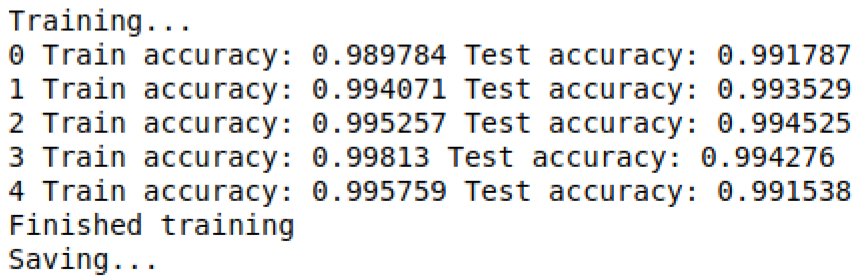
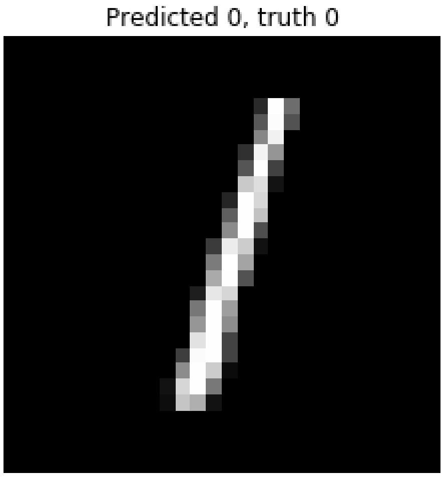
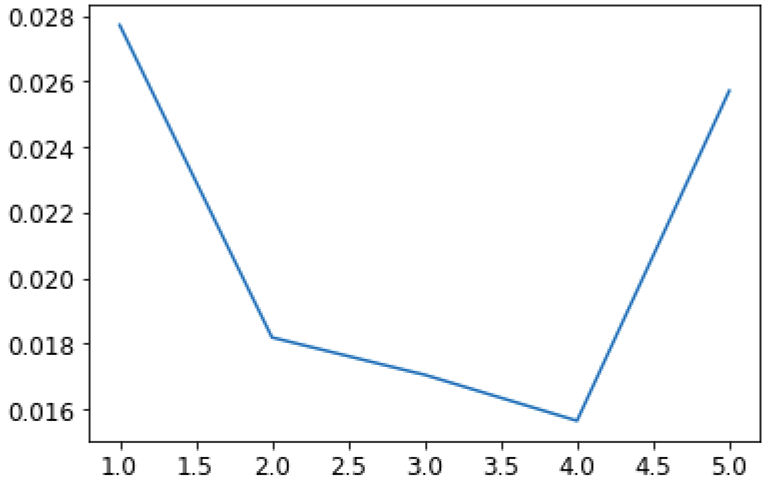
TEST LOSS _1459



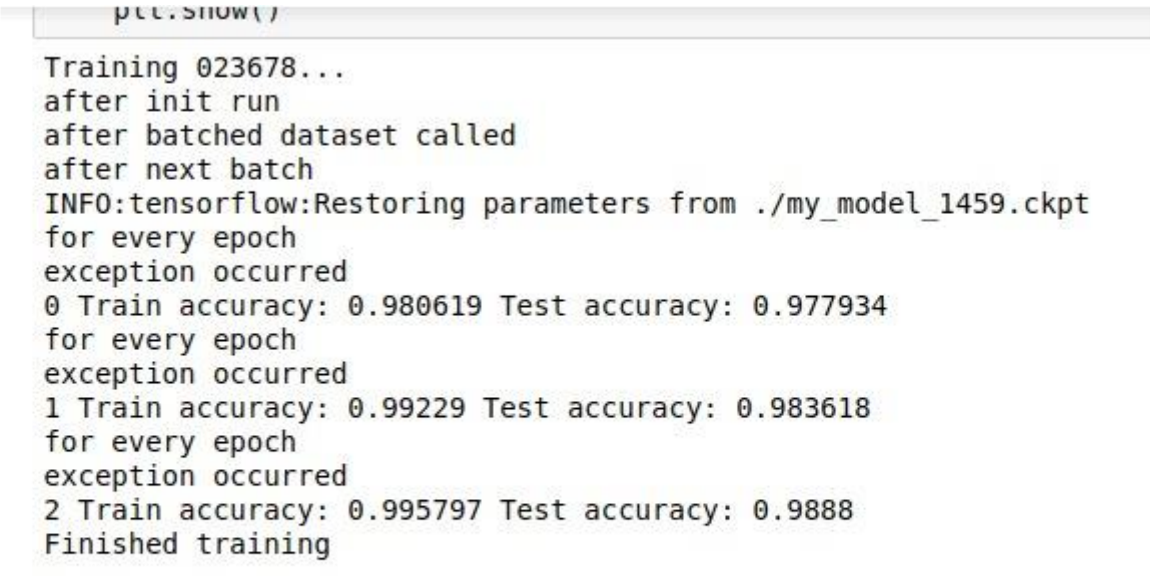
```
Training...
0 Train accuracy: 0.989784 Test accuracy: 0.991787
1 Train accuracy: 0.994071 Test accuracy: 0.993529
2 Train accuracy: 0.995257 Test accuracy: 0.994525
3 Train accuracy: 0.99813 Test accuracy: 0.994276
4 Train accuracy: 0.995759 Test accuracy: 0.991538
Finished training
Saving...
```



Predicted 0, truth 0

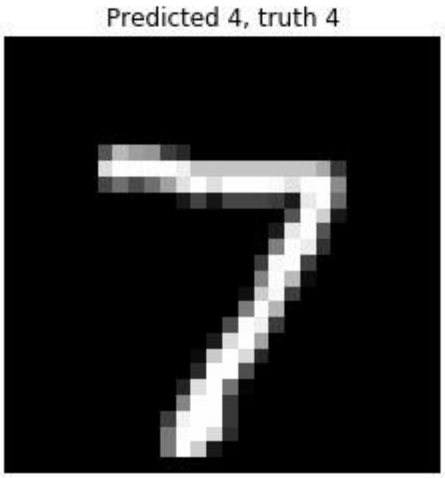


TEST LOSS 023678– NO FREEZE

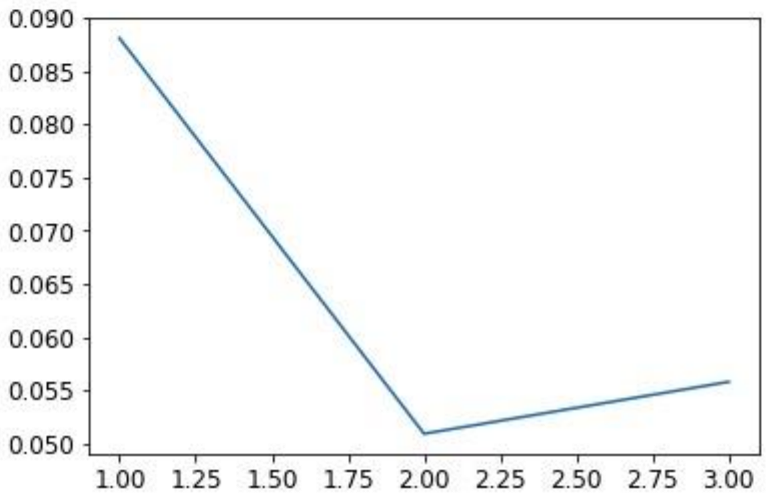


```
    plt.show()
```

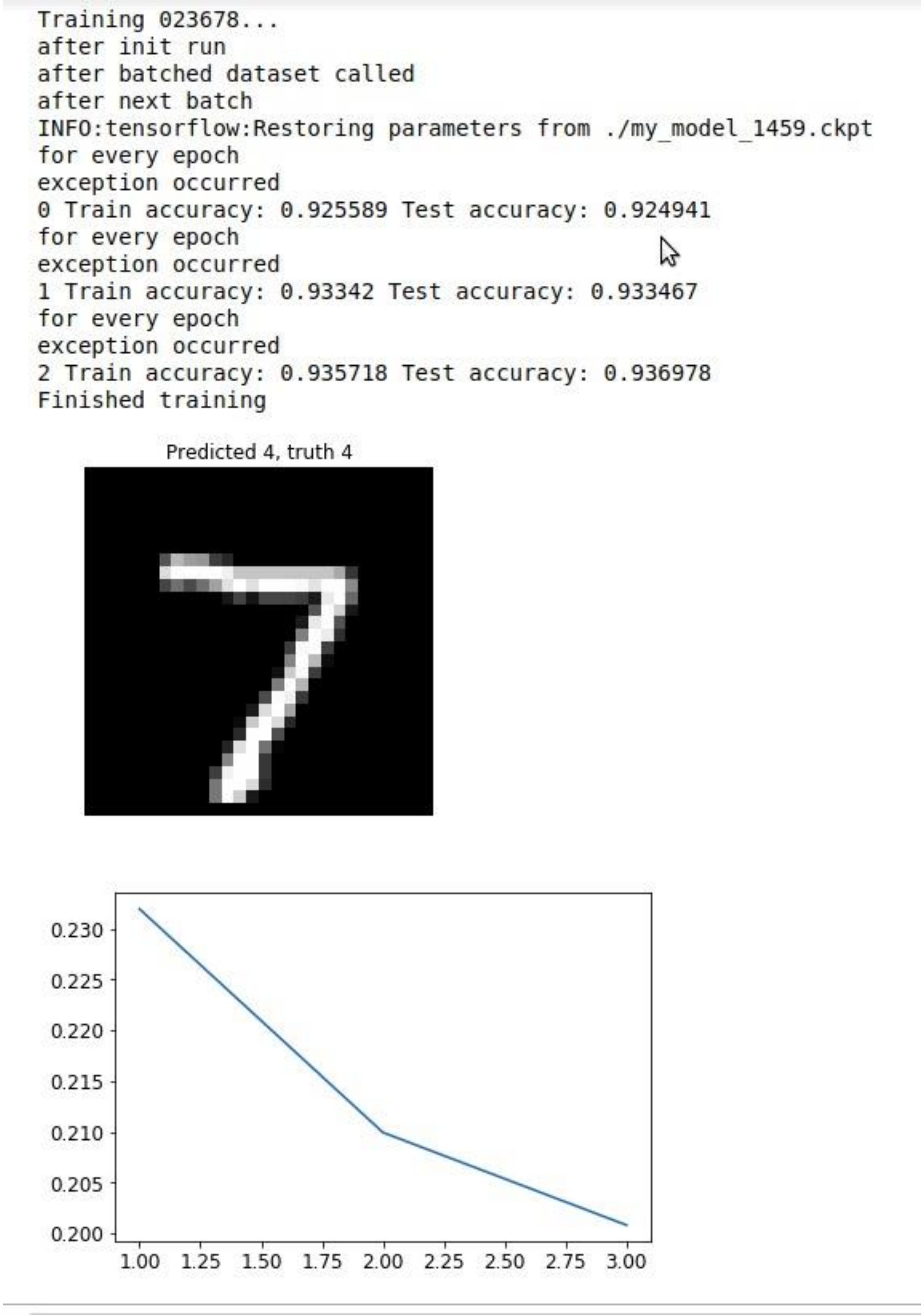
```
Training 023678...
after init run
after batched dataset called
after next batch
INFO:tensorflow:Restoring parameters from ./my_model_1459.ckpt
for every epoch
exception occurred
0 Train accuracy: 0.980619 Test accuracy: 0.977934
for every epoch
exception occurred
1 Train accuracy: 0.99229 Test accuracy: 0.983618
for every epoch
exception occurred
2 Train accuracy: 0.995797 Test accuracy: 0.9888
Finished training
```



Predicted 4, truth 4



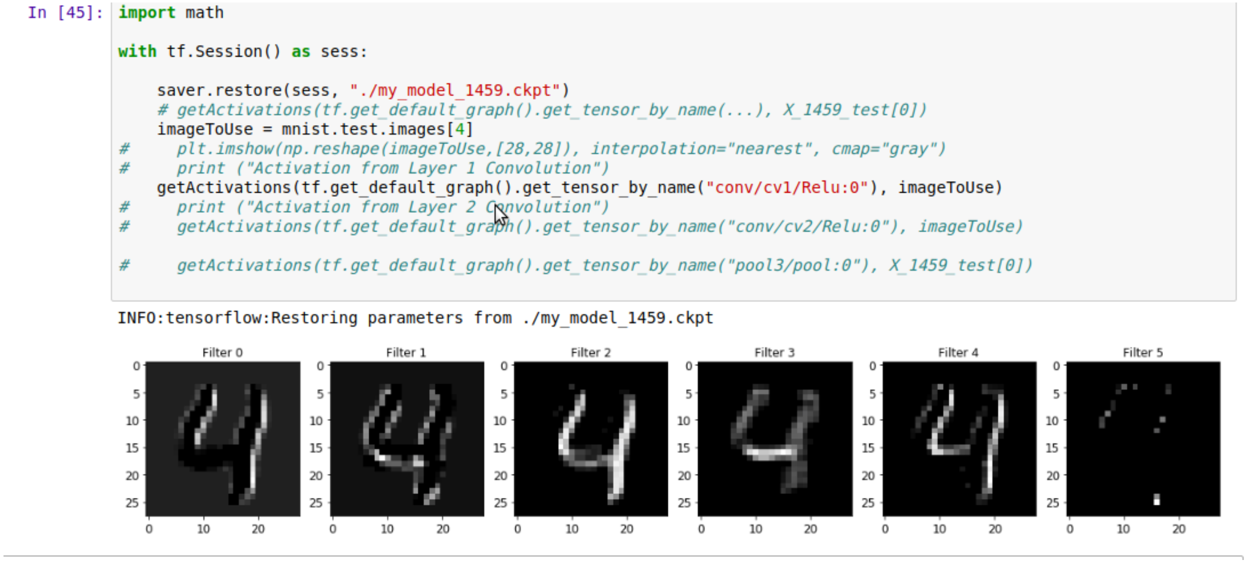
TEST LOSS 023678 – FREEZE



```
Training 023678...
after init run
after batched dataset called
after next batch
INFO:tensorflow:Restoring parameters from ./my_model_1459.ckpt
for every epoch
exception occurred
0 Train accuracy: 0.925589 Test accuracy: 0.924941
for every epoch
exception occurred
1 Train accuracy: 0.93342 Test accuracy: 0.933467
for every epoch
exception occurred
2 Train accuracy: 0.935718 Test accuracy: 0.936978
Finished training
```

Predicted 4, truth 4

ACTIVATION – CONV1

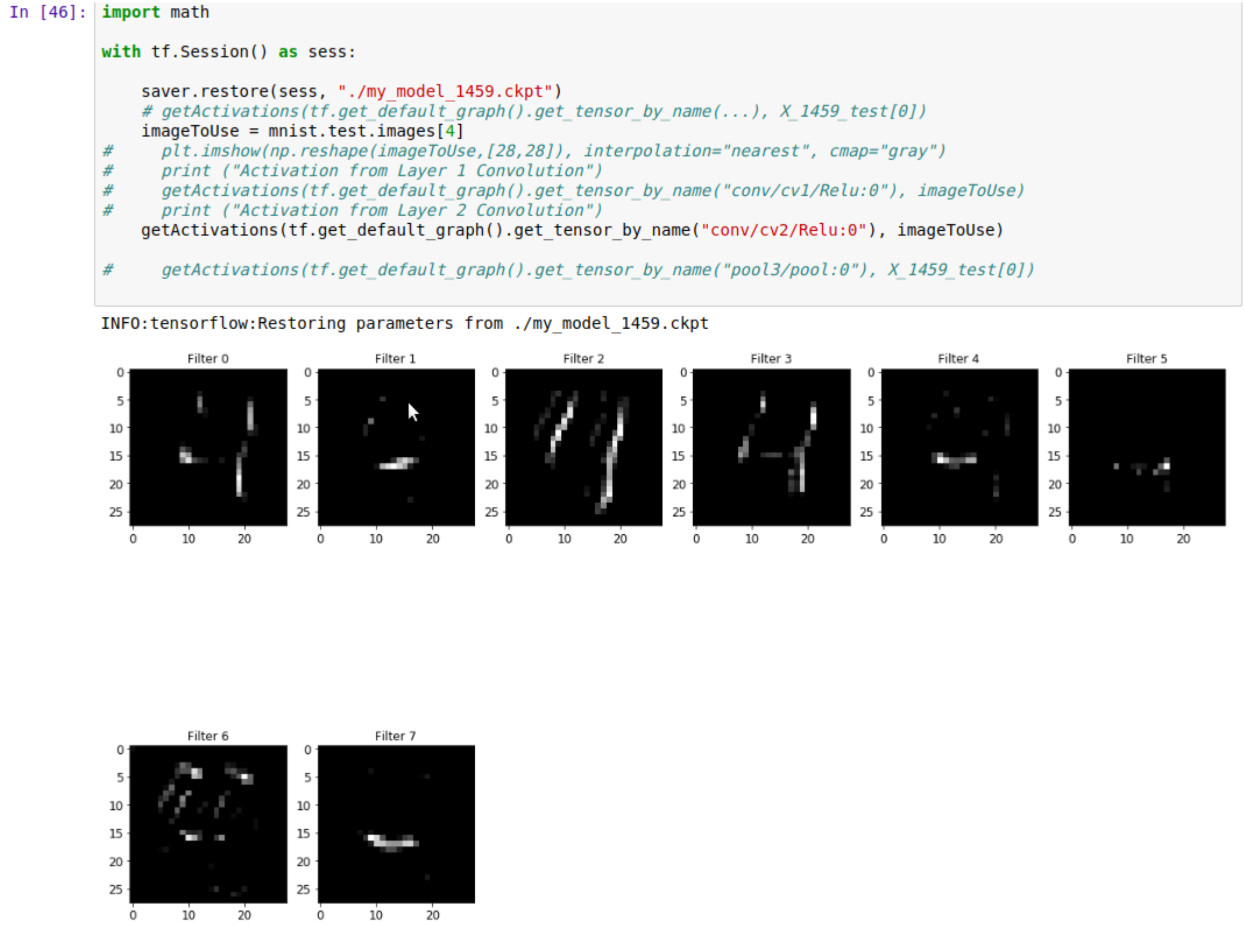


In [45]:

```
import math

with tf.Session() as sess:

    saver.restore(sess, "./my_model_1459.ckpt")
    # getActivations(tf.get_default_graph().get_tensor_by_name(...), X_1459_test[0])
    imageToUse = mnist.test.images[4]
#     plt.imshow(np.reshape(imageToUse,[28,28]), interpolation="nearest", cmap="gray")
#     print ("Activation from Layer 1 Convolution")
    getActivations(tf.get_default_graph().get_tensor_by_name("conv/cv1/Relu:0"), imageToUse)
#     print ("Activation from Layer 2 Convolution")
#     getActivations(tf.get_default_graph().get_tensor_by_name("conv/cv2/Relu:0"), imageToUse)

#     getActivations(tf.get_default_graph().get_tensor_by_name("pool3/pool:0"), X_1459_test[0])
```

```
INFO:tensorflow:Restoring parameters from ./my_model_1459.ckpt
```

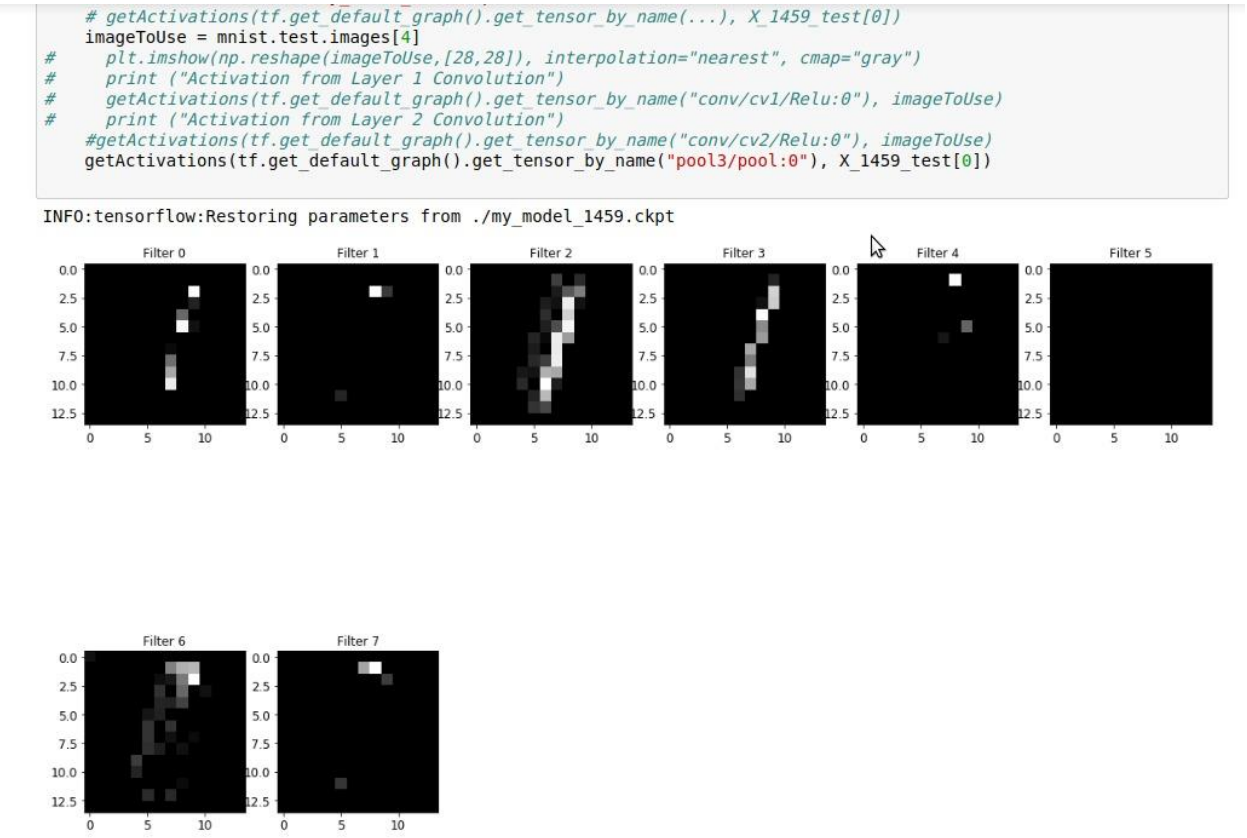
ACTIVATION – CONV2



```
In [46]: import math

         with tf.Session() as sess:

             saver.restore(sess, "./my_model_1459.ckpt")
             # getActivations(tf.get_default_graph().get_tensor_by_name(...), X_1459_test[0])
             imageToUse = mnist.test.images[4]
         #     plt.imshow(np.reshape(imageToUse,[28,28]), interpolation="nearest", cmap="gray")
         #     print ("Activation from Layer 1 Convolution")
         #     getActivations(tf.get_default_graph().get_tensor_by_name("conv/cv1/Relu:0"), imageToUse)
         #     print ("Activation from Layer 2 Convolution")
             getActivations(tf.get_default_graph().get_tensor_by_name("conv/cv2/Relu:0"), imageToUse)

         #     getActivations(tf.get_default_graph().get_tensor_by_name("pool3/pool:0"), X_1459_test[0])
```

```
INFO:tensorflow:Restoring parameters from ./my_model_1459.ckpt
```

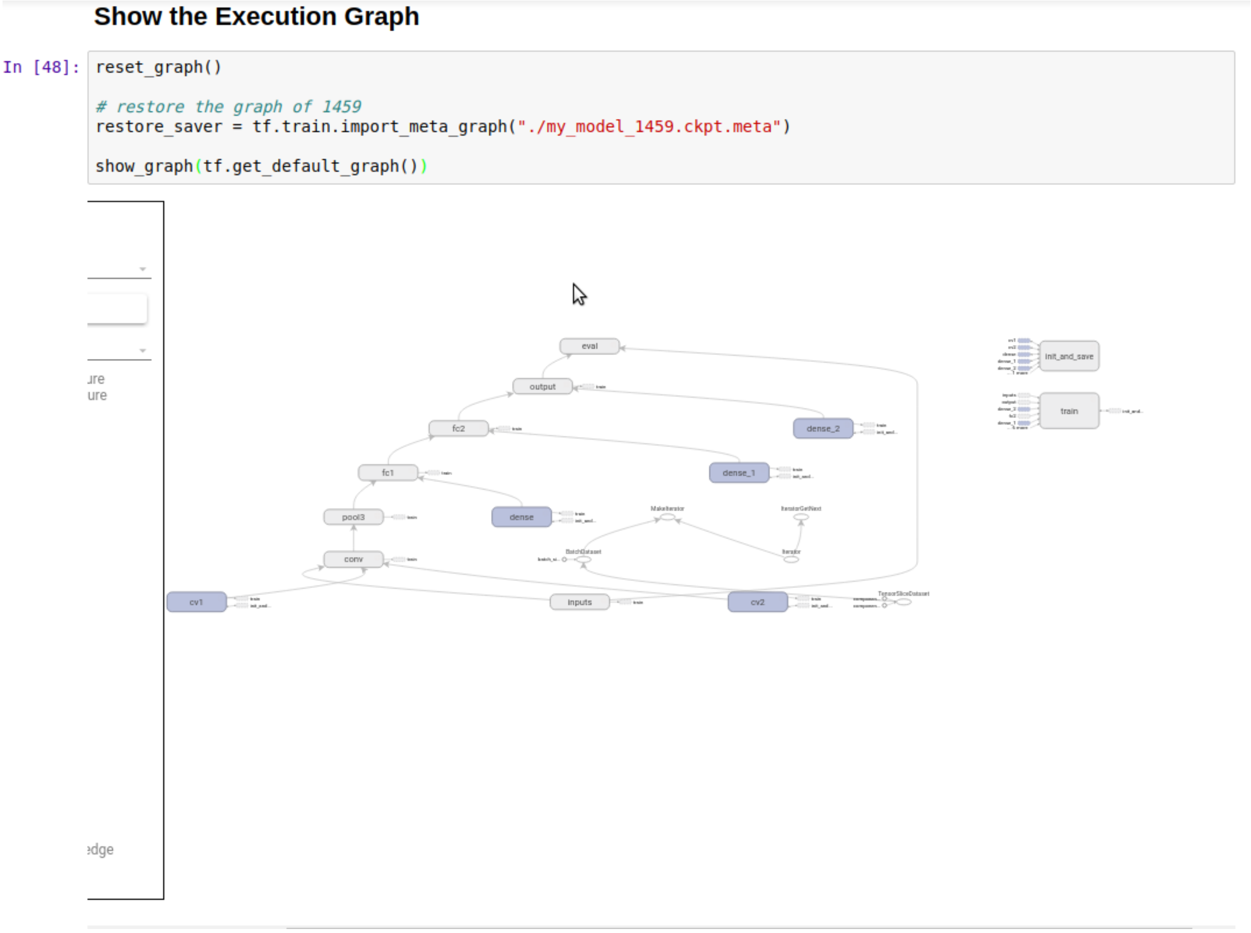
ACTIVATION – POOL



```
    # getActivations(tf.get_default_graph().get_tensor_by_name(...), X_1459_test[0])
    imageToUse = mnist.test.images[4]
#     plt.imshow(np.reshape(imageToUse,[28,28]), interpolation="nearest", cmap="gray")
#     print ("Activation from Layer 1 Convolution")
#     getActivations(tf.get_default_graph().get_tensor_by_name("conv/cv1/Relu:0"), imageToUse)
#     print ("Activation from Layer 2 Convolution")
    #getActivations(tf.get_default_graph().get_tensor_by_name("conv/cv2/Relu:0"), imageToUse)
    getActivations(tf.get_default_graph().get_tensor_by_name("pool3/pool:0"), X_1459_test[0])
```

```
INFO:tensorflow:Restoring parameters from ./my_model_1459.ckpt
```

FINAL EXECUTION GRAPH



Show the Execution Graph

In [48]:
```
reset_graph()

# restore the graph of 1459
restore_saver = tf.train.import_meta_graph("./my_model_1459.ckpt.meta")

show_graph(tf.get_default_graph())
```