Traffic Sign Recognition and Classification

The goals / steps of this project are the following:

* Load the data set (see below for links to the project data set)

* Explore, summarize and visualize the data set

* Design, train and test a model architecture

* Use the model to make predictions on new images

* Analyze the softmax probabilities of the new images

* Summarize the results with a written report

Provide a Writeup / README that includes all the rubric points and how you addressed each one. You can submit your writeup as markdown or pdf. You can use this template as a guide for writing the report. The submission includes the project code.

The project submission includes the following files:

- Ipython notebook with code
- HTML output of the code
- PDF report detailing the rationale behind the current code (currently reading :))
- Images from the web
- Trained Neural network

Data Set Summary & Exploration

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.

The German Traffic sign classifier dataset is a relatively self-sufficient data set to understand the basics of building a neural network. It was a lot of fun working on this project and learning the different methods of building and creating a neural network.

As a starting point, the brief set of information (as requested in the rubric) are described as under:

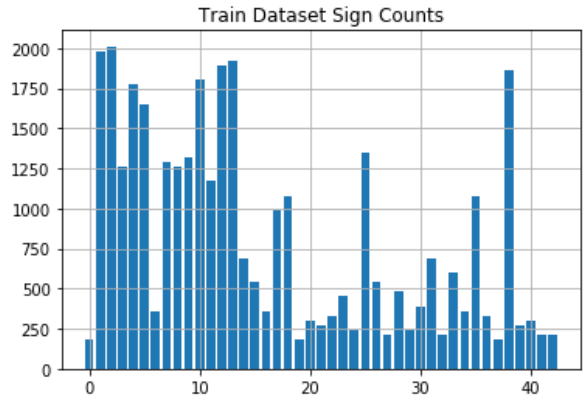
- The size of training set is ***34799***
- The size of the validation set is ***4410***
- The size of test set is ***12630***
- The shape of a traffic sign image is ***32x32x3***
- The number of unique classes/labels in the data set is ***43***

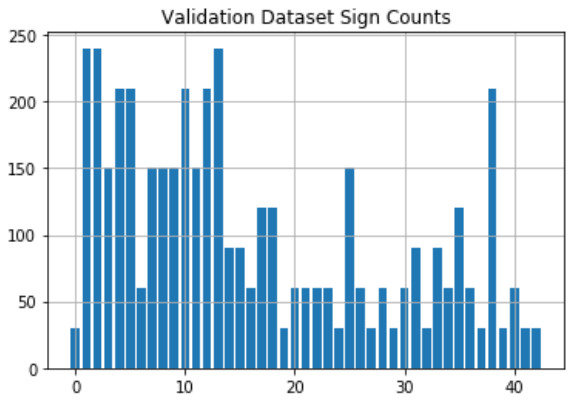
Include an exploratory visualization of the dataset

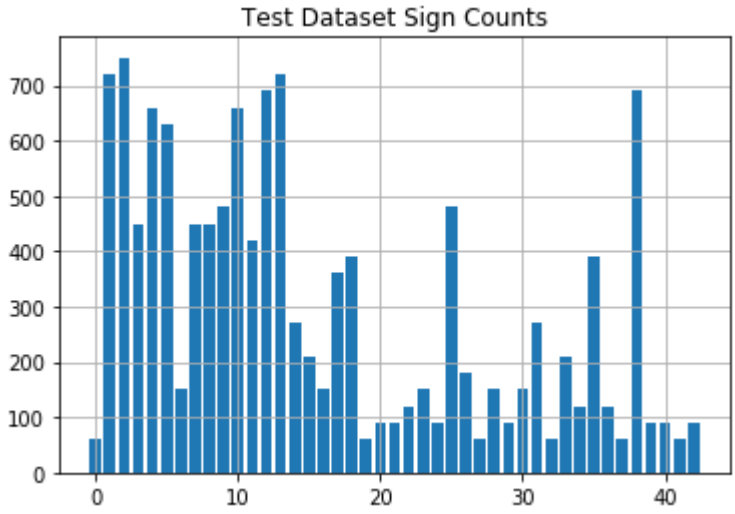
Data set is already classified into training, validation and testing set of pre-determined sizes. However, there is no restriction to use the initial given data set and new data can be added based on initial network performance and other factors. A section on further improvements to the performance of the neural network specifies these points.

