UDACITY: SELF DRIVING NANODEGREE

Project 2: Traffic Sign Classification

Prepared By: Rakesh Paul

15/05/2018

Project Goal:

The Goals/Steps of the project are the following:

- **Dataset Exploration**
- **Design and Test a Model Architecture**
- **Test a Model on New Images**

Dataset Exploration:

Basic Summary: Following are the basic information of the traffic Sign Dataset:

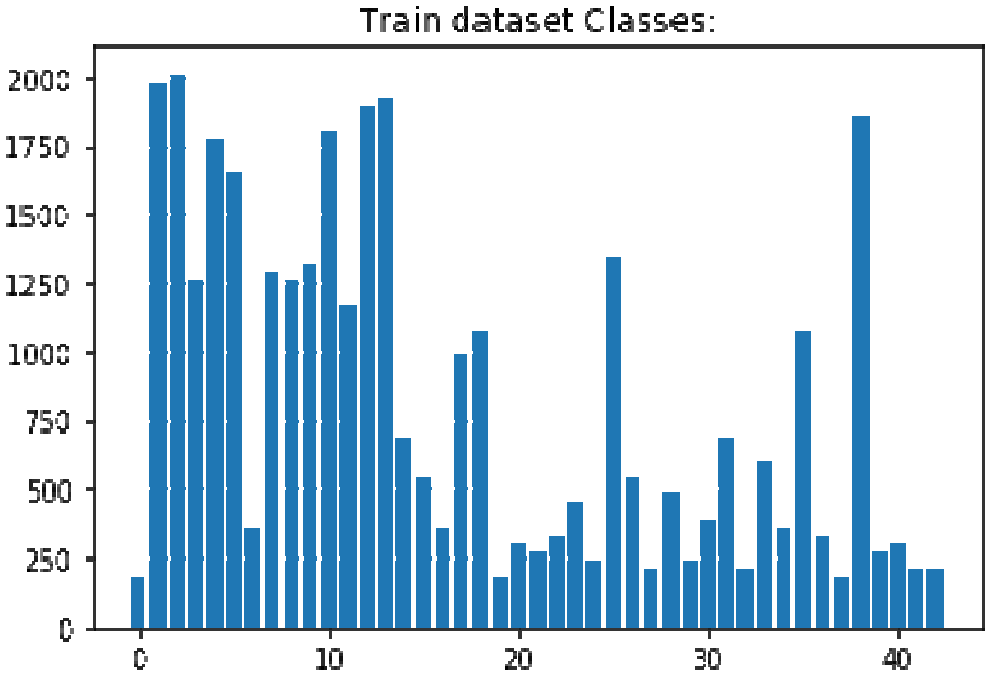
- `Number of training examples = 34799`
- `Number of testing examples = 12630`
- `Image data shape = (32, 32, 3)`
- `Number of classes = 34799`

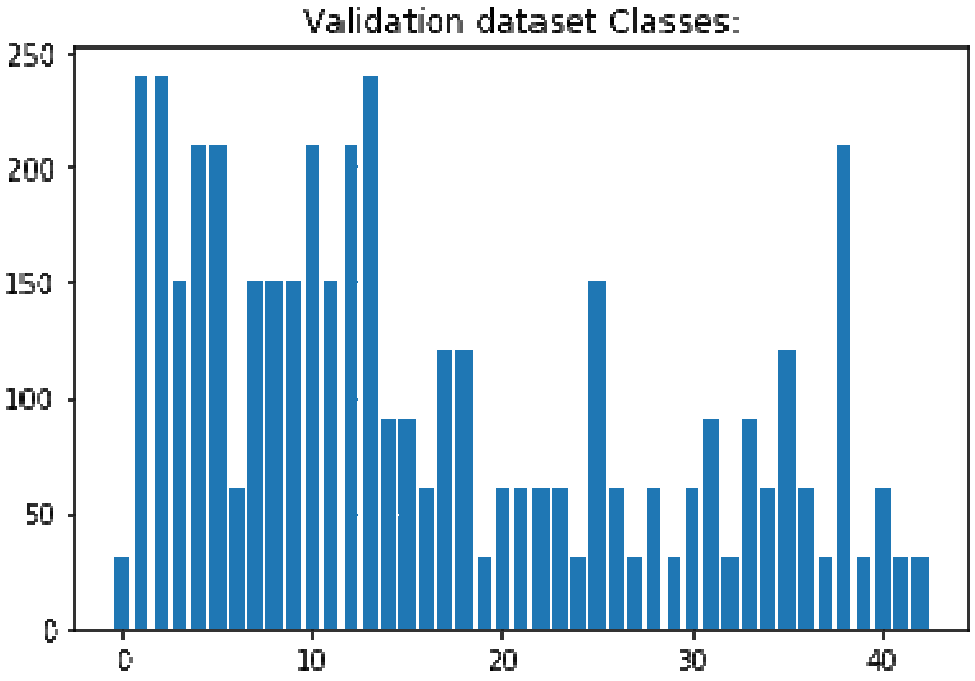
Exploratory Visualization: The visualization performed on the dataset are :

- Displaying 8 random images from the dataset.

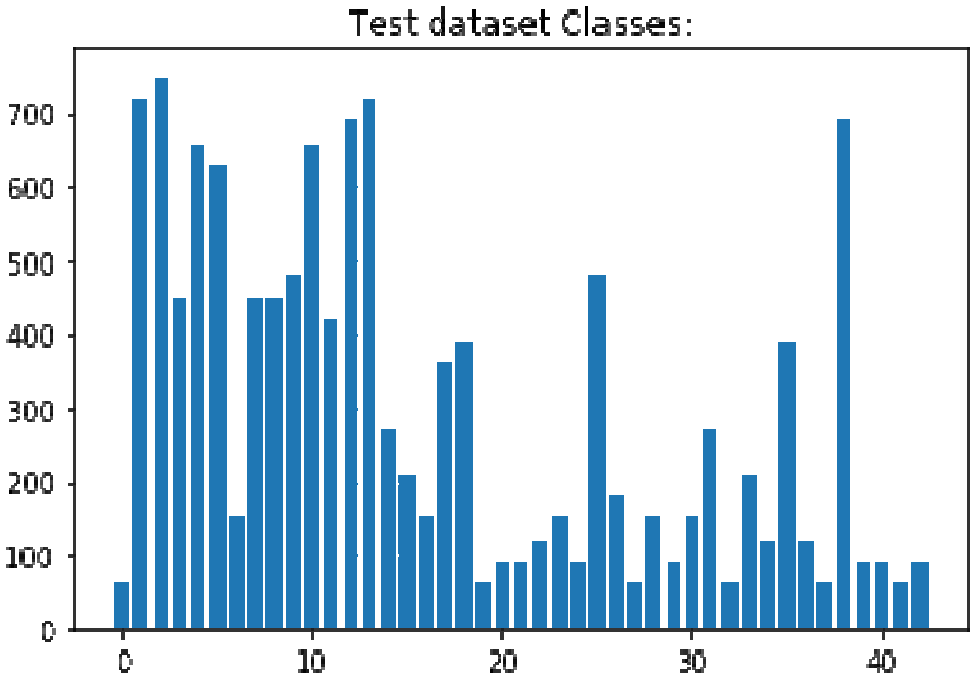
UDACITY: SELF DRIVING NANODEGREE

- Distribution of the 43 Labels in Train/Validation/Test Dataset





UDACITY: SELF DRIVING NANODEGREE



Design and Test a Model Architecture:

Preprocessing:

Convert to Grayscale: The Training set of images are first converted to grayscale as the LeNet architecture(on which the project is based upon) works best with grayscale images.

Normalize Images: The grayed images are then normalized to have 0 mean and equal variance for descending to low cost faster and in turn getting best weights and biases faster.

Parameters: mu = 0, sigma = 0.1

UDACITY: SELF DRIVING NANODEGREE

Model Architecture: The model used is as follows:

Functions	Input	Filter/weights	Output
Conv1	32x32x1	5x5x1x16	28x28x16
Max_pool	28x28x16	1x2x2x1	14x14x16
Conv2	14x14x16	5x5x16x32	10x10x32
Max_pool	10x10x32	1x2x2x1	5x5x32
flatten	5x5x32		1x800
Full connection	1x800	800x240	1x240
Full connection	1x240	240x168	1x168
Full connection	1x168	168x43	1x43

Training Model:

Following are the values used as training parameters:

- Epoch = 20
- Batch Size = 128
- Learning rate = 0.001
- Optimizer used = AdamOptimizer

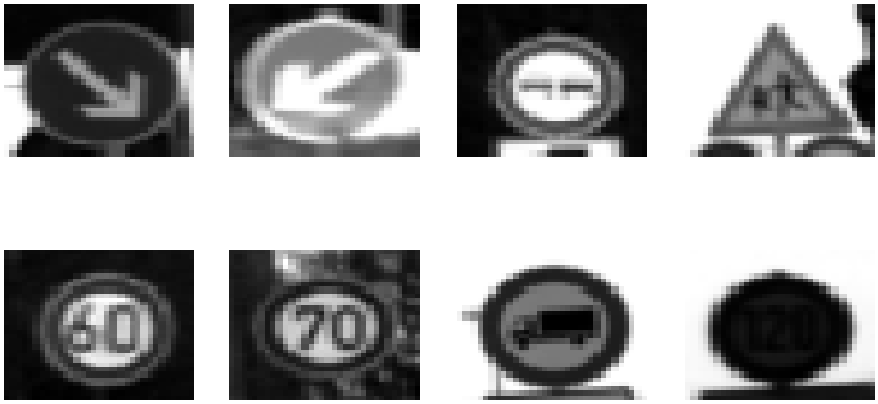
Convolution Neural Network is used to train the dataset. Probabilities for the logits are found using softmax function. These data is used as input to AdamOptimizer to find the minimum cost and correspondingly best weights and biases for testing.

Solution and Validation:

The approach followed for training this CNN is as follows:

- The images are first converted to grayscale as the LeNet architecture works better with grayscale image.

UDACITY: SELF DRIVING NANODEGREE



- The images are then normalized to have zero mean and equal variance which will help finding the minimal loss faster.

- Architecture modifications(w.r.t LeNet architecture):
    - The modified LeNet architecture for this project includes modified convolution kernels with larger output depth to extract more color channel information.
    - Dropout regularization technique is used for forcing the network learn redundant representation and in turn making it robust and prevent over-fitting
- AdamOptimizer is used for determining the best weights and labels for the dataset under consideration using learning rate of 0.001. Smaller learning rate slows down the gradient descent but prevents overshooting and helps to reach the minimum loss steadily.

UDACITY: SELF DRIVING NANODEGREE

Validation: The traffic sign dataset is split into training/validation/test datasets. This helps to have a more appropriate result about how the network will behave with datasets which are not used for training.

- Labels are one_hot encoded using tf.one_hot() function for better comparison.
- tf.argmax() function is applied on the predictions found using CNN for the validation dataset and also on the one-hot encoded labels. These are then checked for equality to find the number of matching outputs to find accuracy of CNN.
- The validation accuracy hence found is 0.944.
- Similar process is applied on test dataset. Test accuracy thus found is 0.923

Test a Model on New Images:

Loading new web Images: 6 german traffic sign images are downloaded from Internet for random testing.

The images are resized/cropped to fit 32x32 resolution, reason being LeNet architecture only support input resolution of 32x32xC. Here C is called the color depth. Moreover, using larger input resolutions, if used, will lead to very computationally intensive and very large neural network. This may need very high end processors with GPU's to compute the network.

As the original images were with very high resolution and had bigger background, the image quality of these images are downgraded and pixilated.

These images are then loaded and displayed.

UDACITY: SELF DRIVING NANODEGREE

Performance on New Images: The images are gray scaled and normalized using similar parameters as before and the test accuracy is 0.833. The resultant accuracy is drifted from the test accuracy found in the German dataset(0.923). The reasons as per my understanding are as follows:

- The training dataset include lesser number of images with label 3 as compared to label 1 image. This might have affected the result(currently label 3 image is predicted as label 1). Using dataset with more image of label 3 will surely improve the performance of the CNN.
- Due to the resizing, the quality of the images got degraded and pixilated. This might have affected the feature extraction on the particular image under consideration.

Softmax Probabilities: Softmax probabilities are calculated using tf.nn.top_k() and the results are as follows:

```
Softmax top 5 probability for Image 1 :
Probability 1 : (100.000%, label=14)
Probability 2 : (0.000%, label=38)
Probability 3 : (0.000%, label=2)
Probability 4 : (0.000%, label=13)
Probability 5 : (0.000%, label=15)

Softmax top 5 probability for Image 2 :
Probability 1 : (100.000%, label=25)
Probability 2 : (0.000%, label=20)
Probability 3 : (0.000%, label=30)
Probability 4 : (0.000%, label=21)
Probability 5 : (0.000%, label=23)
```

```
Softmax top 5 probability for Image 3 :
Probability 1 : (100.000%, label=38)
Probability 2 : (0.000%, label=23)
Probability 3 : (0.000%, label=3)
Probability 4 : (0.000%, label=34)
Probability 5 : (0.000%, label=13)

Softmax top 5 probability for Image 4 :
Probability 1 : (99.961%, label=23)
Probability 2 : (0.038%, label=11)
Probability 3 : (0.001%, label=19)
Probability 4 : (0.000%, label=21)
Probability 5 : (0.000%, label=30)

Softmax top 5 probability for Image 5 :
Probability 1 : (100.000%, label=27)
Probability 2 : (0.000%, label=11)
Probability 3 : (0.000%, label=18)
Probability 4 : (0.000%, label=24)
Probability 5 : (0.000%, label=28)

Softmax top 5 probability for Image 6 :
Probability 1 : (99.266%, label=1)
Probability 2 : (0.677%, label=31)
Probability 3 : (0.057%, label=2)
Probability 4 : (0.000%, label=5)
Probability 5 : (0.000%, label=3)
```

Model Certainity: As per the softmax probabilities, 5 images from new dataset are 100% accurately classified. 6th image is wrongly classified(correct label is 3). Therefore based on 6 images, the model is 83.3% certain.