

## DataSet Exploration

Here is the Dataset Summary from the notebook:

Training Set: 34799 samples  
Validation Set: 4410 samples  
Test Set: 12630 samples

Image data shape = (32, 32, 3)  
Number of classes = 43

## Preprocessing

I converted the images to grayscale and then scaled the images

$X_{\text{train\_gray\_nm}} = (X_{\text{train\_gray}}/128 - 1)$

## Model Architecture

Here is the convolutional neural network model I used.

*Layer 1: Convolutional(Filter size 5X5) Input = 32x32x1. Output = 28x28x20*

*Activation: ReLu*

*Max Pooling Input(Filter size 2X2) = 28x28x20. Output = 14x14x20.*

*Layer 2: Convolutional(Filter size 5X5). Input = 14x14x20 Output = 10x10x36.*

*Activation: ReLu*

*Max Pooling (Filter size 2X2). Input = 10x10x36. Output = 5x5x36*

*Layer 3: Fully Connected. Input = 900. Output = 360.*

*Activation: ReLu*

*Layer 4: Fully Connected. Input = 360. Output = 200.*

*Activation: ReLu*

*Layer 5: Fully Connected. Input = 200. Output = 43= $n_{\text{classes}}$ .*

I tried several other architectures, I initially started with a LeNet with lesser depth (6) in the initial layer. Increasing the number of filters(depth) to 20 and 36 helped with identifying higher level features and gave an improvement in validation accuracy. Final validation accuracy reached was 95.7%

I also tried adding another fully connected layer, used dropout of 0.5 in the last but one FC layer, but validation accuracy didn't improve much, so ended up removing it.

## Model Training

Separate train, valid and test data as given in the pickled data set were used.

I played with hyper parameters learning rate, epochs and batch size and the combination that gives the best result is

rate = 0.002 EPOCHS = 100 BATCH\_SIZE = 256

Also experimented with rate=0.001, but it takes a lot of time to converge. With .002 accuracy is abit jerky in the initial epochs but over time it becomes steady and gives the best performance.

Increasing/decreasing the batch size didn't seem to help much.

I used adam optimizer with default settings for optimization.

## Solution Approach

Using the parameters and the convnet described above, final metrics looks like follows for last few epochs:

```
EPOCH 97 ...
Training Accuracy = 1.000
Validation Accuracy = 0.957
```

```
EPOCH 98 ...
Training Accuracy = 1.000
Validation Accuracy = 0.957
```

```
EPOCH 99 ...
Training Accuracy = 1.000
Validation Accuracy = 0.957
```

```
EPOCH 100 ...
Training Accuracy = 1.000
Validation Accuracy = 0.957
```

---

```
Test Accuracy = 0.944
```

I downloaded few images for german traffic signals from net and tried to run them with the classifier.

Accuracy = 11/13= 84.6%

Last image is not in the training dataset, hence it fails to be identified. Last but one image (60 km/hr) is not identified properly as well, possibly because its a zoomed in image.