

[Open in Colab](#)

(https://colab.research.google.com/github/oliverfoster27/Practical-Machine-Learning/blob/master/Week%206/C6_Autoencoder_Homework.ipynb)

Fashion MNIST Autoencoder

Directives

- Create the autoencoders described below, you can play with the topology, those are just starting points
- Use GPU runtime
- Print 10 inputs and their associated outputs
- Don't forget to normalize your data
- Use the Functional API for Keras
- Plot the model loss over time

Autoencoders to build

1. Stacked
2. Convolutional Encoder

Don't worry if the data you generate looks bad. This is just an exercise, we can't really measure performance on an unsupervised task

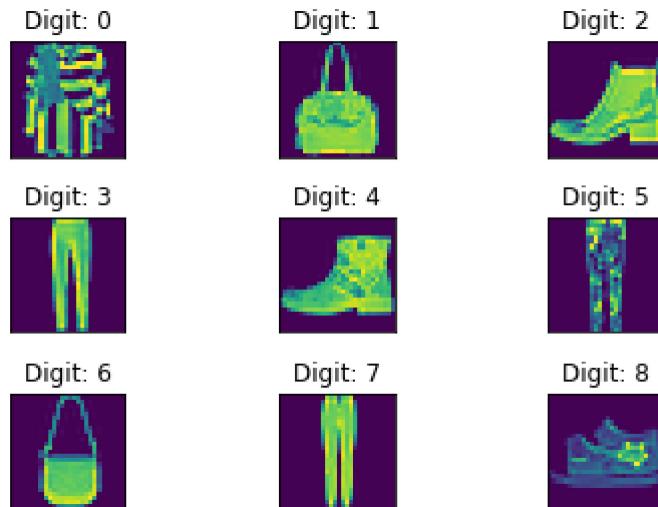
```
In [31]: from keras.datasets import fashion_mnist
import numpy as np
import matplotlib.pyplot as plt

(X_train, _), (X_test, _) = fashion_mnist.load_data()
all_data = np.concatenate((X_train, X_test))
all_data.shape
```

```
Out[31]: (70000, 28, 28)
```

```
In [0]: def print_first_9(data):
    fig = plt.figure()
    for i in range(9):
        plt.subplot(3,3,i+1)
        plt.tight_layout()
        plt.imshow(data[i], interpolation='none')
        plt.title("Digit: {}".format(i))
        plt.xticks([])
        plt.yticks([])
```

```
In [33]: print_first_9(all_data[-9:])
```



```
In [0]: from keras.layers import Dense, Flatten, Conv2D, MaxPool2D, Dropout, BatchNormalization, UpSampling2D
from keras.models import Sequential
from keras.optimizers import Adam
from keras.callbacks import EarlyStopping
from keras import regularizers
import math
import keras.backend as K
```

Stacked Autoencoder

```
In [35]: SA_X_train = X_train.reshape(-1, 784)
SA_X_test = X_test.reshape(-1, 784)
SA_X_train = SA_X_train / 255
SA_X_test = SA_X_test / 255
print(SA_X_train.shape, SA_X_test.shape)

K.clear_session()

model1 = Sequential()
...
Encoder:
...
model1.add(Dense(200, activation='relu', input_shape=(784, )))
model1.add(Dropout(0.2))
model1.add(Dense(10, activation='relu'))

...
Decoder:
...
model1.add(Dense(10, activation='relu'))
model1.add(Dropout(0.2))
model1.add(Dense(200, activation='relu'))

...
Output:
...
model1.add(Dense(784, activation='relu'))

optimizer = Adam(lr=10e-6)
model1.compile(optimizer=optimizer,
               loss='mean_squared_error')

model1.summary()

h1 = model1.fit(SA_X_train, SA_X_train, batch_size=128, epochs=80,
                 verbose=1, validation_split=0.3)
```

(60000, 784) (10000, 784)

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 200)	157000
dropout_1 (Dropout)	(None, 200)	0
dense_2 (Dense)	(None, 10)	2010
dense_3 (Dense)	(None, 10)	110
dropout_2 (Dropout)	(None, 10)	0
dense_4 (Dense)	(None, 200)	2200
dense_5 (Dense)	(None, 784)	157584

