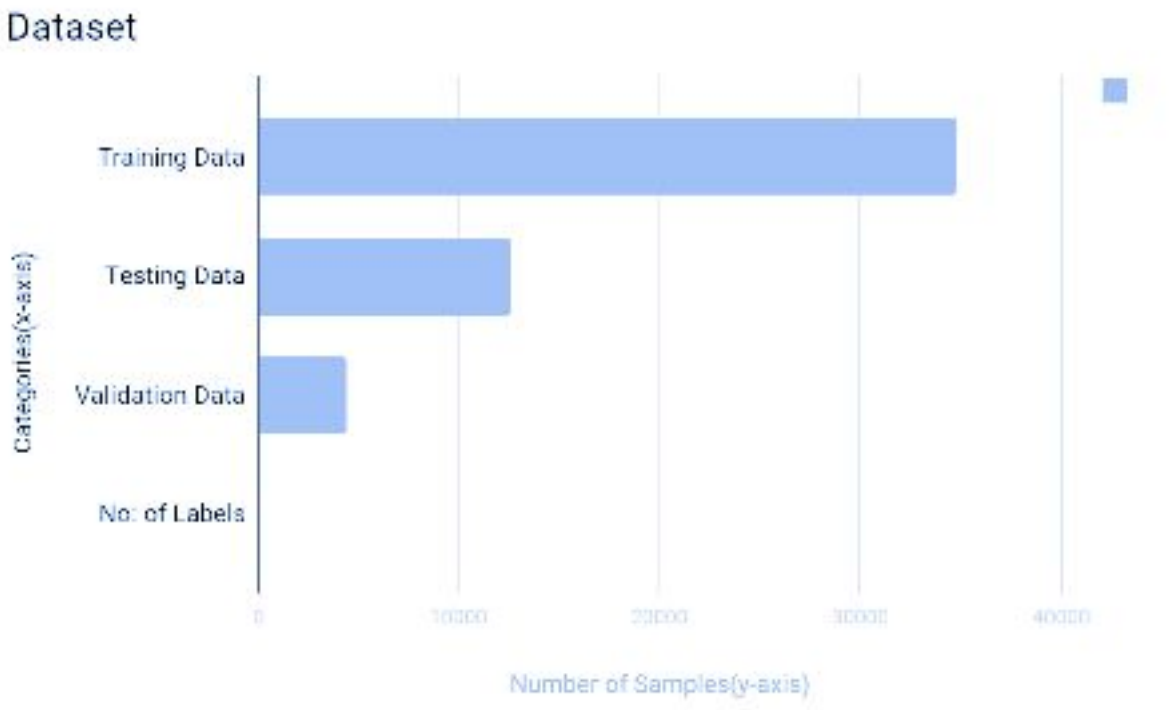
Project-2
Traffic-Sign-Classifier Project1

Dataset Summary

The dataset consists of three files. It has training file, testing file and validation file.

- The size of samples in the training set is 34799 with a dimension of 32*32*3 images
- The size of samples in the validation set is 4410 with dimension of 32*32*3 images
- The size of samples in the test set is 12630 with dimension of 32*32*3 images
- The shape of a traffic sign image is 32*32*3
- The number of unique classes/labels in the dataset is 43

Visualization of the dataset



Here the number of classes/labels is 43. The x-axis in the chart represents the samples and the y-axis represents the number of samples. The training set have the highest amount of sample (34799) . Secondly, the testing set have samples of 12630. Thirdly, the validation set have samples of 4410.

Design and Test a Model Architecture

1. As a first step, i decided to convert the images into grayscale because converting a RGB image into grayscale helps in the deep analysis of the image. It can identify the shape and converting it to vectors. During the conversion to grayscale, the image dimension will change from 32*32*3 to 32*32*1. The technique i used here is:

- For Training Set:

  X_train_gry = np.sum(X_train/3, axis=3, keepdims=True)

- For Validation Set:

  X_valid_gry = np.sum(x_valid/3,axis=3,keepdims=True)

- For Testing Set:

  X_test_gry = np.sum(x_test/3, axis=3, keepdims=True)

Example of traffic sign image before and after grayscaling:

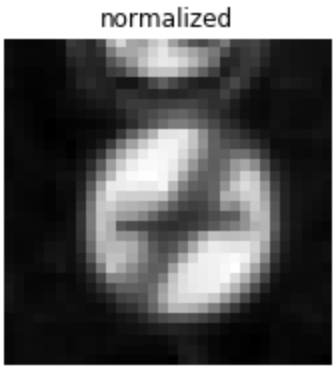
As a last step, I normalized the image data because it helps to reduce the size and to prevent the duplication of data. The normalization technique i used is:

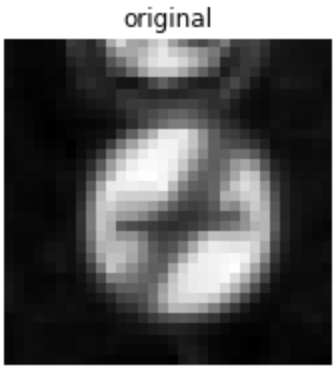
X_train_normalized = (x_train -128)/128

x_test _normalized = (x_test -128)/128

x_valid_normalized = (x_valid-128)/128

Example of original and augmented image:





2.

Layer	Description
Input	32*32*1 RGB image(Preprocessed Image)
Convolution Layer-1 (5*5*1),Output Depth-6	1*1 stride, Valid padding, outputs 28*28*6
RELU	Activation
Max pooling-1	2*2 stride,outputs 14*14*6
Convolutional Layer-2 (5*5*6),Output Depth-16	1*1 stride, Valid padding, outputs 10*10*16
RELU	Activation
Max pooling-2	2*2 stride,outputs 5*5*16
Flatten	400 parameters
Dropout-1	Keep_prob =(0.6 for training set and 1 for validation set)
Fully Connected Layer -1	Shape = (400,120)
RELU	Activation
Dropout-2	Keep_prob =(0.6 for training set and 1 for validation set)
Fully Connected Layer -2	Shape = (120,84)

RELU	Activation
Dropout-3	Keep_prob =(0.6 for training set and 1 for validation set)
Fully Connected Layer -3	Shape = (120,84)
RELU	Activation
Dropout-3	Keep_prob =(0.6 for training set and 1 for validation set
Softmax	tf.nn.softmax_cross_entropy_with_logits()

My model consist of two convolutional layer and three fully connected layer. In the first convolutional layer, the input image with dimension 32*32*1(grayscale) is converted into 28*28*6 with valid padding and filter weight of 5*5*6. The relu activation function is used for the activation and max pooling with stride of 2*2 is used which will make the dimension as 14*14*6. In the second convolutional layer, the image with dimension 14*14*6 is converted into 10*10*16 with valid padding and filter weight of 5*5*16. Then a relu activation function is used for the activation and max pooling is done with stride of 2*2 with valid padding which will make the dimension as 5*5*16. By using flatten we reformat the second convolutional layer and dropout is done which will remove the connection with zero nodes. Atlast we fully connect the layer three time until we reach with a output dimension of shape(120,84).

3.To train the model two tensors will be initialized as x and y with x for the input and y for the labels as one hot tensor. Through the the Lenet model, logits will be outputted and we use this logits with tf.nn.softmax_cross_entropy_with_logits() which will convert the logits to one hot encoded labels or scores. Using the Adam Optimizer algorithm we find the minimum error between each scores and minimize them to the maximum. During training, a training set of images will the passed as input to the lenet which will find the error. When validating we find can the accuracy of the set with the training set. All this be done as batches. At last we get the total accuracy of the set. After doing all this thing we test the set with new images.
To train the model, i used the following parameters of value:

1. BATCH_SIZE = 150
2. EPOCHS = 100
3. Mean(mu) = 0
4. Variance(sigma) =0.1
5. Dropout(keep_prob) = 0.6 for the training set and 1 for the validation set

6. Learningrate = 0.0009

4. The steps which i took to improve the accuracy was by increasing the size of batches to 150, epochs to 100 and by the dropout value of 0.6 for the training set and 1 for the validation. I used the dropout in between every fully connected layer in order to remove the zero valued node.

My final model results were:

- Training set accuracy of 93.66%
- Validation set accuracy of 93.66%
- Test set accuracy of 40%

- What was the first architecture that was tried and why was it chosen?

The first architecture i tried was the lenet itself. It was chosen as it is an image classifier project. The lenet architecture work well on the image classification.

- What were some problems with the initial architecture?

Lot of layers is needed in order to classify the image well. Here we need two convolutional layer and three fully connected layers.

- How was the architecture adjusted and why was it adjusted?

The architecture was adjusted with two convolutional layer and three fully connected layers. In between each convolutional layer we use an activation function and max pooling with size of 5*5 and stride of 2*2. After the convolutional layer we use the dropout in order avoid some zero nodes and to improve the accuracy of the set.

- Which parameters were tuned? How were they adjusted and why?

Adjusting the dropout helped me to acquire an accuracy of 93.66%. During training the dropout has a value of 0.6 and during validation it has a value of 1.

- What are some of the important design choices and why were they chosen? For example, why might a convolution layer work well with this problem? How might a dropout layer help with creating a successful model?

The important design choices are the convolutional layer helps in extracting the image and helps to identify the shapes and size of the images. The dropout layer helps to avoid the unnecessary nodes and helps in improving the accuracy of the model.

- What architecture was chosen?

I do not know more architectures but i have heard that the AlexNet would help in image recognitions and object recognitions.

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

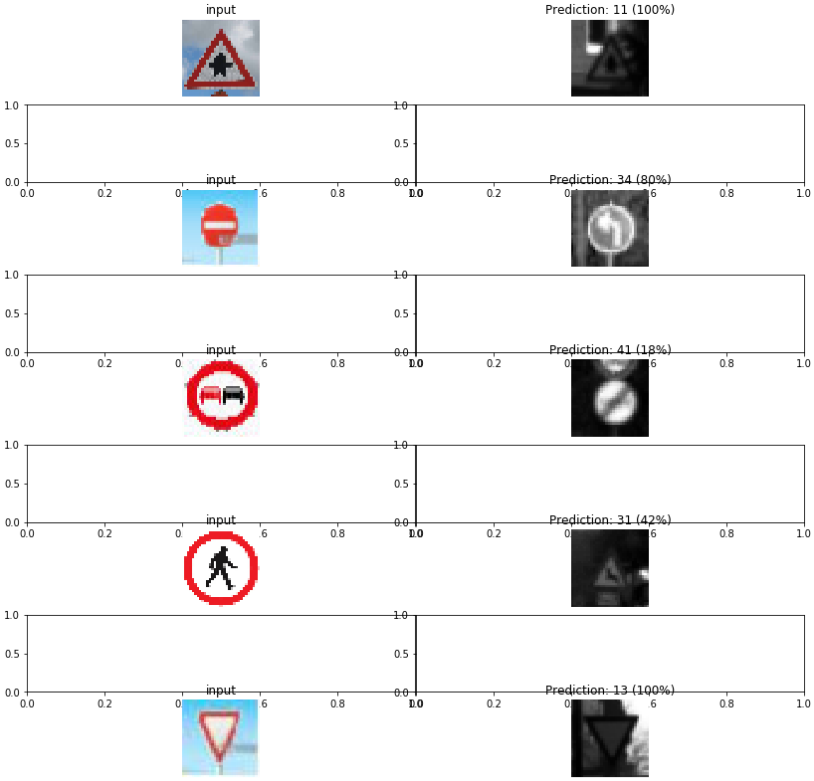
The traffic signs are similar in every country, for example if there is a board showing 30 as number it means that we should travel only by a speed of 30 km/h. So the model trained can be used in different nations will work well expect in some small cases.

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

During training and validation, the set got an accuracy of 93.66% which means the model has an accuracy of 93.66%.. When testing it with new image it can identify the images and recognize them as those image that we have trained. This can clearly state that model is working.

Test a Model on New Images

1. Here are five German traffic signs that i found on the web:



- The fourth image was difficult to classify may be because of the unfamiliar image present in the set.

2.

Image	Prediction
Right-of-way at the next intersection	Right-of-way at the next intersection
No entry	Turn left ahead
No passing	End of no passing
Pedestrians	Wild animals crossing
Yield	Yield

The model predicted 4 out of 5 which gives a test set accuracy of 80%. The accuracy on the test set of first, second, third and fifth were predicted well.

3.

Probability	Prediction
1	Right-of-way at the next intersection
0.34	Turn left ahead
0.18	End of no passing
0.42	Wild animals crossing
1	Yield

The prediction went wrong on the fourth. This may be because of unfamiliar image in the training set.