1. 创建数据集对象

（一）tf.data.Dataset的组成

tensorflow2中通过tf.data.Dataset类对数据集进行了封装。tf.data.Dataset由一系列的可迭代访问的元素组成，每个元素包含一个或多个张量。如图像数据集中，每个元素可以是形状为 长 x 宽 x 通道数 的图片张量，也可以是由图片张量和图片标签张量组成的元组。

（二）创建tf.data.Dataset对象

当数据量较小时，可以通过tf.data.Dataset.from\_tensor\_slices()创建tf.data.Dataset对象，传入一个张量或者多个第0维大小相同的张量，即可按张量的第0维展开构建数据集，数据集的元素数量为张量第0维的大小。当提供多个张量作为输入时，张量的第0维大小必须相同，且必须将多个张量作为元组拼接并作为输入。

如：

import tensorflow as tf

import numpy as np

.

X = tf.constant([2013, 2014, 2015, 2016, 2017])

Y = tf.constant([12000, 14000, 15000, 16500, 17500])

dataset = tf.data.Dataset.from\_tensor\_slices((X, Y))

for x, y in dataset:

print(x.numpy(), y.numpy())

（三）展示mnist\_dataset数据集的图片和标签

import tensorflow as tf  
import matplotlib.pyplot as plt  
import numpy as np  
(train\_data, train\_label), ( test\_data, test\_label ) = tf.keras.datasets.mnist.load\_data()  
train\_data = np.expand\_dims(train\_data.astype(np.float32) / 255., axis=-1)  
mnist\_dataset = tf.data.Dataset.from\_tensor\_slices((train\_data, train\_label))  
  
***#mnist\_dataset = mnist\_dataset.map()***for image, label in mnist\_dataset:  
 plt.title(label.numpy())  
 plt.imshow(image.numpy()[:, :, 0])  
 plt.show()

1. 数据集对象的预处理
2. Dataset.map(fun)

对数据集对象中的每个元素应用函数fun，得到一个新的数据集。

def rot90(image, label):

image = tf.image.rot90(image)

return image, label

mnist\_dataset = mnist\_dataset.map(rot90)

for image, label in mnist\_dataset:

plt.title(label.numpy())

plt.imshow(image.numpy()[:, :, 0])

plt.show()

1. Dataset.shuffle(buffer\_size)

打乱数据集，设定一个buffer\_size大小的缓冲区，取出前buffer\_size个元素放到缓冲区中，再从缓冲区中随机采样，后续的数据填补空缺。当数据集的顺序分布较为随机，则缓冲区的大小可较小，否则需要大的缓冲区。

mnist\_dataset = mnist\_dataset.shuffle(buffer\_size=10000).batch(4)

for images, labels in mnist\_dataset:

fig, axs = plt.subplots(1, 4)

for i in range(4):

axs[i].set\_title(labels.numpy()[i])

axs[i].imshow(images.numpy()[i, :, :, 0])

plt.show()

1. Dataset.batch(batch\_size)

将数据集的数据按batch\_size分成多个批次，每batch\_size个元素，使用tf.stack()在第0位合并为一个元素。

mnist\_dataset = mnist\_dataset.batch(4)

for images, labels in mnist\_dataset: # image: [4, 28, 28, 1], labels: [4]

fig, axs = plt.subplots(1, 4)

for i in range(4):

axs[i].set\_title(labels.numpy()[i])

axs[i].imshow(images.numpy()[i, :, :, 0])

plt.show()

1. 获取数据集的元素
2. for循环

tf.data.Dataset是一个可迭代的对象，可以使用for循环迭代获取数据。

1. iter()

使用iter()显示创建一个迭代器，并使用next()获取下一个元素：

dataset = tf.data.Dataset.from\_tensor\_slices((A, B, C, ...))

it = iter(dataset)

a\_0, b\_0, c\_0, ... = next(it)

a\_1, b\_1, c\_1, ... = next(it)

1. 使用数据集对象

keras支持使用tf.data.Dataset作为输入，如fit()函数需要传入训练数据，标签数据，迭代次数以及每批数据的数量。

model.fit(x=train\_data, y=train\_label, epochs=num\_epochs,

batch\_size=batch\_size)

我们可以直接传入Dataset，在此之前通过Dataset.batch(batch\_size)进行分批，因此，fit（）函数中只需要传入Dataset以及迭代次数即可：

model.fit(mnist\_dataset, epochs=num\_epochs)