Total params: 318,904

Trainable params: 318,904

Non-trainable params: 0

Train on 42000 samples, validate on 18000 samples

Epoch 1/80

42000/42000 [=====] - 2s 48us/step - loss: 0.1596 -
val_loss: 0.1229

Epoch 2/80

42000/42000 [=====] - 2s 42us/step - loss: 0.1090 -
val_loss: 0.0905

Epoch 3/80

42000/42000 [=====] - 2s 42us/step - loss: 0.0908 -
val_loss: 0.0811

Epoch 4/80

42000/42000 [=====] - 2s 42us/step - loss: 0.0821 -
val_loss: 0.0737

Epoch 5/80

42000/42000 [=====] - 2s 41us/step - loss: 0.0751 -
val_loss: 0.0678

Epoch 6/80

42000/42000 [=====] - 2s 42us/step - loss: 0.0702 -
val_loss: 0.0642

Epoch 7/80

42000/42000 [=====] - 2s 42us/step - loss: 0.0671 -
val_loss: 0.0619

Epoch 8/80

42000/42000 [=====] - 2s 47us/step - loss: 0.0647 -
val_loss: 0.0599

Epoch 9/80

42000/42000 [=====] - 2s 47us/step - loss: 0.0623 -
val_loss: 0.0585

Epoch 10/80

42000/42000 [=====] - 2s 47us/step - loss: 0.0610 -
val_loss: 0.0576

Epoch 11/80

42000/42000 [=====] - 2s 47us/step - loss: 0.0600 -
val_loss: 0.0569

Epoch 12/80

```
42000/42000 [=====] - 2s 47us/step - loss: 0.0590 -  
val_loss: 0.0560  
Epoch 13/80  
42000/42000 [=====] - 2s 43us/step - loss: 0.0578 -  
val_loss: 0.0555  
Epoch 14/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0574 -  
val_loss: 0.0551  
Epoch 15/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0564 -  
val_loss: 0.0546  
Epoch 16/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0562 -  
val_loss: 0.0546  
Epoch 17/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0557 -  
val_loss: 0.0541  
Epoch 18/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0554 -  
val_loss: 0.0539  
Epoch 19/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0550 -  
val_loss: 0.0535  
Epoch 20/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0545 -  
val_loss: 0.0528  
Epoch 21/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0542 -  
val_loss: 0.0527  
Epoch 22/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0538 -  
val_loss: 0.0524  
Epoch 23/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0535 -  
val_loss: 0.0518  
Epoch 24/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0531 -  
val_loss: 0.0515  
Epoch 25/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0530 -  
val_loss: 0.0512  
Epoch 26/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0527 -  
val_loss: 0.0513  
Epoch 27/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0523 -  
val_loss: 0.0509  
Epoch 28/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0519 -  
val_loss: 0.0508  
Epoch 29/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0518 -  
val_loss: 0.0505  
Epoch 30/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0514 -  
val_loss: 0.0498  
Epoch 31/80
```

```
42000/42000 [=====] - 2s 42us/step - loss: 0.0505 -  
val_loss: 0.0494  
Epoch 32/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0503 -  
val_loss: 0.0492  
Epoch 33/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0495 -  
val_loss: 0.0481  
Epoch 34/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0494 -  
val_loss: 0.0479  
Epoch 35/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0489 -  
val_loss: 0.0479  
Epoch 36/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0489 -  
val_loss: 0.0476  
Epoch 37/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0485 -  
val_loss: 0.0473  
Epoch 38/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0483 -  
val_loss: 0.0470  
Epoch 39/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0481 -  
val_loss: 0.0468  
Epoch 40/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0478 -  
val_loss: 0.0468  
Epoch 41/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0478 -  
val_loss: 0.0465  
Epoch 42/80  
42000/42000 [=====] - 2s 43us/step - loss: 0.0475 -  
val_loss: 0.0465  
Epoch 43/80  
42000/42000 [=====] - 2s 43us/step - loss: 0.0474 -  
val_loss: 0.0461  
Epoch 44/80  
42000/42000 [=====] - 2s 49us/step - loss: 0.0471 -  
val_loss: 0.0462  
Epoch 45/80  
42000/42000 [=====] - 2s 49us/step - loss: 0.0471 -  
val_loss: 0.0459  
Epoch 46/80  
42000/42000 [=====] - 2s 49us/step - loss: 0.0470 -  
val_loss: 0.0458  
Epoch 47/80  
42000/42000 [=====] - 2s 49us/step - loss: 0.0468 -  
val_loss: 0.0454  
Epoch 48/80  
42000/42000 [=====] - 2s 47us/step - loss: 0.0461 -  
val_loss: 0.0450  
Epoch 49/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0458 -  
val_loss: 0.0450  
Epoch 50/80
```