Data set consists of several thousands of images which fall under the 43 different classes. Some of them are:

A further deep dive was performed to understand the distribution of images in the different data sets. It is depicted as:







Based on the plots, we can understand that the distribution across the training test and validation data sets are similar. However, the distribution across different classes of images is not uniform. The following statistics represents the 5 most common and 5 least common of classes in the training data set:

```
Most common five:
  5.776%, 2010 images - Speed limit (50km/h)
  5.690%, 1980 images - Speed limit (30km/h)
  5.517%, 1920 images - Yield
  5.431%, 1890 images - Priority road
  5.345%, 1860 images - Keep right

Least common five:
  0.603%, 210 images - End of no passing
  0.603%, 210 images - End of no passing by vehicles over 3.5 metric tons
  0.517%, 180 images - Speed limit (20km/h)
  0.517%, 180 images - Dangerous curve to the left
  0.517%, 180 images - Go straight or left
```

This information would be used in the final section of the project to understand how the model performs in detecting images from the web.

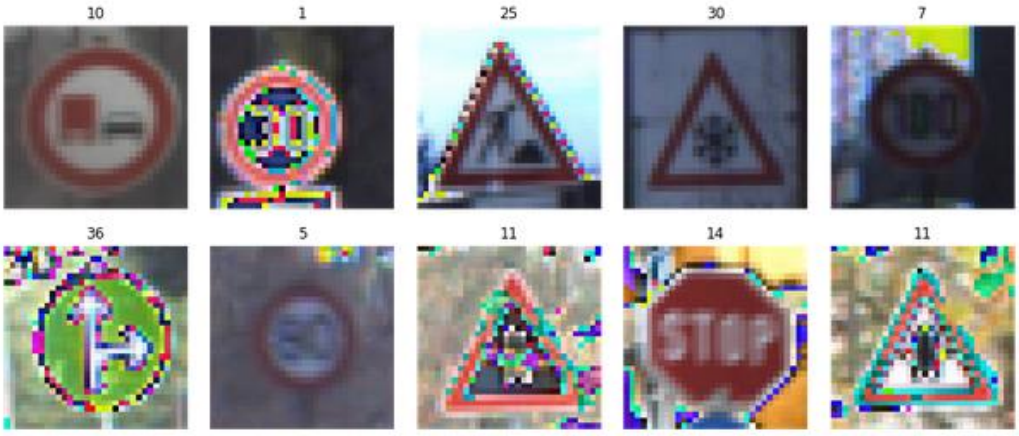
Design and Test a Model Architecture

Describe how you preprocessed the image data. What techniques were chosen and why did you choose these techniques? Consider including images showing the output of each preprocessing technique. Pre-processing refers to techniques such as converting to grayscale, normalization, etc.

The images were pre-processed in 2 steps:

- Normalization of images

  RGB images vary from [0 255] and this might cause the optimizer to not converge to a good solution. This occur s because mean and variance of different layers across different images is not consistent. To bring the mean and variance to be in the same range of values, normalization is performed.



- Gray scaling of images

  Colors in the traffic sign are important in real world for people to recognize different signs. However, the most important thing is the shape and the contents of the traffic sign. This is also based on the dataset which has different distinct shapes and contents and hence the different signs can be learned with gray-scaled images as well.

[***Note***: Augmenting images was not performed since the neural network performance was within the required 0.93 validation accuracy. However, it is discussed in further in further improvements section.]

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.

The final model consists of the following layers:

Layer	***Description***
Input	32x32x1 gray-scaled image
Convolution 5x5	1x1 stride, Valid padding, output 28x28x6
RELU	Activation Layer
Dropout	Dropout layer after every activation to understand overfitting trends
Max Pooling	2x2 stride, output 14x14x6
Convolution 5x5	1x1 stride, Valid padding, output 10x10x16
RELU	Activation Layer
Dropout	Dropout layer after every activation to understand overfitting trends
Max Pooling	2x2 stride, output 5x5x16
Flatten	Flattening layer, output 400
Fully Connected	Output 120

RELU	Activation Layer
Dropout	Dropout layer after every activation to understand overfitting trends
Fully Connected	Output 84
RELU	Activation Layer
Dropout	Dropout layer after every activation to understand overfitting trends
Fully Connected	Output 43

The LeNet5 architecture was initially used as a starting point to understand and dropout layers were added to understand concerns of overfitting to training data.

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.

Adam optimizer was used in the training of the model. The batch size and the number of epochs were tuned through multiple iterations.

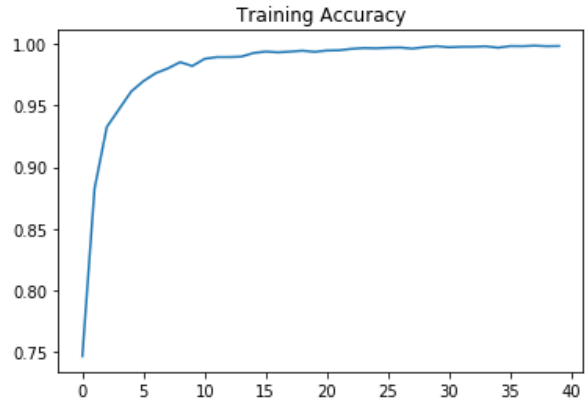
The hyper parameters available based on the above described model architecture was:

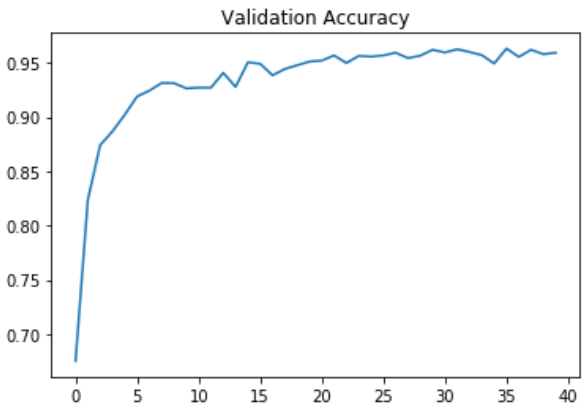
1. ***Epochs***: The number of epochs represent the number of forward and backward passes for optimization of weights. This value was initially started as 10 epochs and increased in steps based on observations to around 40.
2. ***Batch size***: Mini batching tuning parameter to form subsets to reduce training time. This parameter was not used for fine tuning and was set to 128.
3. ***Learning Rate***: Learning rate being an important parameter mentioning how fast the weights can change in a single step. This parameter is very sensitive, hence not tuned much and was set to 0.001.
4. ***Dropout Probability***: Parameter to tune which mentions probability of a node being dropped out. This was initially set at 0.5 and tuned to 0.75. This layer was introduced to understand any signs of overfitting and to improve robustness of the model.

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 **0.998**
- Validation set accuracy of **0.959**
- Test set accuracy of **0.939**





The model is developed based on the well-known LeNet Architecture. The LeNet architecture has a great performance on recognizing handwritings. LeNet Architecture's simplicity and the flexibility to modify the architecture was the convincing reason for using the architecture in the traffic sign application.

The steps involved in arriving at these results were:

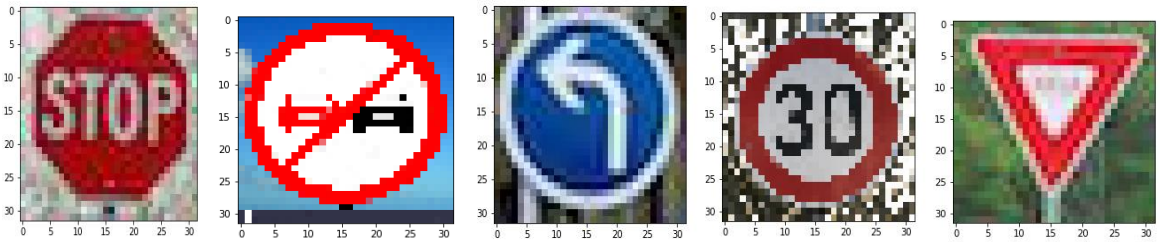
1. The basic LeNet architecture was initially implemented with no normalization and grayscaling – Accuracy values were observed to be around **89** % in validation data set.
2. To make sure the data being provided is well conditioned, normalization was implemented- Accuracy increased to **93** % in the validation data set.
3. High training accuracy with lower validation accuracy introduced concerns of overfitting of the data. At this stage, dropout layers were added to alleviate these concerns – Accuracy values also increased to **95**%
4. Grayscaling of images was also done to introduce no variations with respect to color of the recognized images- Accuracy values increased slightly to ~**96**%

Using the graphs described above, we can further optimize the epoch hyper parameters. This shows the variation in accuracy over the number of epochs.

The test accuracy is around ~94% which is significantly good with the current implemented model.

Test a Model on New Images

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.



The above plotted images represent the 5 images downloaded from the web and converted to 32x32 image. The 5 images are:

1. **Stop sign** - Based on section 1, Not belonging to either most or least common images
2. **End of No passing** - Based on section 1, this is one of the least common five images available in training data set
3. **Left Turn** - Based on section 1, Not belonging to either most or least common images
4. **30 km/h** - Based on section 1, this is one of the most common five images available in training data set
5. **Yield Sign** - Based on section 1, this is one of the most common five images available in training data set

The images used in this classification are semi-easy in a way that most of the images have the signs focused in the image. There are no images with rotated/ translated traffic signs or images with more noise which would make it difficult to classify.

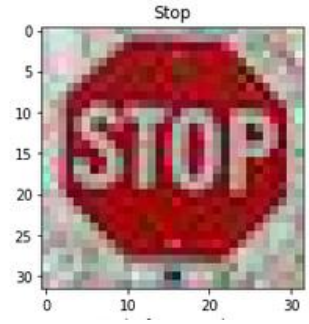
Another important consideration while choosing these images was to make sure they were chosen to have adequate information when converted to 32x32 pixels. There were certain images encountered during this project where the information is lost from the image when compared to 32x32. This would make it difficult to classify into appropriate classes.

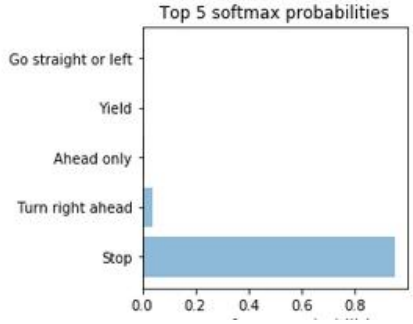
Discuss the model's predictions on these new traffic signs and compare the results to predicting on the test set. At a minimum, discuss what the predictions were, the accuracy on these new predictions, and compare the accuracy to the accuracy on the test set (OPTIONAL: Discuss the results in more detail as described in the "Stand Out Suggestions" part of the rubric).

