ommender-system-using-autoencoders

April 17, 2024

[5]:
```
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.layers import Input, Dense
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Define the path to your CelebA dataset
data_dir = 'path/to/celeba-dataset'  # Replace 'path/to/celeba-dataset' with␣
 ↪the actual path

# Define image dimensions and batch size
img_width, img_height = 64, 64
batch_size = 256

# Use ImageDataGenerator to load and preprocess images
datagen = ImageDataGenerator(rescale=1. / 255)

train_generator = datagen.flow_from_directory(
    data_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode=None,
    shuffle=True)

# Define the autoencoder model
encoding_dim = 64  # Size of the encoded representations
input_img = Input(shape=(img_width * img_height * 3,))  # For CelebA images
encoded = Dense(encoding_dim, activation='relu')(input_img)
decoded = Dense(img_width * img_height * 3, activation='sigmoid')(encoded)

autoencoder = Model(input_img, decoded)

# Compile the autoencoder
autoencoder.compile(optimizer='adam', loss='binary_crossentropy')

# Train the autoencoder
```

```
autoencoder.fit(train_generator, steps_per_epoch=len(train_generator),␣
 ↪epochs=50)

# Display original and reconstructed images
n = 10  # Number of images to display
plt.figure(figsize=(20, 4))
for i in range(n):
    # Original images
    ax = plt.subplot(2, n, i + 1)
    plt.imshow(train_generator[i])
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)

    # Reconstructed images
    ax = plt.subplot(2, n, i + 1 + n)
    reconstructed_img = autoencoder.predict(train_generator[i])
    plt.imshow(reconstructed_img.reshape(img_width, img_height, 3))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)

plt.show()
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-5-88cf1ed02360> in <cell line: 17>()
     15 datagen = ImageDataGenerator(rescale=1. / 255)
     16
---> 17 train_generator = datagen.flow_from_directory(
     18     data_dir,
     19     target_size=(img_width, img_height),

/usr/local/lib/python3.10/dist-packages/keras/src/preprocessing/image.py in␣
 ↪flow_from_directory(self, directory, target_size, color_mode, classes,␣
 ↪class_mode, batch_size, shuffle, seed, save_to_dir, save_prefix, save_format,␣
 ↪follow_links, subset, interpolation, keep_aspect_ratio)
   1647             and `y` is a numpy array of corresponding labels.
   1648         """
-> 1649         return DirectoryIterator(
   1650             directory,
   1651             self,

/usr/local/lib/python3.10/dist-packages/keras/src/preprocessing/image.py in␣
 ↪__init__(self, directory, image_data_generator, target_size, color_mode,␣
 ↪classes, class_mode, batch_size, shuffle, seed, data_format, save_to_dir,␣
 ↪save_prefix, save_format, follow_links, subset, interpolation,␣
 ↪keep_aspect_ratio, dtype)
```

```
    561         if not classes:
    562             classes = []
--> 563             for subdir in sorted(os.listdir(directory)):
    564                 if os.path.isdir(os.path.join(directory, subdir)):
    565                     classes.append(subdir)

FileNotFoundError: [Errno 2] No such file or directory: 'path/to/celeba-dataset
```