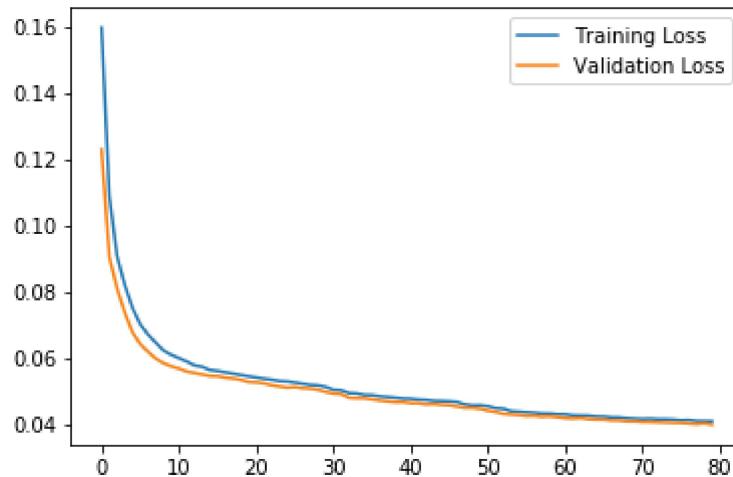
```
42000/42000 [=====] - 2s 42us/step - loss: 0.0458 -  
val_loss: 0.0448  
Epoch 51/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0455 -  
val_loss: 0.0442  
Epoch 52/80  
42000/42000 [=====] - 2s 45us/step - loss: 0.0449 -  
val_loss: 0.0438  
Epoch 53/80  
42000/42000 [=====] - 2s 48us/step - loss: 0.0448 -  
val_loss: 0.0432  
Epoch 54/80  
42000/42000 [=====] - 2s 47us/step - loss: 0.0440 -  
val_loss: 0.0430  
Epoch 55/80  
42000/42000 [=====] - 2s 47us/step - loss: 0.0439 -  
val_loss: 0.0429  
Epoch 56/80  
42000/42000 [=====] - 2s 47us/step - loss: 0.0437 -  
val_loss: 0.0426  
Epoch 57/80  
42000/42000 [=====] - 2s 45us/step - loss: 0.0435 -  
val_loss: 0.0426  
Epoch 58/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0433 -  
val_loss: 0.0424  
Epoch 59/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0433 -  
val_loss: 0.0425  
Epoch 60/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0431 -  
val_loss: 0.0422  
Epoch 61/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0431 -  
val_loss: 0.0419  
Epoch 62/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0428 -  
val_loss: 0.0418  
Epoch 63/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0427 -  
val_loss: 0.0420  
Epoch 64/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0427 -  
val_loss: 0.0417  
Epoch 65/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0424 -  
val_loss: 0.0415  
Epoch 66/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0424 -  
val_loss: 0.0415  
Epoch 67/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0421 -  
val_loss: 0.0412  
Epoch 68/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0421 -  
val_loss: 0.0412  
Epoch 69/80
```

```
42000/42000 [=====] - 2s 42us/step - loss: 0.0419 -  
val_loss: 0.0409  
Epoch 70/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0417 -  
val_loss: 0.0409  
Epoch 71/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0416 -  
val_loss: 0.0408  
Epoch 72/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0417 -  
val_loss: 0.0408  
Epoch 73/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0416 -  
val_loss: 0.0407  
Epoch 74/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0416 -  
val_loss: 0.0406  
Epoch 75/80  
42000/42000 [=====] - 2s 42us/step - loss: 0.0415 -  
val_loss: 0.0405  
Epoch 76/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0413 -  
val_loss: 0.0405  
Epoch 77/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0414 -  
val_loss: 0.0403  
Epoch 78/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0411 -  
val_loss: 0.0401  
Epoch 79/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0410 -  
val_loss: 0.0403  
Epoch 80/80  
42000/42000 [=====] - 2s 41us/step - loss: 0.0410 -  
val_loss: 0.0400
```

View Training Loss

```
In [36]: plt.plot(h1.history['loss'], label='Training Loss')
plt.plot(h1.history['val_loss'], label='Validation Loss')
plt.legend()
```

```
Out[36]: <matplotlib.legend.Legend at 0x7fb6738d7780>
```

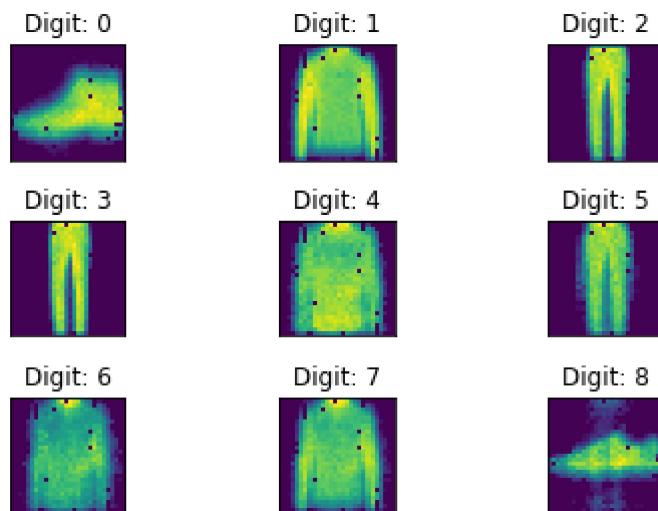


Print Auto-Encoded Data and Real Input Data

```
In [0]: SA_test_encoded = model1.predict(SA_X_test).reshape(-1, 28, 28)
```

```
In [38]: print("Auto-encoded data:")
print_first_9(SA_test_encoded)
```

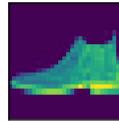
Auto-encoded data:



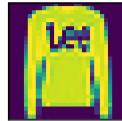
```
In [39]: print("Original data:")
print(first_9(SA_X_test.reshape(-1, 28, 28)))
```

Original data:

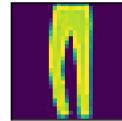
Digit: 0



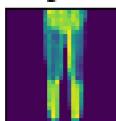
Digit: 1



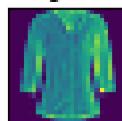
Digit: 2



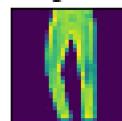
Digit: 3



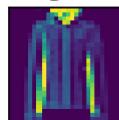
Digit: 4



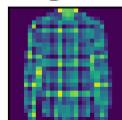
Digit: 5



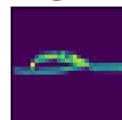
Digit: 6



Digit: 7



Digit: 8



Convolutional Autoencoder

```
In [46]: CA_X_train = X_train.reshape(-1, 28, 28, 1)
CA_X_test = X_test.reshape(-1, 28, 28, 1)
CA_X_train = CA_X_train / 255
CA_X_test = CA_X_test / 255
print(CA_X_train.shape, CA_X_test.shape)

K.clear_session()

model2 = Sequential()
...
Encoder:
...
model2.add(Conv2D(16, (3, 3), activation='relu', padding='same', input_shape=(28, 28, 1)))
model2.add(MaxPool2D(pool_size=(2, 2)))
model2.add(Dropout(0.2))

model2.add(Conv2D(8, (3, 3), padding='same', activation='relu'))
model2.add(MaxPool2D(pool_size=(2, 2)))

...
Decoder:
...
model2.add(Conv2D(8, (3, 3), padding='same', activation='relu'))
model2.add(UpSampling2D(size=(2, 2)))
model2.add(Dropout(0.2))

model2.add(Conv2D(16, (3, 3), padding='same', activation='relu'))
model2.add(UpSampling2D(size=(2, 2)))
model2.add(Dropout(0.3))

...
Output:
...
model2.add(Conv2D(1, (3, 3), padding='same', activation='relu'))

optimizer = Adam(lr=10e-6)
model2.compile(optimizer=optimizer,
               loss='mean_squared_error')

model2.summary()

h2 = model2.fit(CA_X_train, CA_X_train, batch_size=128, epochs=80,
                 verbose=1, validation_split=0.3)
```

```
(60000, 28, 28, 1) (10000, 28, 28, 1)
```

Layer (type)	Output Shape	Param #
<hr/>		
conv2d_1 (Conv2D)	(None, 28, 28, 16)	160
<hr/>		
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 16)	0
<hr/>		
dropout_1 (Dropout)	(None, 14, 14, 16)	0
<hr/>		
conv2d_2 (Conv2D)	(None, 14, 14, 8)	1160
<hr/>		
max_pooling2d_2 (MaxPooling2D)	(None, 7, 7, 8)	0
<hr/>		
conv2d_3 (Conv2D)	(None, 7, 7, 8)	584
<hr/>		
up_sampling2d_1 (UpSampling2D)	(None, 14, 14, 8)	0
<hr/>		
dropout_2 (Dropout)	(None, 14, 14, 8)	0
<hr/>		
conv2d_4 (Conv2D)	(None, 14, 14, 16)	1168
<hr/>		
up_sampling2d_2 (UpSampling2D)	(None, 28, 28, 16)	0
<hr/>		
dropout_3 (Dropout)	(None, 28, 28, 16)	0
<hr/>		
conv2d_5 (Conv2D)	(None, 28, 28, 1)	145
<hr/>		
Total params:	3,217	
Trainable params:	3,217	
Non-trainable params:	0	

Train on 42000 samples, validate on 18000 samples
Epoch 1/80
42000/42000 [=====] - 3s 77us/step - loss: 0.1358 -
val_loss: 0.1145
Epoch 2/80
42000/42000 [=====] - 3s 61us/step - loss: 0.0922 -
val_loss: 0.0793
Epoch 3/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0754 -
val_loss: 0.0631
Epoch 4/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0651 -
val_loss: 0.0538
Epoch 5/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0587 -
val_loss: 0.0491
Epoch 6/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0541 -
val_loss: 0.0460
Epoch 7/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0507 -
val_loss: 0.0437
Epoch 8/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0479 -
val_loss: 0.0423

```
Epoch 9/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0456 -
val_loss: 0.0409
Epoch 10/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0437 -
val_loss: 0.0401
Epoch 11/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0420 -
val_loss: 0.0389
Epoch 12/80
42000/42000 [=====] - 2s 58us/step - loss: 0.0405 -
val_loss: 0.0378
Epoch 13/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0391 -
val_loss: 0.0365
Epoch 14/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0378 -
val_loss: 0.0354
Epoch 15/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0366 -
val_loss: 0.0342
Epoch 16/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0355 -
val_loss: 0.0332
Epoch 17/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0344 -
val_loss: 0.0321
Epoch 18/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0334 -
val_loss: 0.0309
Epoch 19/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0325 -
val_loss: 0.0301
Epoch 20/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0317 -
val_loss: 0.0291
Epoch 21/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0310 -
val_loss: 0.0282
Epoch 22/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0303 -
val_loss: 0.0276
Epoch 23/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0296 -
val_loss: 0.0267
Epoch 24/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0290 -
val_loss: 0.0262
Epoch 25/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0285 -
val_loss: 0.0254
Epoch 26/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0279 -
val_loss: 0.0249
Epoch 27/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0275 -
val_loss: 0.0245
```

```
Epoch 28/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0270 -
val_loss: 0.0239
Epoch 29/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0265 -
val_loss: 0.0235
Epoch 30/80
42000/42000 [=====] - 3s 60us/step - loss: 0.0261 -
val_loss: 0.0232
Epoch 31/80
42000/42000 [=====] - 3s 65us/step - loss: 0.0257 -
val_loss: 0.0227
Epoch 32/80
42000/42000 [=====] - 3s 64us/step - loss: 0.0253 -
val_loss: 0.0222
Epoch 33/80
42000/42000 [=====] - 3s 65us/step - loss: 0.0250 -
val_loss: 0.0219
Epoch 34/80
42000/42000 [=====] - 3s 61us/step - loss: 0.0247 -
val_loss: 0.0217
Epoch 35/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0244 -
val_loss: 0.0213
Epoch 36/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0241 -
val_loss: 0.0211
Epoch 37/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0238 -
val_loss: 0.0207
Epoch 38/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0235 -
val_loss: 0.0206
Epoch 39/80
42000/42000 [=====] - 2s 58us/step - loss: 0.0233 -
val_loss: 0.0203
Epoch 40/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0230 -
val_loss: 0.0201
Epoch 41/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0228 -
val_loss: 0.0200
Epoch 42/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0226 -
val_loss: 0.0197
Epoch 43/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0223 -
val_loss: 0.0195
Epoch 44/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0221 -
val_loss: 0.0193
Epoch 45/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0220 -
val_loss: 0.0191
Epoch 46/80
42000/42000 [=====] - 2s 58us/step - loss: 0.0218 -
val_loss: 0.0189
```

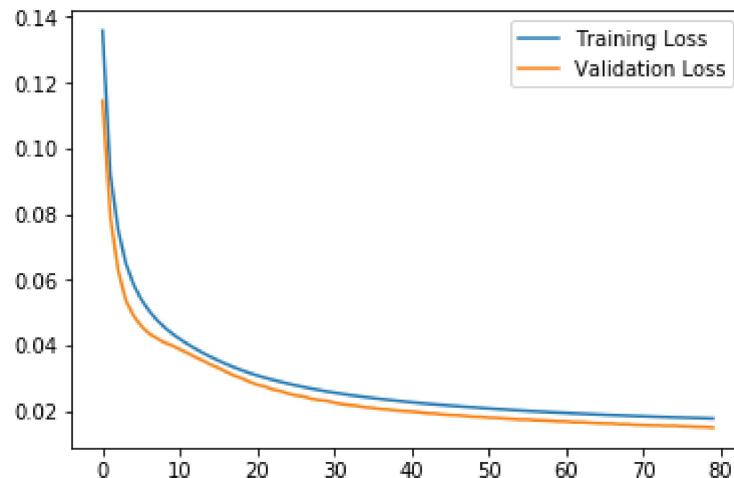
```
Epoch 47/80
42000/42000 [=====] - 3s 65us/step - loss: 0.0216 -
val_loss: 0.0188
Epoch 48/80
42000/42000 [=====] - 3s 66us/step - loss: 0.0214 -
val_loss: 0.0186
Epoch 49/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0212 -
val_loss: 0.0184
Epoch 50/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0211 -
val_loss: 0.0183
Epoch 51/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0209 -
val_loss: 0.0181
Epoch 52/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0207 -
val_loss: 0.0180
Epoch 53/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0206 -
val_loss: 0.0179
Epoch 54/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0204 -
val_loss: 0.0177
Epoch 55/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0203 -
val_loss: 0.0175
Epoch 56/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0202 -
val_loss: 0.0175
Epoch 57/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0200 -
val_loss: 0.0173
Epoch 58/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0199 -
val_loss: 0.0173
Epoch 59/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0198 -
val_loss: 0.0171
Epoch 60/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0196 -
val_loss: 0.0170
Epoch 61/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0195 -
val_loss: 0.0169
Epoch 62/80
42000/42000 [=====] - 3s 60us/step - loss: 0.0194 -
val_loss: 0.0167
Epoch 63/80
42000/42000 [=====] - 3s 65us/step - loss: 0.0193 -
val_loss: 0.0167
Epoch 64/80
42000/42000 [=====] - 3s 64us/step - loss: 0.0192 -
val_loss: 0.0166
Epoch 65/80
42000/42000 [=====] - 3s 64us/step - loss: 0.0191 -
val_loss: 0.0165
```

```
Epoch 66/80
42000/42000 [=====] - 3s 62us/step - loss: 0.0190 -
val_loss: 0.0164
Epoch 67/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0189 -
val_loss: 0.0163
Epoch 68/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0188 -
val_loss: 0.0161
Epoch 69/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0187 -
val_loss: 0.0161
Epoch 70/80
42000/42000 [=====] - 2s 58us/step - loss: 0.0186 -
val_loss: 0.0159
Epoch 71/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0185 -
val_loss: 0.0159
Epoch 72/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0184 -
val_loss: 0.0158
Epoch 73/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0184 -
val_loss: 0.0157
Epoch 74/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0183 -
val_loss: 0.0156
Epoch 75/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0182 -
val_loss: 0.0156
Epoch 76/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0181 -
val_loss: 0.0155
Epoch 77/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0181 -
val_loss: 0.0154
Epoch 78/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0180 -
val_loss: 0.0153
Epoch 79/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0179 -
val_loss: 0.0153
Epoch 80/80
42000/42000 [=====] - 2s 59us/step - loss: 0.0178 -
val_loss: 0.0151
```

View Training Loss

```
In [47]: plt.plot(h2.history['loss'], label='Training Loss')
plt.plot(h2.history['val_loss'], label='Validation Loss')
plt.legend()
```

```
Out[47]: <matplotlib.legend.Legend at 0x7fb644421748>
```

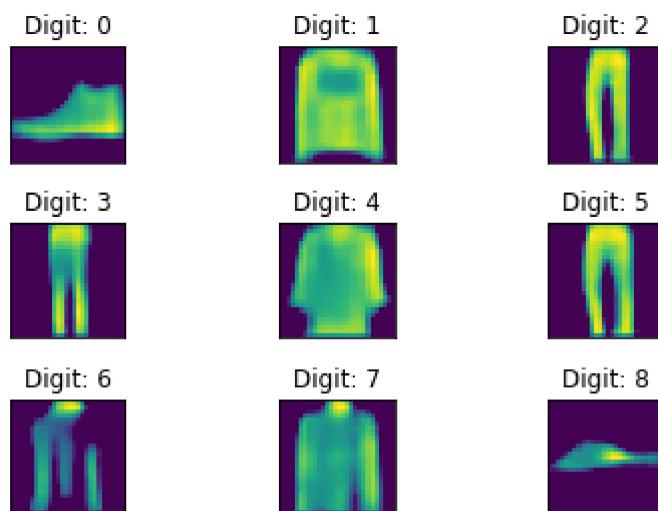


Print Auto-Encoded Data and Real Input Data

```
In [0]: CA_test_encoded = model2.predict(CA_X_test)[:, :, :, :, 0]
```

```
In [49]: print("Auto-encoded data:")
print_first_9(CA_test_encoded)
```

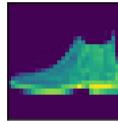
Auto-encoded data:



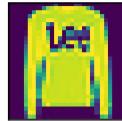
```
In [50]: print("Original data:")
print_first_9(CA_X_test[:, :, :, 0])
```

Original data:

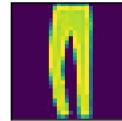
Digit: 0



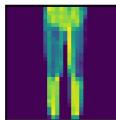
Digit: 1



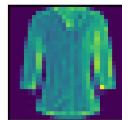
Digit: 2



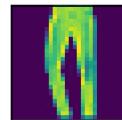
Digit: 3



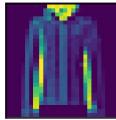
Digit: 4



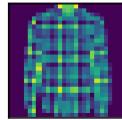
Digit: 5



Digit: 6



Digit: 7



Digit: 8

