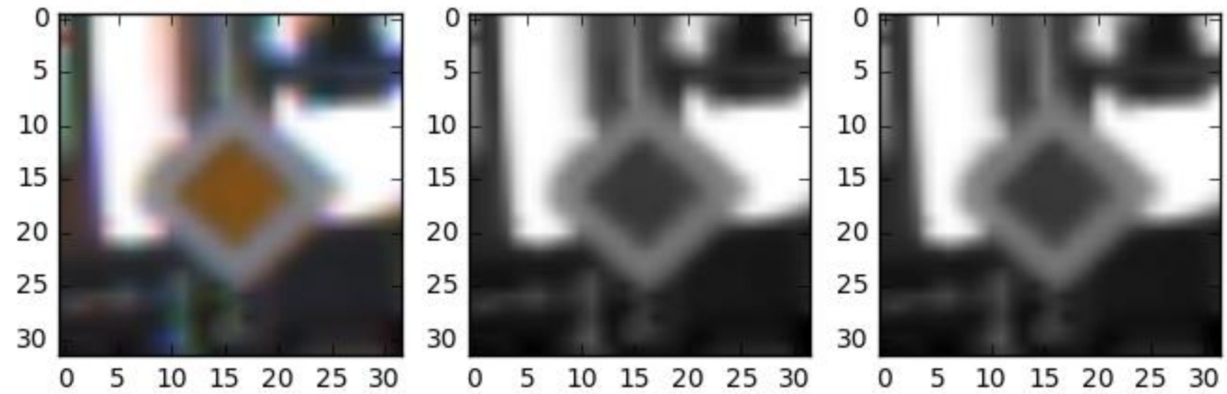
1. Describe how you preprocessed the image data. What techniques were chosen and why did you choose these techniques?

My dataset preprocessing includes:

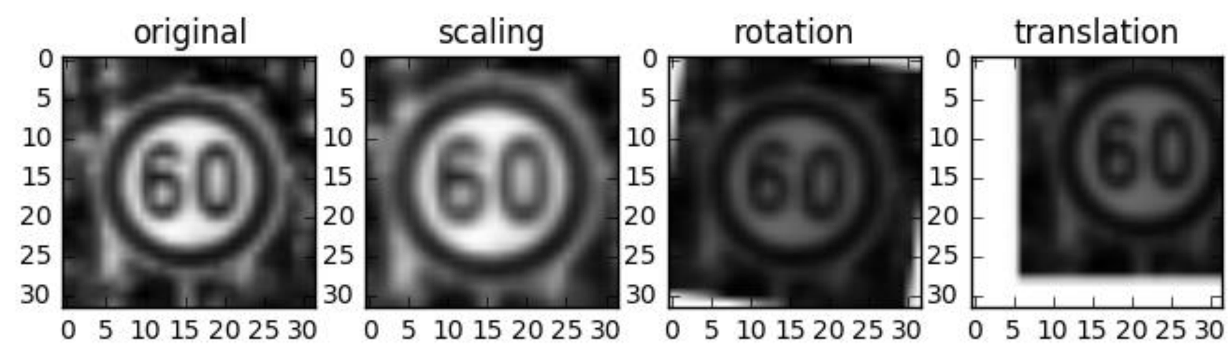
a. Converting to grayscale works well in the written digits classification. It also helps to reduce training time.

b. Once I got the grayscale data set. I scaled the data by using max-min method. Such that all pixel values in all data are scaled in the range of -0.5 to 0.5. This was done using the line of code X_scaled = -0.5 + (X - Xmin) / (Xmax - Xmin). The resulting dataset has a mean of -0.19, while the mean before scaling is 82. I believe another method using (X – 128) / 128 also works. The reason of applying scaling or normalization is to make gradient descent converge to the minimal value fast and stable. A wider distribution in the data would make it more difficult to train using a single learning rate, especially when values of different features may range far from each other and a single learning rate might make some weights diverge.

The following figure shows an example of original, grayscale, and max-min scaled sign image. The grayscale and max-min scaled image doesn’t have obvious difference.



c. I also wrote functions for transforming an image, although I didn't add the transformed images to the training set. The functions are random scaling, random rotation and random translation. The following figure compares these operations.



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.

I applied the same architecture from the LeNet Lab and adding two dropout layers. This model worked quite well with ~96% validation accuracy. The layers are set up like this:

1. 5x5 convolution (32x32x1 in, 28x28x6 out)
2. ReLU
3. 2x2 max pool (28x28x6 in, 14x14x6 out)
4. 5x5 convolution (14x14x6 in, 10x10x16 out)
5. ReLU
6. 2x2 max pool (10x10x16 in, 5x5x16 out)
7. Flatten layers from #6 (5x5x16 -> 400)
8. Fully connected (400 in, 120 out)
9. ReLU
10. Dropout layer

11. Fully connected layer (120 in, 84 out)
12. ReLU
13. Dropout layer
14. Fully connected layer (84 in, 43 out)
15. Softmax

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.

I used the Adam optimizer. The final settings used were:

- batch size: 128
- epochs: 50
- learning rate: 0.0009
- mu: 0
- sigma: 0.1
- dropout keep probability: 0.5

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 96.0%
- test set accuracy of 93.6%

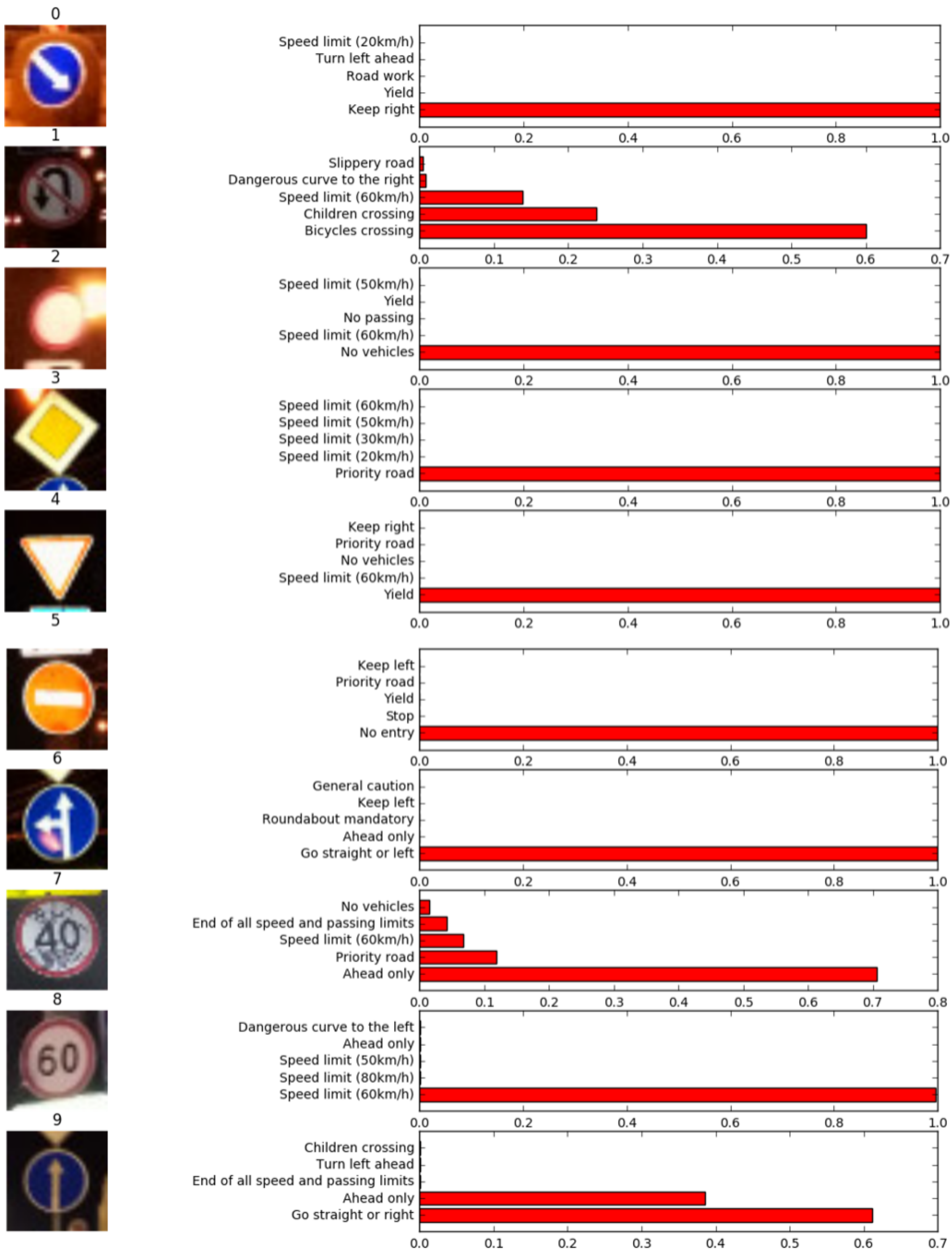
If a well-known architecture was chosen:

- I applied the same architecture from the LeNet Lab and adding two dropout layers.
- The LeNet architecture is a successful in classifying written digits in grayscale. The traffic signs also have some similar features, like numbers, shapes. At beginning it is hard to predict how well the LeNet works for traffic signs. During the training, the accuracy is good for both training set and validation set. The accuracy for test set is a little lower but it is reasonable. The potential improve includes adding the transformed image data to the training set and use original color information.

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.

I have tested 10 images. Here image 1 (no U-turn) and image 7 (40 speed limit) don’t exist in the training set. So the classifier may not predict these patterns accurately.


0
Speed limit (20km/h)
Turn left ahead
Road work
Yield
Keep right
0.0 0.2 0.4 0.6 0.8 1.0
1
Slippery road
Dangerous curve to the right
Speed limit (60km/h)
Children crossing
Bicycles crossing
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7
2
Speed limit (50km/h)
Yield
No passing
Speed limit (60km/h)
No vehicles
0.0 0.2 0.4 0.6 0.8 1.0
3
Speed limit (60km/h)
Speed limit (50km/h)
Speed limit (30km/h)
Speed limit (20km/h)
Priority road
0.0 0.2 0.4 0.6 0.8 1.0
4
Keep right
Priority road
No vehicles
Speed limit (60km/h)
Yield
0.0 0.2 0.4 0.6 0.8 1.0
5
Keep left
Priority road
Yield
Stop
No entry
0.0 0.2 0.4 0.6 0.8 1.0
6
General caution
Keep left
Roundabout mandatory
Ahead only
Go straight or left
0.0 0.2 0.4 0.6 0.8 1.0
7
No vehicles
End of all speed and passing limits
Speed limit (60km/h)
Priority road
Ahead only
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8
8
Dangerous curve to the left
Ahead only
Speed limit (50km/h)
Speed limit (80km/h)
Speed limit (60km/h)
0.0 0.2 0.4 0.6 0.8 1.0
9
Children crossing
Turn left ahead
End of all speed and passing limits
Ahead only
Go straight or right
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7

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.

From the softmax probability, we see all images are in the right class except image 1 and image 7 which are not in the training data set. Image 9 also has two probabilities. Excluding image 1 and 7, the accuracy is 87.5% that is lower than that of test data set of 93.5%. But the accuracy of testing a few images strongly depends on what images I choose.

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)

See above figures.