Image	*Prediction*
Stop sign	**Stop sign**
End of No passing	**End of No passing**
Left Turn	**Keep Right**
30 km/h	**30 km/h**
Yield Sign	**Yield Sign**

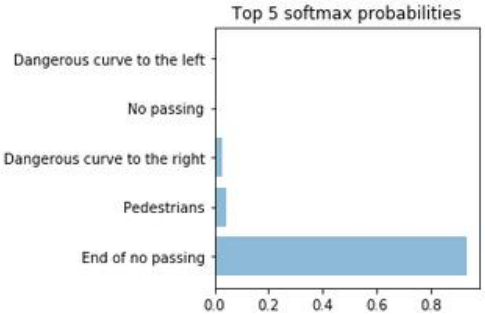
The model was able to correctly guess 4 of the 5 traffic signs, which gives an accuracy of 80%. This compares considerably well the test set accuracy to be 93.9%. The accuracy concerns can be addressed with the implementation of further improvements section.

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)

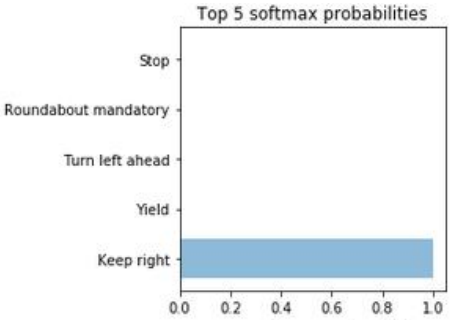

Stop


Top 5 softmax probabilities
Go straight or left
Yield
Ahead only
Turn right ahead
Stop

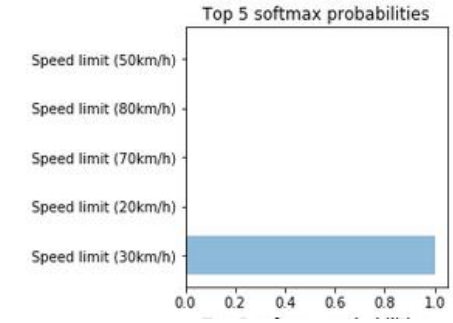

End of no passing

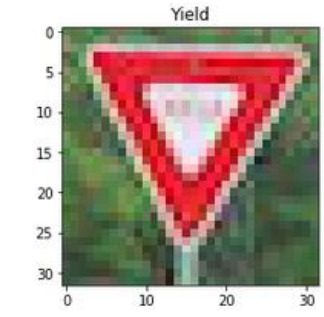

Top 5 softmax probabilities
Dangerous curve to the left
No passing
Dangerous curve to the right
Pedestrians
End of no passing

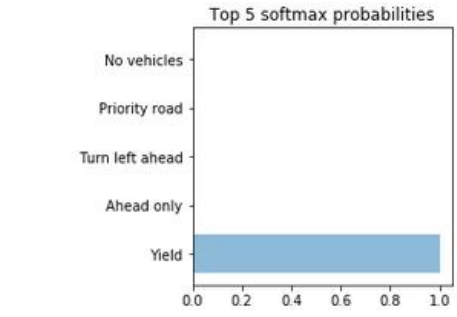

Turn left ahead


Top 5 softmax probabilities
Stop
Roundabout mandatory
Turn left ahead
Yield
Keep right


Speed limit (30km/h)


Top 5 softmax probabilities
Speed limit (50km/h)
Speed limit (80km/h)
Speed limit (70km/h)
Speed limit (20km/h)
Speed limit (30km/h)


Yield


Top 5 softmax probabilities
No vehicles
Priority road
Turn left ahead
Ahead only
Yield

The above images show the softmax top 5 probabilities of the predictions.

Shortcomings & Further Improvements

There are several potential shortcomings of the current implementation:

1. Rotated/Translated images of same sign can become tricky to handle with the current implementation.
2. Number of examples across all classes is not uniform and this can introduce error in classification
3. Other well-known traffic sign implementation network architectures can produce better results

These are some improvements to the neural network that could improve accuracy of the traffic sign classification:

1. More images can be added to the data set to make them uniform across classes.
2. The added images can represent the same signs from different perspective such as rotated, translated and images with more noise to make the prediction more robust.