

Semantic Segmentation Project Reflection

Table of Contents

1	Introduction.....	1
2	Architecture.....	1
3	Files.....	2
4	Running.....	2
5	Training Params	2
6	Results	3
7	Samples	3
8	References.....	4

1 Introduction

The goal of this project is to construct a fully convolutional neural network based on the VGG-16 image classifier architecture for performing semantic segmentation to identify drivable road area from an car dashcam image (trained and tested on the KITTI data set).

2 Architecture

A pre-trained VGG-16 network was converted to a fully convolutional network by converting the final fully connected layer to a 1x1 convolution and setting the depth equal to the number of desired classes

Performance is improved through the use of skip connections, performing 1x1 convolutions on previous VGG layers (in this case, layers 3 and 4) and adding them element-wise to up sampled (through transposed convolution) lower-level layers (i.e. the 1x1-convolved layer 7 is up sampled before being added to the 1x1-convolved layer 4).

Each convolution and transpose convolution layer includes a kernel initializer and regularizer

Optimizer

The loss function for the network is cross-entropy, and an Adam optimizer is used.

3 Files

- vizualize_images.ipynb
visualize the input images
- main.py
main program that trains the FCN
- helper.py
helper functions
- tests.py
test/validation programs

4 Running

```
ubuntu@ip-172-30-2-165:~/projects/semantic_segmentation$  
ubuntu@ip-172-30-2-165:~/projects/semantic_segmentation$  
ubuntu@ip-172-30-2-165:~/projects/semantic_segmentation$ ls -al  
total 456  
drwxrwxr-x 5 ubuntu ubuntu 4096 Dec 7 18:38 .  
drwxrwxr-x 4 ubuntu ubuntu 4096 Dec 7 18:41 ..  
drwxr-xr-x 4 ubuntu ubuntu 4096 Dec 7 17:56 data  
-rw-r--r-- 1 ubuntu ubuntu 5401 Dec 7 18:37 helper.py  
-rw-r--r-- 1 ubuntu ubuntu 9482 Dec 7 18:37 main.py  
drwxrwxr-x 3 ubuntu ubuntu 4096 Dec 7 18:06 output  
drwxrwxr-x 2 ubuntu ubuntu 4096 Dec 7 17:50 __pycache__  
-rw-r--r-- 1 ubuntu ubuntu 6178 Dec 7 18:37 tests.py  
-rw-r--r-- 1 ubuntu ubuntu 415223 Dec 7 18:38 vizualize_images.ipynb  
ubuntu@ip-172-30-2-165:~/projects/semantic_segmentation$
```

```
ubuntu@ip-172-30-2-165:~/projects/semantic_segmentation$  
ubuntu@ip-172-30-2-165:~/projects/semantic_segmentation$ python main.py > output.log 2>&1
```

After the completion, check out the output.log for run time details.

5 Output

./output.log contains the output from training NN

```
TensorFlow Version: 1.4.0
Default GPU Device: /device:GPU:0
Tests Passed
Tests Passed
Tests Passed
Tests Passed
***** Semantic Segmentation Project *****
Tests Passed
Training...

EPOCH 1 ...
Loss: = 3.025
Loss: = 6.254
Loss: = 0.797
Loss: = 0.931
Loss: = 0.775
Loss: = 0.739
Loss: = 0.667
Loss: = 0.653
Loss: = 0.646
Loss: = 0.642
Loss: = 0.636
Loss: = 0.635
Loss: = 0.613
Loss: = 0.638
output.log
```

`./output/{timestamp}` contains the output images. I have not checked in because it is huge size.

`./output/samples` contains sample files

6 Training Params

The hyper-parameters used for training are:

- keep_prob: 0.5
- learning_rate: 0.00095
- epochs: 30
- batch_size: 5

7 Results

Loss per batch tends to average below 0.190 after 5 epochs and below 0.100 after ten epochs. Average loss per batch at epoch 10:0.045, 20: 0.054, at epoch 30: 0.072

The results are created in output directory.

8 Samples

output/samples director contains few sample images from the output of the fully convolutional network, with the segmentation class overlaid upon the original image in green.

1. [KITTI](<http://www.cvlibs.net/datasets/kitti/>) dataset
2. [Cityscapes](<https://www.cityscapes-dataset.com/>) dataset
3. [FCN](https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf)
ANN