Traffic Sign Classification writeup

Yi Zhu

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 32*32
* The number of unique classes/labels in the data set is 43

####2. Include an exploratory visualization of the dataset.
I plot a bar chart (In notebook) showing how the data is formed in training data, from which we can see data is a little bit unbalanced that we have more data with smaller index than data with larger index.

###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? (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
As a first step, I tried to convert the images to grayscale but I didn't implement this in training since I think color information is still useful for traffic sign detection.
In data preprocessing, I shuffle the training data to avoid bias. I tried using normalization which just turned out to perform better.
I decided to generate additional data because the data we have is not enough to learn the traffic sign with different orientation, different distance even same sign in different weather condition. So I add more data by using imgaug package where I can implement cropping, blurring, sharpening, dropout, translation, rotation, etc. To be specific, I didn't augment the data in the beginning but randomly choose some augmentation method during each iteration of training so that it saves storage space a lot.

####2. Describe what your final model architecture looks like including model type, layers, layer sizes, connectivity, etc.)

I mainly used LeNet architecture with some tuning of parameters. The reason why I didn't make much change is that the training performance is really good, so the architecture is strong enough to find the underlying principle of data. The bottleneck is data but not deep learning architecture.

####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.

I used Adam optimizer with 10 epochs, batch size 128 and learning rate 0.001.
####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.
My final model results were:
* training set accuracy of 0.909
* validation set accuracy of 0.943?

###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.

The images are shown in notebook. It turned out we missed one image which is 7th image. It should be in class 38, but model predicts it to be 39. 38 is Keeping right while 39 is keeping left. There are two reasons for misclassification First is the size of data of class 39 is too small and it leads to the data unbalance. Second is because 38 and 39 are too similar which actually could be transformed using a symmetric transformation.

2 The information of 10 predicted images are :

```
test sample 4, prediction is [4]
test sample 3, prediction is [3]
test sample 13, prediction is [13]
test sample 25, prediction is [25]
test sample 38, prediction is [38]
test sample 23, prediction is [23]
test sample 38, prediction is [39]
test sample 5, prediction is [5]
test sample 25, prediction is [25]
test sample 10, prediction is [10]
```

Corresponding top five softmax probabilities are : (You can also see this in notebook)

```
test sample 4, top 5 probability values are TopKV2(values=array([[ 324.4464416
5,  185.63923645,  151.36981201,  150.06211853,
        119.68250275]], dtype=float32), indices=array([[4, 2, 7, 5, 8]]))
test sample 3, top 5 probability values are TopKV2(values=array([[ 586.0341796
9,  527.5401001 ,  250.22422791,  208.72109985,
         35.67368317]], dtype=float32), indices=array([[ 3,  5,  2,  7, 15]]))
```

```
test sample 13, top 5 probability values are TopKV2(values=array([[ 2995.74462
891,   845.92486572,   291.82632446,   233.84037781,
         227.40063477]], dtype=float32), indices=array([[13, 15, 35, 29, 4
0]]))
test sample 25, top 5 probability values are TopKV2(values=array([[ 2540.19165
039,  1104.24987793,   842.35443115,   722.62786865,
         568.46173096]], dtype=float32), indices=array([[25, 24, 21, 30, 2
0]]))
test sample 38, top 5 probability values are TopKV2(values=array([[ 1050.80822
754,   977.43395996,   570.86260986,   428.48538208,
         408.04437256]], dtype=float32), indices=array([[38, 34, 40, 35, 3
6]]))
test sample 23, top 5 probability values are TopKV2(values=array([[ 2221.50708
008,  1797.640625  ,  1433.14501953,  1356.17102051,
        1068.37670898]], dtype=float32), indices=array([[23, 19, 30, 21, 2
0]]))
test sample 38, top 5 probability values are TopKV2(values=array([[ 210.160598
75,  182.50698853,   28.47053528,   14.297472  ,
         14.11289978]], dtype=float32), indices=array([[39, 38, 16, 33, 34]]))

test sample 5, top 5 probability values are TopKV2(values=array([[ 2544.138916
02,  1373.23937988,  1193.88452148,  1023.70526123,
         677.02874756]], dtype=float32), indices=array([[5, 3, 7, 2, 1]]))
test sample 25, top 5 probability values are TopKV2(values=array([[ 593.061035
16,   83.85523224,   45.83409119,   29.83340645,
          3.51120663]], dtype=float32), indices=array([[25, 10, 28, 22, 29]]))

test sample 10, top 5 probability values are TopKV2(values=array([[ 690.703430
18,  295.23553467,  201.80430603,   67.79099274,
         32.89255524]], dtype=float32), indices=array([[10,  9, 16, 14,  5]]))
```