Rubric Points
###Here I will consider the [rubric points](https://review.udacity.com/#!/rubrics/481/view) individually and describe how I addressed each point in my implementation.

###Writeup / README

####1. 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.

Answer:

This is my writeup file and all the required files are attached in the zip file along with it.

Main files

writeup_report.pdf (this file).

Traffic_Sign_Classifier.ipynb

Report.html

###Data Set Summary & Exploration

####1. Provide a basic summary of the data set and identify where in your code the summary was done. In the code, the analysis should be done using python, numpy and/or pandas methods rather than hardcoding results manually.

The code for this step is contained in the second code cell of the IPython notebook.

Answer (code is in the 2nd code cell of IPython notebook):

I used the python functions to calculate summary statistics of the traffic signs data set:

(copying the output below)

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

####2. Include an exploratory visualization of the dataset and identify where the code is in your code file.

The code for this step is contained in the third code cell of the IPython notebook.

Answer(code is in the 3rd code cell of IPython notebook):

1) one example for each label is displayed for the visualization of the data.

2) A bar chart is shown to illustrate the no. of training examples available for each label.

###Design and Test a Model Architecture

####1. Describe how, and identify where in your code, you preprocessed the image data. What tecniques 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 code for this step is contained in the fourth code cell of the IPython notebook.

Answer(code is in the 4th code cell of IPython notebook):

Pre-processing steps

1) Training data is shuffled because training data for each label are clusterred together, as during training batches this could cause overfitting problem.

2) After looking at the plots of the data, color doesn't seem to help in classification and as it might complicate the classification, converted the images to gray.

3) For better classification rate, histogram equalization is performed on the data.

4) All the data is normalized, as un-normalized data can take longer to converge.

####2. Describe how, and identify where in your code, you set up training, validation and testing data. How much data was in each set? Explain what techniques were used to split the data into these sets. (OPTIONAL: As described in the "Stand Out Suggestions" part of the rubric, if you generated additional data for training, describe why you decided to generate additional data, how you generated the data, identify where in your code, and provide example images of the additional data)

Answer(code is in the 1st code cell of IPython notebook while loading the data):

1) I have used the same classification as the data set provided.

Training set : 34799

Validation set : 4410

Test set : 12630

####3. Describe, and identify where in your code, 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.

Answer(code is in the 5th code cell of IPython notebook):

1) I have used a model similar to Lenet.

My final model consisted of the following 5 layers:

Input : 32x32x1 - grayscale image

Layer 1 :

```
1) convolution layer - 5x5x1x15 - output size: 28x28x15
     stride - [1,1,1,1], padding = valid
2)Activation function - relu
3) Pooling - 2x2 - output size:14x14x15
```

Layer 2 : convolution layer

```
1) convolution layer - 5x5x15x30 - output size: 10x10x30
     stride - [1,1,1,1], padding = valid
2)Activation function - relu
3) Pooling - 2x2 - output size:5x5x30
```

Flattening - output: 750

Layer 3 : Fully connected layer + activation function(relu). output:400

Layer 4 : Fully connected layer + activation function(relu). output:200

Layer 5 : Fully connected layer. output:43

####4. Describe how, and identify where in your code, 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.

Answer(code is in the 5th,6th,7th code cell of IPython notebook):

1) Adamoptimizer is used similar to Lenet architecture.

2) Learning rate = 0.001 (did it vary the value around finally settled with 0.001).

3) mu = 0, as it is better to have a zero-mean, didn't change the value.

4) sigma = 0.1, tested with few other values, finally settled with 0.1 based on the test accuracy.

5) Batch_size = 128.

6) Epochs = 10 (as the accuracy not increasing, even if the epochs are increased at the current learning rate, settled with 10).

####5. Describe the approach taken for finding a solution. 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.

Answer(code is in the 7th code cell of IPython notebook):

1) Most of the network variables are decided based on trail and error.
2) Lenet architecture is selected as it also performance image classification task.
3) No. of convolution layers and fully connected layers are experimented by increasing and decreasing by one layer and the test accuracy is used as a metric to settle with the current network.
4) Same experiment is performed with no. of nodes in each layer and current nodes are selected based on test accuracy.

