assignment12

August 14, 2021

[1]:
```
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

"""
## Create a sampling layer
"""


class Sampling(layers.Layer):
    """Uses (z_mean, z_log_var) to sample z, the vector encoding a digit."""

    def call(self, inputs):
        z_mean, z_log_var = inputs
        batch = tf.shape(z_mean)[0]
        dim = tf.shape(z_mean)[1]
        epsilon = tf.keras.backend.random_normal(shape=(batch, dim))
        return z_mean + tf.exp(0.5 * z_log_var) * epsilon


"""
## Build the encoder
"""

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)
z = Sampling()([z_mean, z_log_var])
encoder = keras.Model(encoder_inputs, [z_mean, z_log_var, z], name="encoder")
```

```
encoder.summary()

"""
## Build the decoder
"""

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.Conv2DTranspose(1, 3, activation="sigmoid",␣
↪padding="same")(x)
decoder = keras.Model(latent_inputs, decoder_outputs, name="decoder")
decoder.summary()

"""
## Define the VAE as a `Model` with a custom `train_step`
"""


class VAE(keras.Model):
    def __init__(self, encoder, decoder, **kwargs):
        super(VAE, self).__init__(**kwargs)
        self.encoder = encoder
        self.decoder = decoder
        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, z = self.encoder(data)
            reconstruction = self.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))
            kl_loss = tf.reduce_mean(tf.reduce_sum(kl_loss, axis=1))
            total_loss = reconstruction_loss + 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 {
            "loss": self.total_loss_tracker.result(),
            "reconstruction_loss": self.reconstruction_loss_tracker.result(),
            "kl_loss": self.kl_loss_tracker.result(),
        }


"""
## Train the VAE
"""

(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())
vae.fit(mnist_digits, epochs=30, batch_size=128)

"""
## Display a grid of sampled digits
"""

import matplotlib.pyplot as plt


def plot_latent_space(vae, n=30, figsize=15):
    # display a n*n 2D manifold of digits
    digit_size = 28
    scale = 1.0
    figure = np.zeros((digit_size * n, digit_size * n))
    # linearly spaced coordinates corresponding to the 2D plot
    # of digit classes in the latent space
```

```
    grid_x = np.linspace(-scale, scale, n)
    grid_y = np.linspace(-scale, scale, 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=(figsize, figsize))
    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.imshow(figure, cmap="Greys_r")
    plt.show()


plot_latent_space(vae)

"""
## Display how the latent space clusters different digit classes
"""


def plot_label_clusters(vae, data, labels):
    # display a 2D plot of the digit classes in the latent space
    z_mean, _, _ = vae.encoder.predict(data)
    plt.figure(figsize=(12, 10))
    plt.scatter(z_mean[:, 0], z_mean[:, 1], c=labels)
    plt.colorbar()
    plt.xlabel("z[0]")
    plt.ylabel("z[1]")
    plt.show()


(x_train, y_train), _ = keras.datasets.mnist.load_data()
x_train = np.expand_dims(x_train, -1).astype("float32") / 255
```

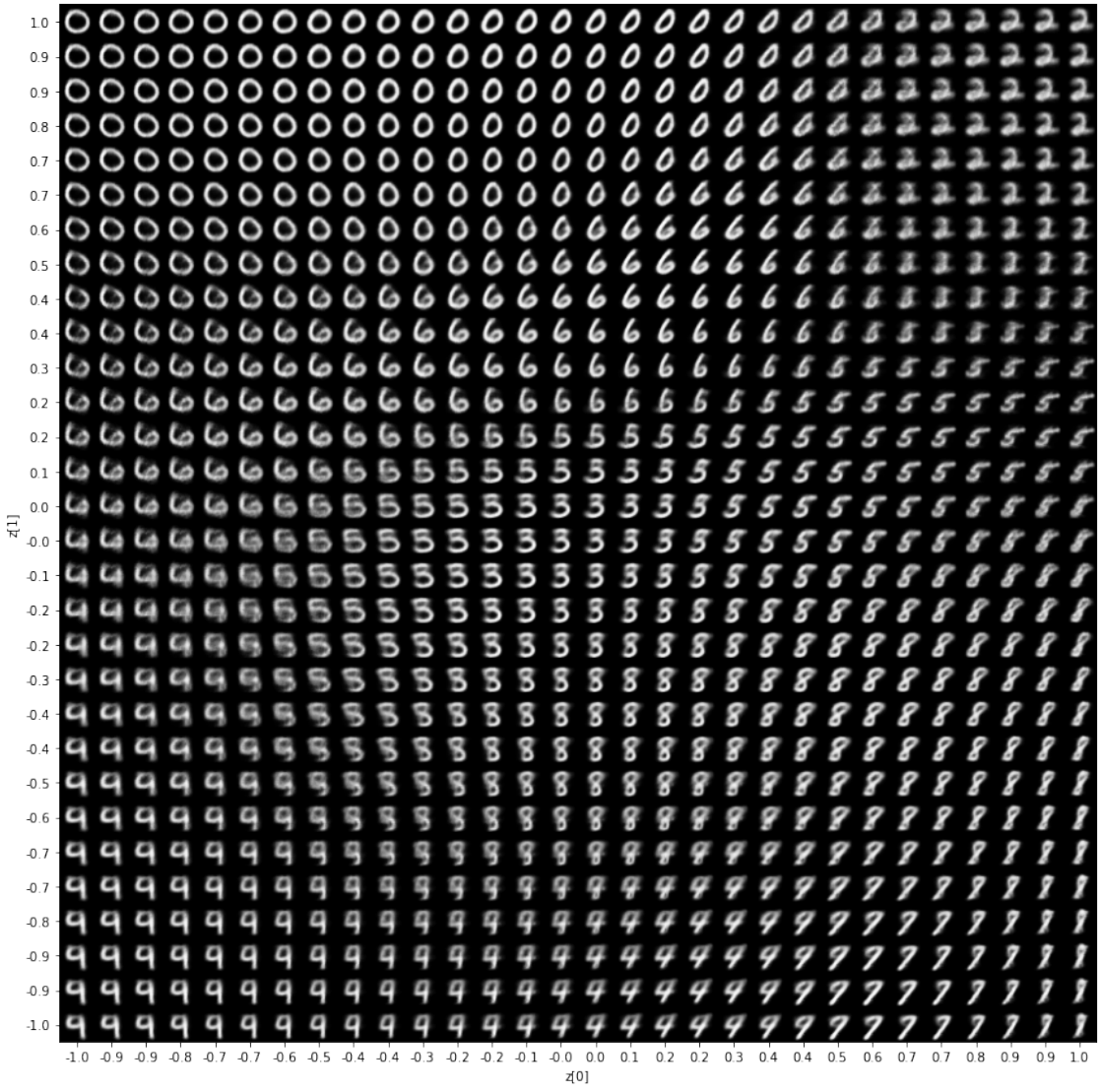
```
plot_label_clusters(vae, x_train, y_train)
```

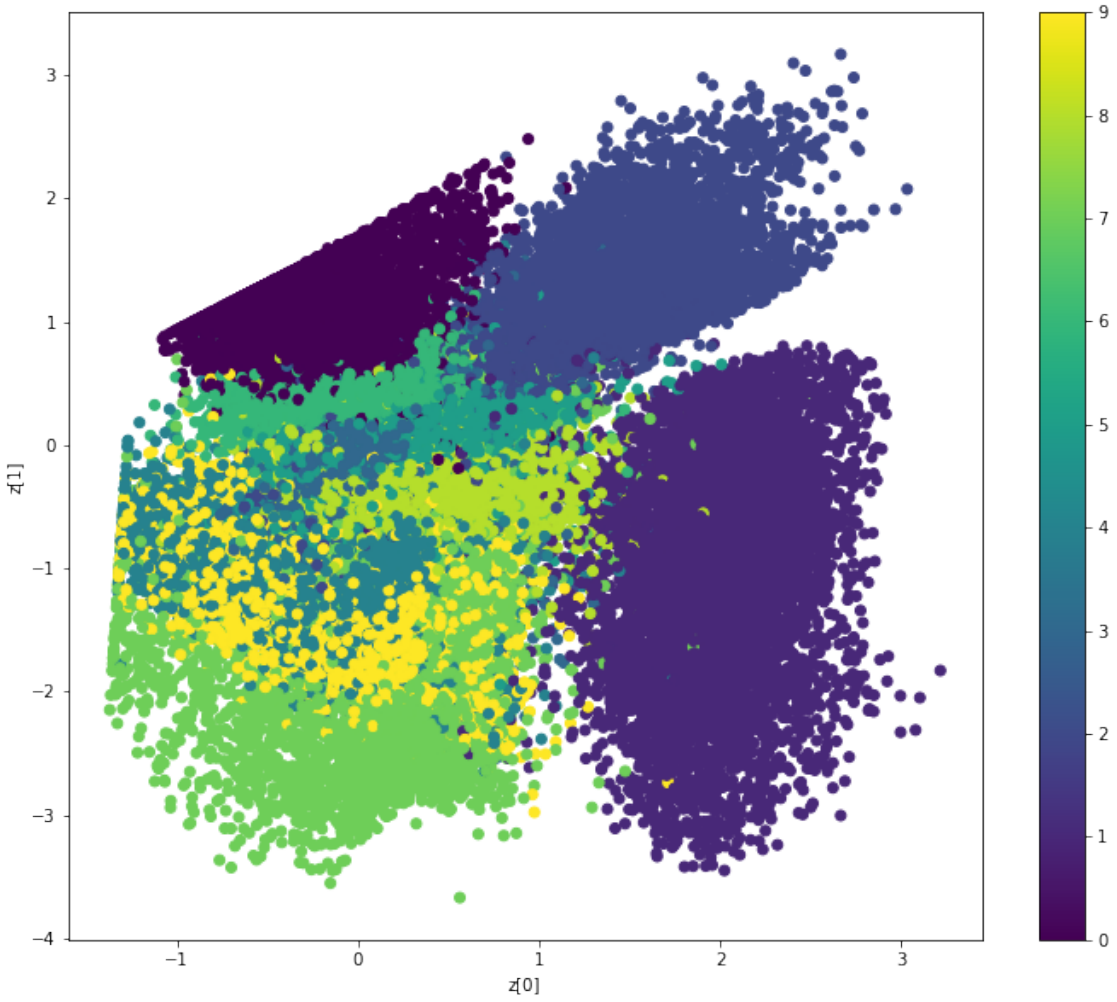
```
Model: "encoder"
__________________________________________________________________________________
__________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================
==================
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]
__________________________________________________________________________________
__________________
sampling (Sampling)             (None, 2)            0           z_mean[0][0]
                                                                 z_log_var[0][0]
==================================================================================
==================
Total params: 69,076
Trainable params: 69,076
Non-trainable params: 0
__________________________________________________________________________________
__________________
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_transpose_2 (Conv2DTr (None, 28, 28, 1)         289
=================================================================
Total params: 65,089
Trainable params: 65,089
Non-trainable params: 0
_________________________________________________________________
Epoch 1/30
547/547 [==============================] - 17s 29ms/step - loss: 250.8808 -
reconstruction_loss: 206.3580 - kl_loss: 3.2322
Epoch 2/30
547/547 [==============================] - 16s 30ms/step - loss: 177.4263 -
reconstruction_loss: 168.5917 - kl_loss: 5.6146
Epoch 3/30
547/547 [==============================] - 16s 29ms/step - loss: 167.5385 -
reconstruction_loss: 160.6515 - kl_loss: 5.9297
Epoch 4/30
547/547 [==============================] - 16s 29ms/step - loss: 162.7022 -
reconstruction_loss: 155.8202 - kl_loss: 6.1876
Epoch 5/30
547/547 [==============================] - 16s 29ms/step - loss: 159.4916 -
reconstruction_loss: 152.8403 - kl_loss: 6.3574
Epoch 6/30
547/547 [==============================] - 16s 29ms/step - loss: 158.1453 -
reconstruction_loss: 151.1727 - kl_loss: 6.4392
Epoch 7/30
547/547 [==============================] - 16s 29ms/step - loss: 156.3537 -
reconstruction_loss: 149.8345 - kl_loss: 6.4910
Epoch 8/30
547/547 [==============================] - 16s 29ms/step - loss: 155.7306 -
reconstruction_loss: 148.8605 - kl_loss: 6.5518
Epoch 9/30
547/547 [==============================] - 16s 29ms/step - loss: 154.9450 -
reconstruction_loss: 148.0338 - kl_loss: 6.5791
Epoch 10/30
547/547 [==============================] - 16s 29ms/step - loss: 154.3637 -
reconstruction_loss: 147.3387 - kl_loss: 6.5946
Epoch 11/30
547/547 [==============================] - 16s 29ms/step - loss: 153.6141 -
reconstruction_loss: 146.7730 - kl_loss: 6.6144
Epoch 12/30
547/547 [==============================] - 16s 29ms/step - loss: 152.4868 -
reconstruction_loss: 146.2099 - kl_loss: 6.6373
```

```
Epoch 13/30
547/547 [==============================] - 16s 29ms/step - loss: 152.6626 -
reconstruction_loss: 145.8254 - kl_loss: 6.6273
Epoch 14/30
547/547 [==============================] - 15s 28ms/step - loss: 152.2488 -
reconstruction_loss: 145.5249 - kl_loss: 6.6361
Epoch 15/30
547/547 [==============================] - 16s 29ms/step - loss: 151.2918 -
reconstruction_loss: 145.1136 - kl_loss: 6.6321
Epoch 16/30
547/547 [==============================] - 16s 29ms/step - loss: 151.3065 -
reconstruction_loss: 144.7920 - kl_loss: 6.6384
Epoch 17/30
547/547 [==============================] - 16s 29ms/step - loss: 150.9131 -
reconstruction_loss: 144.3936 - kl_loss: 6.6207
Epoch 18/30
547/547 [==============================] - 16s 29ms/step - loss: 150.6731 -
reconstruction_loss: 144.2220 - kl_loss: 6.6320
Epoch 19/30
547/547 [==============================] - 16s 29ms/step - loss: 150.7368 -
reconstruction_loss: 144.0112 - kl_loss: 6.6128
Epoch 20/30
547/547 [==============================] - 16s 29ms/step - loss: 150.2649 -
reconstruction_loss: 143.7254 - kl_loss: 6.6160
Epoch 21/30
547/547 [==============================] - 16s 29ms/step - loss: 150.2480 -
reconstruction_loss: 143.4682 - kl_loss: 6.6343
Epoch 22/30
547/547 [==============================] - 16s 29ms/step - loss: 149.9981 -
reconstruction_loss: 143.2682 - kl_loss: 6.6325
Epoch 23/30
547/547 [==============================] - 16s 29ms/step - loss: 150.0634 -
reconstruction_loss: 143.1118 - kl_loss: 6.6343
Epoch 24/30
547/547 [==============================] - 16s 29ms/step - loss: 149.7317 -
reconstruction_loss: 143.0139 - kl_loss: 6.6165
Epoch 25/30
547/547 [==============================] - 16s 29ms/step - loss: 149.1103 -
reconstruction_loss: 142.7590 - kl_loss: 6.6367
Epoch 26/30
547/547 [==============================] - 16s 29ms/step - loss: 149.2443 -
reconstruction_loss: 142.6397 - kl_loss: 6.6328
Epoch 27/30
547/547 [==============================] - 15s 28ms/step - loss: 148.8007 -
reconstruction_loss: 142.4261 - kl_loss: 6.6203
Epoch 28/30
547/547 [==============================] - 15s 27ms/step - loss: 148.7791 -
reconstruction_loss: 142.3780 - kl_loss: 6.6310
```

```
Epoch 29/30
547/547 [==============================] - 16s 29ms/step - loss: 149.0317 -
reconstruction_loss: 142.1983 - kl_loss: 6.6441
Epoch 30/30
547/547 [==============================] - 16s 29ms/step - loss: 148.5467 -
reconstruction_loss: 142.1444 - kl_loss: 6.6355
```




z[0]
z[1]

[ ]: