UDACITY

‹ Back to Self-Driving Car Engineer

Traffic Sign Classification

REVIEW

CODE REVIEW

HISTORY

Requires Changes

2 SPECIFICATIONS REQUIRE CHANGES

Lately on slack few students asked for a good Deep Learning book. So here's a book which is also recommended by Elon Musk:

- Deep Learning (Adaptive Computation and Machine Learning series) Github and on Amazon
- Fast.ai
- A Guide to Deep Learning

Really nice job with this first submission. Your next will certainly be the ONE. Keep up the good work !!!

Files Submitted

The project submission includes all required files.

- **Ipython notebook with code**
- **HTML output of the code**
- **A writeup report (either pdf or markdown)**

Dataset Exploration

The submission includes a basic summary of the data set.

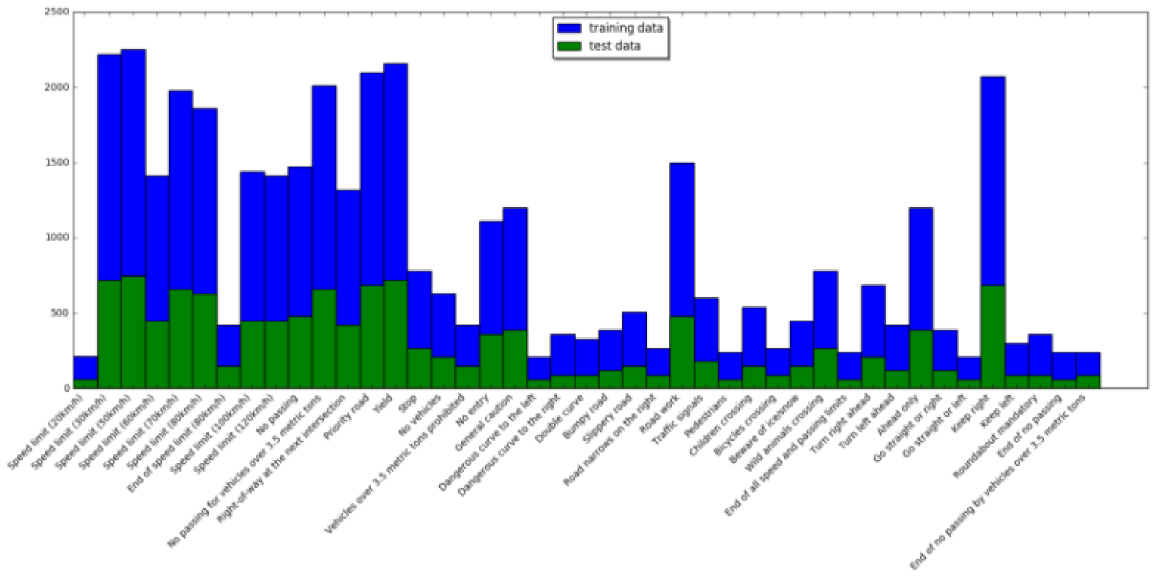
- A basic summary statistics of the dataset has been printed out
- Nice job !!!

The submission includes an exploratory visualization on the dataset.

- Very nice job with the visualizations here

Suggestion

- Here's a histogram showing the distribution inside both the training and the testing set



Design and Test a Model Architecture

The submission describes the preprocessing techniques used and why these techniques were chosen.

- Preprocessing has been discussed and justified

Suggestions & Comments

Here are some preprocessing techniques that have proven to work on this dataset (some of which you did think of so good job on that)

- Performing translations, scalings and rotations of the training set considerably improves generalization
- Values for translation, rotation and scaling can be drawn from a uniform distribution in a specified range, i.e. ±T % of the image size for translation, 1±S/100 for scaling and ±R○ for rotation.
- A normalization method that yields very impressive results is the Contrast-limited Adaptive Histogram Equalization (CLAHE)
- Normalization can help in case of High contrast variation among the images
- These resources (1 & 2 & 3) might provide some more intuition on the subject here.

The submission provides details of the characteristics and qualities of the architecture, including the type of model used, the number of layers, and the size of each layer. Visualizations emphasizing particular qualities of the architecture are encouraged.

Sufficient details about the model's architecture is provided

Suggestions & Comments

- In order to further improve this work, it would be better to visualize the architecture that is adopted. And TensorBoard is a very good visualization tool.

The submission describes how the model was trained by discussing what optimizer was used, batch size, number of epochs and values for hyperparameters.

Suggestions & Comments

- Using StratifiedShuffleSplit over the traditional train-test-split, enables the use of the entire training data while keeping data for cross-validation: the distribution of the labels in both would be similar. Around 1% of the data can be used for the cross-validation set.
- Here's a nice discussion on how to choose the batch_size of Stochastic Gradient Decent
- And a Discussion on Adam Optimizer

The submission describes the approach to finding a solution. Accuracy on the validation set is 0.93 or greater.

The approach taken is nicely discussed. I left some comments below.

Suggestions & Comments

Here are few questions to think about while designing the architecture:

- How would I choose the optimizer? What is its Pros & Cons and how would I evaluate it?
- How would I decide the number and type of layers?
- How would I tune the hyperparameter? How many values should I test and how to decide the values?
- How would I preprocess my data? Why do I need to apply a certain technique?
- How would I train the model?
- How would I evaluate the model? What is the metric? How do I set the benchmark?
- In case you're interested, here's an interesting Inception's example.

Test a Model on New Images

The submission includes five new German Traffic signs found on the web, and the images are visualized. Discussion is made as to particular qualities of the images or traffic signs in the images that are of interest, such as whether they would be difficult for the model to classify.

- A good discussion to pinpoint the qualities of images that lead to classification difficulties is made. Great!
- This paper might provide further intuitions on the subject here.

The submission documents the performance of the model when tested on the captured images. The performance on the new images is compared to the accuracy results of the test set.

Required

- You provide a good analysis of the results on the newly acquired images but forgot to compare the accuracy on the new set of images to that on the old test set.
- Your explanation can look something like: the accuracy on the captured images is X% while it was Y% on the testing set thus It seems the model is overfitting

The top five softmax probabilities of the predictions on the captured images are outputted. The submission discusses how certain or uncertain the model is of its predictions.

Required

- Good job using top_k, but it was applied to the logits instead of the softmax probabilities.
- Make sure to apply softmax to the logits before applying tf.nn.top_k
- The softmax probabilities are calculated from the resulting logits, in this case the results from the output layer in your model, as in tf.nn.softmax(logits)
- These resources (1 & 2) might provide further intuitions on the subject here.

Suggestion

- Here is an example code from a student

```
### Print out the top five softmax probabilities for the predictions on the German traffic sign i
mages found on the web.
###sess.run(tf.nn.top_K(tf.constant(classification),k=5))
keep_drop_out = 1.0
predicted_labels = tf.nn.softmax(logits)

with tf.Session() as sess:
    saver.restore(sess,save_file)
    output = sess.run(predicted_labels,feed_dict={x:test_images,keep_prob:keep_drop_out})
    final_out = sess.run(tf.nn.top_k(output,k=5))
    print(final_out)
```

```
TopKV2(values=array([[  1.00000000e+00,   1.60835317e-10,   1.13804157e-11,
          2.41506207e-12,   9.92635878e-13],
       [  1.00000000e+00,   4.57480516e-13,   1.92221459e-20,
          1.10328777e-20,   2.26027424e-24],
       [  4.23467577e-01,   2.43671894e-01,   1.55723602e-01,
          9.51780155e-02,   5.53437807e-02],
       [  1.00000000e+00,   1.66713162e-15,   6.94942676e-16,
          5.40751094e-17,   2.97571803e-21],
       [  9.99983549e-01,   9.06850892e-06,   7.06000947e-06,
          4.03408364e-07,   3.22741940e-08]], dtype=float32), indices=array([[25, 31, 22, 29, 2
6],
       [40, 18, 37, 38, 39],
       [29, 26, 28,  0,  8],
       [22, 25, 26, 20, 18],
       [24, 26, 27, 25, 18]]))
```

RESUBMIT

DOWNLOAD PROJECT

Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

Watch Video (3:01)

RETURN TO PATH