Final accuracy:

Test accuracy = 91.9

Validation accuracy = 94.6

Training accuracy = 99.8

###Test a Model on New Images

####1. 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.

Answer(code is in the 8th code cell of IPython notebook):

Following 5 images are downloaded from the web for this exercise

1)http://i.istockimg.com/file_thumbview_approve/5240692/3/stock-photo-5240692-roundabout-traffic-sign-spain.jpg

2)http://bicyclegermany.com/Images/Laws/Arterial.jpg

3)http://a.rgbimg.com/cache1nHmS6/users/s/su/sundstrom/300/ mifuUb0.jpg

4) http://bicyclegermany.com/Images/Laws/100_1607.jpg

5)https://www.allstate.com/resources/Allstate/images/tools-resources-articles/car/handling-a-deer-accident.jpg

I predicted that final image might be difficult for the network to predict as the wild animal sign looks a bit different from what we have in the dataset. But I wanted to see how the network performs.

While loading the images itself, I am converting them to grayscale and resizing them to 32x32 as needed. And the final images are displayed in the python notebook.

####2. Discuss the model's predictions on these new traffic signs and compare the results to predicting on the test set. Identify where in your code predictions were made. 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).

Answer(code is in the 9th, 10th code cell of IPython notebook):

web test images accuracy: 80

Actual image	Predicted as
Roundabout mandatory	Roundabout mandatory
Priority road	Priority road
Road work	Road work
Right-of-way at the next intersection	Right-of-way at the next intersection

Wild animals crossing   Right-of-way at the next intersection

The model was able to correctly guess 4 of the 5 traffic signs, which gives an accuracy of 80%. This compares favorably to the accuracy on the test set of 90%

As expected the final wild animal image was wrongly classified, as this image looks significantly different from the ones available in the dataset used during training.

####3. Describe how certain the model is when predicting on each of the five new images by looking at the softmax probabilities for each prediction and identify where in your code softmax probabilities were outputted. 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)

Answer(code is in the 11th code cell of IPython notebook):

Top five predictions for each test image are as follows and a bar graph is drawn for each test image in the ipython notebook.

1) actual image: Roundabout mandatory
    Top 5 pridictions

  prediction 1 Roundabout mandatory
  prediction 2 Vehicles over 3.5 metric tons prohibited
  prediction 3 Speed limit (100km/h)
  prediction 4 No passing
  prediction 5 Priority road

This is a good prediction as rest of the probabilities are quite lower and there is no confusion in the decision

2) actual image: Priority road
    Top 5 pridictions

  prediction 1 Priority road
  prediction 2 No passing
  prediction 3 Roundabout mandatory
  prediction 4 Vehicles over 3.5 metric tons prohibited
  prediction 5 End of no passing by vehicles over 3.5 metric tons

This is a good prediction as rest of the probabilities are quite lower and there is no confusion in the decision

3) actual image: Road work
    Top 5 pridictions

  prediction 1 Wild animals crossing

```
  prediction 2 Road work
  prediction 3 Dangerous curve to the left
  prediction 4 Double curve
  prediction 5 Bicycles crossing
```

Even though this is a wrong prediction, right label is in close proximity. This might be because of how similar the close one looks.

4) actual image: Right-of-way at the next intersection

Top 5 pridictions

```
  prediction 1 Right-of-way at the next intersection
  prediction 2 Dangerous curve to the left
  prediction 3 Road narrows on the right
  prediction 4 Double curve
  prediction 5 General caution
```

This is a good prediction as rest of the probabilities are quite lower and there is no confusion in the decision

5) actual image: Wild animals crossing

Top 5 pridictions

```
  prediction 1 Wild animals crossing
  prediction 2 Roundabout mandatory
  prediction 3 Speed limit (50km/h)
  prediction 4 Double curve
  prediction 5 Speed limit (30km/h)
```

Even though this is a correct prediction there is one more label in close proximity. This might be because of how similar the close one looks.