

show_results

October 11, 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") conv3_2 =
new_conv_layer(conv3_1, [3, 3, 16, 2*10], "conv3_2") gap = tf.reduce_mean( conv3_2, [1,2] )
ccn = tf.reshape(gap,[-1,10,nb_CCN]) ccn = tf.reduce_mean(ccn, 2)
```

1.2 train params

L2 on weights (5e-5) L1 on GAP

```
In [35]: 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   = 100
noise       = .1
crop_pos    = (10,10)
```

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

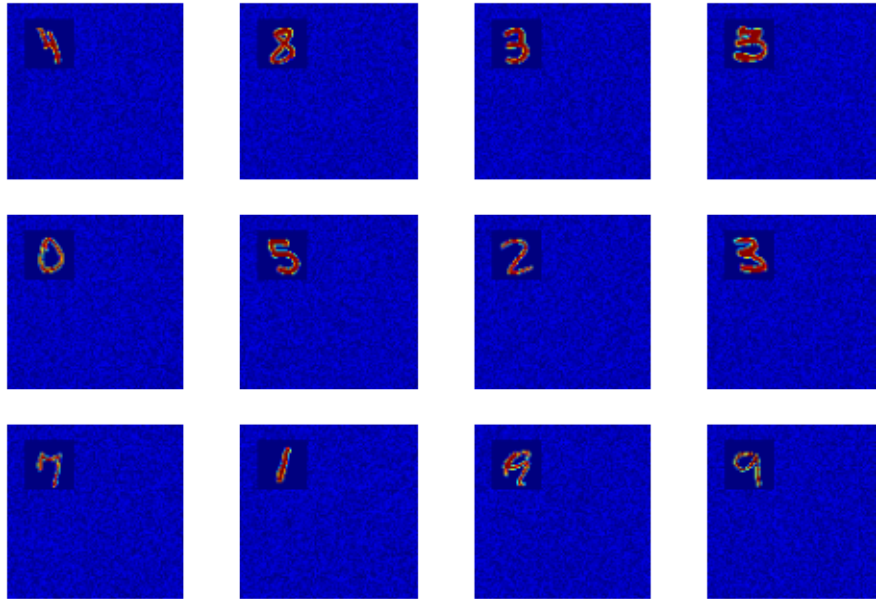
```
In [36]: """Plot training samples"""
batch = utils.get_batch('train', im_size=back_size, noise=noise, crop_pos=

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 [37]: """Perform training"""
gen = training_generator(lr=lr, back_size=back_size, noise=noise, crop_pos=
for _ in range(10):
    _,accs = gen.next()
    print "max acc so far : "+str(max(accs)*100)

```

```

***** EPOCH 0 *****
1-Loss on testset is 2.296860
1-Accuracy now is 11.63
2-Loss on testset is 2.296276
2-Accuracy now is 11.62
lr now is 0.00475
max acc so far : 11.63

```

```

***** EPOCH 1 *****
1-Loss on testset is 1.565749
1-Accuracy now is 50.71
2-Loss on testset is 1.511480
2-Accuracy now is 52.88

```

lr now is 0.00451
max acc so far : 50.71

***** EPOCH 2 *****
1-Loss on testset is 0.453971
1-Accuracy now is 91.31
2-Loss on testset is 0.437376
2-Accuracy now is 91.98
lr now is 0.00429
max acc so far : 91.31

***** EPOCH 3 *****
1-Loss on testset is 0.376442
1-Accuracy now is 91.73
2-Loss on testset is 0.359728
2-Accuracy now is 92.38
lr now is 0.00407
max acc so far : 91.73

***** EPOCH 4 *****
1-Loss on testset is 0.290531
1-Accuracy now is 95.04
2-Loss on testset is 0.273884
2-Accuracy now is 95.36
lr now is 0.00387
max acc so far : 95.04

***** EPOCH 5 *****
1-Loss on testset is 0.179444
1-Accuracy now is 97.87
2-Loss on testset is 0.176238
2-Accuracy now is 97.95
lr now is 0.00368
max acc so far : 97.87

***** EPOCH 6 *****
1-Loss on testset is 0.209377
1-Accuracy now is 96.82
2-Loss on testset is 0.203660
2-Accuracy now is 96.93
lr now is 0.00349
max acc so far : 97.87

***** EPOCH 7 *****
1-Loss on testset is 0.152638
1-Accuracy now is 98.48
2-Loss on testset is 0.149315
2-Accuracy now is 98.42

```
lr now is 0.00332
max acc so far : 98.48
```

```
***** EPOCH 8 *****
1-Loss on testset is 0.200854
1-Accuracy now is 96.95
2-Loss on testset is 0.193384
2-Accuracy now is 97.12
lr now is 0.00315
max acc so far : 98.48
```

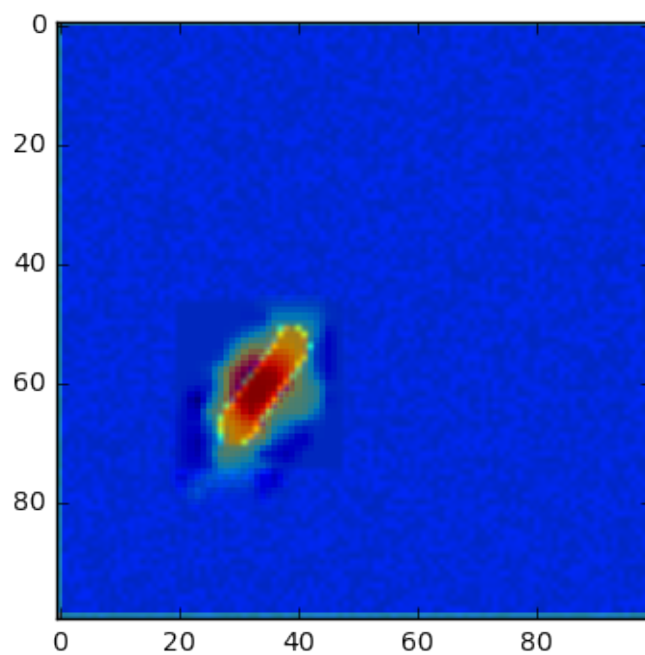
```
***** EPOCH 9 *****
1-Loss on testset is 0.157136
1-Accuracy now is 98.02
2-Loss on testset is 0.154534
2-Accuracy now is 98.12
lr now is 0.00299
max acc so far : 98.48
```

```
In [38]: back_size = 100
```

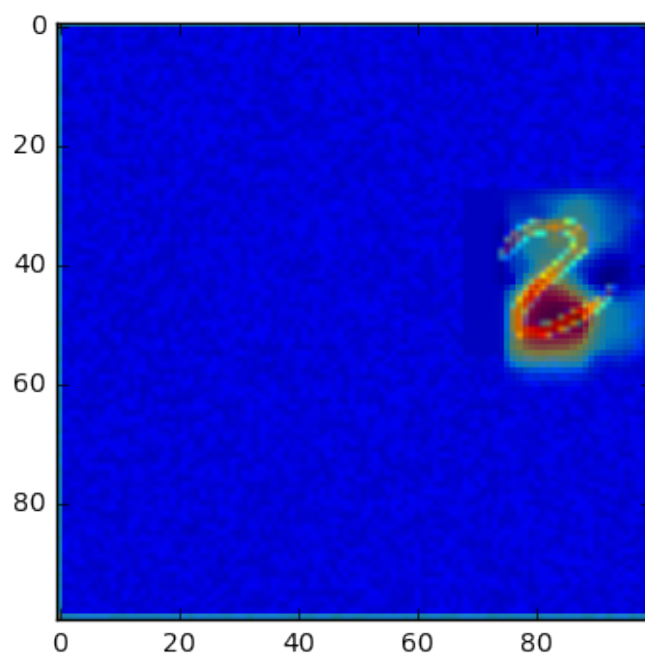
```
    for _ in range(10):
        print '-----'*5
        imgs, lbls = utils.get_batch('test', 1, back_size, .1).next()

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

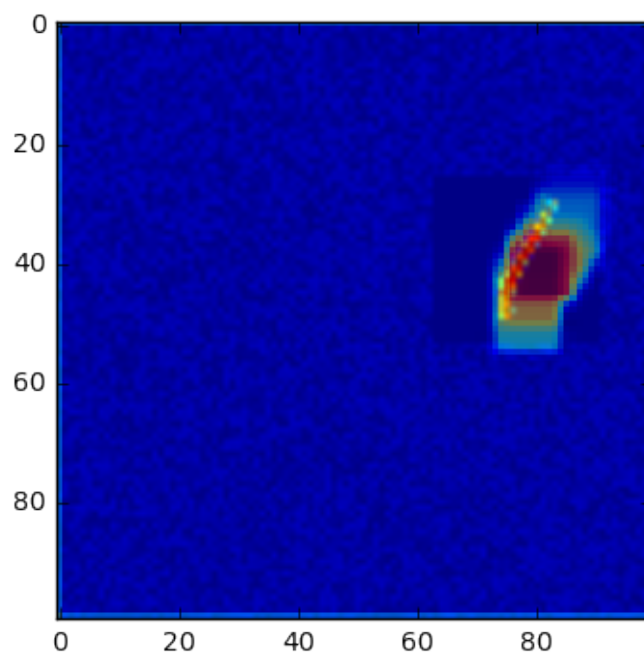
```
prediction is : 1 with 5.336
```



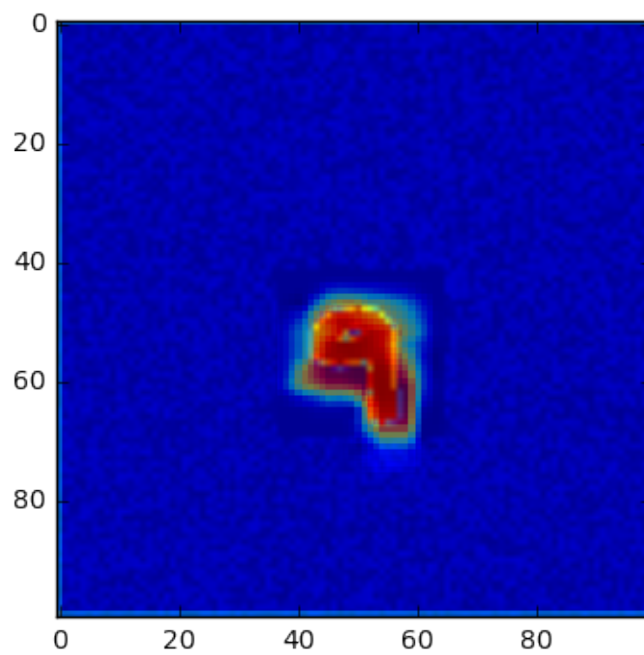
prediction is : 2 with 11.183



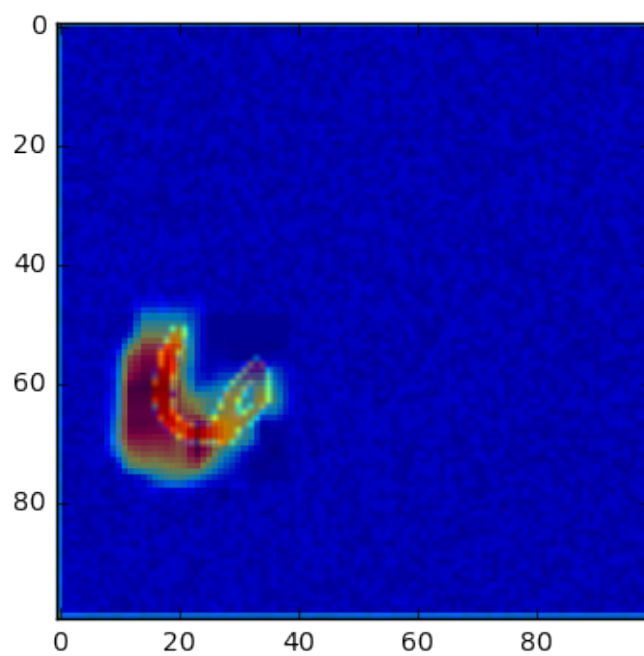
prediction is : 1 with 6.847



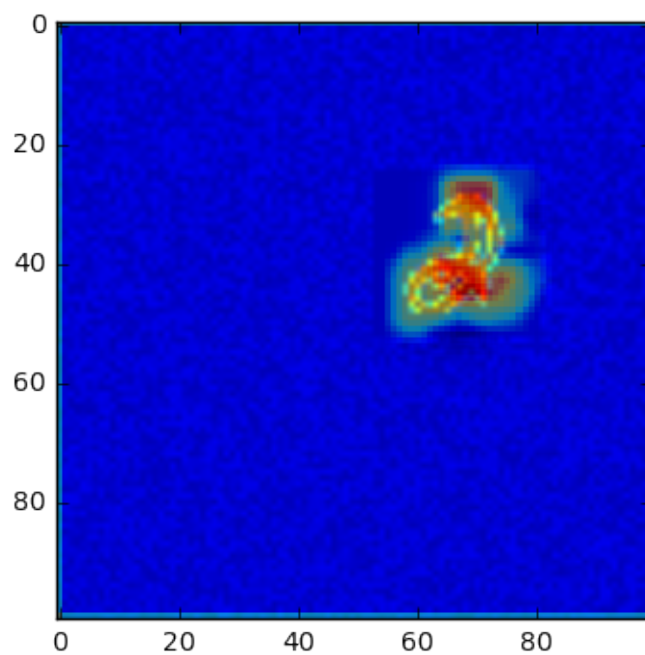
prediction is : 9 with 10.079



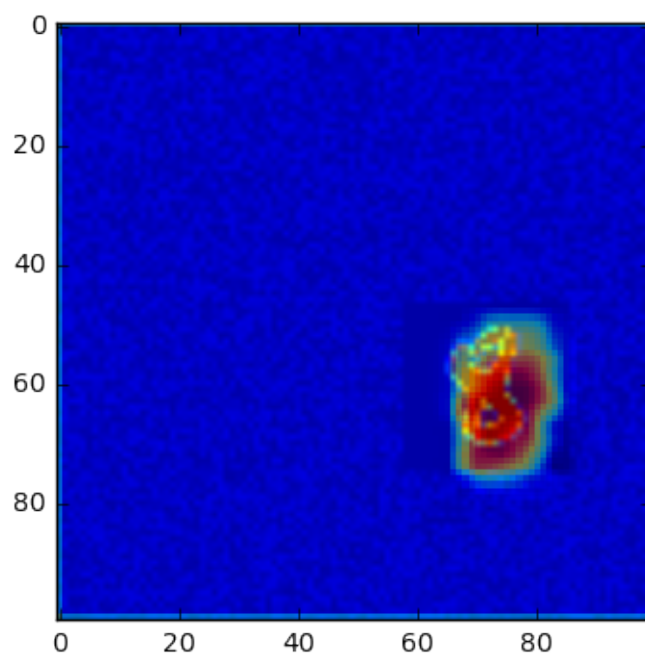
prediction is : 6 with 10.809



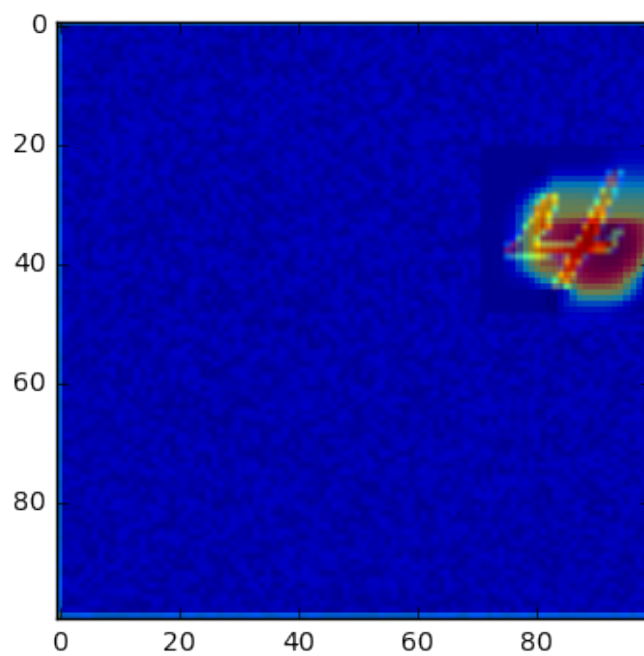
prediction is : 2 with 12.892



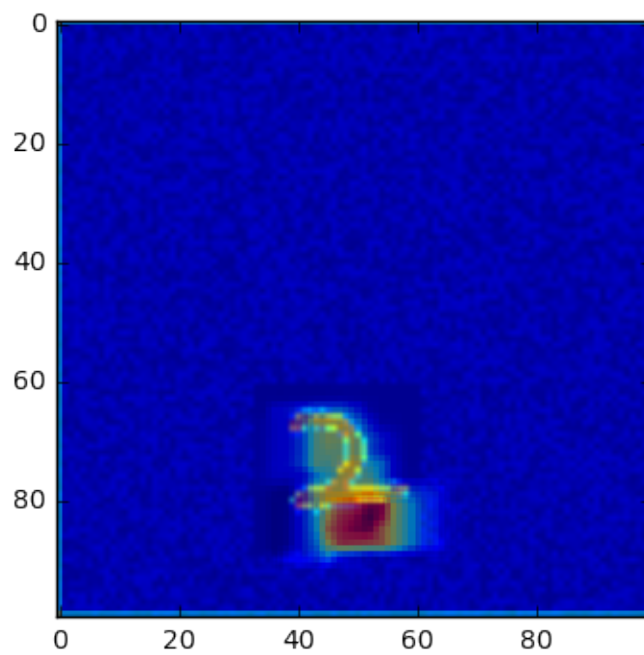
prediction is : 8 with 8.509



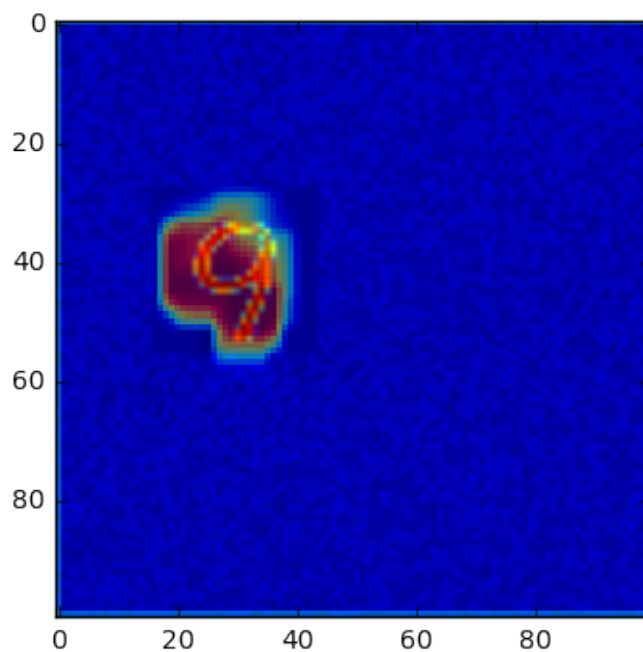
prediction is : 4 with 17.319



prediction is : 2 with 13.384



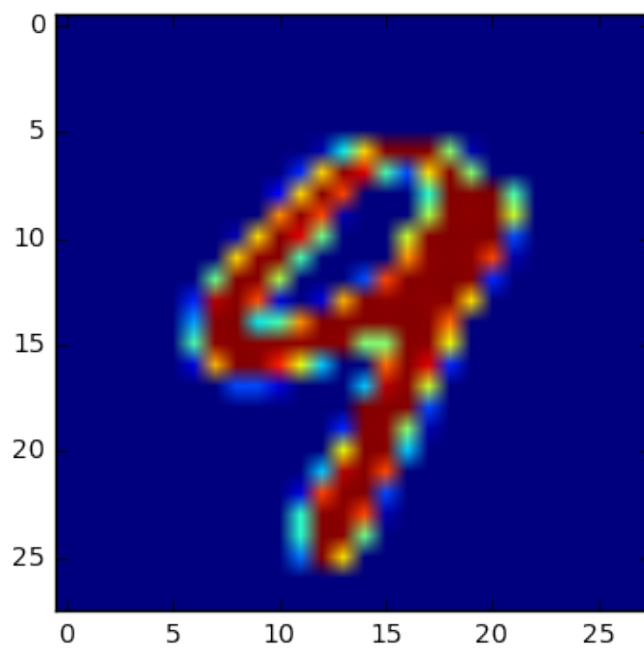
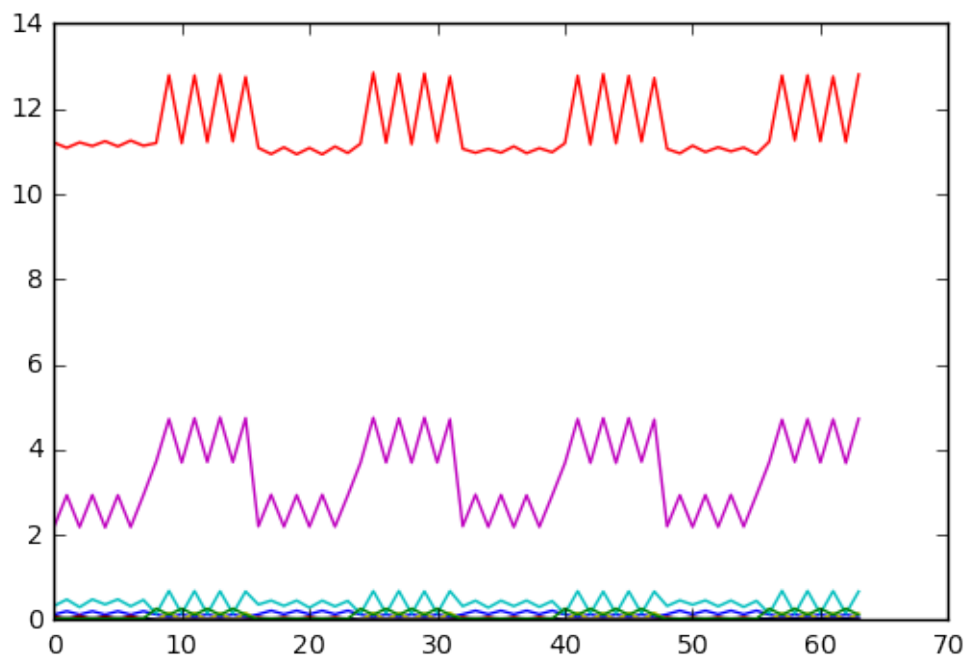
prediction is : 9 with 11.015



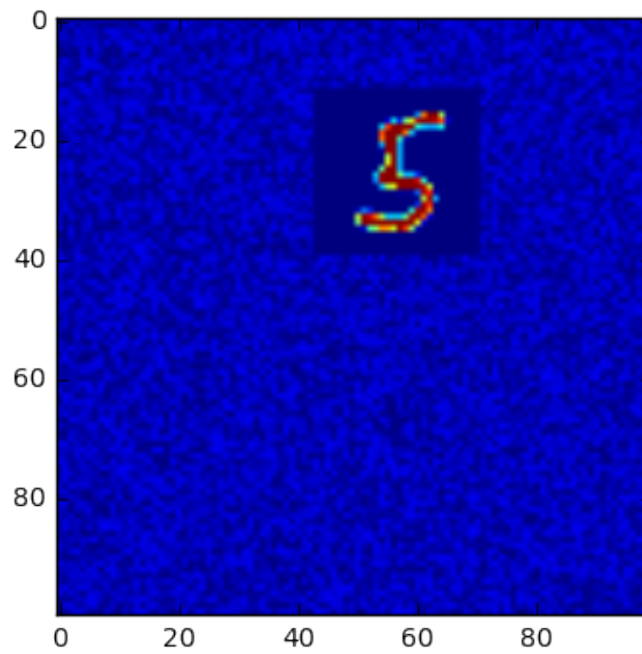
```
In [39]: accuracies = []
         for _ in range(100):
             digit, lbl = utils.get_batch('test', 1, .1).next()
             digit = digit[0].reshape(28,28)
             img = np.random.random((100,100))*0.1
             height = img.shape[0]
             width = img.shape[1]
             box_size = 28
             step_size = 10
             n_x_boxes = (width - box_size) / step_size + 1
             n_y_boxes = (height - box_size) / step_size + 1
             imgs = np.tile(img, (n_x_boxes*n_y_boxes, 1, 1))

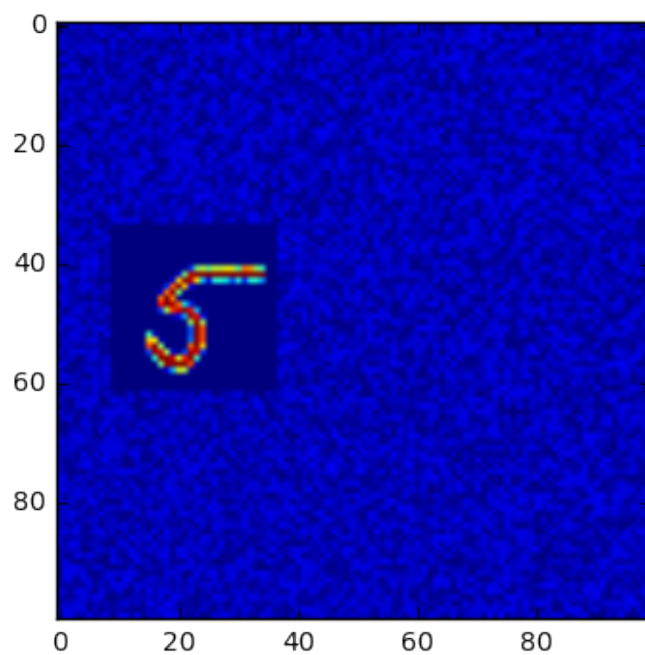
             for xx in range(0, n_x_boxes):
                 for yy in range(0, n_y_boxes):
                     idx = xx*n_x_boxes+yy
                     x = xx*step_size
                     y = yy*step_size
                     imgs[idx, x:x+box_size, y:y+box_size] = digit
```

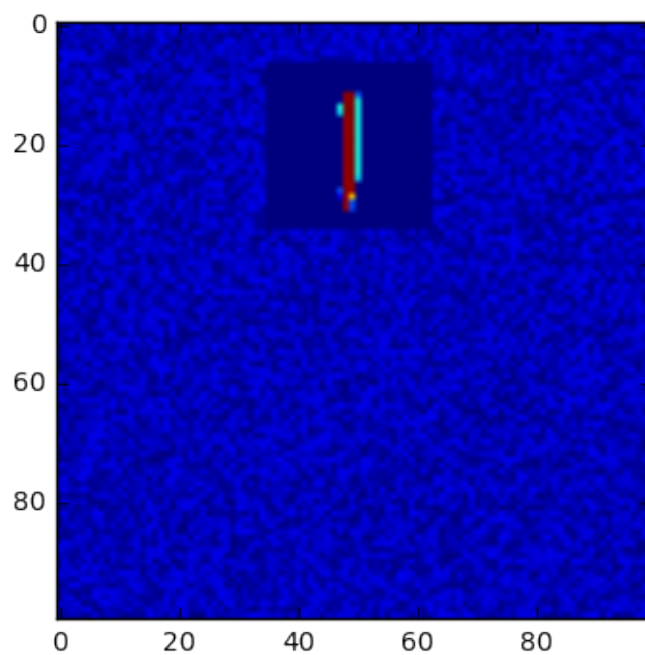

100.0

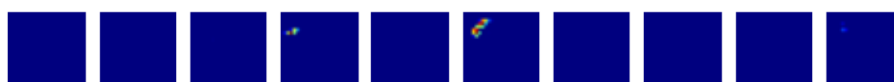
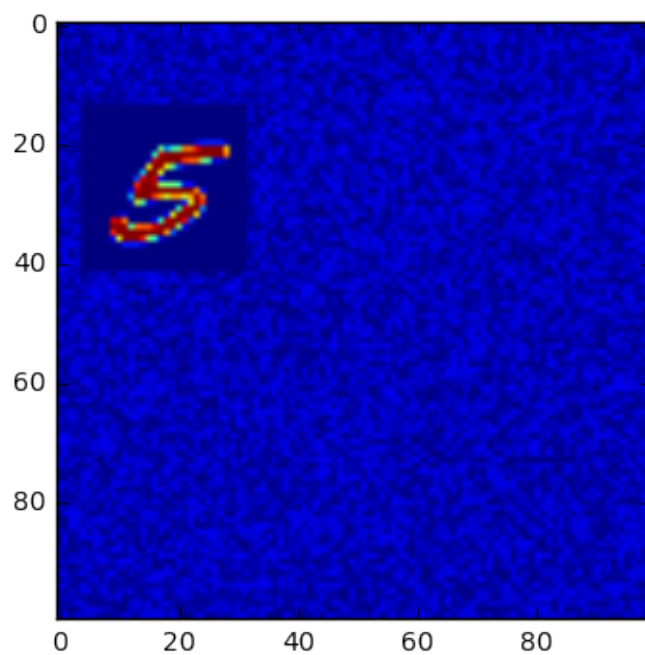


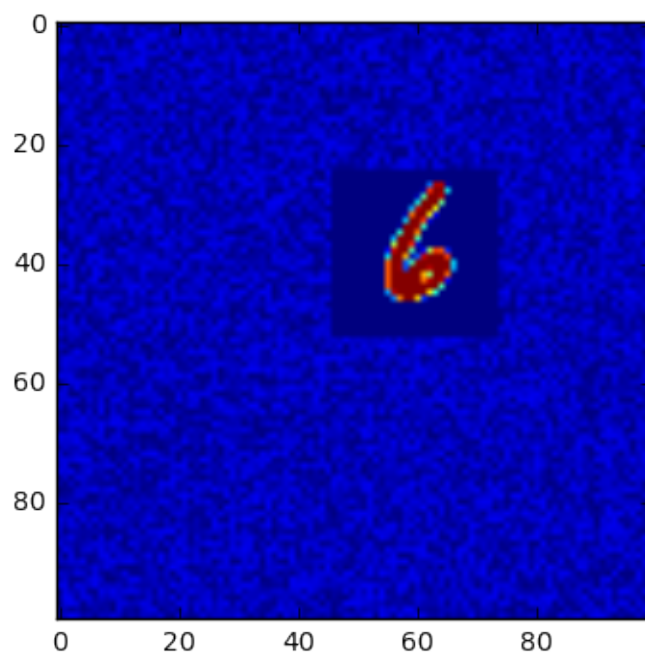
```
In [41]: nb_CCN = 2
         for _ in range(10):
             simple_model.plot_classes_maps(100, 2)
```

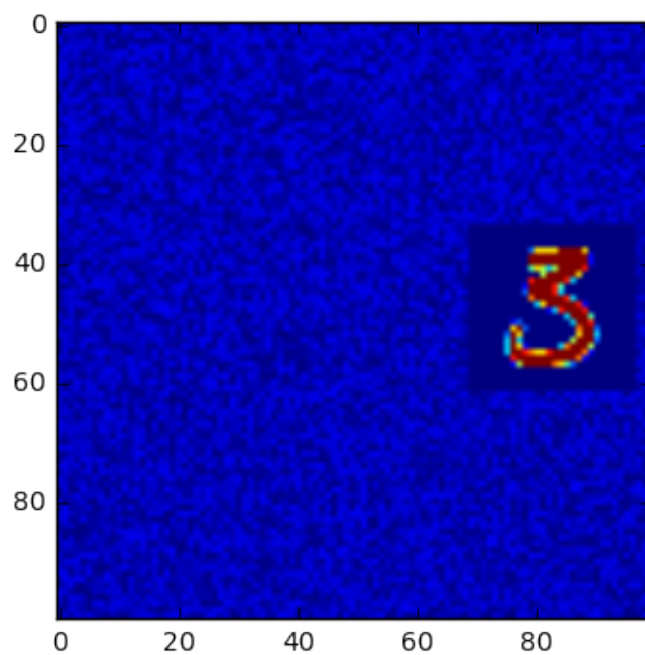


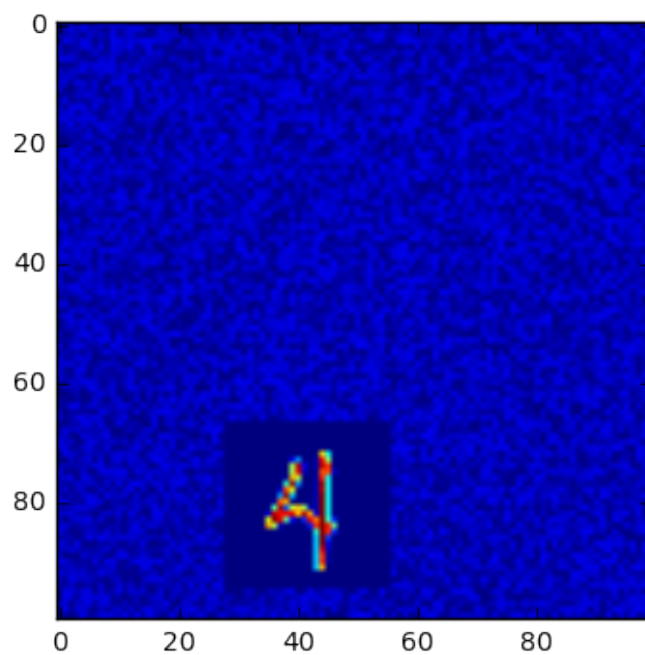


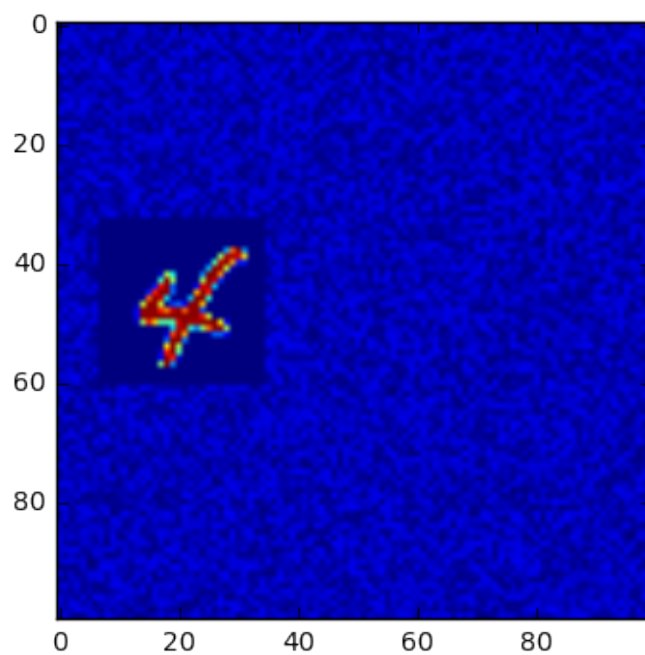


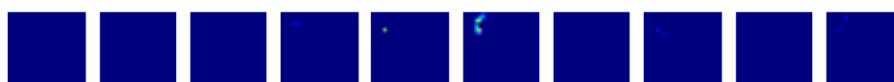
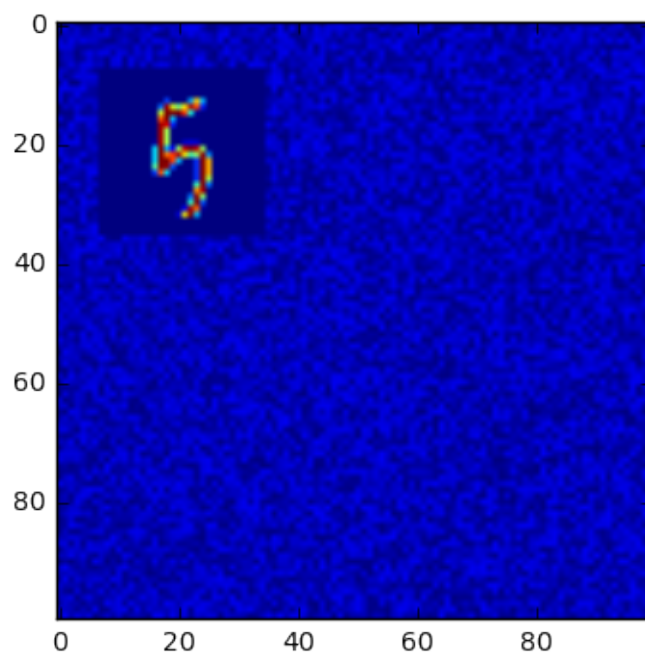


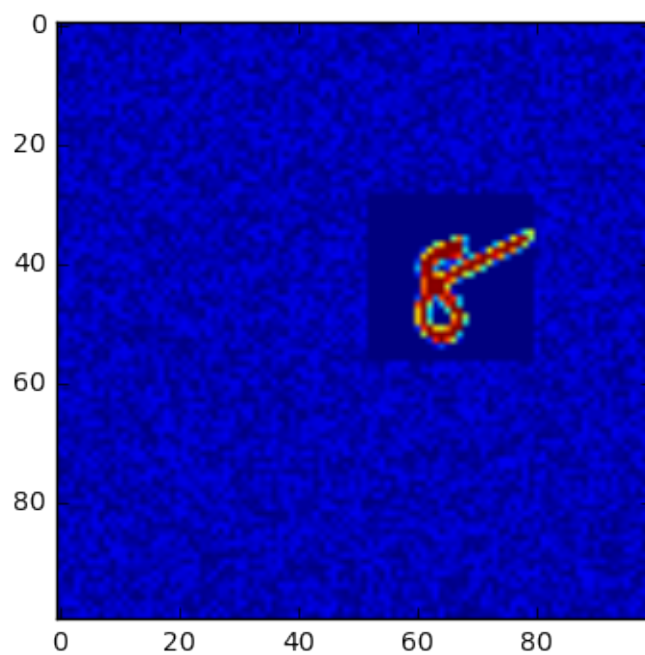












In []: