assignment12_KingRamsey

March 4, 2022

This is a companion notebook for the book Deep Learning with Python, Second Edition. For readability, it only contains runnable code blocks and section titles, and omits everything else in the book: text paragraphs, figures, and pseudocode.

If you want to be able to follow what's going on, I recommend reading the notebook side by side with your copy of the book.

This notebook was generated for TensorFlow 2.6.

0.1 Generating images with variational autoencoders

0.1.1 Sampling from latent spaces of images

0.1.2 Concept vectors for image editing

0.1.3 Variational autoencoders

0.1.4 Implementing a VAE with Keras

VAE encoder network

[2]:
```
from tensorflow import keras
from tensorflow.keras import layers

latent_dim = 2

encoder_inputs = keras.Input(shape=(28, 28, 1))
x = layers.Conv2D(32, 3, activation="relu", strides=2,␣
 ↪padding="same")(encoder_inputs)
x = layers.Conv2D(64, 3, activation="relu", strides=2, padding="same")(x)
x = layers.Flatten()(x)
x = layers.Dense(16, activation="relu")(x)
z_mean = layers.Dense(latent_dim, name="z_mean")(x)
z_log_var = layers.Dense(latent_dim, name="z_log_var")(x)
encoder = keras.Model(encoder_inputs, [z_mean, z_log_var], name="encoder")
```

[3]:
```
encoder.summary()
```

```
Model: "encoder"
________________________________________________________________________________
__________________
Layer (type)                    Output Shape         Param #     Connected to
```

```
==================================================================================
==================
input_1 (InputLayer)           [(None, 28, 28, 1)]  0

----------------------------------------------------------------------------------
------------------
conv2d (Conv2D)                (None, 14, 14, 32)   320         input_1[0][0]

----------------------------------------------------------------------------------
------------------
conv2d_1 (Conv2D)              (None, 7, 7, 64)     18496       conv2d[0][0]

----------------------------------------------------------------------------------
------------------
flatten (Flatten)              (None, 3136)         0           conv2d_1[0][0]

----------------------------------------------------------------------------------
------------------
dense (Dense)                  (None, 16)           50192       flatten[0][0]

----------------------------------------------------------------------------------
------------------
z_mean (Dense)                 (None, 2)            34          dense[0][0]

----------------------------------------------------------------------------------
------------------
z_log_var (Dense)              (None, 2)            34          dense[0][0]
==================================================================================
==================
Total params: 69,076
Trainable params: 69,076
Non-trainable params: 0

----------------------------------------------------------------------------------
------------------
```

Latent-space-sampling layer

[4]:
```python
import tensorflow as tf

class Sampler(layers.Layer):
    def call(self, z_mean, z_log_var):
        batch_size = tf.shape(z_mean)[0]
        z_size = tf.shape(z_mean)[1]
        epsilon = tf.random.normal(shape=(batch_size, z_size))
        return z_mean + tf.exp(0.5 * z_log_var) * epsilon
```

VAE decoder network, mapping latent space points to images

[5]:
```python
latent_inputs = keras.Input(shape=(latent_dim,))
x = layers.Dense(7 * 7 * 64, activation="relu")(latent_inputs)
x = layers.Reshape((7, 7, 64))(x)
x = layers.Conv2DTranspose(64, 3, activation="relu", strides=2,␣
 ↪padding="same")(x)
```

```
x = layers.Conv2DTranspose(32, 3, activation="relu", strides=2,␣
↪padding="same")(x)
decoder_outputs = layers.Conv2D(1, 3, activation="sigmoid", padding="same")(x)
decoder = keras.Model(latent_inputs, decoder_outputs, name="decoder")
```

[6]:
```
decoder.summary()
```

```
Model: "decoder"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 2)]               0
_________________________________________________________________
dense_1 (Dense)              (None, 3136)              9408
_________________________________________________________________
reshape (Reshape)            (None, 7, 7, 64)          0
_________________________________________________________________
conv2d_transpose (Conv2DTran (None, 14, 14, 64)        36928
_________________________________________________________________
conv2d_transpose_1 (Conv2DTr (None, 28, 28, 32)        18464
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 28, 28, 1)         289
=================================================================
Total params: 65,089
Trainable params: 65,089
Non-trainable params: 0
_________________________________________________________________
```

VAE model with custom train_step()

