

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, 5)
```

1.2 train params

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

```
In [48]: 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)
n_CCN      = 5
```

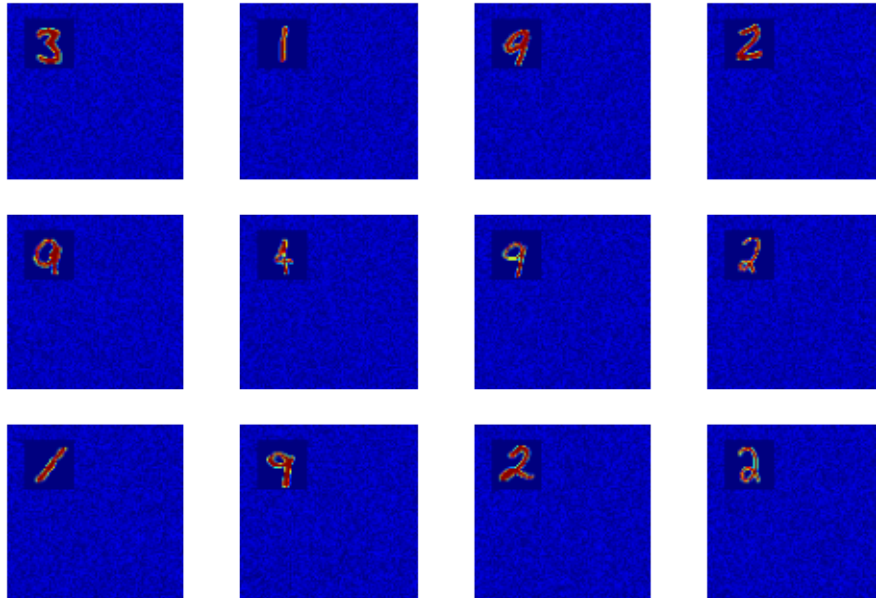
Exception AssertionError: AssertionError("Nesting violated for default stack of <ty

```
In [49]: """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 [50]: """Perform training"""
gen = training_generator(lr=lr, back_size=back_size, noise=noise, crop_pos=
for _ in range(25):
    _,accs = gen.next()
    print "max acc so far : "+str(max(accs)*100)

```

```

***** EPOCH 0 *****
1-Loss on testset is 2.489643
1-Accuracy now is 10.20
2-Loss on testset is 2.489449
2-Accuracy now is 10.16
lr now is 0.00475
max acc so far : 10.2

```

```

***** EPOCH 1 *****
1-Loss on testset is 1.973896
1-Accuracy now is 43.86
2-Loss on testset is 1.907417

```

2-Accuracy now is 45.36
lr now is 0.00451
max acc so far : 43.86

***** EPOCH 2 *****
1-Loss on testset is 1.074933
1-Accuracy now is 76.03
2-Loss on testset is 1.019283
2-Accuracy now is 77.77
lr now is 0.00429
max acc so far : 76.03

***** EPOCH 3 *****
1-Loss on testset is 0.528466
1-Accuracy now is 92.17
2-Loss on testset is 0.512067
2-Accuracy now is 92.74
lr now is 0.00407
max acc so far : 92.17

***** EPOCH 4 *****
1-Loss on testset is 0.468169
1-Accuracy now is 92.25
2-Loss on testset is 0.444276
2-Accuracy now is 92.92
lr now is 0.00387
max acc so far : 92.25

***** EPOCH 5 *****
1-Loss on testset is 0.508854
1-Accuracy now is 90.77
2-Loss on testset is 0.484943
2-Accuracy now is 91.46
lr now is 0.00368
max acc so far : 92.25

***** EPOCH 6 *****
1-Loss on testset is 0.359007
1-Accuracy now is 96.75
2-Loss on testset is 0.355956
2-Accuracy now is 96.86
lr now is 0.00349
max acc so far : 96.75

***** EPOCH 7 *****
1-Loss on testset is 0.364711
1-Accuracy now is 95.78
2-Loss on testset is 0.348414

2-Accuracy now is 96.37
lr now is 0.00332
max acc so far : 96.75

***** EPOCH 8 *****
1-Loss on testset is 0.312506
1-Accuracy now is 97.24
2-Loss on testset is 0.310179
2-Accuracy now is 97.30
lr now is 0.00315
max acc so far : 97.24

***** EPOCH 9 *****
1-Loss on testset is 0.239351
1-Accuracy now is 98.29
2-Loss on testset is 0.235691
2-Accuracy now is 98.22
lr now is 0.00299
max acc so far : 98.29

***** EPOCH 10 *****
1-Loss on testset is 0.257086
1-Accuracy now is 97.65
2-Loss on testset is 0.254358
2-Accuracy now is 97.58
lr now is 0.00284
max acc so far : 98.29

***** EPOCH 11 *****
1-Loss on testset is 0.215574
1-Accuracy now is 98.58
2-Loss on testset is 0.213214
2-Accuracy now is 98.65
lr now is 0.00270
max acc so far : 98.58

***** EPOCH 12 *****
1-Loss on testset is 0.240551
1-Accuracy now is 98.40
2-Loss on testset is 0.233189
2-Accuracy now is 98.53
lr now is 0.00257
max acc so far : 98.58

***** EPOCH 13 *****
1-Loss on testset is 0.257476
1-Accuracy now is 97.87
2-Loss on testset is 0.249801

2-Accuracy now is 98.09
lr now is 0.00244
max acc so far : 98.58

***** EPOCH 14 *****
1-Loss on testset is 0.214279
1-Accuracy now is 98.76
2-Loss on testset is 0.208502
2-Accuracy now is 98.76
lr now is 0.00232
max acc so far : 98.76

***** EPOCH 15 *****
1-Loss on testset is 0.217039
1-Accuracy now is 98.70
2-Loss on testset is 0.210192
2-Accuracy now is 98.83
lr now is 0.00220
max acc so far : 98.76

***** EPOCH 16 *****
1-Loss on testset is 0.210039
1-Accuracy now is 98.86
2-Loss on testset is 0.206373
2-Accuracy now is 98.82
lr now is 0.00209
max acc so far : 98.86

***** EPOCH 17 *****
1-Loss on testset is 0.205967
1-Accuracy now is 98.87
2-Loss on testset is 0.198867
2-Accuracy now is 98.97
lr now is 0.00199
max acc so far : 98.87

***** EPOCH 18 *****
1-Loss on testset is 0.197678
1-Accuracy now is 98.95
2-Loss on testset is 0.190971
2-Accuracy now is 99.02
lr now is 0.00189
max acc so far : 98.95

***** EPOCH 19 *****
1-Loss on testset is 0.196025
1-Accuracy now is 98.93
2-Loss on testset is 0.187407

```
2-Accuracy now is 99.08
lr now is 0.00179
max acc so far : 98.95
```

```
***** EPOCH 20 *****
1-Loss on testset is 0.215798
1-Accuracy now is 98.32
2-Loss on testset is 0.210243
2-Accuracy now is 98.50
lr now is 0.00170
max acc so far : 98.95
```

```
***** EPOCH 21 *****
1-Loss on testset is 0.189173
1-Accuracy now is 99.05
2-Loss on testset is 0.184063
2-Accuracy now is 99.16
lr now is 0.00162
max acc so far : 99.05
```

```
***** EPOCH 22 *****
1-Loss on testset is 0.184798
1-Accuracy now is 99.08
2-Loss on testset is 0.178782
2-Accuracy now is 99.14
lr now is 0.00154
max acc so far : 99.08
```

```
***** EPOCH 23 *****
1-Loss on testset is 0.204412
1-Accuracy now is 98.93
2-Loss on testset is 0.190236
2-Accuracy now is 98.89
lr now is 0.00146
max acc so far : 99.08
```

```
***** EPOCH 24 *****
1-Loss on testset is 0.192797
1-Accuracy now is 99.03
2-Loss on testset is 0.184844
2-Accuracy now is 99.10
lr now is 0.00139
max acc so far : 99.08
```

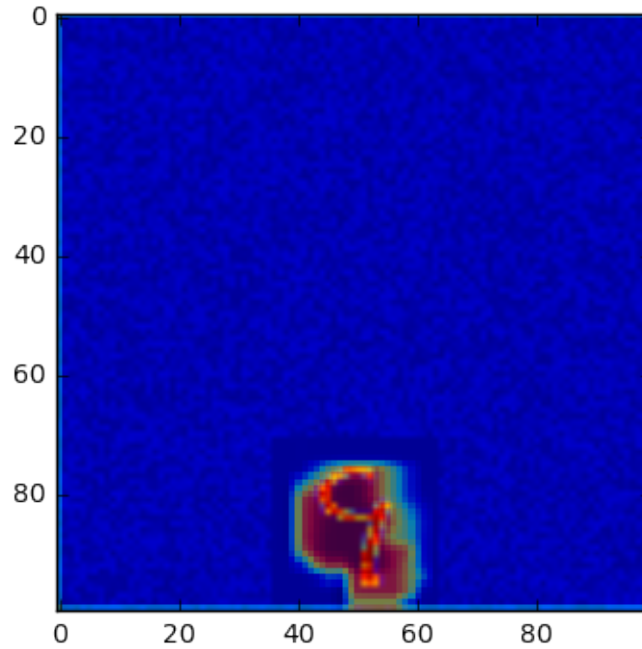
```
In [51]: back_size = 100
```

```
for _ in range(10):
```

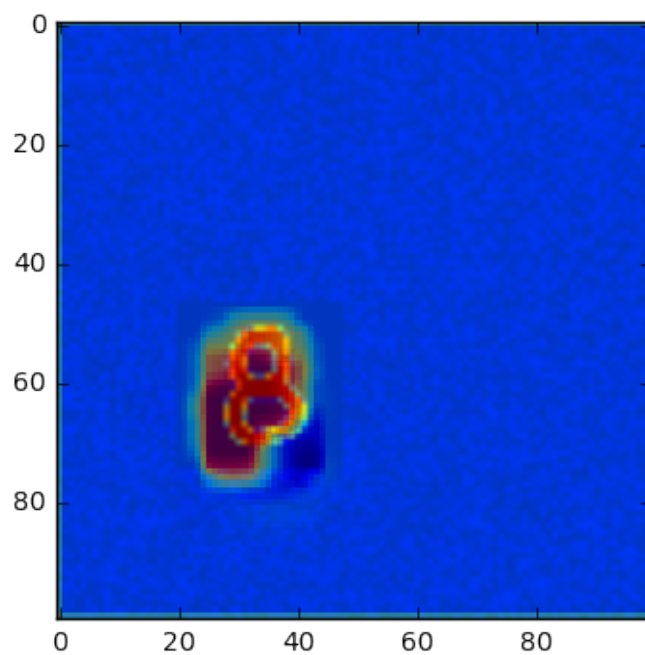
```
print '-----'*5
imgs, lbls = utils.get_batch('test', 1, back_size, .1).next()

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

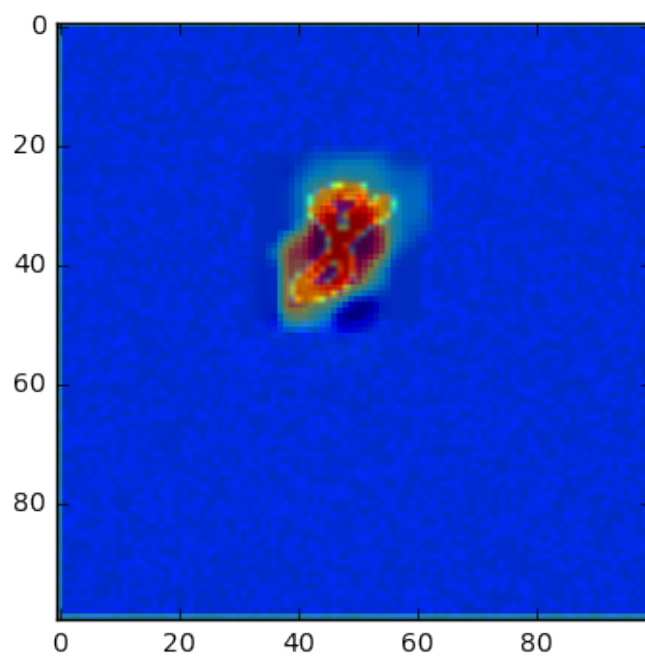
prediction is : 9 with 6.649



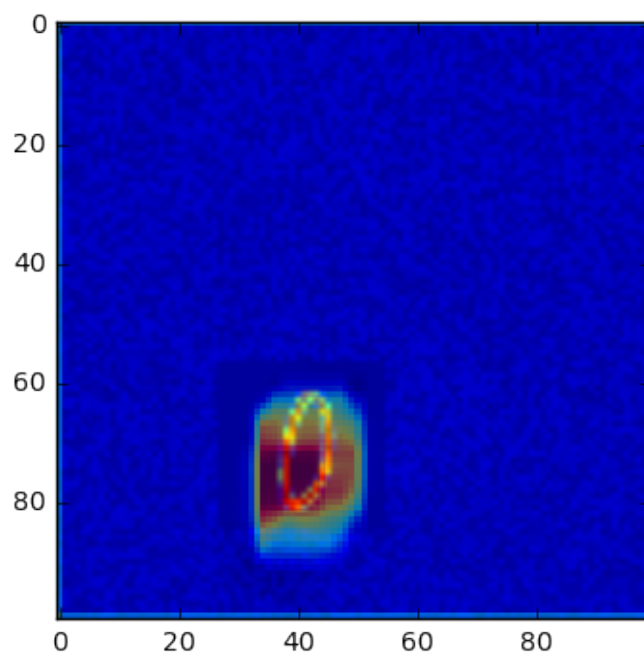
prediction is : 8 with 9.760



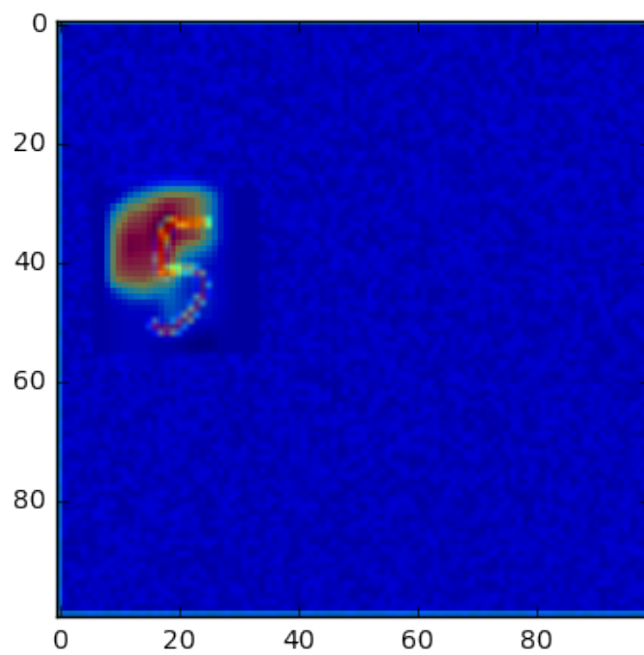
prediction is : 8 with 9.905



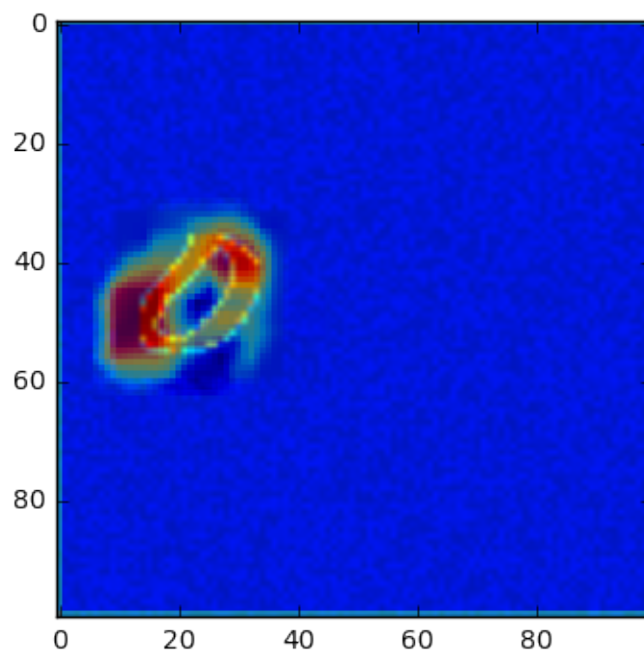
prediction is : 0 with 4.597



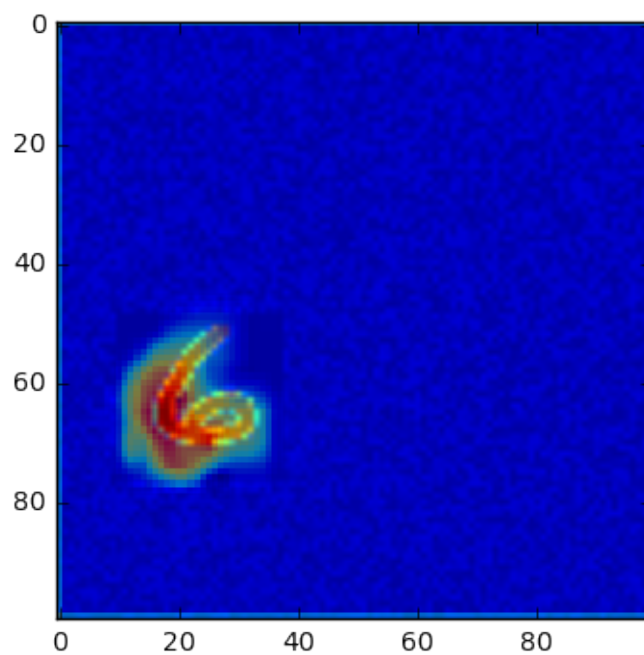
prediction is : 5 with 7.316



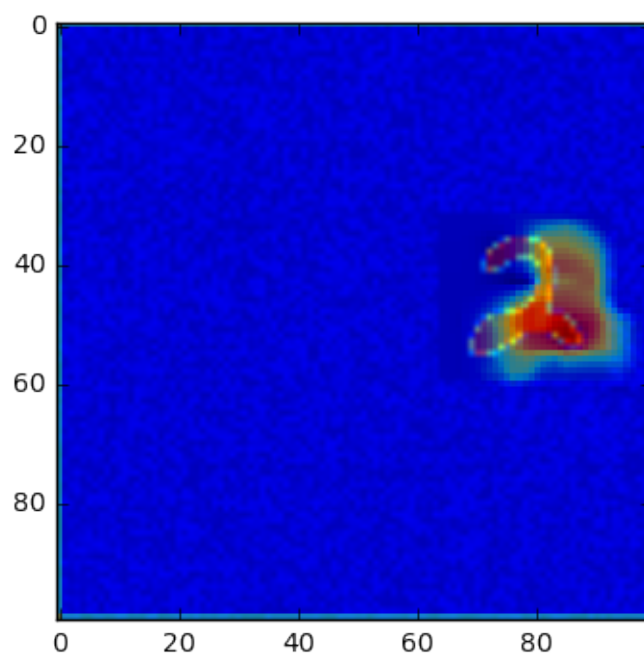
prediction is : 0 with 9.095



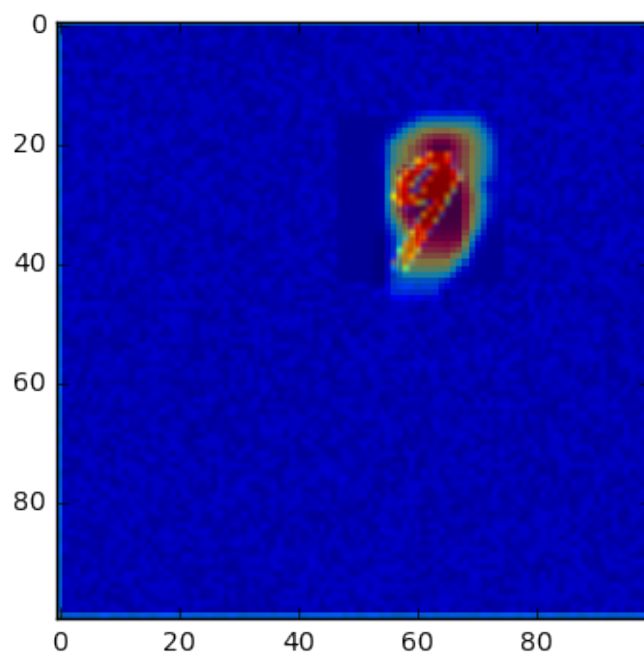
prediction is : 6 with 10.804



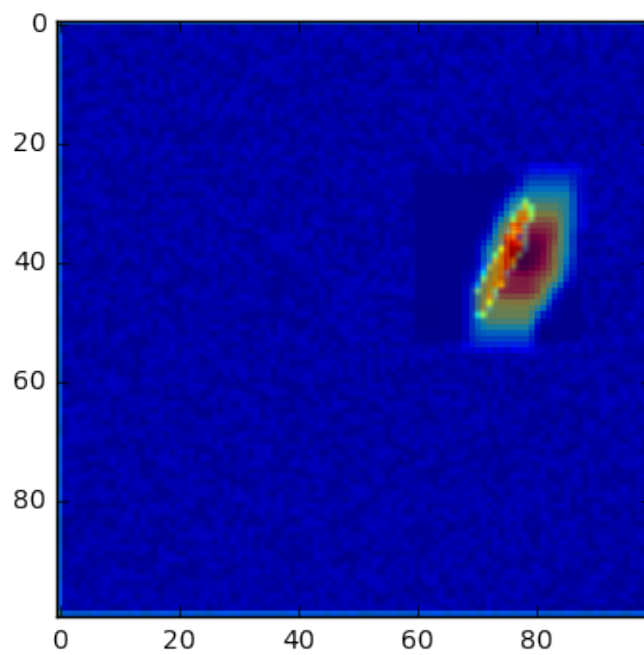
prediction is : 2 with 14.082



prediction is : 9 with 9.942



prediction is : 1 with 7.829



```

In [52]: 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

             imgs = imgs.reshape((-1,100,100,1))
             preds = simple_model.sess.run(simple_model.tf_out, feed_dict={simple_model.x: imgs})
             pred = np.argmax(preds[-1])

             accuracy = sum(preds.argmax(axis=1) == lbl) / float(len(preds))
             accuracies.append(accuracy)

         print sum(accuracies)/len(accuracies)

0.99328125

```

```

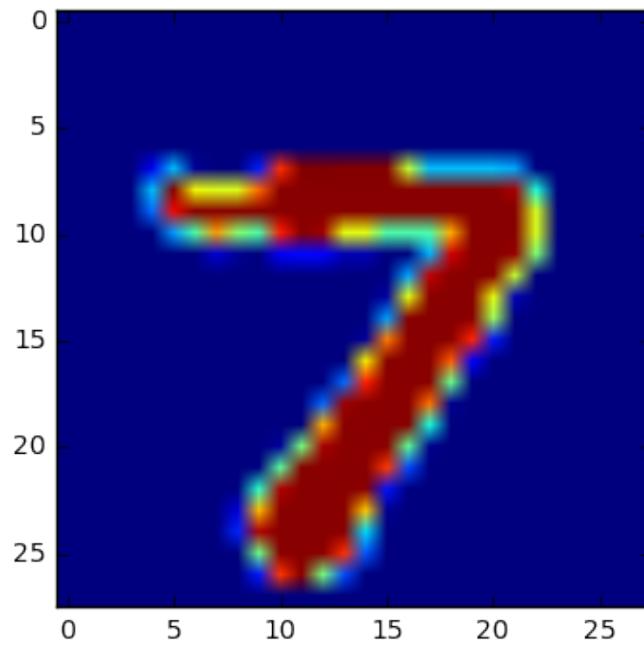
In [53]: digit, lbl = utils.get_batch('test', 1, .1).next()
         digit = digit[0].reshape(28,28)
         img = np.random.random((100,100))*0.1
         height = img.shape[0]
         width = img.shape[1]
         box_size = 28
         step_size = 10
         n_x_boxes = (width - box_size) / step_size + 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):

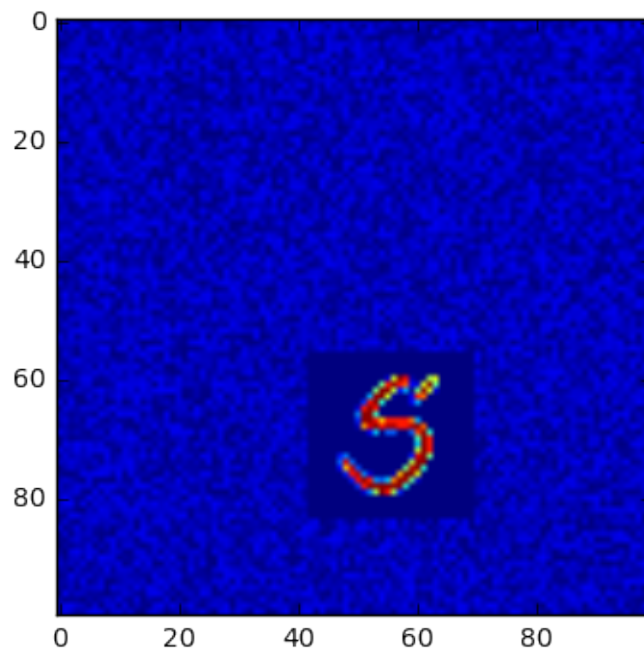
```

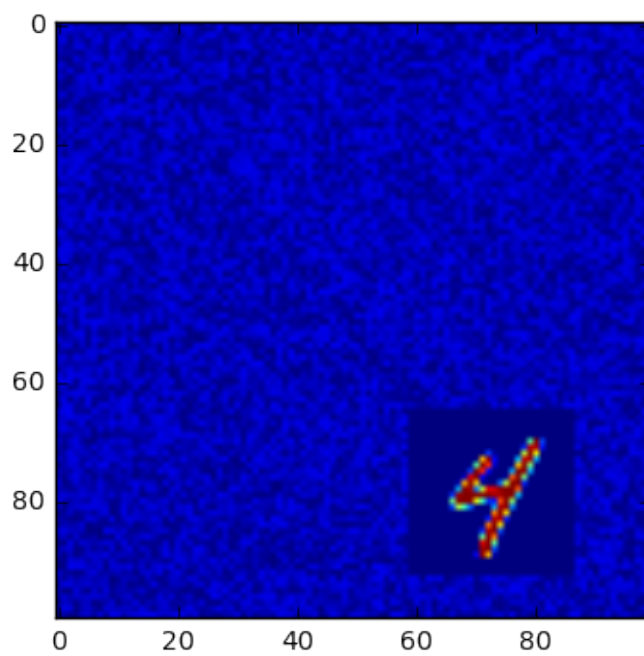
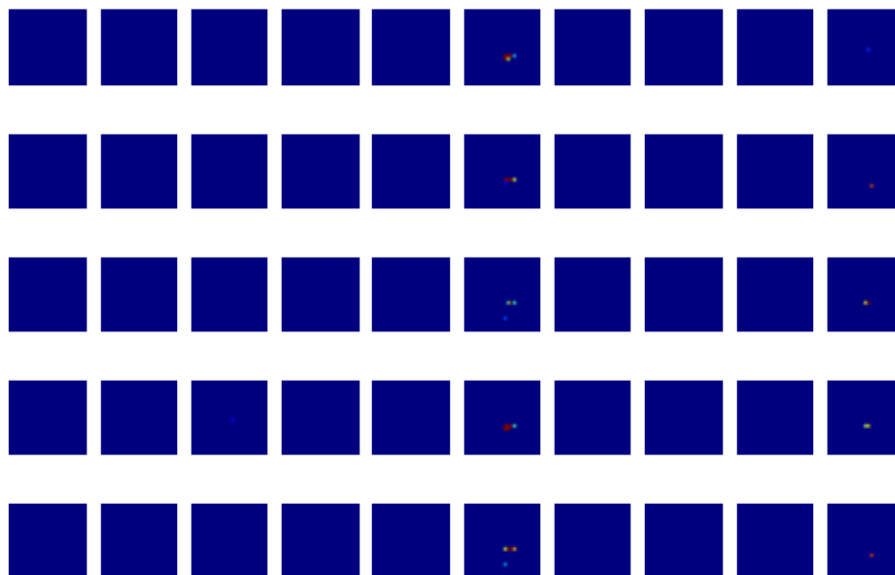
```
[7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7  
 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7]  
100.0
```

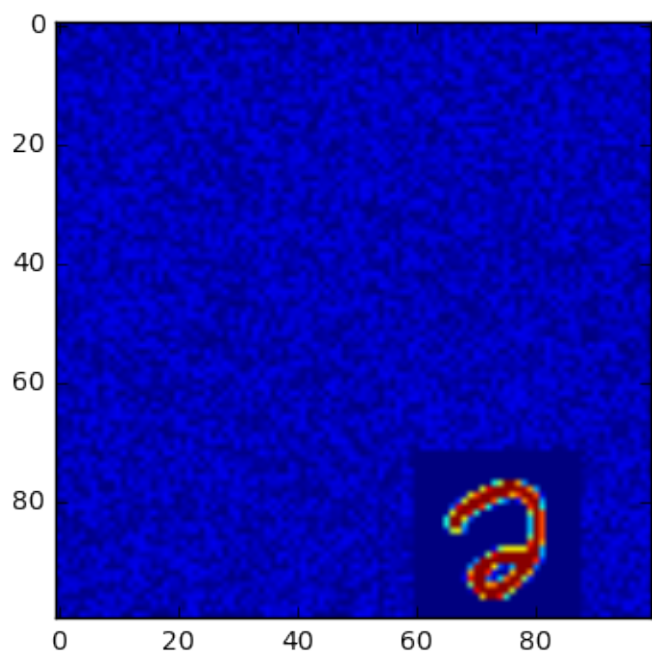
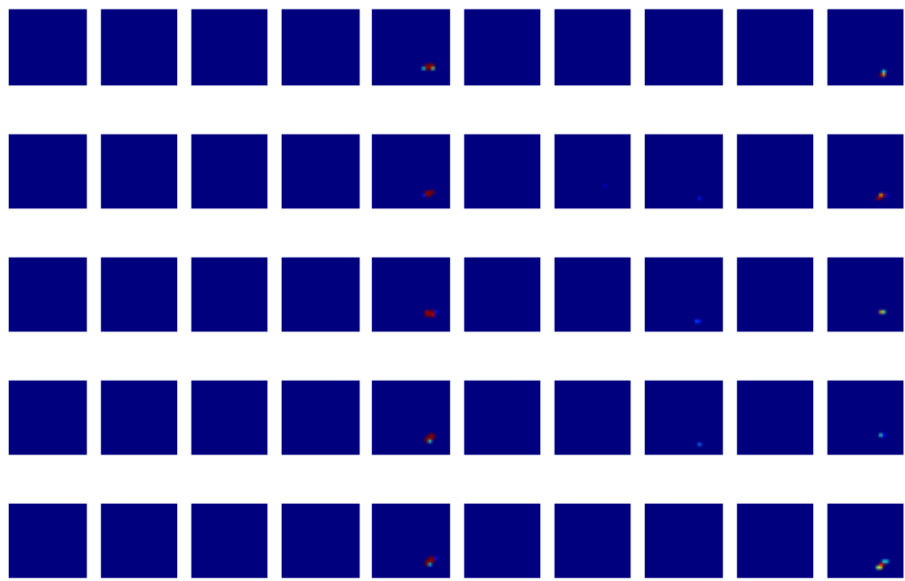


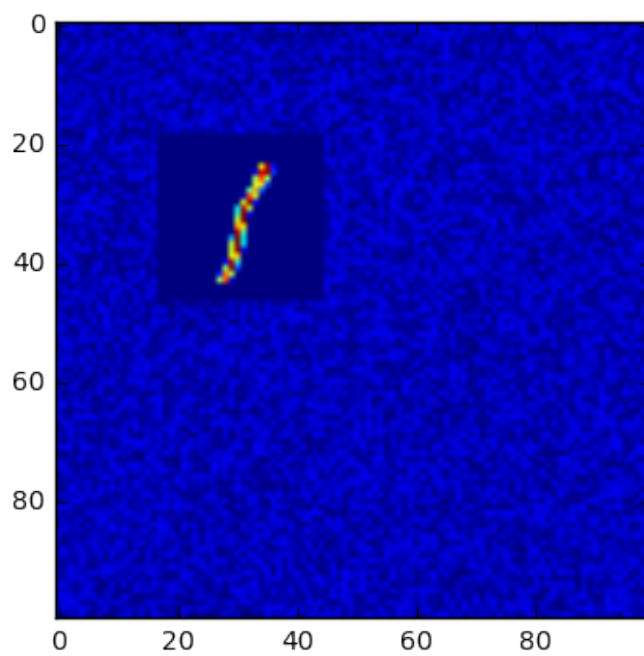
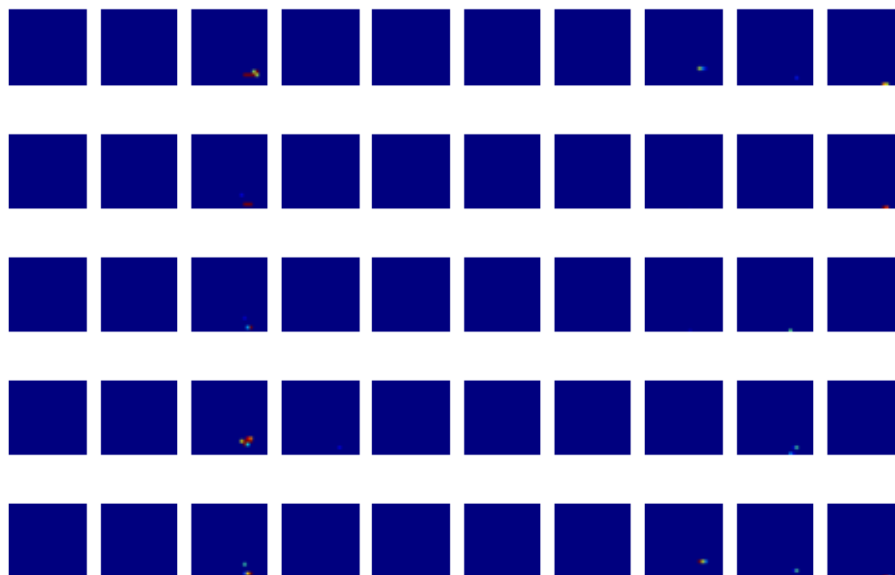


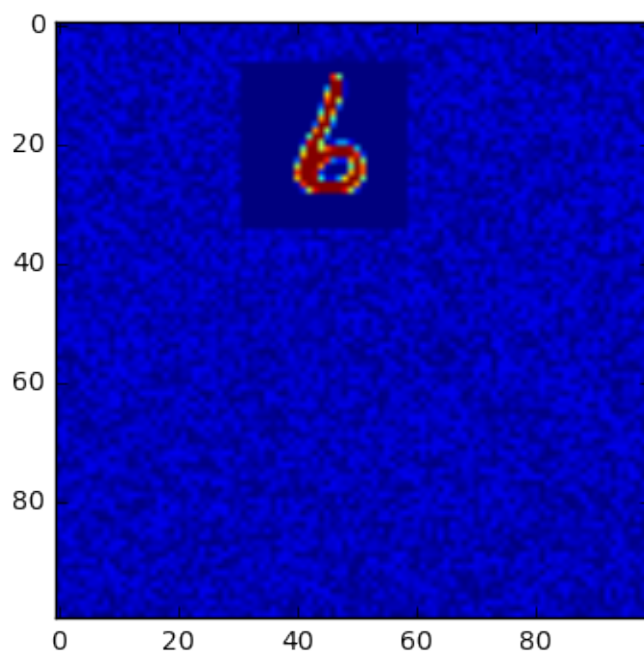
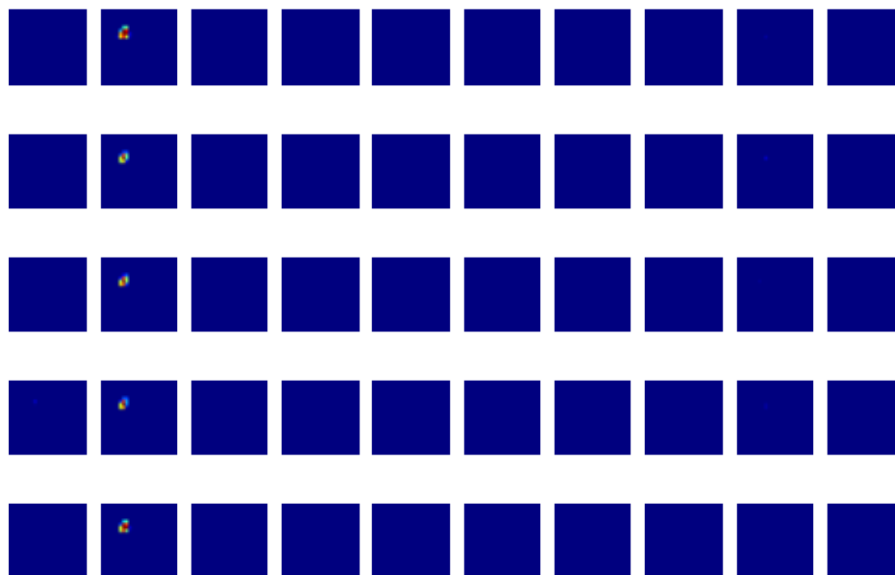
```
In [54]: for _ in range(10):  
         simple_model.plot_classes_maps(100, n_CCN)
```

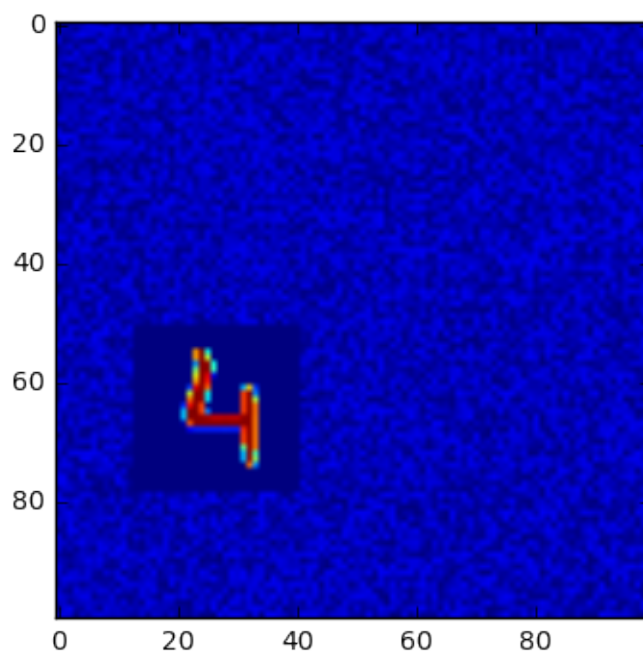
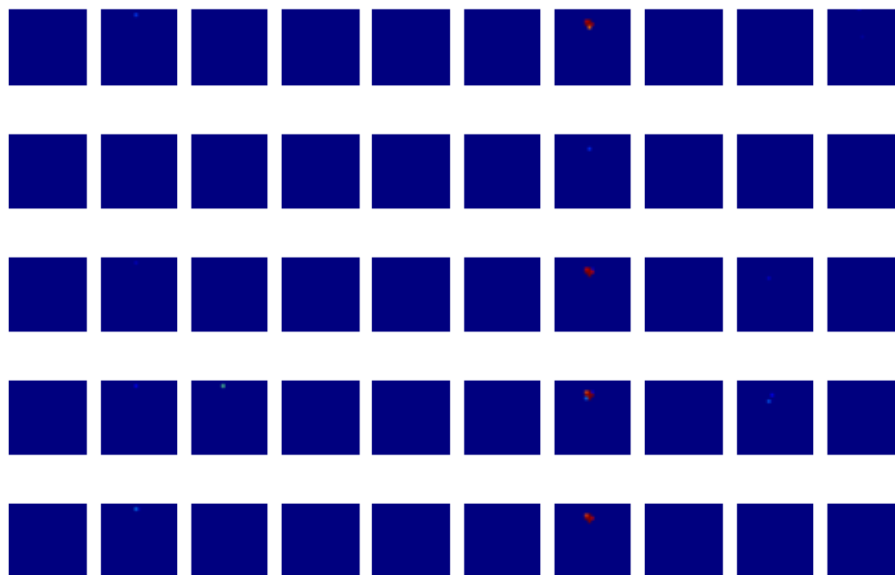


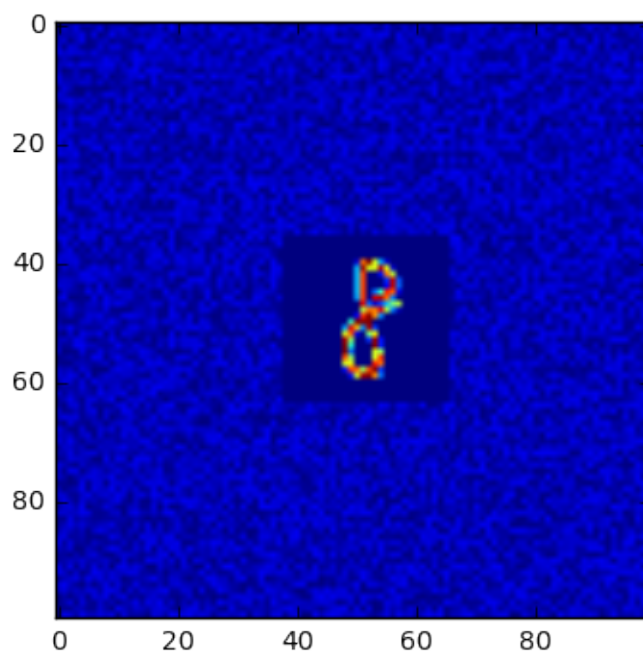
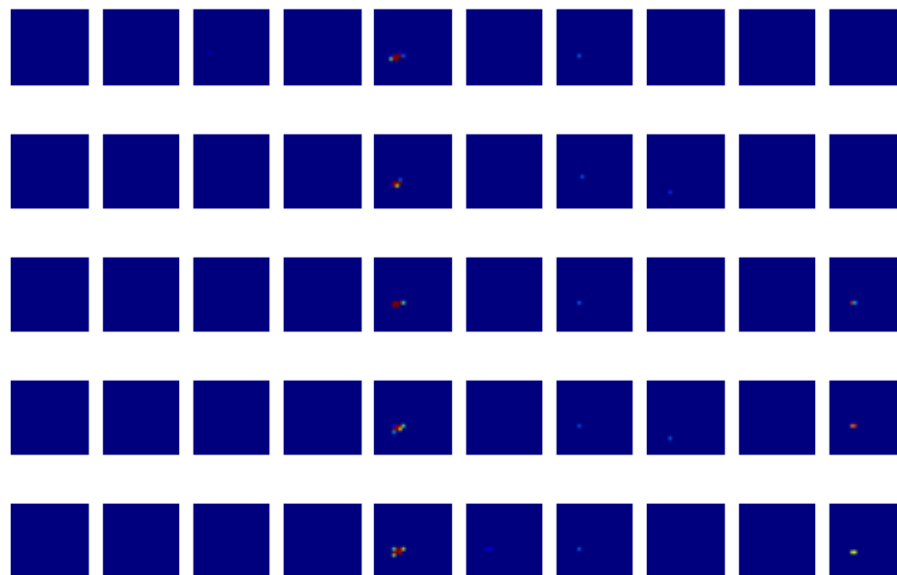


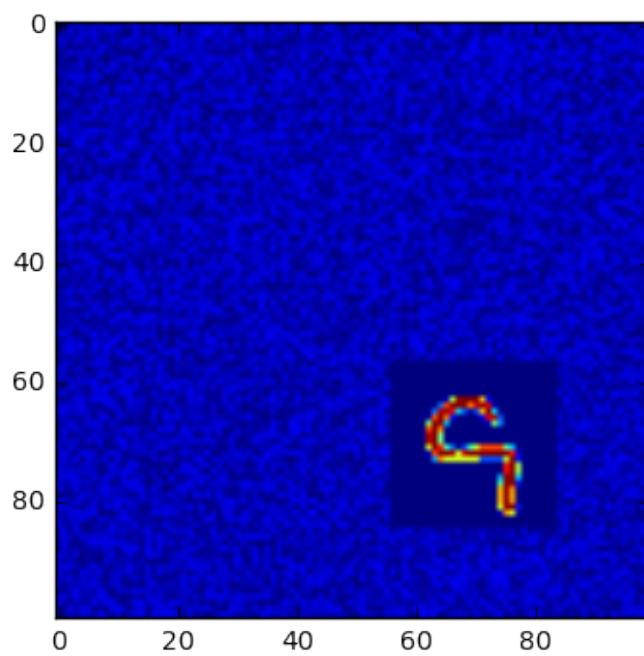
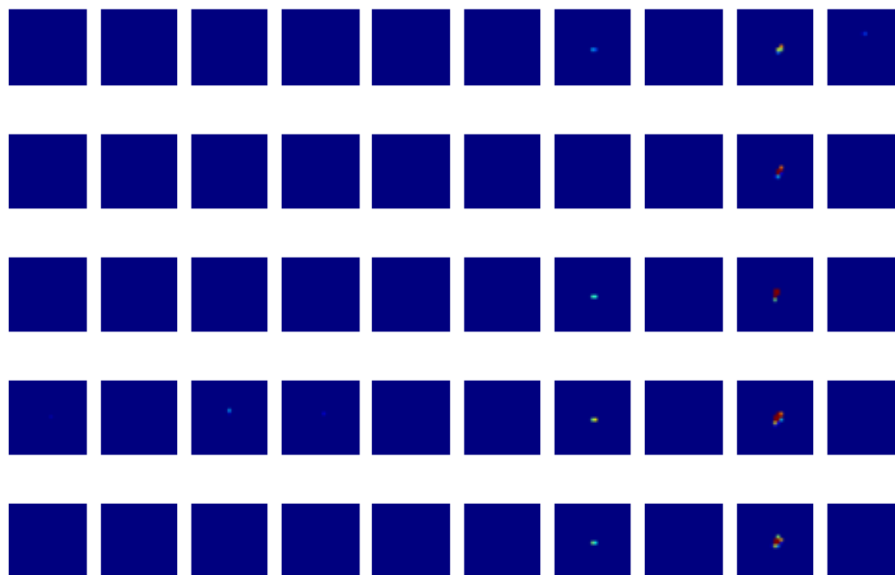


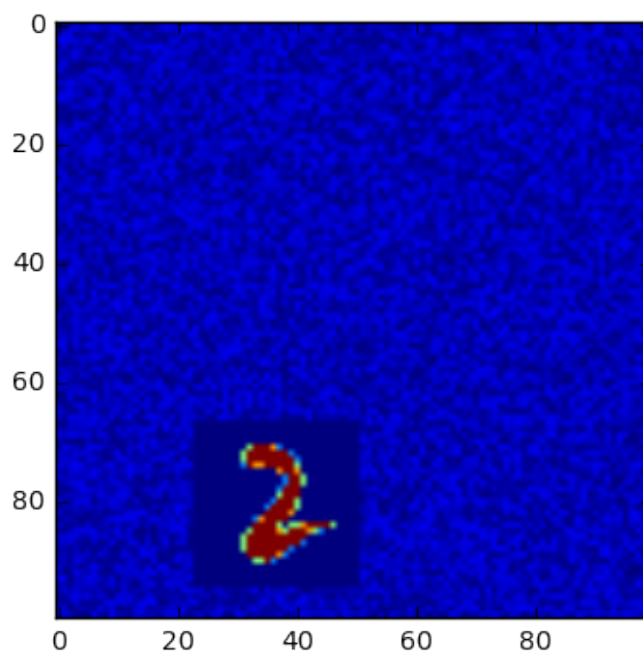
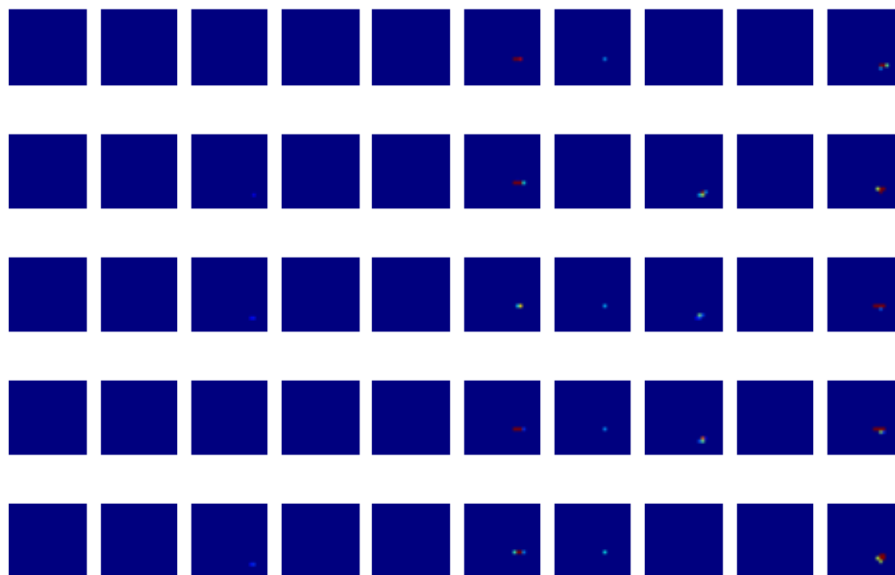


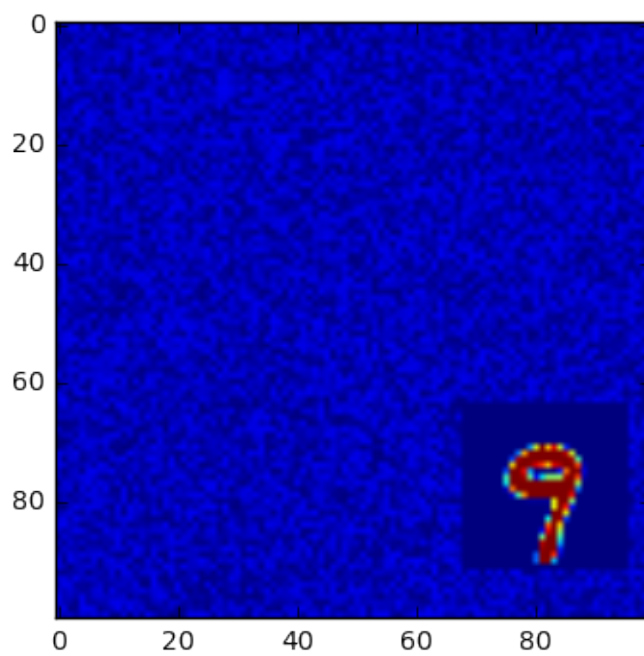
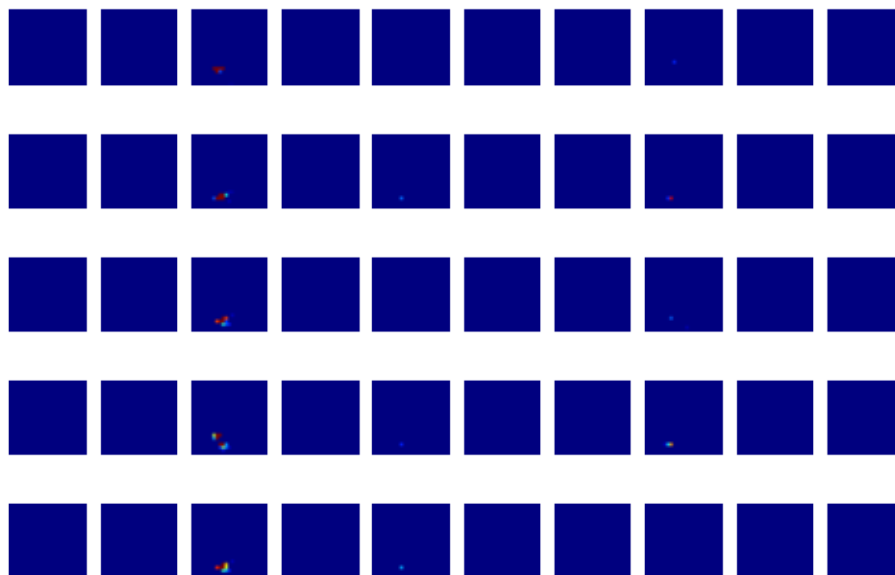


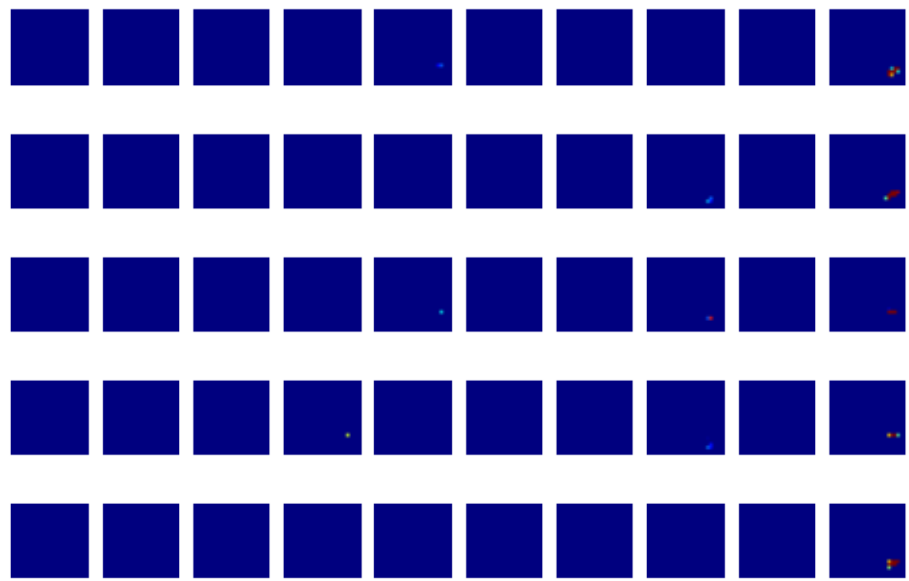












In []: