Udacity CarND Traffic Sign Recognition project writeup

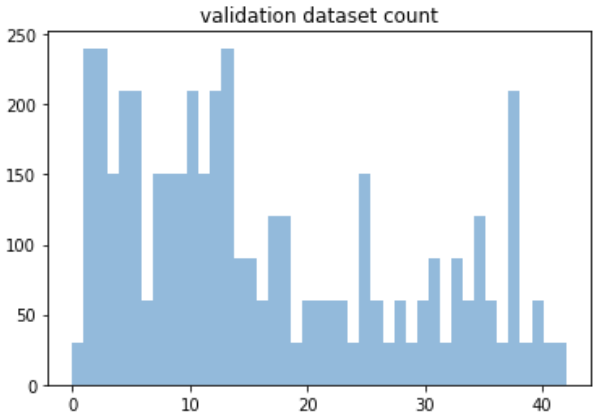
Data Set Summary & Exploration

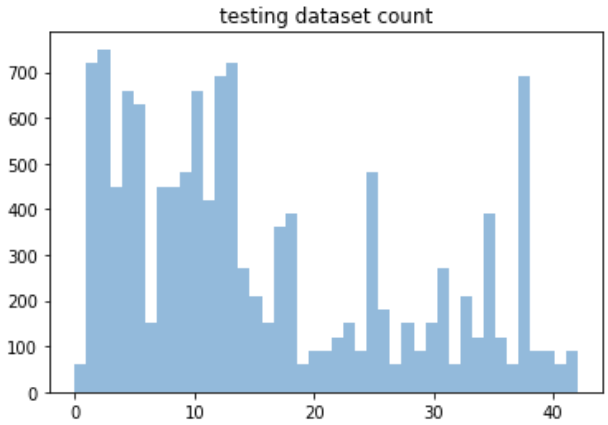
The dataset we are using is the German Traffic Signs Dataset. Below is a brief summary of the dataset used in this project:

Number of training examples = 34799
Number of testing examples = 12630
Image data shape = (32, 32, 3)
Number of classes = 43

This graph is an exploratory of the dataset. I drew the first encountered picture of each label.

Below is bar chart showing how the data is distributed (frequency of each label):



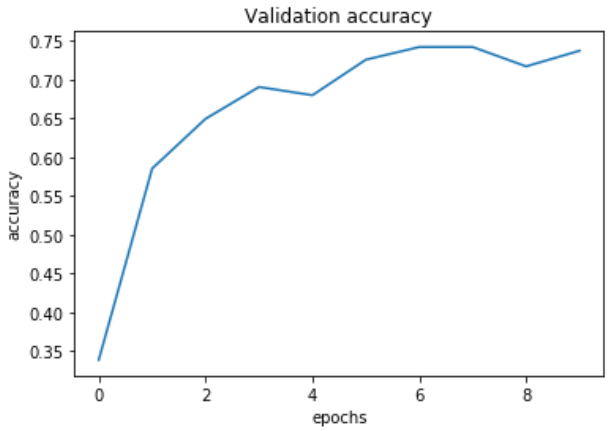


Design and Test a Model Architecture

Data preprocessing

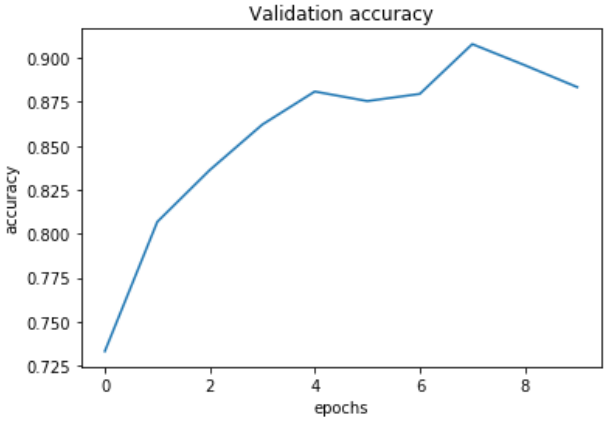
1. What techniques were chosen and why did you choose these techniques

Normalization: I first implemented normalization method using: (pixel -128)/128.



Only using this preprocessing method only gives us 75% accuracy max (using the same LeNet architecture given in the lecture). So, clearly only using normalization is not enough.

Grayscale: I then convert each training image to grayscale.

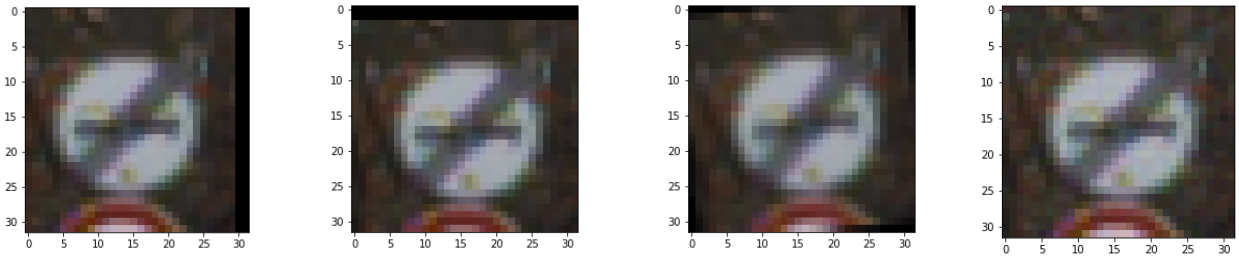


Combined with normalization, the performance significantly improved. The validation accuracy can be as high as 90%.

Data augmentation: Since our dataset is not balanced, we need to create some fake images by data augmentation to boost accuracy even higher. I implemented 3 types of data augmentation. The first is transition. It randomly transit the image to its left, right, up or down within a range of [-2, 2] pixels.

The second type is random rotation. It randomly rotate the image within range of [-15, 15].

The third type is random noise. It adds random noise within 10 to the original images.



From left to right: horizontal transition, vertical transition, rotation, noise

Next, we appended the new created images to the original training dataset. We now have a relative balanced dataset.

Final model:

Layer	Description
Input	32x32x1 (grayscale)
Conv 3x3	5x5 stride, same padding, outputs 28x28x6
ReLU	
MaxPooling	2x2 stride, same padding, outputs 14x14x6
Conv 3x3	5x5 stride, same padding, outputs 10x10x16
ReLU	
MaxPooling	2x2 stride, same padding, outputs 5x5x16
Flatten	
Fully connected	Input: 400, output: 120
Fully connected	Input: 120, output: 84
ReLU	
Fully connected	Input: 84, output: 43

How I trained my model:

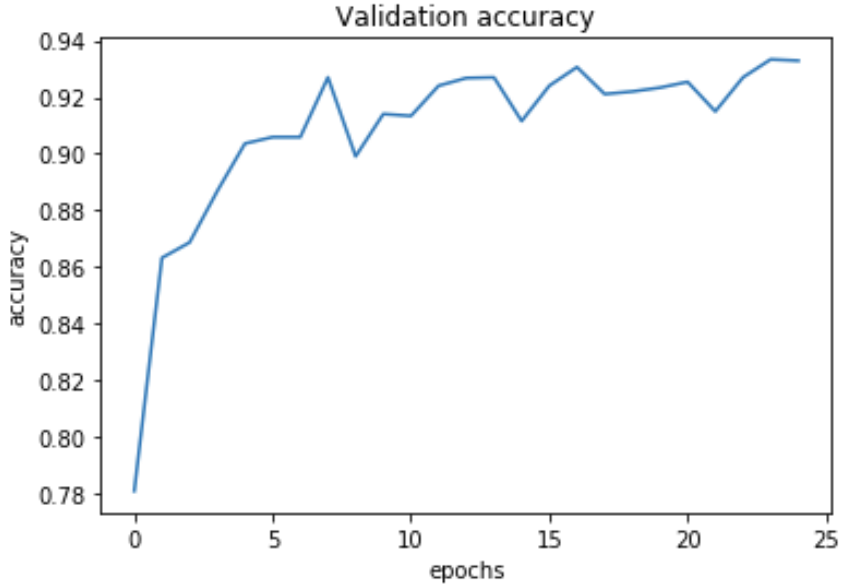
learning rate: 0.0005

Epochs: 25

Batch size: 64

To save the best mode only, I only save the model of the validation accuracy is higher than the best validation accuracy in the training history.

Below is the final validation accuracy plot:



Testing on new images:

The model seems to perform pretty well on the images that it never sees before. The accuracy is 100% in this case.


```
Prediction:
13: Yield
True label:
13: Yield

Prediction:
42: End of no passing by vehicles over 3.5 metric tons
True label:
42: End of no passing by vehicles over 3.5 metric tons

Prediction:
16: Vehicles over 3.5 metric tons prohibited
True label:
16: Vehicles over 3.5 metric tons prohibited

Prediction:
35: Ahead only
True label:
35: Ahead only

Prediction:
33: Turn right ahead
True label:
33: Turn right ahead
```