[7]:
```
class VAE(keras.Model):
    def __init__(self, encoder, decoder, **kwargs):
        super().__init__(**kwargs)
        self.encoder = encoder
        self.decoder = decoder
        self.sampler = Sampler()
        self.total_loss_tracker = keras.metrics.Mean(name="total_loss")
        self.reconstruction_loss_tracker = keras.metrics.Mean(
            name="reconstruction_loss")
        self.kl_loss_tracker = keras.metrics.Mean(name="kl_loss")

    @property
    def metrics(self):
        return [self.total_loss_tracker,
                self.reconstruction_loss_tracker,
                self.kl_loss_tracker]

    def train_step(self, data):
```

```
        with tf.GradientTape() as tape:
            z_mean, z_log_var = self.encoder(data)
            z = self.sampler(z_mean, z_log_var)
            reconstruction = decoder(z)
            reconstruction_loss = tf.reduce_mean(
                tf.reduce_sum(
                    keras.losses.binary_crossentropy(data, reconstruction),
                    axis=(1, 2)
                )
            )
            kl_loss = -0.5 * (1 + z_log_var - tf.square(z_mean) - tf.
↪exp(z_log_var))
            total_loss = reconstruction_loss + tf.reduce_mean(kl_loss)
        grads = tape.gradient(total_loss, self.trainable_weights)
        self.optimizer.apply_gradients(zip(grads, self.trainable_weights))
        self.total_loss_tracker.update_state(total_loss)
        self.reconstruction_loss_tracker.update_state(reconstruction_loss)
        self.kl_loss_tracker.update_state(kl_loss)
        return {
            "total_loss": self.total_loss_tracker.result(),
            "reconstruction_loss": self.reconstruction_loss_tracker.result(),
            "kl_loss": self.kl_loss_tracker.result(),
        }
```

Training the VAE

[8]:
```
import numpy as np

(x_train, _), (x_test, _) = keras.datasets.mnist.load_data()
mnist_digits = np.concatenate([x_train, x_test], axis=0)
mnist_digits = np.expand_dims(mnist_digits, -1).astype("float32") / 255

vae = VAE(encoder, decoder)
vae.compile(optimizer=keras.optimizers.Adam(), run_eagerly=True)
vae.fit(mnist_digits, epochs=30, batch_size=128)
```

```
Epoch 1/30
547/547 [==============================] - 71s 129ms/step - total_loss: 943.2984
- reconstruction_loss: 936.2541 - kl_loss: 7.0439
Epoch 2/30
547/547 [==============================] - 71s 130ms/step - total_loss: 814.1749
- reconstruction_loss: 805.0542 - kl_loss: 9.1214
Epoch 3/30
547/547 [==============================] - 71s 130ms/step - total_loss: 765.6122
- reconstruction_loss: 757.4304 - kl_loss: 8.1818
Epoch 4/30
547/547 [==============================] - 71s 130ms/step - total_loss: 727.9304
- reconstruction_loss: 720.3558 - kl_loss: 7.5751
```

```
Epoch 5/30
547/547 [==============================] - 71s 130ms/step - total_loss: 711.3848
- reconstruction_loss: 704.2355 - kl_loss: 7.1491
Epoch 6/30
547/547 [==============================] - 71s 130ms/step - total_loss: 701.4910
- reconstruction_loss: 694.5892 - kl_loss: 6.9016
Epoch 7/30
547/547 [==============================] - 70s 129ms/step - total_loss: 694.1884
- reconstruction_loss: 687.5301 - kl_loss: 6.6583
Epoch 8/30
547/547 [==============================] - 70s 128ms/step - total_loss: 689.4326
- reconstruction_loss: 682.8683 - kl_loss: 6.5641
Epoch 9/30
547/547 [==============================] - 70s 129ms/step - total_loss: 684.9191
- reconstruction_loss: 678.5387 - kl_loss: 6.3807
Epoch 10/30
547/547 [==============================] - 70s 129ms/step - total_loss: 680.9442
- reconstruction_loss: 674.7290 - kl_loss: 6.2152
Epoch 11/30
547/547 [==============================] - 71s 130ms/step - total_loss: 678.2233
- reconstruction_loss: 672.1674 - kl_loss: 6.0560
Epoch 12/30
547/547 [==============================] - 72s 132ms/step - total_loss: 675.9580
- reconstruction_loss: 670.0017 - kl_loss: 5.9564
Epoch 13/30
547/547 [==============================] - 69s 127ms/step - total_loss: 673.0218
- reconstruction_loss: 667.1535 - kl_loss: 5.8684
Epoch 14/30
547/547 [==============================] - 60s 110ms/step - total_loss: 670.9811
- reconstruction_loss: 665.1780 - kl_loss: 5.8028
Epoch 15/30
547/547 [==============================] - 61s 112ms/step - total_loss: 669.5266
- reconstruction_loss: 663.8165 - kl_loss: 5.7104
Epoch 16/30
547/547 [==============================] - 61s 111ms/step - total_loss: 667.6951
- reconstruction_loss: 662.0244 - kl_loss: 5.6704
Epoch 17/30
547/547 [==============================] - 61s 111ms/step - total_loss: 666.0848
- reconstruction_loss: 660.4729 - kl_loss: 5.6120
Epoch 18/30
547/547 [==============================] - 61s 112ms/step - total_loss: 664.5005
- reconstruction_loss: 658.9274 - kl_loss: 5.5732
Epoch 19/30
547/547 [==============================] - 61s 111ms/step - total_loss: 663.7745
- reconstruction_loss: 658.2141 - kl_loss: 5.5604
Epoch 20/30
547/547 [==============================] - 61s 111ms/step - total_loss: 662.2164
- reconstruction_loss: 656.6661 - kl_loss: 5.5501
```

```
Epoch 21/30
547/547 [==============================] - 61s 111ms/step - total_loss: 660.9861
- reconstruction_loss: 655.5096 - kl_loss: 5.4772
Epoch 22/30
547/547 [==============================] - 61s 111ms/step - total_loss: 660.2222
- reconstruction_loss: 654.7386 - kl_loss: 5.4837
Epoch 23/30
547/547 [==============================] - 61s 111ms/step - total_loss: 659.2493
- reconstruction_loss: 653.7598 - kl_loss: 5.4893
Epoch 24/30
547/547 [==============================] - 60s 111ms/step - total_loss: 657.8759
- reconstruction_loss: 652.4306 - kl_loss: 5.4453
Epoch 25/30
547/547 [==============================] - 60s 110ms/step - total_loss: 657.4072
- reconstruction_loss: 652.0070 - kl_loss: 5.4008
Epoch 26/30
547/547 [==============================] - 61s 111ms/step - total_loss: 656.4970
- reconstruction_loss: 651.0792 - kl_loss: 5.4177
Epoch 27/30
547/547 [==============================] - 61s 111ms/step - total_loss: 655.4706
- reconstruction_loss: 650.0897 - kl_loss: 5.3817
Epoch 28/30
547/547 [==============================] - 61s 111ms/step - total_loss: 655.1507
- reconstruction_loss: 649.7756 - kl_loss: 5.3753
Epoch 29/30
547/547 [==============================] - 61s 111ms/step - total_loss: 654.0383
- reconstruction_loss: 648.6729 - kl_loss: 5.3647
Epoch 30/30
547/547 [==============================] - 61s 111ms/step - total_loss: 653.1835
- reconstruction_loss: 647.8334 - kl_loss: 5.3494
```

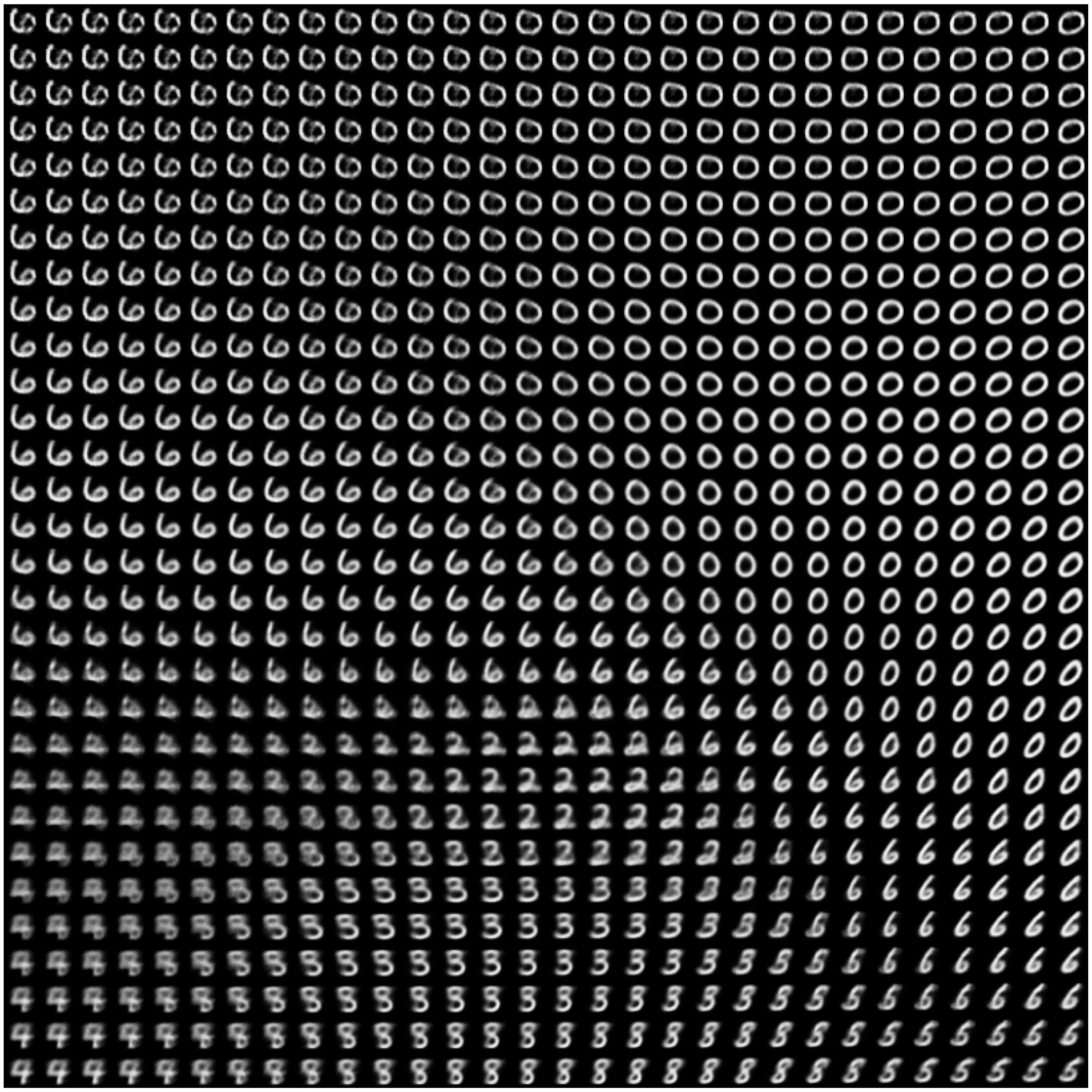
[8]:
```
<tensorflow.python.keras.callbacks.History at 0x7fbca8393190>
```

Sampling a grid of images from the 2D latent space

[9]:
```python
import matplotlib.pyplot as plt

n = 30
digit_size = 28
figure = np.zeros((digit_size * n, digit_size * n))

grid_x = np.linspace(-1, 1, n)
grid_y = np.linspace(-1, 1, n)[::-1]

for i, yi in enumerate(grid_y):
    for j, xi in enumerate(grid_x):
        z_sample = np.array([[xi, yi]])
        x_decoded = vae.decoder.predict(z_sample)
```

```
        digit = x_decoded[0].reshape(digit_size, digit_size)
        figure[
            i * digit_size : (i + 1) * digit_size,
            j * digit_size : (j + 1) * digit_size,
        ] = digit

plt.figure(figsize=(15, 15))
start_range = digit_size // 2
end_range = n * digit_size + start_range
pixel_range = np.arange(start_range, end_range, digit_size)
sample_range_x = np.round(grid_x, 1)
sample_range_y = np.round(grid_y, 1)
plt.xticks(pixel_range, sample_range_x)
plt.yticks(pixel_range, sample_range_y)
plt.xlabel("z[0]")
plt.ylabel("z[1]")
plt.axis("off")
plt.imshow(figure, cmap="Greys_r")
```

```
      ␣
↪---------------------------------------------------------------------------

       NameError                                 Traceback (most recent call␣
↪last)

       <ipython-input-9-0480eec5a4e5> in <module>
        31 plt.imshow(figure, cmap="Greys_r")
        32
   ---> 33 results_dir = Path('results').joinpath('vae')
        34 results_dir.mkdir(parents=True,exist_ok=True)
        35 results_dir.joinpath(figure)


       NameError: name 'Path' is not defined
```



[12]:
```
import os
from pathlib import Path

results_dir = Path('results').joinpath('vae')
results_dir.mkdir(parents=True,exist_ok=True)
results_dir.joinpath('figure.png')
```

[12]:
```
PosixPath('results/vae/figure.png')
```

0.1.5 Wrapping up