WuBenjaminAssignment9

May 2, 2022

1 CS156 (Introduction to AI), Spring 2022

2 Homework 9 submission

2.0.1 Roster Name: Benjamin Wu

2.0.2 Student ID: 013607880

2.0.3 Email address: benjamin.wu01@sjsu.edu

2.1 Solution

2.2 Import libraries, setup random seed

[ ]:
```
import numpy as np
import keras.datasets.fashion_mnist
import pandas as pd
from sklearn.model_selection import train_test_split
from tensorflow.keras import layers
from tensorflow.keras.layers import Input
import matplotlib.pyplot as plt
```

[ ]:
```
np.random.seed(42)
```

2.3 References and sources

List all your references and sources here. This includes all sites/discussion boards/blogs/posts/etc. where you grabbed some code examples.

2.4 Code the solution

[ ]:
```
(x_train_valid, y_train_valid), (x_test, y_test) = keras.datasets.fashion_mnist.
↪load_data()
```

[ ]:
```
x_train, x_validation, y_train, y_validation = train_test_split(x_train_valid,␣
↪y_train_valid, test_size=0.2, random_state=0)
```

[ ]:
```
input_shape = (28, 28, 1)

x_train = x_train.astype("float32") / 255
```

```
x_validation = x_validation.astype("float32") / 255
x_test = x_test.astype("float32") / 255


x_train = x_train.reshape(-1, 784)
x_validation = x_validation.reshape(-1, 784)
x_test = x_test.reshape(-1, 784)
```

[ ]:
```
input_layer = Input(shape=(784,))
encoded = layers.Dense(128, activation='relu')(input_layer)
encoded = layers.Dense(64, activation='relu')(encoded)
encoded = layers.Dense(32, activation='relu')(encoded)

decoded = layers.Dense(64, activation='relu')(encoded)
decoded = layers.Dense(128, activation='relu')(decoded)
decoded = layers.Dense(784, activation='sigmoid')(decoded)

autoencoder = keras.Model(input_layer, decoded)

encoder = keras.Model(input_layer, encoded)

encoded_input = keras.Input(shape=(32,))
decoder_layer = autoencoder.layers[-3]
decoder = keras.Model(encoded_input, decoder_layer(encoded_input))
```

[ ]:
```
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))
```

```
Epoch 1/30
24/24 [==============================] - 2s 77ms/step - loss: 0.5751 - val_loss:
0.4652
Epoch 2/30
24/24 [==============================] - 2s 70ms/step - loss: 0.4238 - val_loss:
0.3924
Epoch 3/30
24/24 [==============================] - 2s 75ms/step - loss: 0.3770 - val_loss:
0.3676
Epoch 4/30
24/24 [==============================] - 2s 71ms/step - loss: 0.3571 - val_loss:
0.3497
Epoch 5/30
24/24 [==============================] - 2s 80ms/step - loss: 0.3417 - val_loss:
0.3401
```

```
Epoch 6/30
24/24 [==============================] - 2s 80ms/step - loss: 0.3313 - val_loss:
0.3278
Epoch 7/30
24/24 [==============================] - 2s 83ms/step - loss: 0.3239 - val_loss:
0.3283
Epoch 8/30
24/24 [==============================] - 2s 68ms/step - loss: 0.3198 - val_loss:
0.3190
Epoch 9/30
24/24 [==============================] - 2s 64ms/step - loss: 0.3154 - val_loss:
0.3159
Epoch 10/30
24/24 [==============================] - 2s 74ms/step - loss: 0.3129 - val_loss:
0.3139
Epoch 11/30
24/24 [==============================] - 2s 77ms/step - loss: 0.3113 - val_loss:
0.3121
Epoch 12/30
24/24 [==============================] - 2s 86ms/step - loss: 0.3100 - val_loss:
0.3103
Epoch 13/30
24/24 [==============================] - 2s 80ms/step - loss: 0.3078 - val_loss:
0.3089
Epoch 14/30
24/24 [==============================] - 2s 80ms/step - loss: 0.3062 - val_loss:
0.3093
Epoch 15/30
24/24 [==============================] - 2s 71ms/step - loss: 0.3054 - val_loss:
0.3063
Epoch 16/30
24/24 [==============================] - 2s 72ms/step - loss: 0.3037 - val_loss:
0.3052
Epoch 17/30
24/24 [==============================] - 2s 74ms/step - loss: 0.3029 - val_loss:
0.3042
Epoch 18/30
24/24 [==============================] - 2s 76ms/step - loss: 0.3018 - val_loss:
0.3032
Epoch 19/30
24/24 [==============================] - 2s 67ms/step - loss: 0.3009 - val_loss:
0.3023
Epoch 20/30
24/24 [==============================] - 2s 79ms/step - loss: 0.2997 - val_loss:
0.3015
Epoch 21/30
24/24 [==============================] - 2s 75ms/step - loss: 0.3011 - val_loss:
0.3016
```

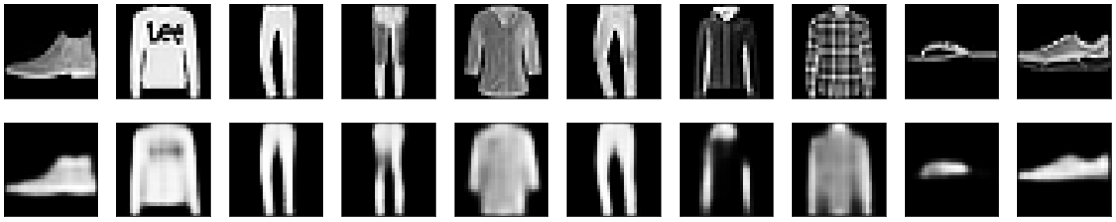
```
Epoch 22/30
24/24 [==============================] - 2s 86ms/step - loss: 0.2986 - val_loss:
0.3002
Epoch 23/30
24/24 [==============================] - 2s 71ms/step - loss: 0.2978 - val_loss:
0.2996
Epoch 24/30
24/24 [==============================] - 2s 80ms/step - loss: 0.2973 - val_loss:
0.3009
Epoch 25/30
24/24 [==============================] - 2s 68ms/step - loss: 0.2971 - val_loss:
0.2987
Epoch 26/30
24/24 [==============================] - 2s 80ms/step - loss: 0.2964 - val_loss:
0.2984
Epoch 27/30
24/24 [==============================] - 2s 83ms/step - loss: 0.2960 - val_loss:
0.2978
Epoch 28/30
24/24 [==============================] - 2s 78ms/step - loss: 0.2963 - val_loss:
0.2975
Epoch 29/30
24/24 [==============================] - 2s 76ms/step - loss: 0.2950 - val_loss:
0.2969
Epoch 30/30
24/24 [==============================] - 2s 75ms/step - loss: 0.2949 - val_loss:
0.2966
```

[ ]:
```
<keras.callbacks.History at 0x28b31b6ebc0>
```

[ ]:
```
predictions = autoencoder.predict(x_test)

n = 10
plt.figure(figsize=(20, 4))
for i in range(n):
    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)

    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()
```



[ ]:
```
x_train = x_train.reshape(-1, 28, 28, 1)
x_validation = x_validation.reshape(-1, 28, 28, 1)
x_test = 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.shape)
x_validation_noisy = x_validation + noise_factor * np.random.normal(loc=0.0,␣
 ↪scale=1.0, size=x_validation.shape)
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.)
```

[ ]:
```
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)

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,
                epochs=30,
                batch_size=2048,
                shuffle=True,
                validation_data=(x_validation_noisy, x_validation))
```

```
Epoch 1/30
24/24 [==============================] - 64s 3s/step - loss: 0.5423 - val_loss:
0.3825
Epoch 2/30
24/24 [==============================] - 63s 3s/step - loss: 0.3504 - val_loss:
0.3320
Epoch 3/30
24/24 [==============================] - 67s 3s/step - loss: 0.3259 - val_loss:
0.3192
Epoch 4/30
24/24 [==============================] - 61s 3s/step - loss: 0.3162 - val_loss:
0.3127
Epoch 5/30
24/24 [==============================] - 61s 3s/step - loss: 0.3114 - val_loss:
0.3097
Epoch 6/30
24/24 [==============================] - 67s 3s/step - loss: 0.3088 - val_loss:
0.3074
Epoch 7/30
24/24 [==============================] - 63s 3s/step - loss: 0.3068 - val_loss:
0.3058
Epoch 8/30
24/24 [==============================] - 65s 3s/step - loss: 0.3052 - val_loss:
0.3041
Epoch 9/30
24/24 [==============================] - 66s 3s/step - loss: 0.3039 - val_loss:
0.3033
Epoch 10/30
24/24 [==============================] - 66s 3s/step - loss: 0.3025 - val_loss:
0.3016
Epoch 11/30
24/24 [==============================] - 67s 3s/step - loss: 0.3015 - val_loss:
0.3006
Epoch 12/30
24/24 [==============================] - 67s 3s/step - loss: 0.3006 - val_loss:
0.2998
Epoch 13/30
24/24 [==============================] - 66s 3s/step - loss: 0.2997 - val_loss:
0.2992
Epoch 14/30
24/24 [==============================] - 66s 3s/step - loss: 0.2991 - val_loss:
0.2983
Epoch 15/30
24/24 [==============================] - 64s 3s/step - loss: 0.2983 - val_loss:
0.2983
Epoch 16/30
24/24 [==============================] - 64s 3s/step - loss: 0.2979 - val_loss:
0.2972
```

```
Epoch 17/30
24/24 [==============================] - 64s 3s/step - loss: 0.2972 - val_loss:
0.2967
Epoch 18/30
24/24 [==============================] - 64s 3s/step - loss: 0.2967 - val_loss:
0.2962
Epoch 19/30
24/24 [==============================] - 63s 3s/step - loss: 0.2962 - val_loss:
0.2958
Epoch 20/30
24/24 [==============================] - 63s 3s/step - loss: 0.2958 - val_loss:
0.2955
Epoch 21/30
24/24 [==============================] - 63s 3s/step - loss: 0.2956 - val_loss:
0.2950
Epoch 22/30
24/24 [==============================] - 65s 3s/step - loss: 0.2952 - val_loss:
0.2947
Epoch 23/30
24/24 [==============================] - 63s 3s/step - loss: 0.2948 - val_loss:
0.2944
Epoch 24/30
24/24 [==============================] - 62s 3s/step - loss: 0.2944 - val_loss:
0.2941
Epoch 25/30
24/24 [==============================] - 65s 3s/step - loss: 0.2942 - val_loss:
0.2938
Epoch 26/30
24/24 [==============================] - 70s 3s/step - loss: 0.2939 - val_loss:
0.2936
Epoch 27/30
24/24 [==============================] - 63s 3s/step - loss: 0.2939 - val_loss:
0.2935
Epoch 28/30
24/24 [==============================] - 57s 2s/step - loss: 0.2934 - val_loss:
0.2931
Epoch 29/30
24/24 [==============================] - 57s 2s/step - loss: 0.2932 - val_loss:
0.2930
Epoch 30/30
24/24 [==============================] - 57s 2s/step - loss: 0.2930 - val_loss:
0.2926
```

[ ]:
```
<keras.callbacks.History at 0x28b364117e0>
```

[ ]:
```
predictions = autoencoder.predict(x_test)
```

```
n = 10
plt.figure(figsize=(20, 4))
for i in range(n):
    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)

    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()
```