# CS156 (Introduction to AI), Spring 2021

## Assignment_10

Roster Name: GURSIMRAN SINGH

Preferred Name (if different): SIMRAN

Student ID: 015212210

Email address: gursimransingh@sjsu.edu

### References and sources

https://keras.io/

## Solution

```
import tensorflow as tf
from tensorflow import keras
from sklearn.model_selection import train_test_split
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense, Input, Conv2D, LSTM, MaxPool2D, UpSampling2D
import matplotlib.pyplot as plt
import numpy as np
```

## Importing the fashion datsets

```
input_shape = (28, 28, 1)
(x_train_valid, y_train_valid), (x_test, y_test)=keras.datasets.fashion_mnist.load_data()
print(x_train_valid[0][15])
```

```
[  0   3   0   0   0   0   0   0   0  62 145 204 228 207 213 221 218 208
 211 218 224 223 219 215 224 244 159   0]
```

## Scale images to the [0, 1] range

```
x_train_valid = x_train_valid.astype("float32") / 255
y_train_valid=y_train_valid.astype("float32") / 255
print(x_train_valid[0][15])
```

```
[0.         0.01176471 0.         0.         0.         0.
 0.         0.         0.         0.24313726 0.5686275  0.8
 0.89411765 0.8117647  0.8352941  0.8666667  0.85490197 0.8156863
 0.827451   0.85490197 0.8784314  0.8745098  0.85882354 0.84313726
 0.8784314  0.95686275 0.62352943 0.        ]
```

```
# splitting the x_Train and Y_Train using train test split
x_train, x_validation, y_train, y_validation  = train_test_split(x_train_valid, y_train_valid
x_train.shape,x_validation.shape, x_test.shape
```

```
((48000, 28, 28), (12000, 28, 28), (10000, 28, 28))
```

# Building a Autoencoder

## Flattening it into single layer

```
# Reshape the images into flat ANN layers
x_train = x_train.reshape(-1, 784)
x_validation = x_validation.reshape(-1, 784)
x_test = x_test.reshape(-1, 784)

x_train.shape, x_validation.shape, x_test.shape
```

```
((48000, 784), (12000, 784), (10000, 784))
```

```
input_layer = Input(shape=(784,))   # 28*28
encoded = layers.Dense(128, activation='relu')(input_layer)
#print(encoded.shape)
encoded = layers.Dense(64, activation='relu')(encoded)
encoded = layers.Dense(32, activation='relu')(encoded)
#print(encoded.shape)

decoded = layers.Dense(64, activation='sigmoid')(encoded)
decoded = layers.Dense(128, activation='sigmoid')(decoded)
decoded2 = layers.Dense(784, activation='sigmoid')(decoded)
#print(decoded.shape)

# reconstruction model:
autoencoder = keras.Model(input_layer, decoded2)
```

```
# encoder model:
encoder = keras.Model(input_layer, encoded)

# decoder model:
encoded_input = keras.Input(shape=(32,))
decoder_layer = autoencoder.layers[-3]  # last layer of autoencoder model
print(encoded_input.shape)
decoder = keras.Model(encoded_input, decoder_layer(encoded_input))
```

```
(None, 32)
```

```
autoencoder.summary()
```

```
Model: "model_41"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_29 (InputLayer)        [(None, 784)]             0
_________________________________________________________________
dense_84 (Dense)             (None, 128)               100480
_________________________________________________________________
dense_85 (Dense)             (None, 64)                8256
_________________________________________________________________
dense_86 (Dense)             (None, 32)                2080
_________________________________________________________________
dense_87 (Dense)             (None, 64)                2112
_________________________________________________________________
dense_88 (Dense)             (None, 128)               8320
_________________________________________________________________
dense_89 (Dense)             (None, 784)               101136
=================================================================
Total params: 222,384
Trainable params: 222,384
Non-trainable params: 0
_________________________________________________________________
```

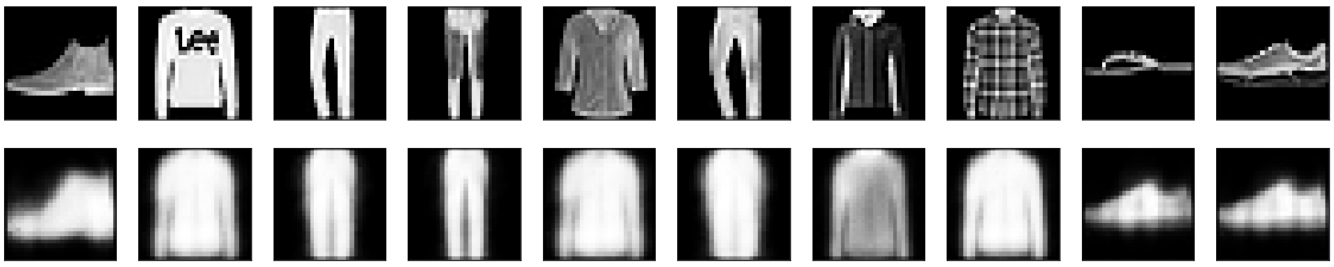
```
autoencoder.compile(optimizer='adam', loss='binary_crossentropy')

autoencoder.fit(x_train, x_train,
                epochs=30,
                batch_size=2048,
                shuffle=True,
                validation_data=(x_test, x_test))
```

```
24/24 [==============================] - 0s 17ms/step - loss: 0.5045 - val_loss: 29.4
Epoch 3/30
24/24 [==============================] - 0s 16ms/step - loss: 0.4886 - val_loss: 27.
Epoch 4/30
24/24 [==============================] - 0s 17ms/step - loss: 0.4813 - val_loss: 23.
Epoch 5/30
24/24 [==============================] - 0s 16ms/step - loss: 0.4640 - val_loss: 2.6
```

```
Epoch 6/30
24/24 [==============================] - 0s 16ms/step - loss: 0.4465 - val_loss: -13
Epoch 7/30
24/24 [==============================] - 0s 16ms/step - loss: 0.4328 - val_loss: -18
Epoch 8/30
24/24 [==============================] - 0s 16ms/step - loss: 0.4178 - val_loss: -32
Epoch 9/30
24/24 [==============================] - 0s 17ms/step - loss: 0.4014 - val_loss: -46
Epoch 10/30
24/24 [==============================] - 0s 16ms/step - loss: 0.3911 - val_loss: -59
Epoch 11/30
24/24 [==============================] - 0s 16ms/step - loss: 0.3867 - val_loss: -67
Epoch 12/30
24/24 [==============================] - 0s 17ms/step - loss: 0.3820 - val_loss: -73
Epoch 13/30
24/24 [==============================] - 0s 16ms/step - loss: 0.3798 - val_loss: -74
Epoch 14/30
24/24 [==============================] - 0s 16ms/step - loss: 0.3772 - val_loss: -78
Epoch 15/30
24/24 [==============================] - 0s 16ms/step - loss: 0.3734 - val_loss: -83
Epoch 16/30
24/24 [==============================] - 0s 16ms/step - loss: 0.3701 - val_loss: -88
Epoch 17/30
24/24 [==============================] - 0s 16ms/step - loss: 0.3670 - val_loss: -94
Epoch 18/30
24/24 [==============================] - 0s 16ms/step - loss: 0.3628 - val_loss: -98
Epoch 19/30
24/24 [==============================] - 0s 16ms/step - loss: 0.3593 - val_loss: -10
Epoch 20/30
24/24 [==============================] - 0s 16ms/step - loss: 0.3559 - val_loss: -11
Epoch 21/30
24/24 [==============================] - 0s 16ms/step - loss: 0.3525 - val_loss: -11
Epoch 22/30
24/24 [==============================] - 0s 16ms/step - loss: 0.3506 - val_loss: -11
Epoch 23/30
24/24 [==============================] - 0s 16ms/step - loss: 0.3484 - val_loss: -12
Epoch 24/30
24/24 [==============================] - 0s 17ms/step - loss: 0.3458 - val_loss: -12
Epoch 25/30
24/24 [==============================] - 0s 16ms/step - loss: 0.3442 - val_loss: -13
Epoch 26/30
24/24 [==============================] - 0s 16ms/step - loss: 0.3434 - val_loss: -12
Epoch 27/30
24/24 [==============================] - 0s 16ms/step - loss: 0.3411 - val_loss: -13
Epoch 28/30
24/24 [==============================] - 0s 16ms/step - loss: 0.3398 - val_loss: -13
Epoch 29/30
24/24 [==============================] - 0s 16ms/step - loss: 0.3386 - val_loss: -13
Epoch 30/30
24/24 [==============================] - 0s 16ms/step - loss: 0.3364 - val_loss: -14
<tensorflow.python.keras.callbacks.History at 0x7efd926df990>
```

# Plotting the image

```
#x_test = x_test.reshape(-1,28,28,1)
predictions = autoencoder.predict(x_test)

n = 10
plt.figure(figsize=(20, 4))
for i in range(n):
    # noisy
    ax = plt.subplot(2, n, i + 1)
    plt.imshow(x_test[i].reshape(28, 28))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)

    # reconstruction
    ax = plt.subplot(2, n, i + 1 + n)
    plt.imshow(predictions[i].reshape(28, 28))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)
plt.show()
```



```
encoded_imgs = encoder.predict(x_test)
decoded_imgs = decoder.predict(encoded_imgs)
decoder.summary()
```

```
Model: "model_40"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_28 (InputLayer)        [(None, 32)]              0         
_________________________________________________________________
dense_81 (Dense)             (None, 64)                2112      
=================================================================
Total params: 2,112
Trainable params: 2,112
Non-trainable params: 0
_________________________________________________________________
```

# TensorBoard

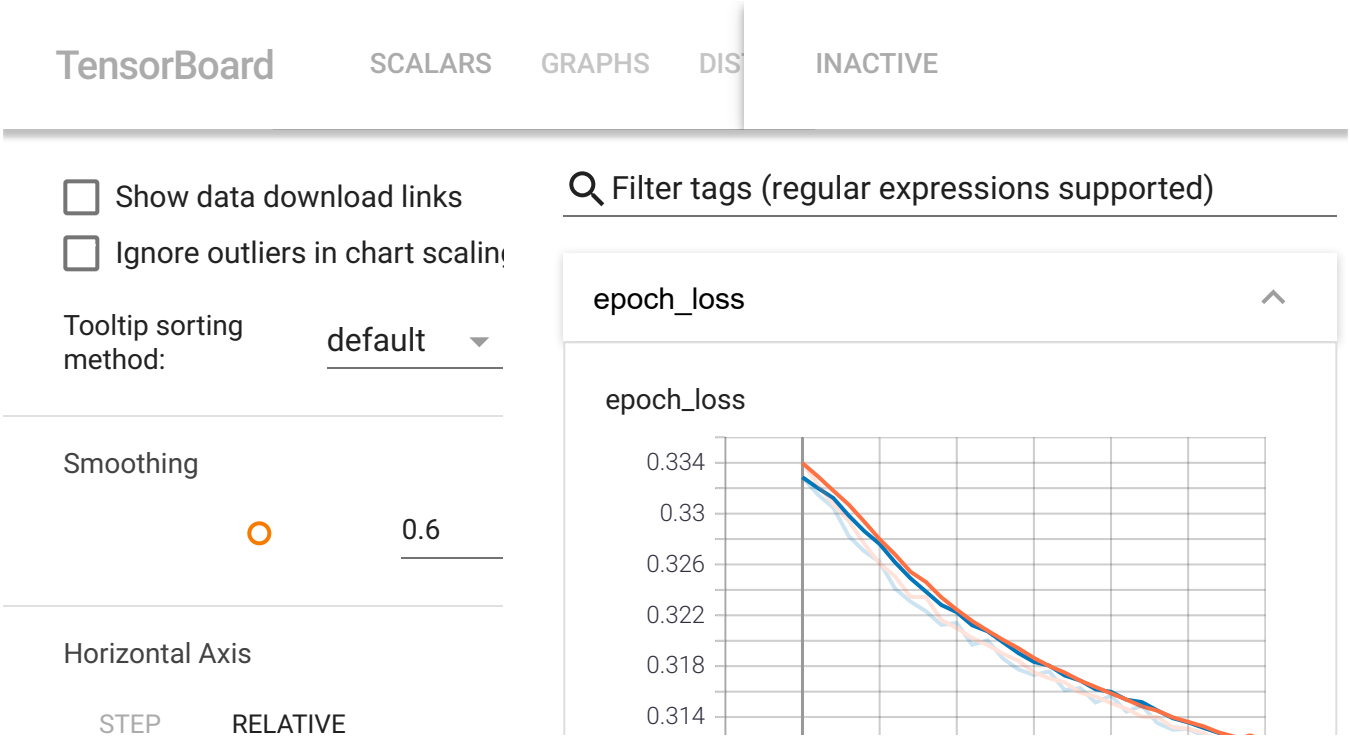
```
from tensorflow.keras.callbacks import TensorBoard

tboard_callback = tf.keras.callbacks.TensorBoard(log_dir = "/tmp/autoencoder",
                                                 histogram_freq = 1,
                                                 profile_batch = '500,520')

autoencoder.fit(x_train, x_train,
                epochs=30,
                batch_size=2048,
                shuffle=True,
                validation_data=(x_validation, x_validation),
                callbacks = [tboard_callback])

#[TensorBoard(log_dir='/tmp/autoencoder', histogram_freq=0, write_graph=True)]
```

```
Epoch 2/30
24/24 [==============================] - 0s 12ms/step - loss: 0.3323 - val_loss: 0.3
Epoch 3/30
24/24 [==============================] - 0s 12ms/step - loss: 0.3307 - val_loss: 0.3
Epoch 4/30
24/24 [==============================] - 0s 12ms/step - loss: 0.3294 - val_loss: 0.3
Epoch 5/30
24/24 [==============================] - 0s 12ms/step - loss: 0.3276 - val_loss: 0.3
Epoch 6/30
24/24 [==============================] - 0s 12ms/step - loss: 0.3261 - val_loss: 0.3
Epoch 7/30
24/24 [==============================] - 0s 12ms/step - loss: 0.3251 - val_loss: 0.3
Epoch 8/30
24/24 [==============================] - 0s 12ms/step - loss: 0.3235 - val_loss: 0.3
Epoch 9/30
24/24 [==============================] - 0s 12ms/step - loss: 0.3234 - val_loss: 0.3
Epoch 10/30
24/24 [==============================] - 0s 12ms/step - loss: 0.3216 - val_loss: 0.3
Epoch 11/30
24/24 [==============================] - 0s 11ms/step - loss: 0.3210 - val_loss: 0.3
Epoch 12/30
24/24 [==============================] - 0s 13ms/step - loss: 0.3202 - val_loss: 0.3
Epoch 13/30
24/24 [==============================] - 0s 11ms/step - loss: 0.3197 - val_loss: 0.3
Epoch 14/30
24/24 [==============================] - 0s 11ms/step - loss: 0.3190 - val_loss: 0.3
Epoch 15/30
24/24 [==============================] - 0s 11ms/step - loss: 0.3184 - val_loss: 0.3
Epoch 16/30
24/24 [==============================] - 0s 11ms/step - loss: 0.3175 - val_loss: 0.3
Epoch 17/30
24/24 [==============================] - 0s 12ms/step - loss: 0.3170 - val_loss: 0.3
Epoch 18/30
24/24 [==============================] - 0s 12ms/step - loss: 0.3167 - val_loss: 0.3
Epoch 19/30
```

```
24/24 [==============================] - 0s 11ms/step - loss: 0.3159 - val_loss: 0.3
Epoch 20/30
24/24 [==============================] - 0s 11ms/step - loss: 0.3156 - val_loss: 0.3
Epoch 21/30
24/24 [==============================] - 0s 18ms/step - loss: 0.3151 - val_loss: 0.3
Epoch 22/30
24/24 [==============================] - 1s 29ms/step - loss: 0.3146 - val_loss: 0.3
Epoch 23/30
24/24 [==============================] - 0s 12ms/step - loss: 0.3140 - val_loss: 0.3
Epoch 24/30
24/24 [==============================] - 0s 12ms/step - loss: 0.3140 - val_loss: 0.3
Epoch 25/30
24/24 [==============================] - 0s 12ms/step - loss: 0.3132 - val_loss: 0.3
Epoch 26/30
24/24 [==============================] - 0s 12ms/step - loss: 0.3131 - val_loss: 0.3
Epoch 27/30
24/24 [==============================] - 0s 12ms/step - loss: 0.3127 - val_loss: 0.3
Epoch 28/30
24/24 [==============================] - 0s 11ms/step - loss: 0.3121 - val_loss: 0.3
Epoch 29/30
24/24 [==============================] - 0s 12ms/step - loss: 0.3120 - val_loss: 0.3
Epoch 30/30
24/24 [==============================] - 0s 12ms/step - loss: 0.3115 - val_loss: 0.3
<tensorflow.python.keras.callbacks.History at 0x7f2a28d95b50>
```

```
# Load the TensorBoard notebook extension.
%load_ext tensorboard
%tensorboard --logdir=/tmp/autoencoder
```

```
The tensorboard extension is already loaded. To reload it, use:
  %reload_ext tensorboard
Reusing TensorBoard on port 6006 (pid 95), started 0:01:01 ago. (Use '!kill 95' to kill
```




```
### Trying to output the encoded and decoded image
```

```
encoded_imgs = encoder.predict(x_test)
decoded_imgs = decoder.predict(encoded_imgs)
decoder.summary()
```

```
Model: "model_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_2 (InputLayer)         [(None, 32)]              0         
_________________________________________________________________
dense_3 (Dense)              (None, 64)                2112      
=================================================================
Total params: 2,112
Trainable params: 2,112
Non-trainable params: 0
_________________________________________________________________
```

# Adding noise to data

```
x_train = x_train.reshape(-1, 28, 28, 1)
x_validation.reshape(-1, 28, 28, 1)
```

```
x_test.reshape(-1, 28, 28, 1)
noise_factor = 0.4
x_train_noisy = x_train + noise_factor * np.random.normal(loc=0.0, scale=1.0, size=x_train.sh
x_validation_noisy = x_validation + noise_factor * np.random.normal(loc=0.0, scale=1.0, size=
x_test_noisy = x_test + noise_factor * np.random.normal(loc=0.0, scale=1.0, size=x_test.shape

x_train_noisy = np.clip(x_train_noisy, 0., 1.)
x_validation_noisy = np.clip(x_validation_noisy, 0., 1.)
x_test_noisy = np.clip(x_test_noisy, 0., 1.)
x_validation = x_validation.reshape(-1, 28, 28, 1)
x_validation.shape
```

```
(12000, 28, 28, 1)
```

```
n = 10
plt.figure(figsize=(30, 30))
for i in range(1, n + 1):
    ax = plt.subplot(1, n, i)
    plt.imshow(x_test_noisy[i].reshape(28, 28))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)
plt.show()
```

```
input_layer = keras.Input(shape=(28, 28, 1))

x = layers.Conv2D(32, (3, 3), activation='relu', padding='same')(input_layer)
x = layers.MaxPooling2D((2, 2), padding='same')(x)
x = layers.Conv2D(32, (3, 3), activation='relu', padding='same')(x)
encoded = layers.MaxPooling2D((2, 2), padding='same')(x)

# At this point the representation is (7, 7, 32)

x = layers.Conv2D(32, (3, 3), activation='relu', padding='same')(encoded)
x = layers.UpSampling2D((2, 2))(x)
x = layers.Conv2D(32, (3, 3), activation='relu', padding='same')(x)
x = layers.UpSampling2D((2, 2))(x)
decoded = layers.Conv2D(1, (3, 3), activation='sigmoid', padding='same')(x)

autoencoder = keras.Model(input_layer, decoded)
autoencoder.compile(optimizer='adam', loss='binary_crossentropy')
```

```
autoencoder.fit(x_train_noisy, x_train
```

```
autoencoder.fit(x_train_noisy, x_train,
                epochs=30,
                batch_size=2048,
                shuffle=True,
                validation_data=(x_validation_noisy, x_validation))
```

```
Epoch 1/30
24/24 [==============================] - 5s 169ms/step - loss: 0.6621 - val_loss: 0.4
Epoch 2/30
24/24 [==============================] - 3s 145ms/step - loss: 0.4352 - val_loss: 0.
Epoch 3/30
24/24 [==============================] - 3s 145ms/step - loss: 0.3415 - val_loss: 0.
Epoch 4/30
24/24 [==============================] - 3s 145ms/step - loss: 0.3214 - val_loss: 0.
Epoch 5/30
24/24 [==============================] - 3s 144ms/step - loss: 0.3152 - val_loss: 0.
Epoch 6/30
24/24 [==============================] - 3s 143ms/step - loss: 0.3118 - val_loss: 0.
Epoch 7/30
24/24 [==============================] - 3s 144ms/step - loss: 0.3084 - val_loss: 0.
Epoch 8/30
24/24 [==============================] - 3s 144ms/step - loss: 0.3070 - val_loss: 0.
Epoch 9/30
24/24 [==============================] - 3s 145ms/step - loss: 0.3057 - val_loss: 0.
Epoch 10/30
24/24 [==============================] - 3s 145ms/step - loss: 0.3034 - val_loss: 0.
Epoch 11/30
24/24 [==============================] - 3s 145ms/step - loss: 0.3028 - val_loss: 0.
Epoch 12/30
24/24 [==============================] - 3s 144ms/step - loss: 0.3015 - val_loss: 0.
Epoch 13/30
24/24 [==============================] - 3s 144ms/step - loss: 0.3006 - val_loss: 0.
Epoch 14/30
24/24 [==============================] - 3s 144ms/step - loss: 0.2993 - val_loss: 0.
Epoch 15/30
24/24 [==============================] - 3s 145ms/step - loss: 0.2989 - val_loss: 0.
Epoch 16/30
24/24 [==============================] - 3s 144ms/step - loss: 0.2992 - val_loss: 0.
Epoch 17/30
24/24 [==============================] - 3s 144ms/step - loss: 0.2978 - val_loss: 0.
Epoch 18/30
24/24 [==============================] - 3s 145ms/step - loss: 0.2981 - val_loss: 0.
Epoch 19/30
24/24 [==============================] - 3s 145ms/step - loss: 0.2972 - val_loss: 0.
Epoch 20/30
24/24 [==============================] - 3s 144ms/step - loss: 0.2968 - val_loss: 0.
Epoch 21/30
24/24 [==============================] - 3s 143ms/step - loss: 0.2968 - val_loss: 0.
Epoch 22/30
24/24 [==============================] - 3s 144ms/step - loss: 0.2967 - val_loss: 0.
Epoch 23/30
24/24 [==============================] - 3s 144ms/step - loss: 0.2950 - val_loss: 0.
Epoch 24/30
24/24 [==============================] - 3s 145ms/step - loss: 0.2951 - val_loss: 0.
Epoch 25/30
24/24 [==============================] - 3s 144ms/step - loss: 0.2954 - val_loss: 0.
Epoch 26/30
```

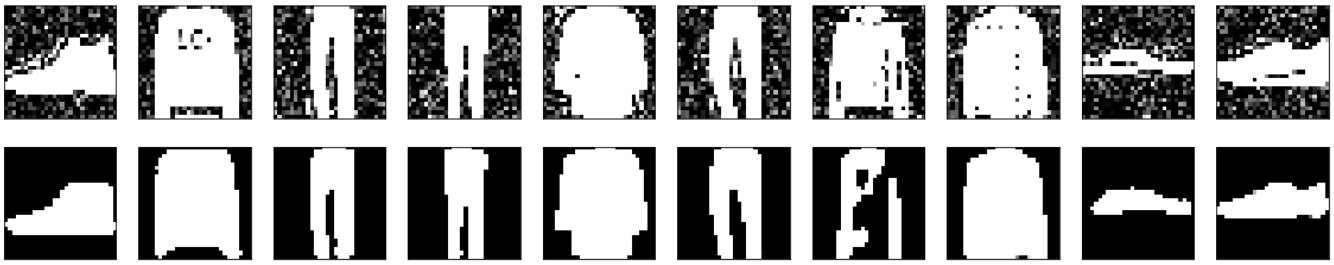
```
24/24 [==============================] - 3s 146ms/step - loss: 0.2940 - val_loss: 0.
Epoch 27/30
24/24 [==============================] - 3s 145ms/step - loss: 0.2940 - val_loss: 0.
Epoch 28/30
24/24 [==============================] - 3s 145ms/step - loss: 0.2944 - val_loss: 0.
Epoch 29/30
```

```
autoencoder.summary()
```

```
Model: "model_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_6 (InputLayer)         [(None, 28, 28, 1)]       0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 28, 28, 32)        320
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 14, 14, 32)        0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 14, 14, 32)        9248
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 7, 7, 32)          0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 7, 7, 32)          9248
_________________________________________________________________
up_sampling2d_2 (UpSampling2 (None, 14, 14, 32)        0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 14, 14, 32)        9248
_________________________________________________________________
up_sampling2d_3 (UpSampling2 (None, 28, 28, 32)        0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 28, 28, 1)         289
=================================================================
Total params: 28,353
Trainable params: 28,353
Non-trainable params: 0
_________________________________________________________________
```

```
x_test = x_test.reshape(-1,28,28,1)
predictions = autoencoder.predict(x_test)

n = 10
plt.figure(figsize=(20, 4))
for i in range(n):
    # noisy
    ax = plt.subplot(2, n, i + 1)
    plt.imshow(x_test_noisy[i].reshape(28, 28))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)

    # reconstruction
    ax = plt.subplot(2, n, i + 1 + n)
```

```
        plt.imshow(predictions[i].reshape(28, 28))
        plt.gray()
        ax.get_xaxis().set_visible(False)
        ax.get_yaxis().set_visible(False)
    plt.show()
```



✓ 0s completed at 12:33 PM