AI学习笔记--Tensorflow--使用 tf.data 加载 pandas dataframes

本教程提供了如何将 pandas dataframes 加载到 tf.data.Dataset。

本教程使用了一个小型数据集，由克利夫兰诊所心脏病基金会（Cleveland Clinic Foundation for Heart Disease）提供. 此数据集中有几百行CSV。每行表示一个患者，每列表示一个属性（describe）。我们将使用这些信息来预测患者是否患有心脏病，这是一个二分类问题

```
。
from __future__ import absolute_import, division, print_function,
unicode_literals

import pandas as pd
import tensorflow as tf
import ssl

def get_compiled_model():
  model = tf.keras.Sequential([
    tf.keras.layers.Dense(10, activation='relu'),
    tf.keras.layers.Dense(10, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
  ])

model.compile(optimizer='adam',
                loss='binary_crossentropy',
                metrics=['accuracy'])
  return model

ssl._create_default_https_context = ssl._create_unverified_context

csv_file = tf.keras.utils.get_file('heart.csv',
'https://storage.googleapis.com/applied-dl/heart.csv')
df = pd.read_csv(csv_file)
print(df.head())

df['thal'] = pd.Categorical(df['thal'])
df['thal'] = df.thal.cat.codes
print(df.head())

target = df.pop('target')
dataset = tf.data.Dataset.from_tensor_slices((df.values, target.values))
for feat, targ in dataset.take(5):
  print ('Features: {}, Target: {}'.format(feat, targ))
tf.constant(df['thal'])

train_dataset = dataset.shuffle(len(df)).batch(1)
model = get_compiled_model()
model.fit(train_dataset, epochs=15)
```

```
inputs = {key: tf.keras.layers.Input(shape=(), name=key) for key in
df.keys()}
x = tf.stack(list(inputs.values()), axis=-1)

x = tf.keras.layers.Dense(10, activation='relu')(x)
output = tf.keras.layers.Dense(1, activation='sigmoid')(x)

model_func = tf.keras.Model(inputs=inputs, outputs=output)

model_func.compile(optimizer='adam',
                   loss='binary_crossentropy',
                   metrics=['accuracy'])
dict_slices = tf.data.Dataset.from_tensor_slices((df.to_dict('list'),
target.values)).batch(16)

for dict_slice in dict_slices.take(1):
  print (dict_slice)
model_func.fit(dict_slices, epochs=15)
#
# test_loss, test_accuracy = model.evaluate(dict_slices)
# print('\n\nTest Loss {}, Test Accuracy {}'.format(test_loss,
test_accuracy))
```