UDACITY

Back to Self-Driving Car Engineer

Traffic Sign Classification

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Here are some comments I have that might help you improve on your work:

- For developing an intuition, I think it comes down more to developing more experience on how to approach a problem.
  CNNs are predominantly used for anything related to images. That's what they were created for, although there has been work to try to use them elsewhere too I believe.
- Building an architecture depends a lot on your data - If I have images of faces, that's more features than a single hand written number, so you need a deeper model to be able to extract the features and then as you go deeper you get more features (as we saw in the lectures)
- As for tuning parameters based on results - I have been recommending students this resource **cs231n.github.io/neural-networks-3/#baby** to get an idea on how to analyse the results; but it then comes down to experimenting and seeing what affects what
- I have been following one rule till now however, and can't say it's definite but more good data is better than trying to finely tune a particular model. So as per me some form of data augmentation based on the data you have will help. Which is easier when it comes to working with images.
- I would recommend you read up on research papers if you are serious on pursuing this field (DL or ML). Or find implementations of similar projects outside of this ND so that you can learn from them and understand why something specific is done. There's too many approaches and too many parameters, and trying everything out on your own is not always feasible. So it's better to learn from others and that will help develop an intuition in the long run.
- To increase the ability of your model to generalize, a good preprocessing yields impressive results.
- Hope that helps :)

Nice job improving on the sections of your report that needed changes. Your project now meets all specifications defined by the project rubric for this project. Congratulations on passing the second project on the Self Driving Car Engineer Nanodegree and best of lucks in the ones ahead.
Cheers and 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.

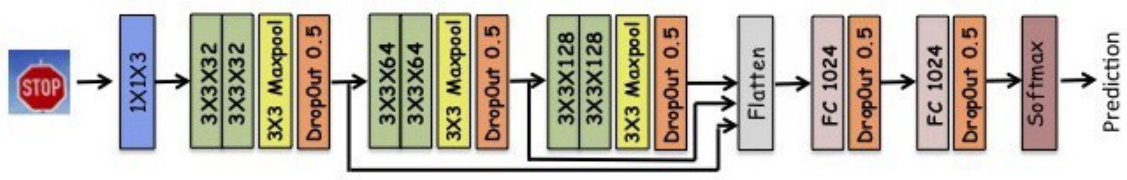
The submission includes an exploratory visualization on the dataset.

Design and Test a Model Architecture

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

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.

- In case you're interested, this is another way of going about the project by Vivek Yadav, a student in the SDCND:



- And here's a link to his article on it.
- Another interesting project is this one from another student too.

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

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

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.

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.

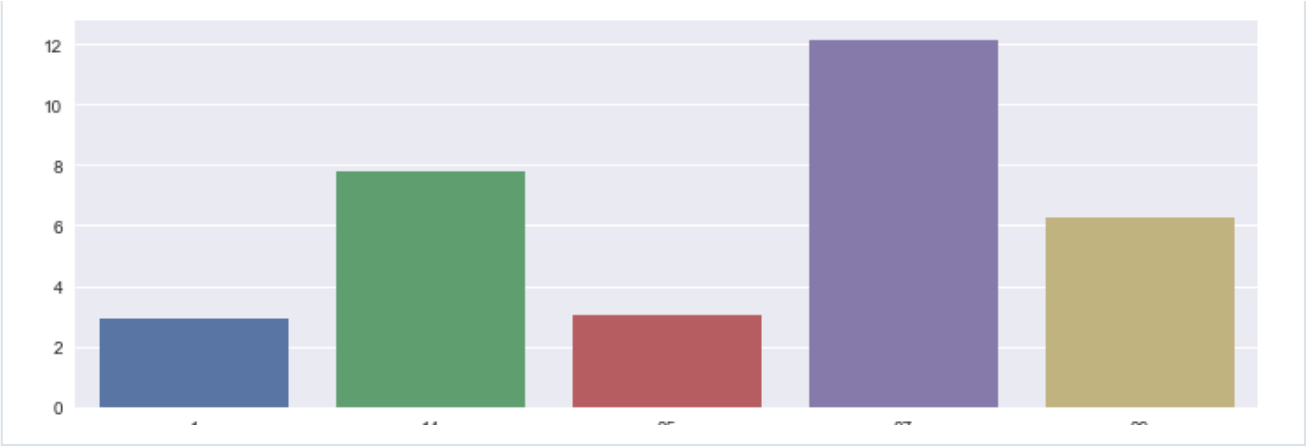
The top five softmax probabilities of the predictions on the captured images are outputted. The submission discusses how certain or uncertain the model is of its predictions.

Suggestions & Comments

- Here is an example code from a student for visualizing the softmax probabilities:

```
### Print out the top five softmax probabilities for the predictions on the German traffic sign ima
### Feel free to use as many code cells as needed.
for i in range(6):
    print(new_images[i])
    print("Predicted Class: {} | True Class: {}".format(preds[i], y_new[i]))
    fig = plt.figure(figsize=(12,4))
    sns.barplot(x=top_k[1][i], y=top_k[0][i])
    plt.show()
```

- And this is an example output from the code above:



DOWNLOAD PROJECT

RETURN TO PATH

Rate this review