Traffic Sign Recognition

Writeup

You can use this file as a template for your writeup if you want to submit it as a markdown file, but feel free to use some other method and submit a pdf if you prefer.

Build a Traffic Sign Recognition Project

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

Data Set Summary &Exploration

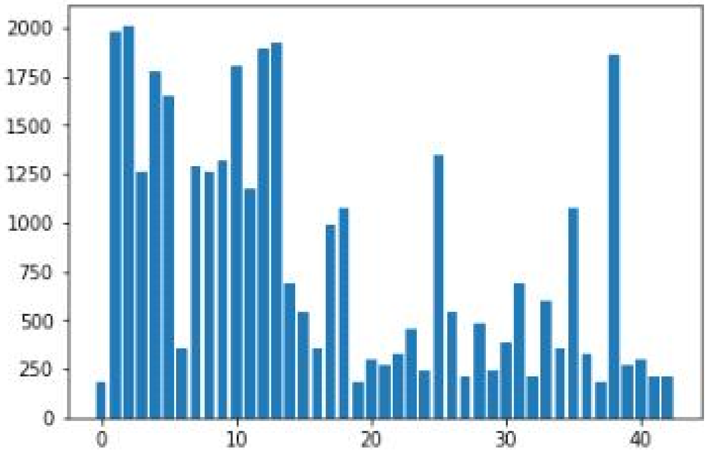
1. Provide a basic summary of the data set. In the code, the analysis should be done using python, numpy and/or pandas methods rather than hardcoding results manually.

I used the pandas library to calculate summary statistics of the traffic signs data set:

- 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

2. Include an exploratory visualization of the dataset.

Here is an exploratory visualization of the data set. It is a bar chart showing how the data



Design and Test a Model Architecture

1. 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. (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, and provide example images of the additional data. Then describe the characteristics of the augmented training set like number of images in the set, number of images for each class, etc.)

As a first step, I decided to normalize the images from [0, 255] to [0, 1]. If the pixel value of the image is large, the value may not be accurate when internalizing or calculating other things

An image is initially 784 features. We reshape this to (28, 28, 3), and pad the image with 0s such that the height and width are 32.

Thus, the input shape going into the first convolutional layer is 32x32x3.

2. 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.

My final model consisted of the following layers:

Layer		Description
	Input	32x32x3 RGB image
Layer1	Convolution	1x1 stride, valid padding, outputs 28x28x6
	RELU	
	Max pooling	2x2 stride, outputs 14x14x6
Layer2	Convolution	1x1 stride, valid padding, outputs 10x10x16
	RELU	
	Max pooling	2x2 stride, outputs 5x5x16
	Flatten	Input = 5x5x16. Output = 400
Layer3	Fully connected	Input = 400. Output = 120
	RELU	
Layer4	Fully connected	Input = 120. Output = 84
	RELU	
Layer5	Fully connected	Input = 84. Output = n_classes

3. 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.

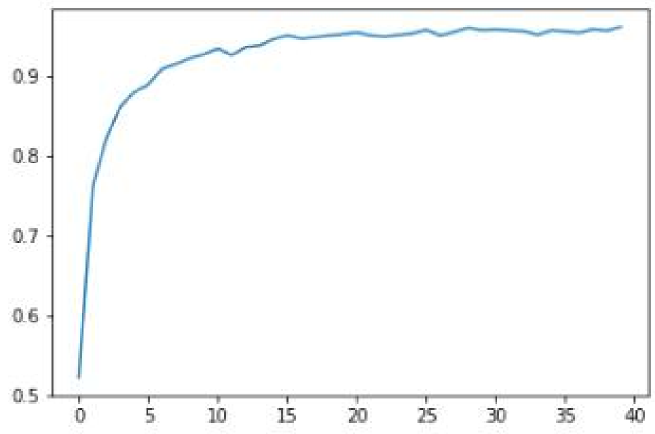
To train the model, I used Adam Optimizer. Adam Optimizer uses the Adam algorithm to minimize loss function as stochastic gradient descent does. Adam Optimizer is a little more sophisticated than stochastic gradient descent.

The EPOCH and BATCH_SIZE values affect the training speed and model accuracy. And, In general, the more EPOCHS there are, the better our model needs to train, and the longer the training will be required. The larger the deployment size, the faster our model will train. So, I set the batch size to 128, and set the epochs value to 40. And

As a result of experimenting with this pipeline, I set the learning rate to the appropriate value of 0.0008.

4. 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.

Here is my network accuracy by epoch:



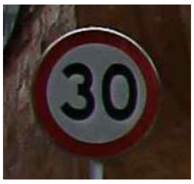
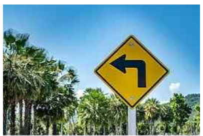
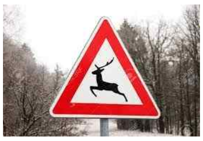
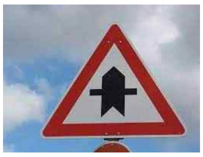
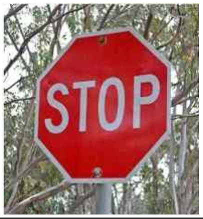
My final model results were:

- training set accuracy of ? 0.998
- validation set accuracy of ? 0.961
- test set accuracy of ? 0.937

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.

Here are five German traffic signs that I found on the web:

	© CanStockPhoto.com			

The third image might be difficult to classify. Because the size was 32×32×3, and in the process, the shape of the wild animal was distorted. So The prediction of the third image is wrong.

2. 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.

Here are the results of the prediction:

Image	Prediction
30 km/h	30 km/h
left-turn	Priority road
Wild animals crossing	double curve
Right of way at the next intersection	Right of way at the next intersection
Stop	Stop

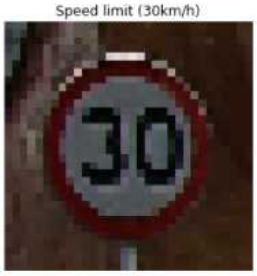
The model was able to correctly guess 3 of the 5 traffic signs, which gives an accuracy of 60%. I expected that the predictions of the first and fifth images would be correct. In addition, since I brought any traffic signal image that did not exist in the data set, I could see that it was significantly inferior to the accuracy of the test set.

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. 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)

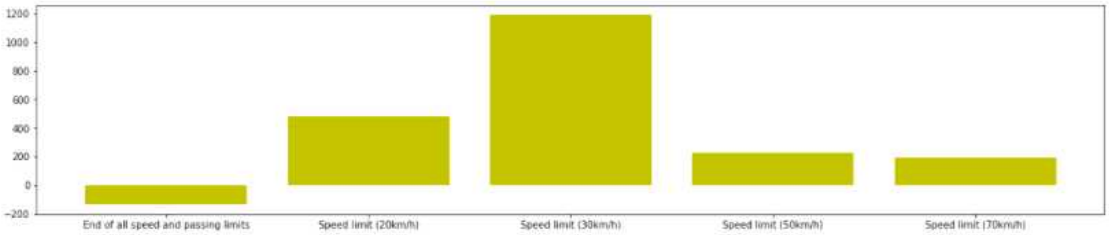
The code for making predictions on my final model is located in the 14th cell of the Ipython notebook.

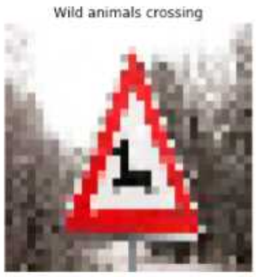
Probability	Prediction
0.61	30 km/h
0.51	Priority road
0.40	double curve
0.51	Right of way at the next intersection
0.81	Stop

Here are the top five softmax probabilities for them and their name values:

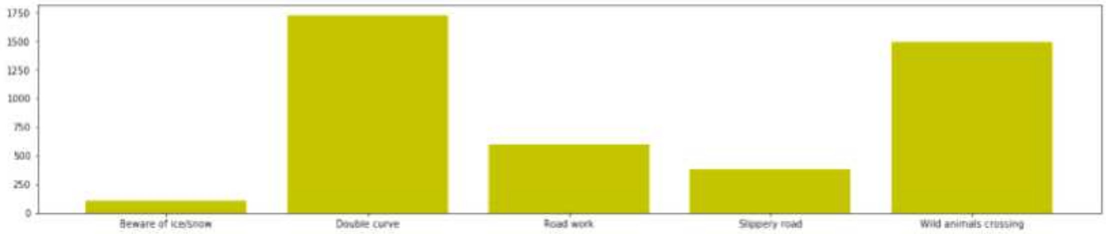


```
[ 0.60812628  0.24672316  0.11499724  0.0986786  -0.06852517] ['Speed limit (30km/h)', 'Speed limit (20km/h)', 'Speed limit (50km/h)',
'Speed limit (70km/h)', 'End of all speed and passing limits']
```

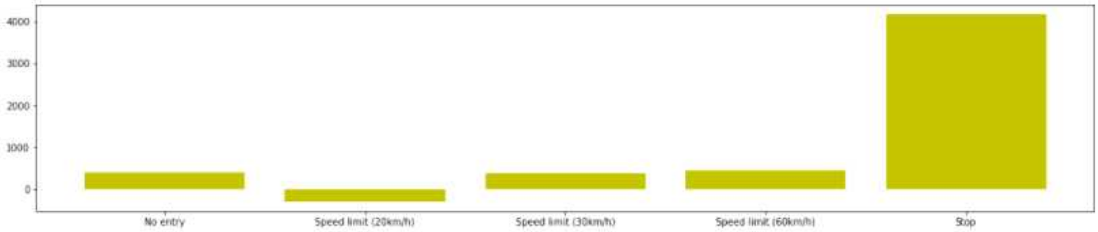




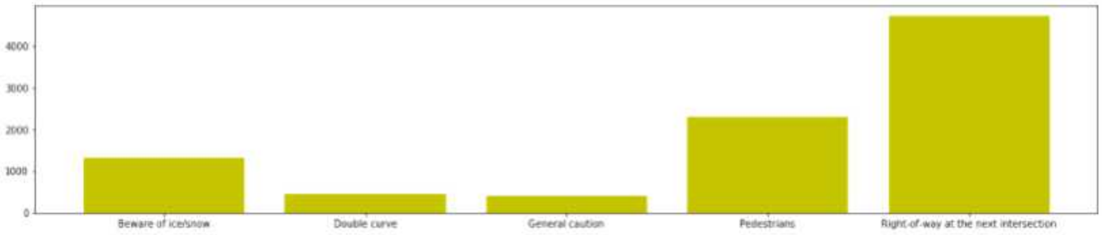
```
[ 0.40055764  0.34598449  0.13964885  0.08932953  0.02447945] ['Double curve', 'Wild animals crossing', 'Road work', 'Slippery road',
'Beware of ice/snow']
```



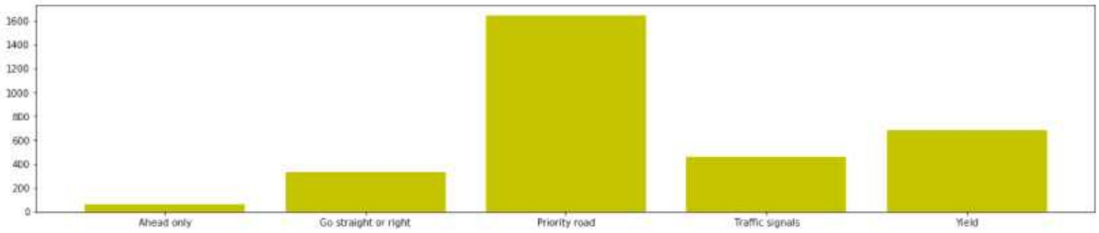
```
[ 0.81279045  0.0900435   0.07871869  0.07471117 -0.05626375] ['Stop', 'Speed limit (60km/h)', 'No entry', 'Speed limit (30km/h)', 'Spe
ed limit (20km/h)']
```



```
[ 0.51371241  0.25002518  0.14303975  0.04839222  0.04483054] ['Right-of-way at the next intersection', 'Pedestrians', 'Beware of ice/s
now', 'Double curve', 'General caution']
```



```
[ 0.51893133  0.21376789  0.14426447  0.10357027  0.01946608] ['Priority road', 'Yield', 'Traffic signals', 'Go straight or right', 'Ah
ead only']
```


Ahead only
Go straight or right
Priority road
Traffic signals
Yield