

show_results

October 3, 2016

1 Load model

1.1 Model

```
conv1_1 = new_conv_layer( image, [3, 3, 1, 16], "conv1_1" )
conv1_2 = new_conv_layer( conv1_1, [3, 3, 16, 16], "conv1_2" )
pool1 = tf.nn.max_pool(conv1_2, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding='SAME',
name='pool1')
conv2_1 = new_conv_layer(pool1, [3, 3, 16, 16], "conv2_1")
conv2_2 = new_conv_layer(conv2_1, [3, 3, 16, 16], "conv2_2")
pool2 = tf.nn.max_pool(conv2_2, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding='SAME',
name='pool2')
conv3_1 = new_conv_layer(pool2, [3, 3, 16, 16], "conv3_1")
fc1 = self.new_fc_layer(conv3_1, 25*25*16, 512, 'fc1')
fc2 = self.new_fc_layer(fc1, 512, 10, 'fc2')
```

1.2 train params

L2 on weights (5e-5)

```
In [7]: import matplotlib.pyplot as plt
import numpy as np
import simple_model
from simple_model import training_generator
import utils

simple_model=reload(simple_model)
lr=.005
back_size=50
crop_pos=(20,20)
```

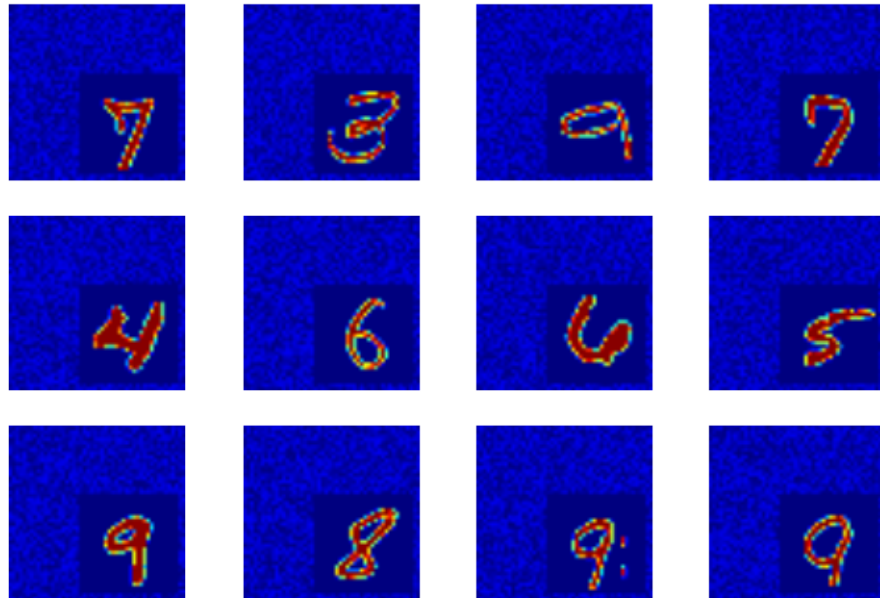
Exception AssertionError: AssertionError("Nesting violated for default stack of <ty

```
In [8]: """Plot training samples"""
batch = utils.get_batch('train', 3*4, back_size, .1, crop_pos).next()
```

```

fig, axs = plt.subplots(3,4)
for ax,img in zip([b for a in axs for b in a],batch[0]):
    ax.imshow(img.reshape(back_size,back_size), vmin=0, vmax=1)
    ax.set_axis_off()
plt.show()

```



```

In [9]: """Perform training"""
        gen = training_generator(lr=.005, back_size=back_size, crop_pos=(20,20) )
        for _ in range(15):
            _,accs = gen.next()
            print "max acc so far : "+str(max(accs)*100)

```

```

***** EPOCH 0 *****
Loss on testset is 2.301503
Accuracy now is 11.74
lr now is 0.00475
max acc so far : 11.74

```

```

***** EPOCH 1 *****
Loss on testset is 0.752772
Accuracy now is 76.18
lr now is 0.00451
max acc so far : 76.18

```

```

***** EPOCH 2 *****

```

Loss on testset is 0.310328
Accuracy now is 92.02
lr now is 0.00429
max acc so far : 92.02

***** EPOCH 3 *****
Loss on testset is 0.224481
Accuracy now is 94.35
lr now is 0.00407
max acc so far : 94.35

***** EPOCH 4 *****
Loss on testset is 0.184939
Accuracy now is 95.36
lr now is 0.00387
max acc so far : 95.36

***** EPOCH 5 *****
Loss on testset is 0.197542
Accuracy now is 95.25
lr now is 0.00368
max acc so far : 95.36

***** EPOCH 6 *****
Loss on testset is 0.125965
Accuracy now is 97.09
lr now is 0.00349
max acc so far : 97.09

***** EPOCH 7 *****
Loss on testset is 0.117109
Accuracy now is 97.32
lr now is 0.00332
max acc so far : 97.32

***** EPOCH 8 *****
Loss on testset is 0.152032
Accuracy now is 96.36
lr now is 0.00315
max acc so far : 97.32

***** EPOCH 9 *****
Loss on testset is 0.170708
Accuracy now is 95.11
lr now is 0.00299
max acc so far : 97.32

***** EPOCH 10 *****

```
Loss on testset is 0.069501
Accuracy now is 98.54
lr now is 0.00284
max acc so far : 98.54
```

```
***** EPOCH 11 *****
Loss on testset is 0.099702
Accuracy now is 97.46
lr now is 0.00270
max acc so far : 98.54
```

```
***** EPOCH 12 *****
Loss on testset is 0.070404
Accuracy now is 98.40
lr now is 0.00257
max acc so far : 98.54
```

```
***** EPOCH 13 *****
Loss on testset is 0.083144
Accuracy now is 98.08
lr now is 0.00244
max acc so far : 98.54
```

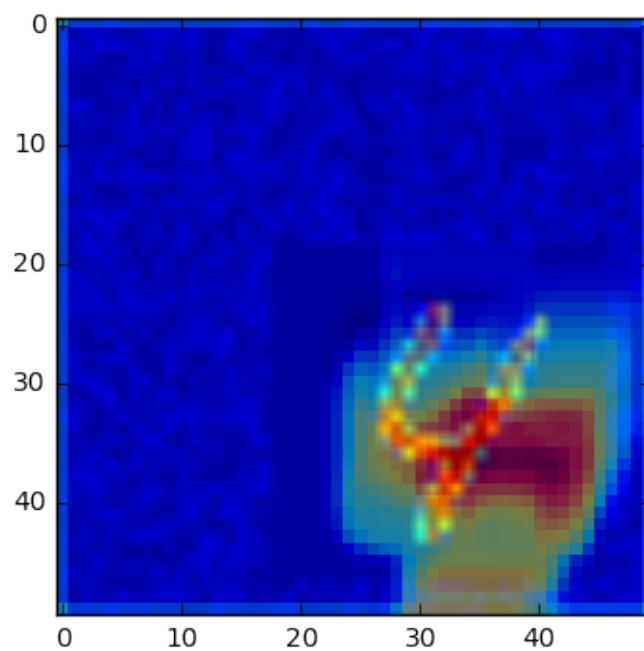
```
***** EPOCH 14 *****
Loss on testset is 0.120123
Accuracy now is 96.93
lr now is 0.00232
max acc so far : 98.54
```

```
In [10]: back_size = 50
```

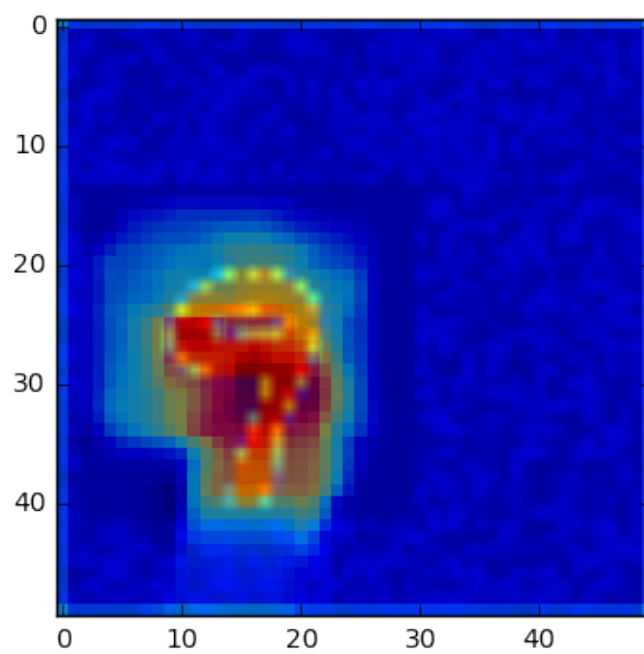
```
    for _ in range(5):
        print '-----'*5
        imgs, lbls = utils.get_batch('test', 1, back_size, .1).next()

        simple_model.show_activation(imgs[0])
```

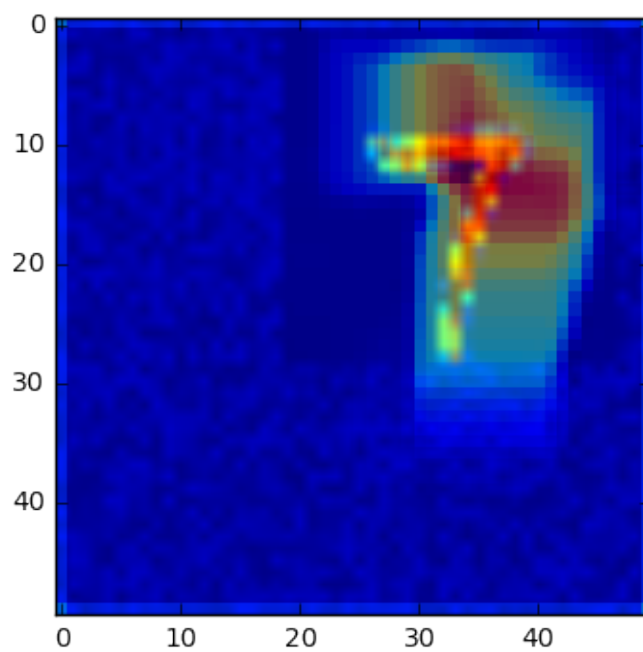
```
-----
prediction is : 4 with 21.036
```



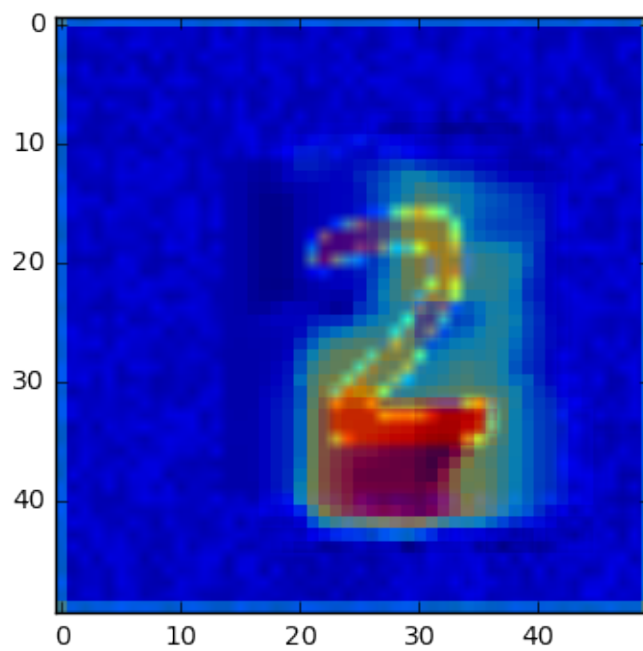
prediction is : 9 with 17.771



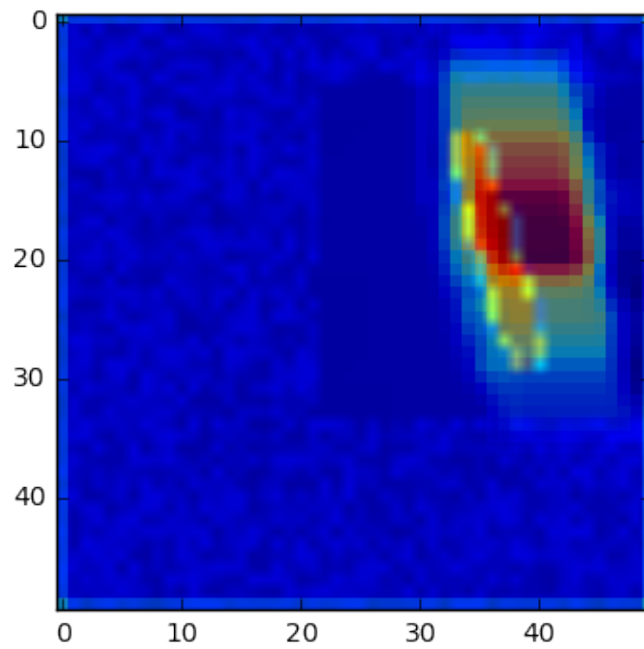
prediction is : 7 with 19.263



prediction is : 2 with 9.508



prediction is : 1 with 16.276



In []:

In []: