

show_results

November 4, 2016

1 Load model

1.1 Model

```
conv1_1 = new_conv_layer( image, [11, 11, 1 , 16], "conv1_1" ) conv1_2 = new_conv_layer(
conv1_1, [11, 11, 16, 16], "conv1_2" )
conv2_1 = new_conv_layer(conv1_2, [11, 11, 16, 16], "conv2_1") conv2_2 =
new_conv_layer(conv2_1, [11, 11, 16, 16], "conv2_2")
gap = tf.reduce_mean( conv2_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 [15]: 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
lr_decay    = .9
back_size   = 100
noise       = .1
crop_pos    = (10,10)
n_CCN       = 5
```

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

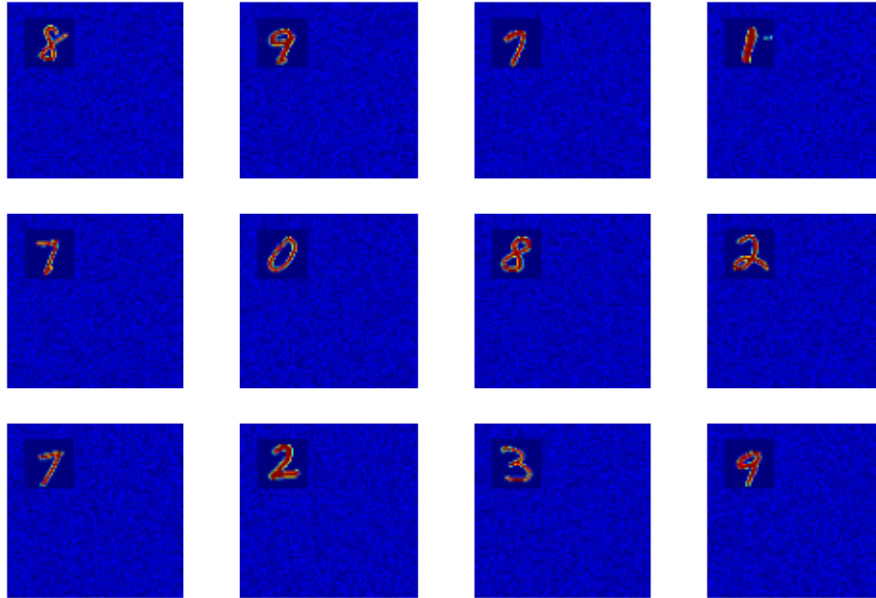
```
In [16]: """Plot training samples"""
batch = utils.get_batch('train', im_size=back_size, noise=noise, crop_pos=

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 [17]: """Perform training"""
        gen = training_generator(lr=lr, lr_decay=lr_decay, back_size=back_size, no
        for _ in range(25):
            _,accs = gen.next()
            print "max acc so far : "+str(max(accs)*100)

```

```

***** EPOCH 0 *****
1-Loss on testset is 3.003971
1-Accuracy now is 17.22
2-Loss on testset is 2.904872
2-Accuracy now is 17.71
lr now is 0.00450
max acc so far : 17.22

```

```

***** EPOCH 1 *****
1-Loss on testset is 23.401737
1-Accuracy now is 10.10
2-Loss on testset is 0.792909
2-Accuracy now is 84.98
lr now is 0.00405

```

max acc so far : 17.22

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

1-Loss on testset is 77.422287

1-Accuracy now is 9.74

2-Loss on testset is 0.310768

2-Accuracy now is 96.95

lr now is 0.00365

max acc so far : 17.22

***** EPOCH 3 *****

1-Loss on testset is 16.258734

1-Accuracy now is 16.01

2-Loss on testset is 0.288334

2-Accuracy now is 97.25

lr now is 0.00328

max acc so far : 17.22

***** EPOCH 4 *****

1-Loss on testset is 16.353914

1-Accuracy now is 9.74

2-Loss on testset is 0.245318

2-Accuracy now is 98.28

lr now is 0.00295

max acc so far : 17.22

***** EPOCH 5 *****

1-Loss on testset is 0.314475

1-Accuracy now is 97.76

2-Loss on testset is 0.266281

2-Accuracy now is 97.30

lr now is 0.00266

max acc so far : 97.76

***** EPOCH 6 *****

1-Loss on testset is 0.299648

1-Accuracy now is 97.77

2-Loss on testset is 0.205482

2-Accuracy now is 98.80

lr now is 0.00239

max acc so far : 97.77

***** EPOCH 7 *****

1-Loss on testset is 0.272260

1-Accuracy now is 98.00

2-Loss on testset is 0.210082

2-Accuracy now is 98.58

lr now is 0.00215

max acc so far : 98.0

***** EPOCH 8 *****

1-Loss on testset is 0.290396

1-Accuracy now is 98.51

2-Loss on testset is 0.183244

2-Accuracy now is 99.23

lr now is 0.00194

max acc so far : 98.51

***** EPOCH 9 *****

1-Loss on testset is 0.888374

1-Accuracy now is 86.48

2-Loss on testset is 0.302338

2-Accuracy now is 96.08

lr now is 0.00174

max acc so far : 98.51

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

1-Loss on testset is 6.626028

1-Accuracy now is 19.35

2-Loss on testset is 0.190548

2-Accuracy now is 98.91

lr now is 0.00157

max acc so far : 98.51

***** EPOCH 11 *****

1-Loss on testset is 3.915804

1-Accuracy now is 13.18

2-Loss on testset is 0.188646

2-Accuracy now is 98.70

lr now is 0.00141

max acc so far : 98.51

***** EPOCH 12 *****

1-Loss on testset is 0.276771

1-Accuracy now is 98.76

2-Loss on testset is 0.174057

2-Accuracy now is 99.29

lr now is 0.00127

max acc so far : 98.76

***** EPOCH 13 *****

1-Loss on testset is 0.338936

1-Accuracy now is 97.93

2-Loss on testset is 0.189160

2-Accuracy now is 98.87

lr now is 0.00114

max acc so far : 98.76

***** EPOCH 14 *****

1-Loss on testset is 0.399669

1-Accuracy now is 96.21

2-Loss on testset is 0.168163

2-Accuracy now is 99.39

lr now is 0.00103

max acc so far : 98.76

***** EPOCH 15 *****

1-Loss on testset is 0.366408

1-Accuracy now is 98.45

2-Loss on testset is 0.176792

2-Accuracy now is 99.29

lr now is 0.00093

max acc so far : 98.76

***** EPOCH 16 *****

1-Loss on testset is 4.574247

1-Accuracy now is 22.33

2-Loss on testset is 0.177397

2-Accuracy now is 99.20

lr now is 0.00083

max acc so far : 98.76

***** EPOCH 17 *****

1-Loss on testset is 0.410534

1-Accuracy now is 97.13

2-Loss on testset is 0.175613

2-Accuracy now is 99.28

lr now is 0.00075

max acc so far : 98.76

***** EPOCH 18 *****

1-Loss on testset is 0.490249

1-Accuracy now is 94.56

2-Loss on testset is 0.170913

2-Accuracy now is 99.27

lr now is 0.00068

max acc so far : 98.76

***** EPOCH 19 *****

1-Loss on testset is 0.495712

1-Accuracy now is 95.49

2-Loss on testset is 0.162000

2-Accuracy now is 99.51

lr now is 0.00061

max acc so far : 98.76

***** EPOCH 20 *****

1-Loss on testset is 0.338924

1-Accuracy now is 98.24

2-Loss on testset is 0.162740

2-Accuracy now is 99.32

lr now is 0.00055

max acc so far : 98.76

***** EPOCH 21 *****

1-Loss on testset is 0.348883

1-Accuracy now is 98.42

2-Loss on testset is 0.157413

2-Accuracy now is 99.41

lr now is 0.00049

max acc so far : 98.76

***** EPOCH 22 *****

1-Loss on testset is 0.307564

1-Accuracy now is 98.80

2-Loss on testset is 0.154183

2-Accuracy now is 99.45

lr now is 0.00044

max acc so far : 98.8

***** EPOCH 23 *****

1-Loss on testset is 0.338892

1-Accuracy now is 98.90

2-Loss on testset is 0.157120

2-Accuracy now is 99.34

lr now is 0.00040

max acc so far : 98.9

***** EPOCH 24 *****

1-Loss on testset is 0.349275

1-Accuracy now is 98.75

2-Loss on testset is 0.153915

2-Accuracy now is 99.45

lr now is 0.00036

max acc so far : 98.9

In [18]: back_size = 100

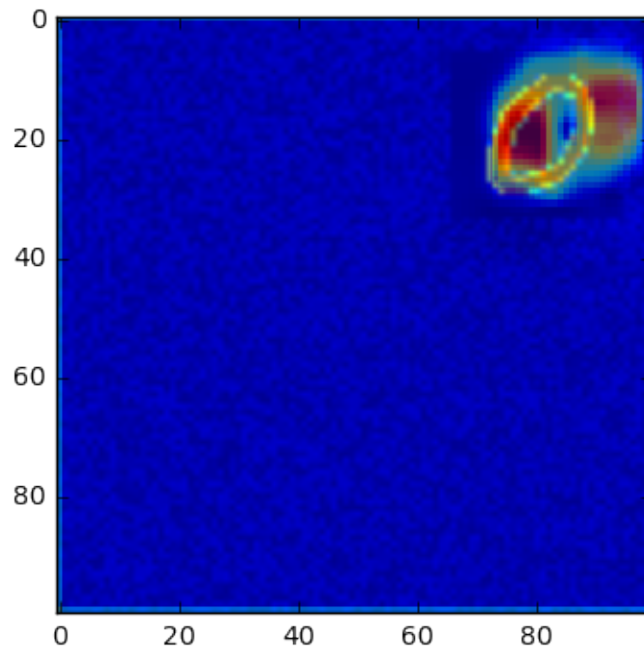
```
for _ in range(10):
```

```
    print '-----'*5
```

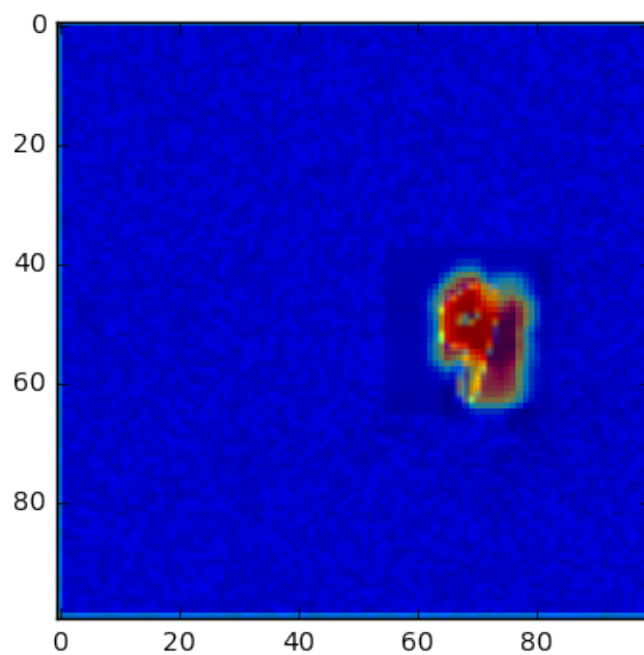
```
    imgs, lbls = utils.get_batch('test', 1, back_size, .1).next()
```

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

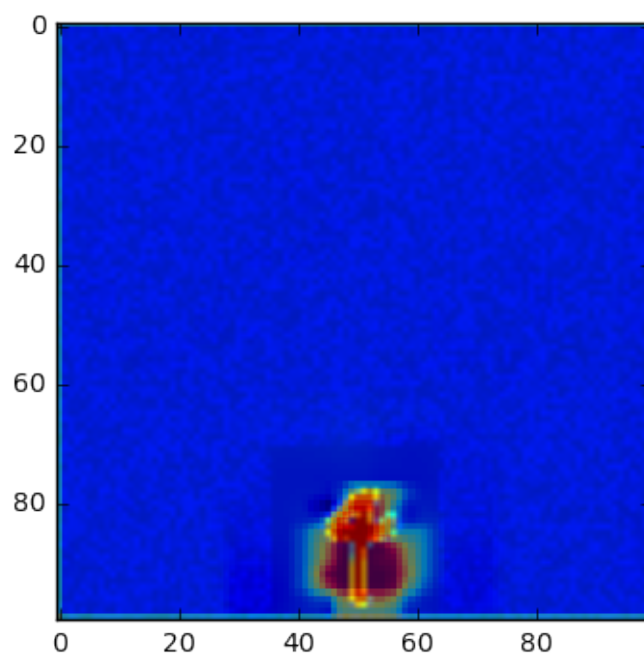
```
prediction is : 0 with 8.177
```



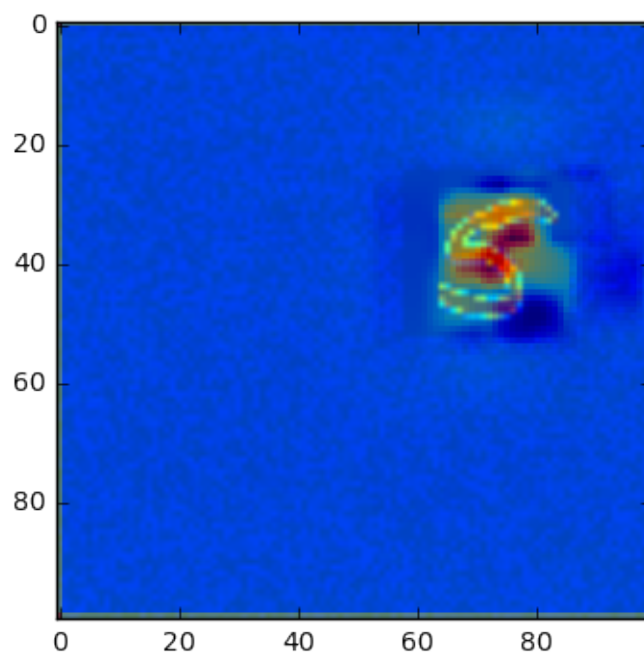
```
prediction is : 9 with 8.858
```



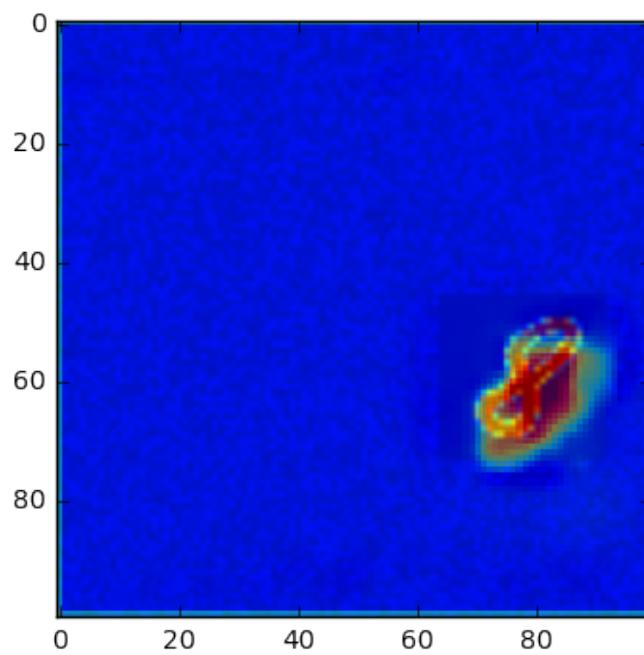
prediction is : 4 with 5.744



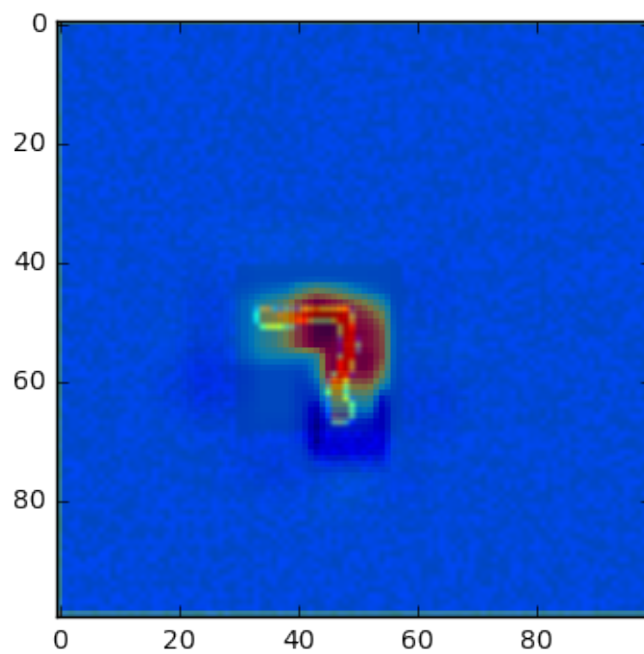
prediction is : 5 with 7.634



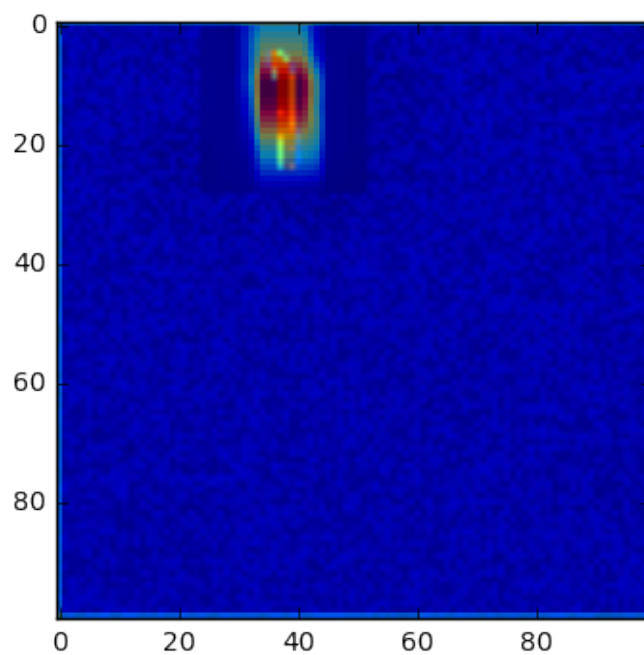
prediction is : 8 with 8.229



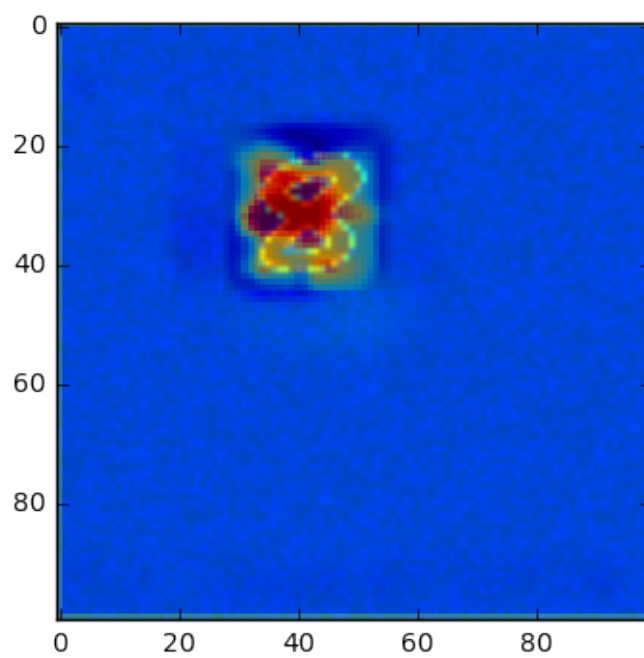
prediction is : 7 with 7.244



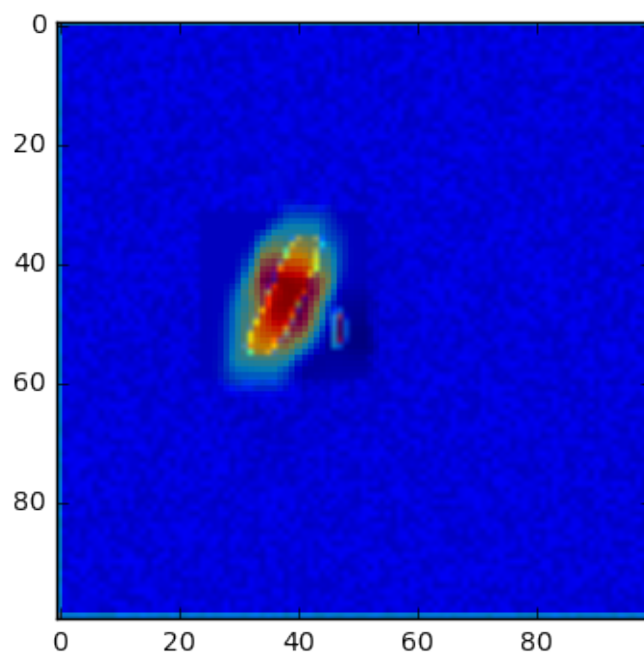
prediction is : 1 with 9.137



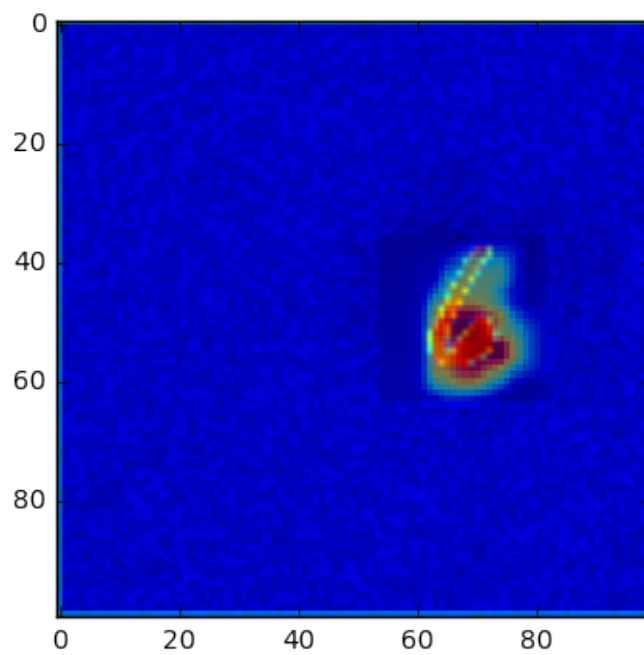
prediction is : 8 with 6.575



prediction is : 1 with 7.938



prediction is : 6 with 9.386



```

In [19]: 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)

```

0.99875

```

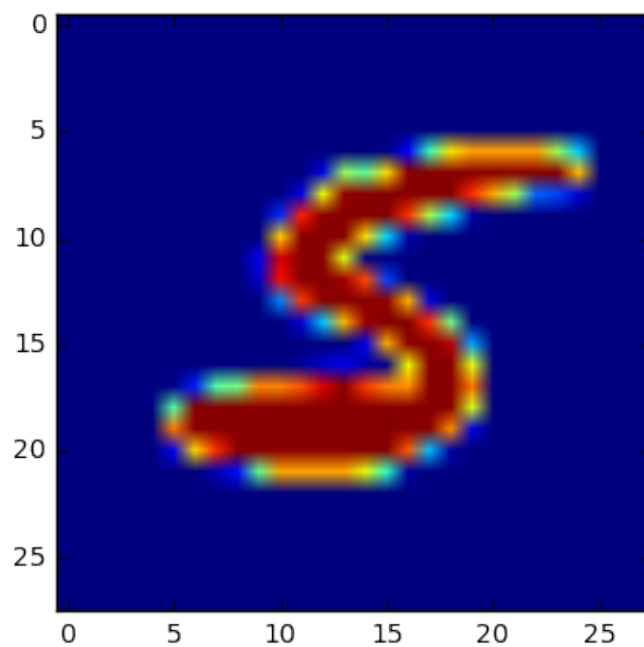
In [20]: digit, lbl = utils.get_batch('test', 1, .1).next()
         digit = digit[0].reshape(28,28)
         img = np.random.random((100,100))*0.1
         height = img.shape[0]
         width = img.shape[1]
         box_size = 28
         step_size = 10
         n_x_boxes = (width - box_size) / step_size + 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):

```

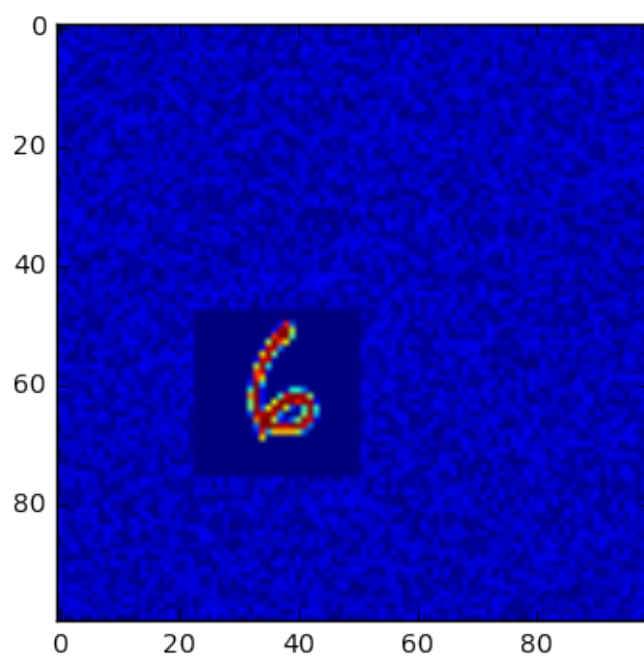
```
[5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5  
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5]
100.0
```

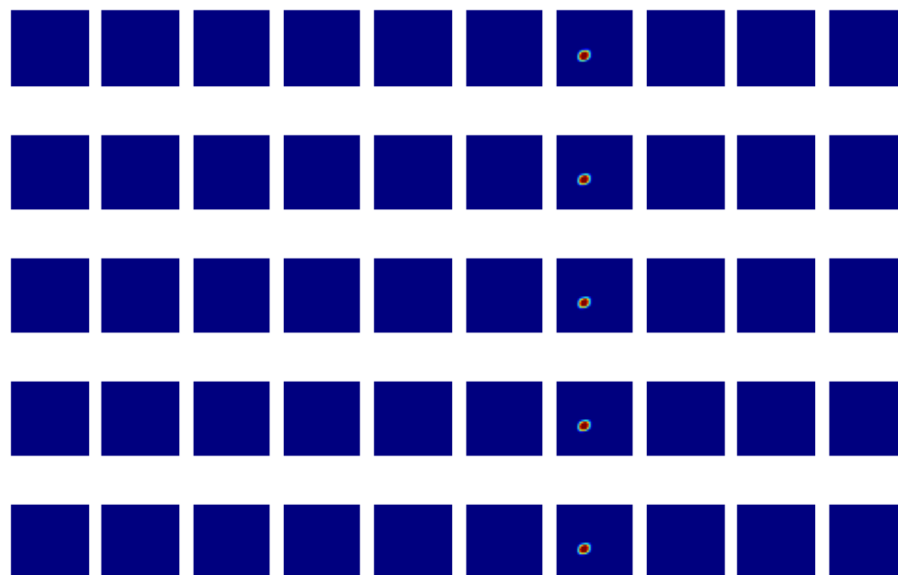




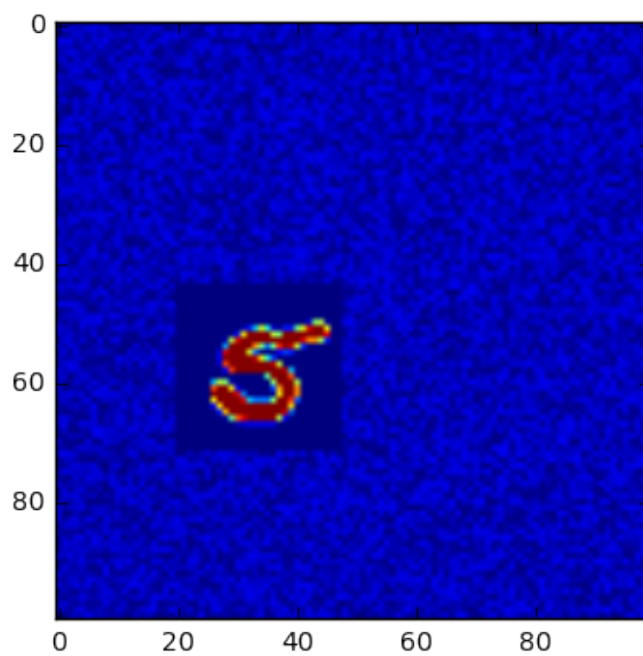
```
In [21]: for _ in range(10):
         simple_model.plot_classes_maps(100, n_CCN)
```

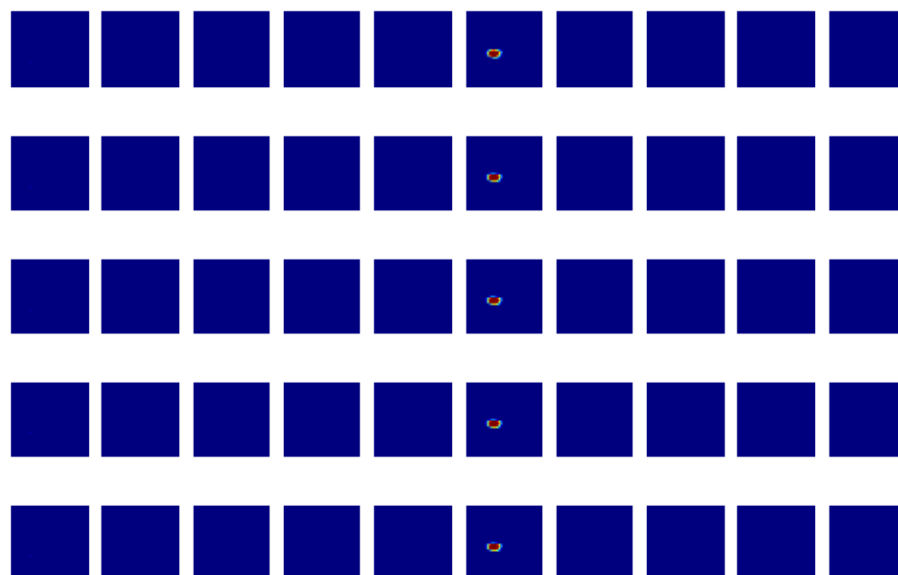
6



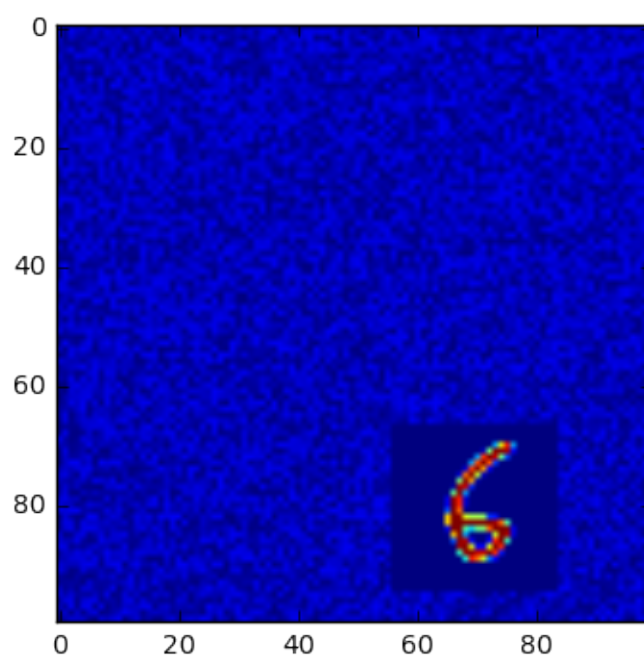


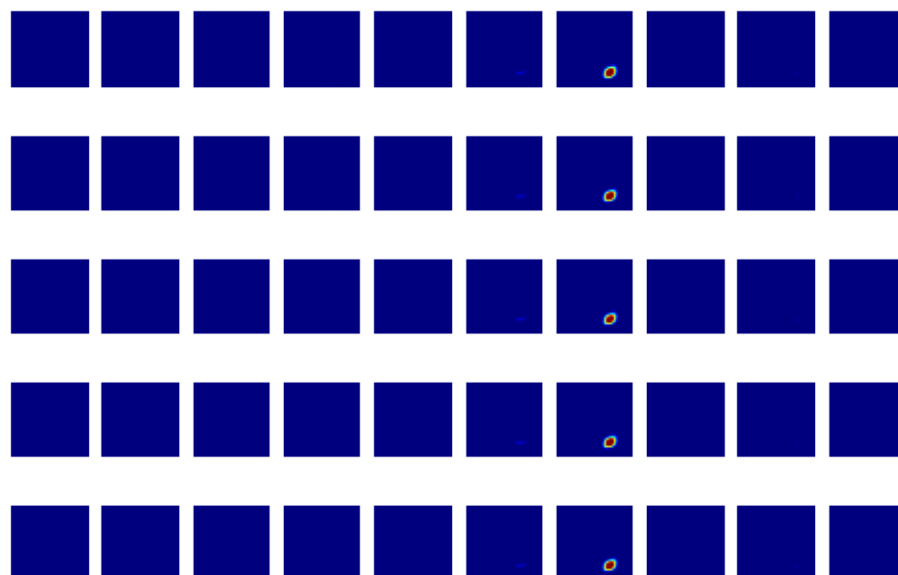
5



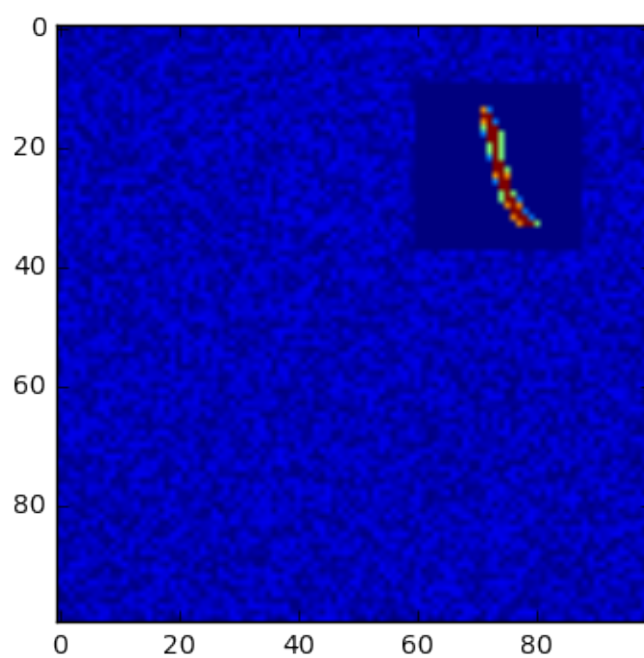


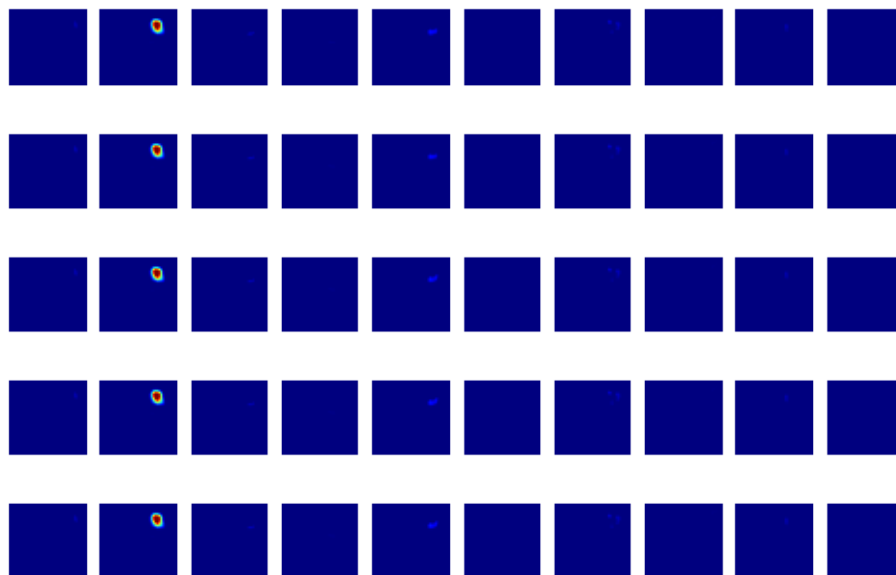
6



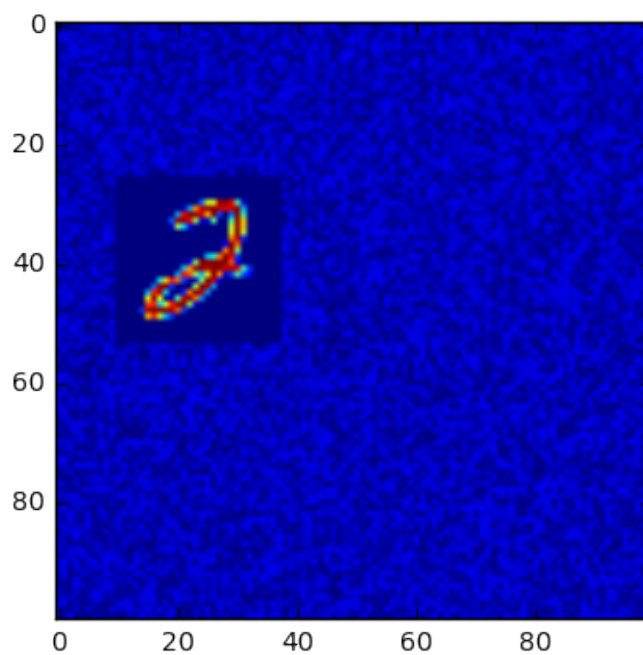


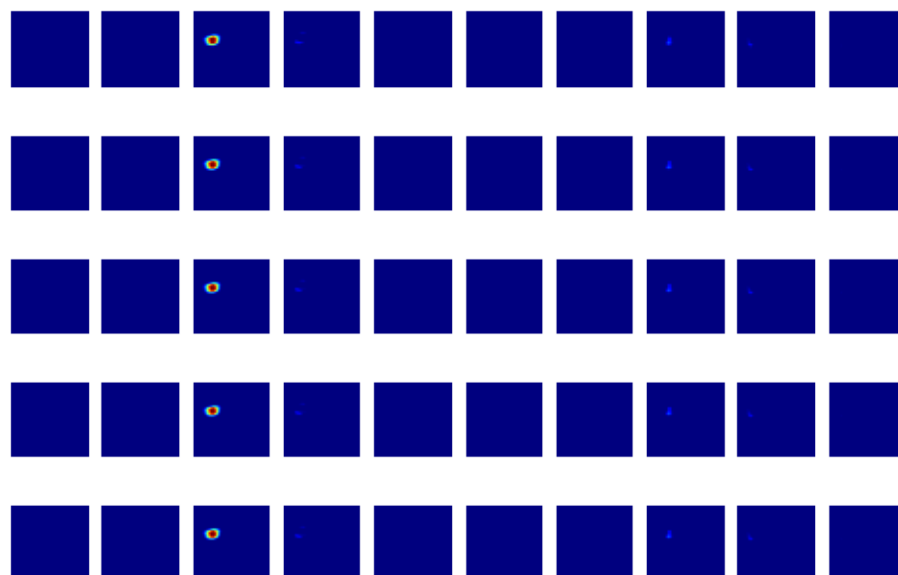
1



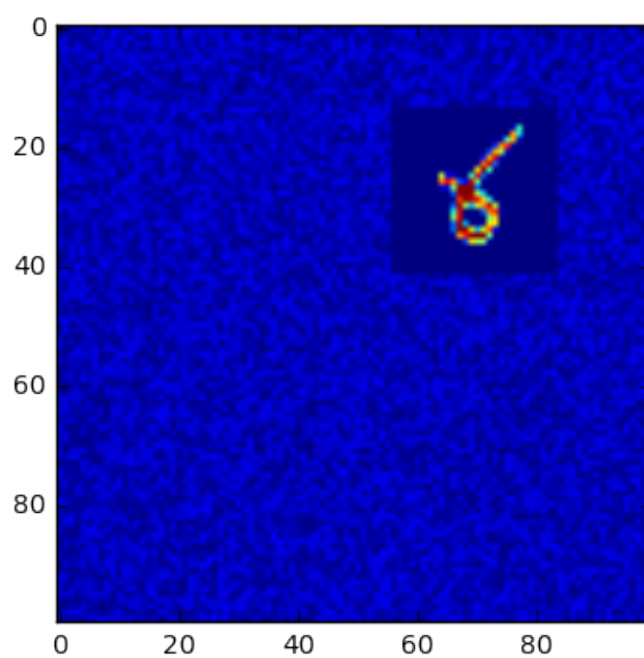


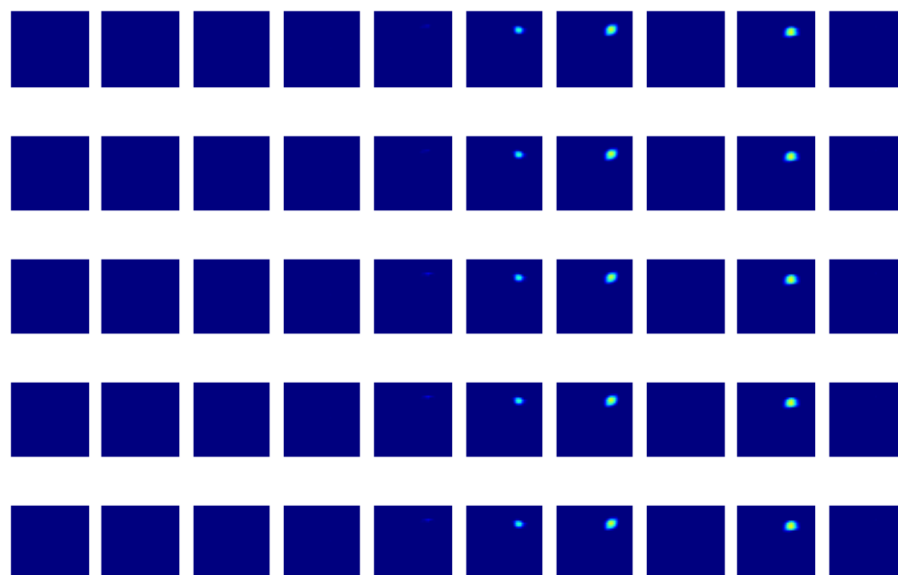
2



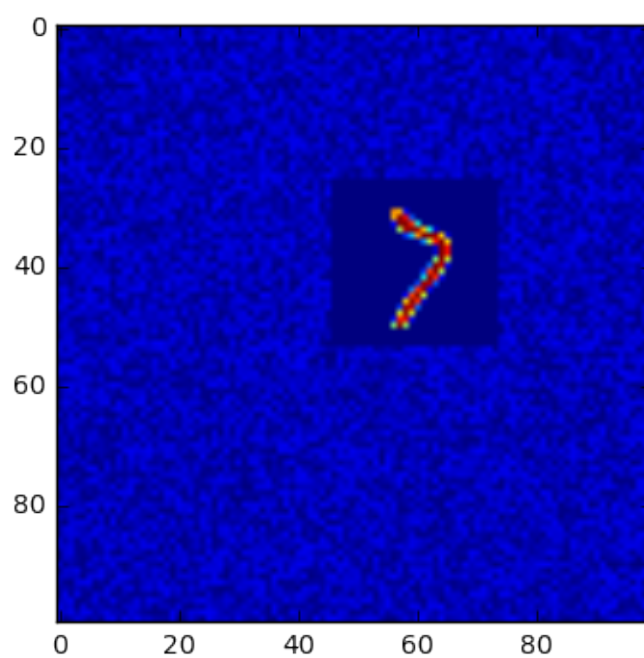


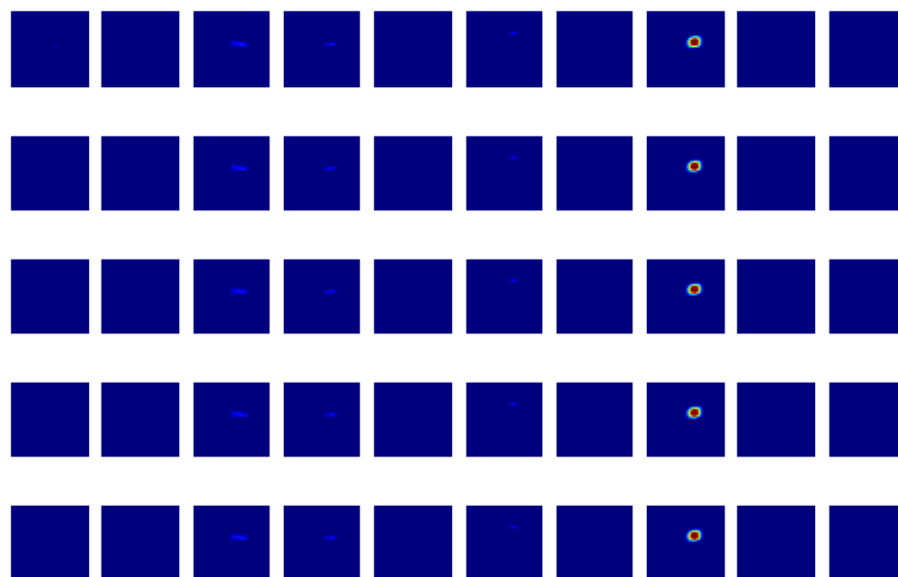
8



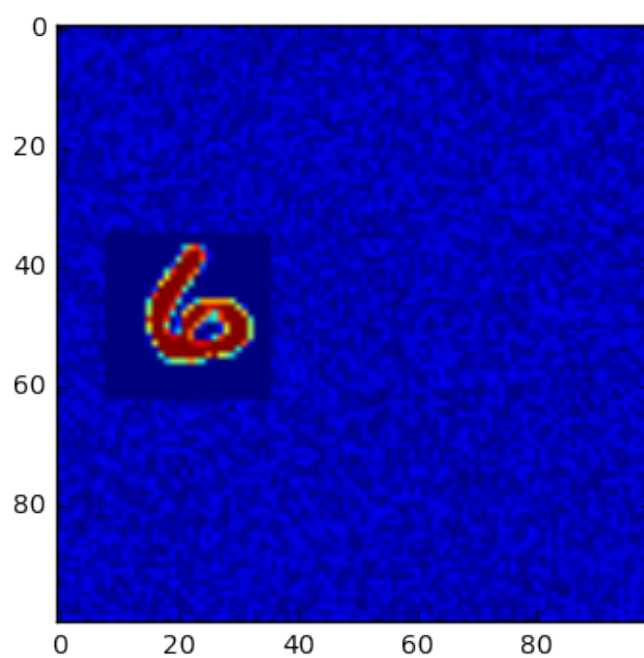


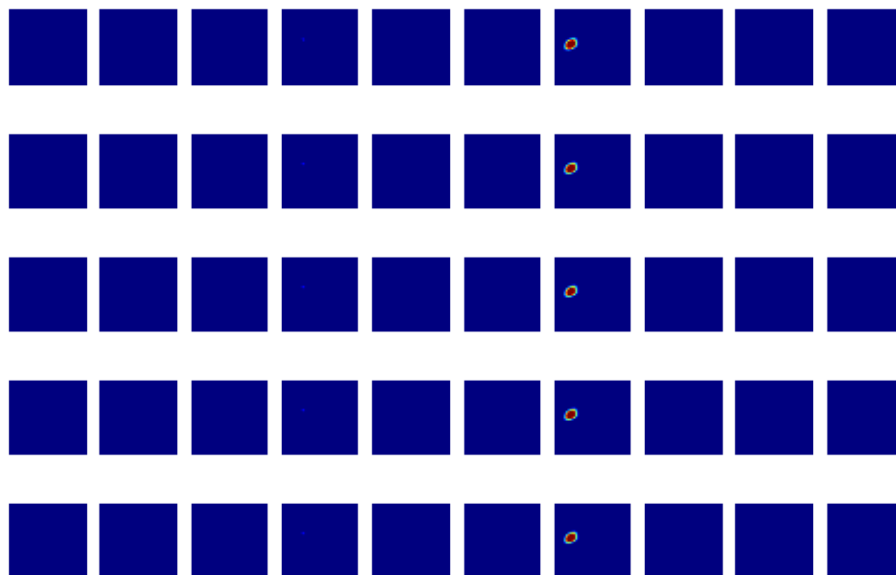
7



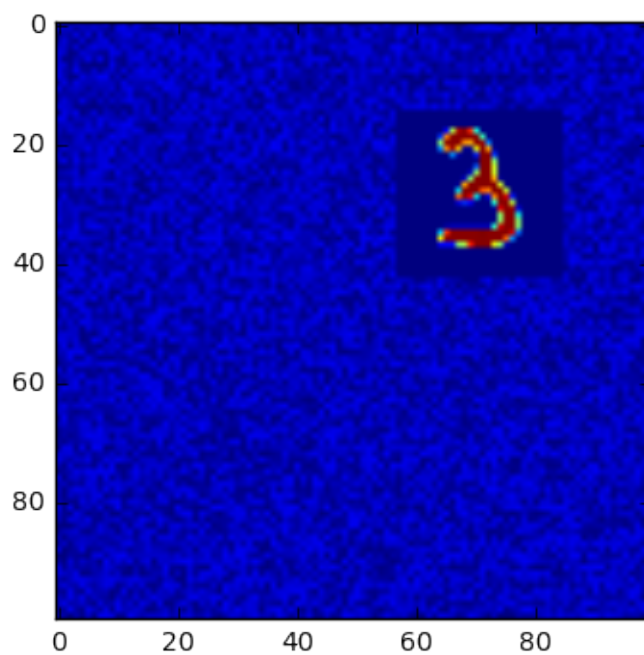


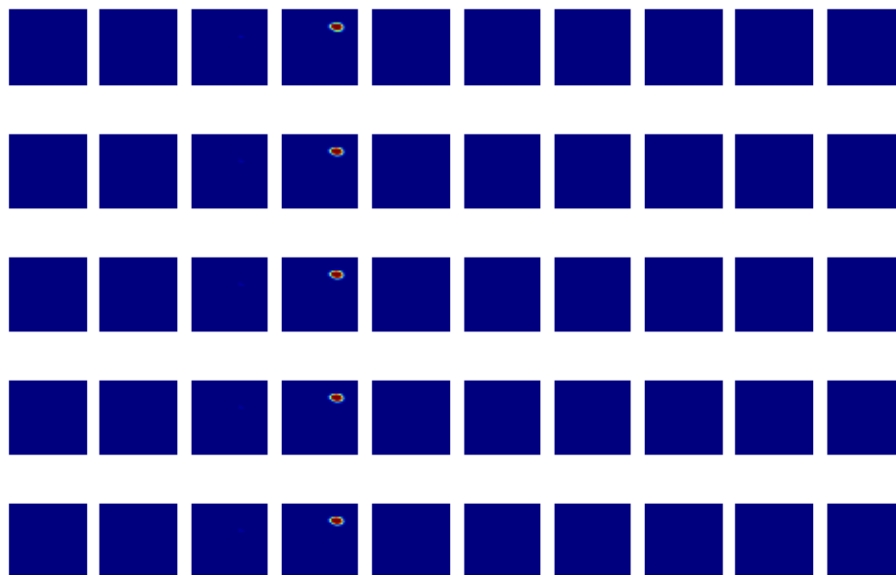
6



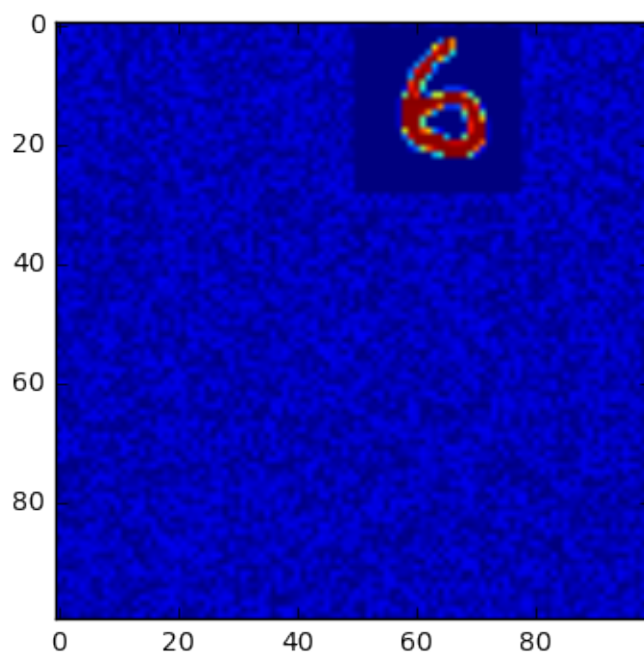


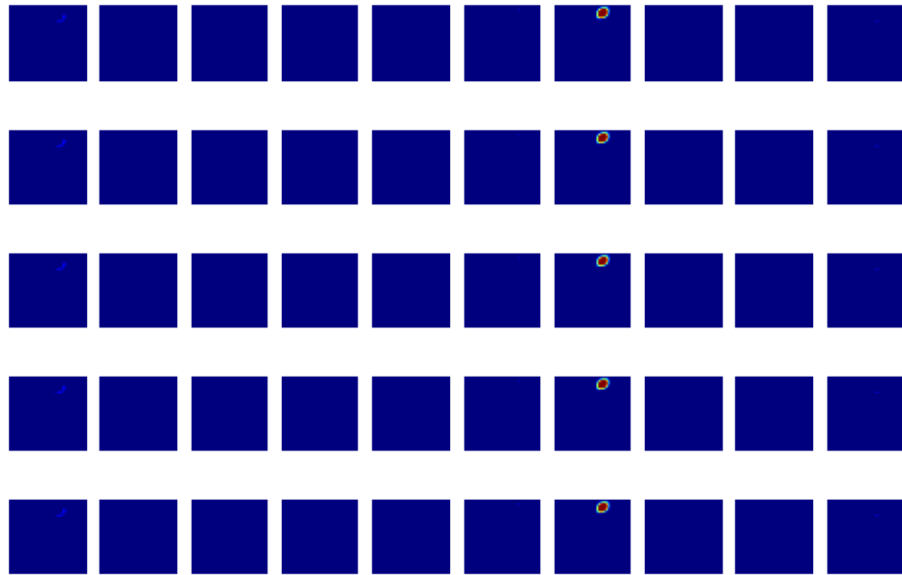
3





6





In []: