

MNIST_display

April 12, 2021

```
[1]: from MNIST_ACAI import ACAI_autoencoder
path = "./mnist_acai/"
acai = ACAI_autoencoder(hidden_dim=200)
acai.decoder.load_weights(path+'decoder/')
acai.encoder.load_weights(path+'encoder/')
acai.critic.load_weights(path+'critic/')
```

Using TensorFlow backend.

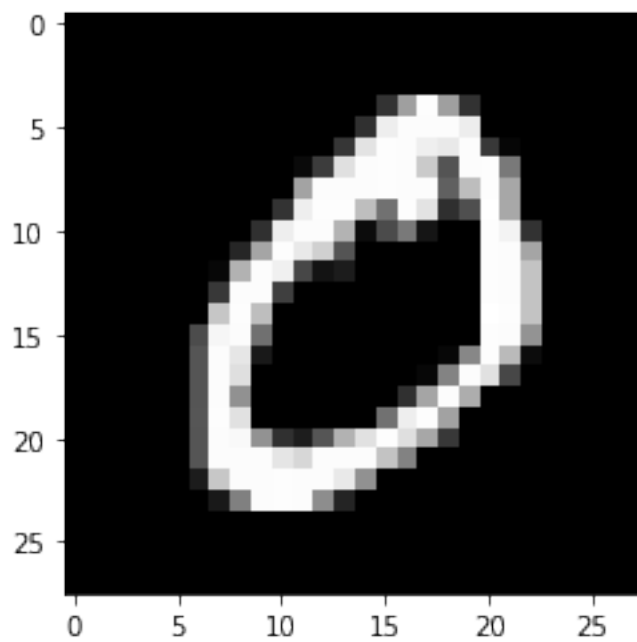
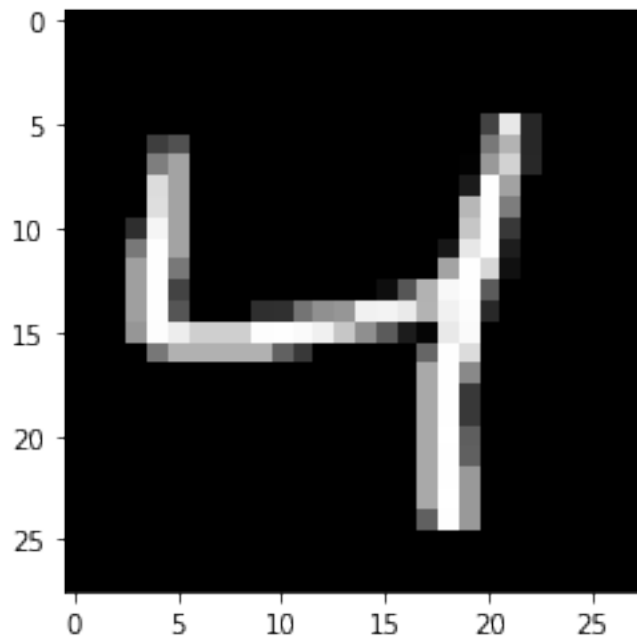
```
[1]: <tensorflow.python.training.tracking.util.CheckpointLoadStatus at
0x7fb2b89c6a50>
```

```
[2]: import matplotlib.pyplot as plt
import tensorflow as tf
from MNIST_utils import get_mnist_data
data = get_mnist_data()
```

```
[3]: d1 = data[2, :, :]
d2 = data[1, :, :]
```

1 Original Digits

```
[4]: plt.figure()
plt.imshow(d1, cmap='gray');
plt.figure()
plt.imshow(d2, cmap='gray');
```

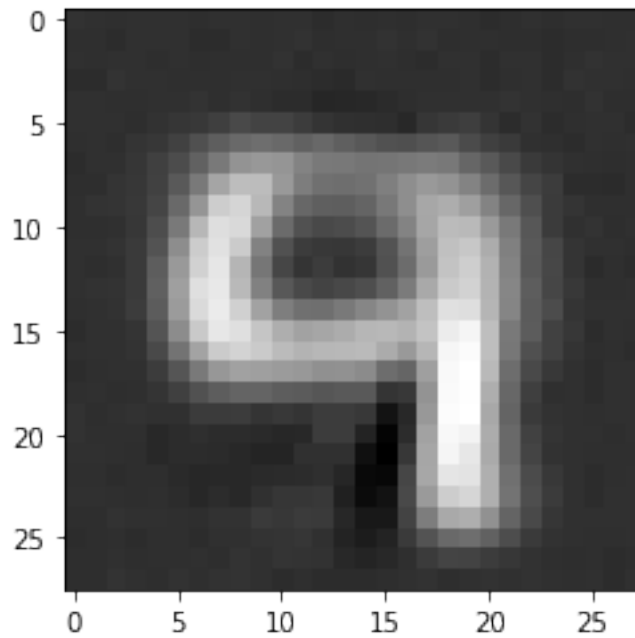


```
[5]: d1z = acai.encoder(tf.reshape(d1, [1, 784]))  
     d2z = acai.encoder(tf.reshape(d2, [1, 784]))
```

2 Reconstructing digits from autoencoder

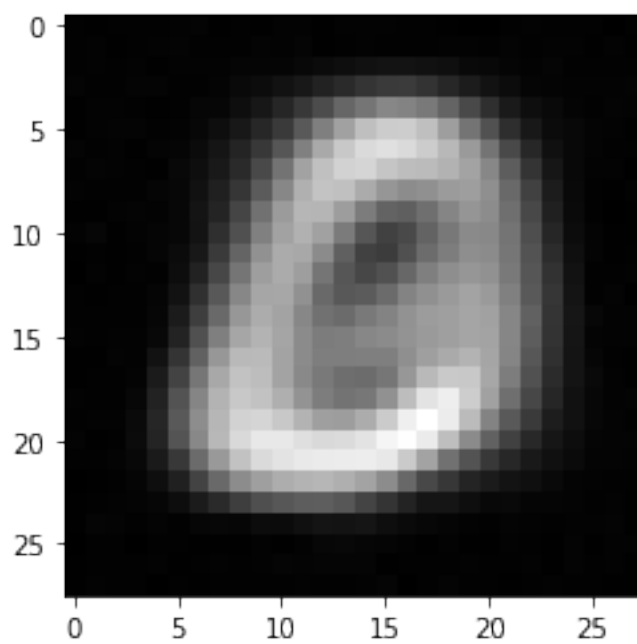
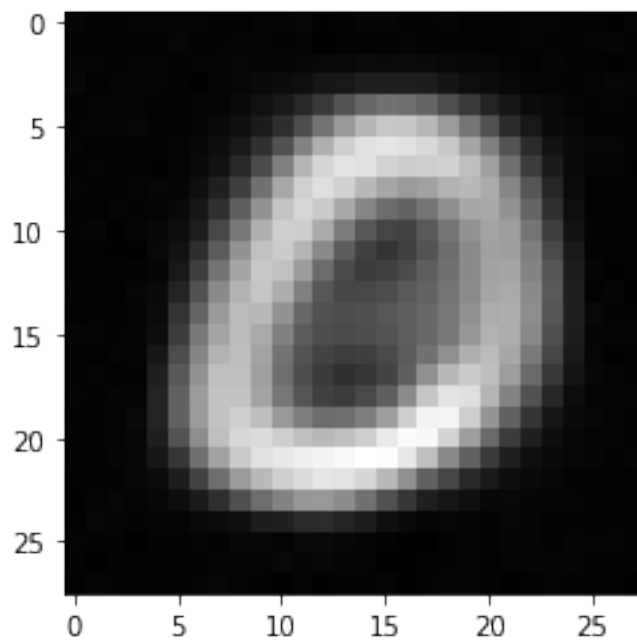
```
[6]: # Autoencoder's reconstruction

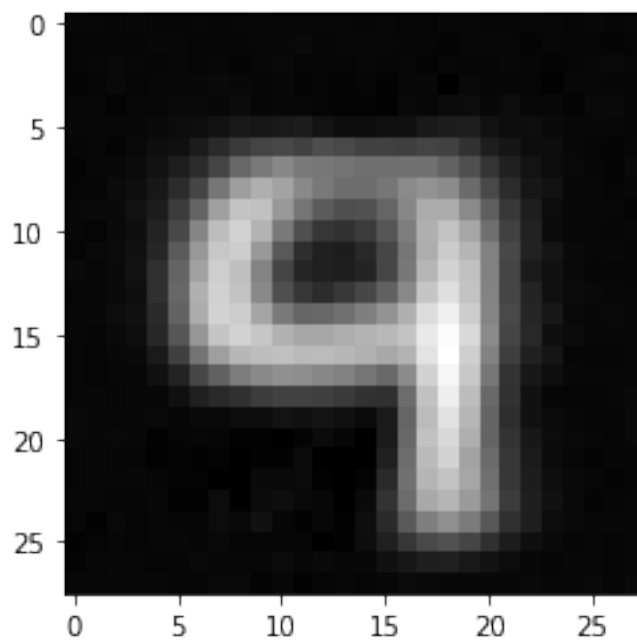
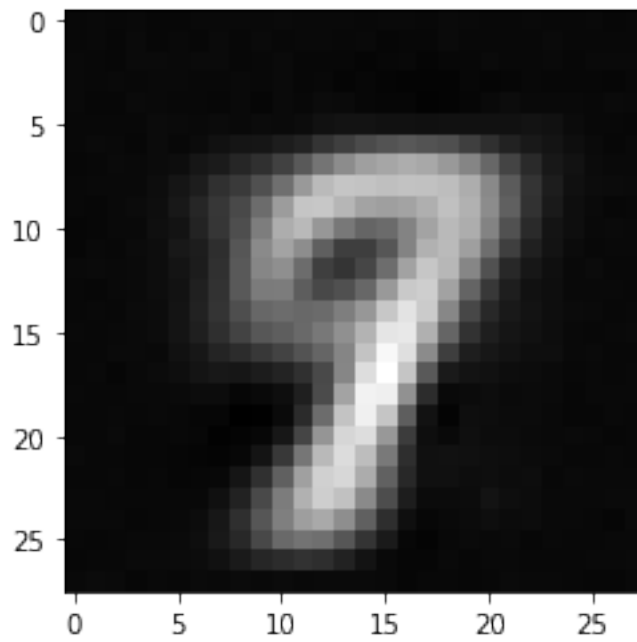
d1_out = tf.reshape(acai.decoder(d1z), [1, 28, 28])
d2_out = tf.reshape(acai.decoder(d2z), [1, 28, 28])
#print(d1_out)
plt.imshow(d1_out[0, :, :], cmap='gray');
```

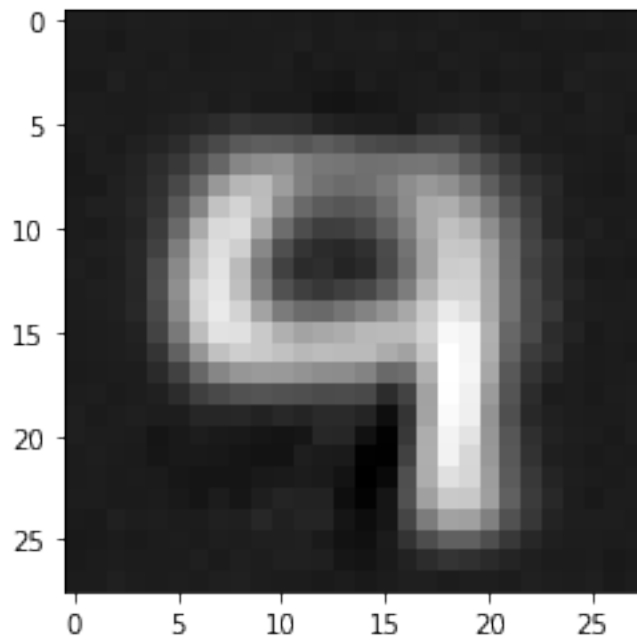


3 Interpolating between 0 and 4

```
[15]: # Interpolating
for alpha in [0.4, 0.5, 0.6, 0.7, 0.8]:
    plt.figure()
    dmixz = d1z * alpha + (1 - alpha) * d2z
    dmix_out = tf.reshape(acai.decoder(dmixz), [1, 28, 28])
    plt.imshow(dmix_out[0, :, :], cmap='gray');
```







[]: