

机器学习春作业

班级：2017211106 学号：2017212116 姓名：杨诺诚

目录

问题简述：	3
操作步骤详解：	3
1.安装配置 TensorFlow 环境	3
2.基于深度学习网络训练 MNIST 数据集，获得精度增长情况	3
3.使用 tensorboard 辅助工具观察精度和误差的变化曲线。	3
4.输出 MNIST 训练集中的一张照片，输出训练集的大小	3
5.输入三张手写体照片(28*28)，查看训练结果	4
代码实现：	4
代码一(test_env.py):	4
代码二(tensorboard_test.py):	4
代码三(tensorboard_test.py):	6
代码四(Download_Photo.py):	6
代码五(prediction.py):	7
效果图展示：	8
代码一(test_env.py):	8
代码二(tensorboard_test.py):	10
代码三(tensorboard_test.py):	11
代码四(Download_Photo.py):	12
代码五(prediction.py):	13
说明：	15

操作者和 Git 地址:

组员 (1/1) 2017212116 杨诺诚

Github 此实验库 https://github.com/YangNuoCheng/TF2_MNIST

问题重述:

查阅 kaggle 网站 (<https://www.kaggle.com/c/digit-recognizer/kernels>) 关于 mnist 手写体识别的问题之后, 我决定使用 Tensorflow2.0 和 `tf.keras.datasets.mnist` 中的 mnist 数据集训练, TensorBoard 查看训练效果, 并使用自己的手写体图片预测结果。

我将以以下的步骤完成实验:

1. 安装配置 TensorFlow 环境;
2. 基于深度学习网络训练 MNIST 数据集, 获得精度增长情况;
3. 使用 tensorboard 辅助工具观察精度和误差的变化曲线;
4. 输出 MNIST 训练集中的一张图片, 输出训练集的大小;
5. 输入手写的三个数字并使用训练集预测结果。

操作步骤详解:

1. 安装配置 TensorFlow 环境

- 1) 在 anaconda3 中新建一个环境命名为 tensorflow, 安装 TensorFlow2.0 和其他相关函数库。
- 2) 在 pycharm 中使用 `anaconda3/envs/tensorflow/bin/python` 作为编译器。
- 3) `import tensorflow as tf`, 可以输出 tf 的版本号和具体信息。

2. 基于深度学习网络训练 MNIST 数据集, 获得精度增长情况

- 1) 制作三层结构的模型 (784 维输入, 512 维输入, 10 维输出, 在这个过程中需要将二维数组集合成一维)。
- 2) 将结果的十维结果转化为概率分布。
- 3) 做五次训练
- 4) 将训练好的结果(包括模型和参数值)保存在桌面上。
- 5) 可以在其他环境中调用保存好的模型做结果训练。

3.使用 tensorboard 辅助工具观察精度和误差的变化曲线。

在训练过程中使用 `tf.summary.create_file_writer(train_log_dir)` 在指定位置生成 `tensorboard` 文件，在环境中使用 `tensorboard --logdir=dir` 的方法生成一个浏览器接口，就可以使用浏览器直接查看模型在训练过程中各项参数（`loss`，`accuracy` 等）的变化情况，是以后优化模型架构的常用参数指标。可以直接使用 `pycharm` 中调用好的 `python` 环境和库文件打开，避免了在 `MacOS` 中使用的 `python` 编译器与 `Anaconda3` 中的环境不匹配的问题。

4.输出 MNIST 训练集中的一张照片，输出训练集的大小

- 1)导入 MNIST 数据文件
- 2)定义文件存储的地址
- 3)使用 `28*28` 的方阵标识图片大小
- 4)使用 `PIL.putpixel` 函数为每一个像素点位置赋值。

5.输入三张手写体照片(28*28)，查看训练结果.

- 1)将手写体照片导入 `python` 程序中
- 2)将照片处理成灰度图片
- 3)重新创建空模型
- 4)载入问题 2 中保存的模型
- 4)将照片输入模型中得到预测结果

代码实现：

代码一(test_env.py):

```
import tensorflow as tf
print("tf.__version__: ",tf.__version__)
print("tf.__path__:",tf.__path__)
```

代码二(tensorboard_test.py):

```
import tensorflow as tf
import datetime

mnist = tf.keras.datasets.mnist
```

```

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

def create_model():
    return tf.keras.models.Sequential([
        tf.keras.layers.Flatten(input_shape=(28, 28)),
        tf.keras.layers.Dense(512, activation='relu'),
        tf.keras.layers.Dropout(0.2),
        tf.keras.layers.Dense(10, activation='softmax')
    ])

model = create_model()
# 保存模型
#                                restored_model                                =
tf.keras.models.load_model('/Users/yangnuocheng/Desktop/models/tensorboard.h5')
# 重载模型

train_dataset = tf.data.Dataset.from_tensor_slices((x_train, y_train))
test_dataset = tf.data.Dataset.from_tensor_slices((x_test, y_test))

train_dataset = train_dataset.shuffle(60000).batch(64)
test_dataset = test_dataset.batch(64)

loss_object = tf.keras.losses.SparseCategoricalCrossentropy()
optimizer = tf.keras.optimizers.Adam()

train_loss = tf.keras.metrics.Mean('train_loss', dtype=tf.float32)
train_accuracy = tf.keras.metrics.SparseCategoricalAccuracy('train_accuracy')
test_loss = tf.keras.metrics.Mean('test_loss', dtype=tf.float32)
test_accuracy = tf.keras.metrics.SparseCategoricalAccuracy('test_accuracy')

def train_step(x_train, y_train):
    with tf.GradientTape() as tape:
        predictions = model(x_train, training=True)
        loss = loss_object(y_train, predictions)
        grads = tape.gradient(loss, model.trainable_variables)
        optimizer.apply_gradients(zip(grads, model.trainable_variables))

    train_loss(loss)
    train_accuracy(y_train, predictions)

```

```

def test_step(x_test, y_test):
    predictions = model(x_test)
    loss = loss_object(y_test, predictions)
    test_loss(loss)
    test_accuracy(y_test, predictions)

current_time = datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
train_log_dir = '/Users/yangnuocheng/Desktop/' + current_time + '/train'
test_log_dir = '/Users/yangnuocheng/Desktop/' + current_time + '/test'
train_summary_writer = tf.summary.create_file_writer(train_log_dir)
test_summary_writer = tf.summary.create_file_writer(test_log_dir)

EPOCHS = 5

for epoch in range(EPOCHS):
    for (x_train, y_train) in train_dataset:
        train_step(x_train, y_train)
    with train_summary_writer.as_default():
        tf.summary.scalar('loss', train_loss.result(), step=epoch)
        tf.summary.scalar('accuracy', train_accuracy.result(), step=epoch)

    for (x_test, y_test) in test_dataset:
        test_step(x_test, y_test)
    with test_summary_writer.as_default():
        tf.summary.scalar('loss', test_loss.result(), step=epoch)
        tf.summary.scalar('accuracy', test_accuracy.result(), step=epoch)

    template = 'Epoch {}, Loss: {}, Accuracy: {}, Test Loss: {}, Test Accuracy: {}'
    print(template.format(epoch + 1,
                           train_loss.result(),
                           train_accuracy.result() * 100,
                           test_loss.result(),
                           test_accuracy.result() * 100))

    # Reset metrics every epoch
    train_loss.reset_states()
    test_loss.reset_states()
    train_accuracy.reset_states()
    test_accuracy.reset_states()
model.save('/Users/yangnuocheng/Desktop/models/tensorboard.h5')
# 保存训练好的模型

```

代码三(tensorboard_test.py):

同代码二。运行结束后在命令行中添加:

tensorboard --logdir=/Users/yangnuocheng/Desktop/20200516-214807/
即可。

代码四(Download_Photo.py):

```
import os
import tensorflow as tf
from tensorflow.examples.tutorials.mnist import input_data

from PIL import Image

# 声明图片宽高
rows = 28
cols = 28

# 要提取的图片数量
images_to_extract = 5

# 当前路径下的保存目录
save_dir = "./mnist_digits_images"

# 读入 mnist 数据
mnist = input_data.read_data_sets("MNIST_data/", one_hot=False)
print("mnist.train.images.shape:",mnist.train.images.shape)
print("mnist.train.labels.shape:",mnist.train.labels.shape)

# 通过 python 图片处理库, 生成图片

labels = mnist.train.labels
if mnist.train.images.dtype == "float32":
    for i in range(0,images_to_extract):
        for n in range(28*28):
            if mnist.train.images[i][n] != 0:
                mnist.train.images[i][n] = 255

indices = [0 for x in range(0, 10)]
for i in range(0, images_to_extract):
    img = Image.new("L", (cols, rows))
    for m in range(rows):
        for n in range(cols):
```

```

        # print("mnist.train.images[i][n + m * cols]:",mnist.train.images[i][n + m *
cols])

        img.putpixel((n, m), int(mnist.train.images[i][n + m * cols]))
    # 根据图片所代表的数字 label 生成对应的保存路径
    digit = labels[i]
    path = "%s-%s-%s.bmp" % (save_dir, str(digit), str(i))
    img.save(path)

```

代码五(prediction.py):

```

import tensorflow as tf
import numpy as np
from PIL import Image

def getTestPicArray(filename):
    # 生成文件的 28*28 矩阵
    im = Image.open(filename)
    x_s = 28
    y_s = 28
    out = im.resize((x_s, y_s), Image.ANTIALIAS)
    # 调整图片的大小
    im_arr = np.array(out.convert('L'))
    # 转换为灰度图，值在 0~255 之间
    return im_arr.reshape((1, 784))

model=tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation='softmax')
])

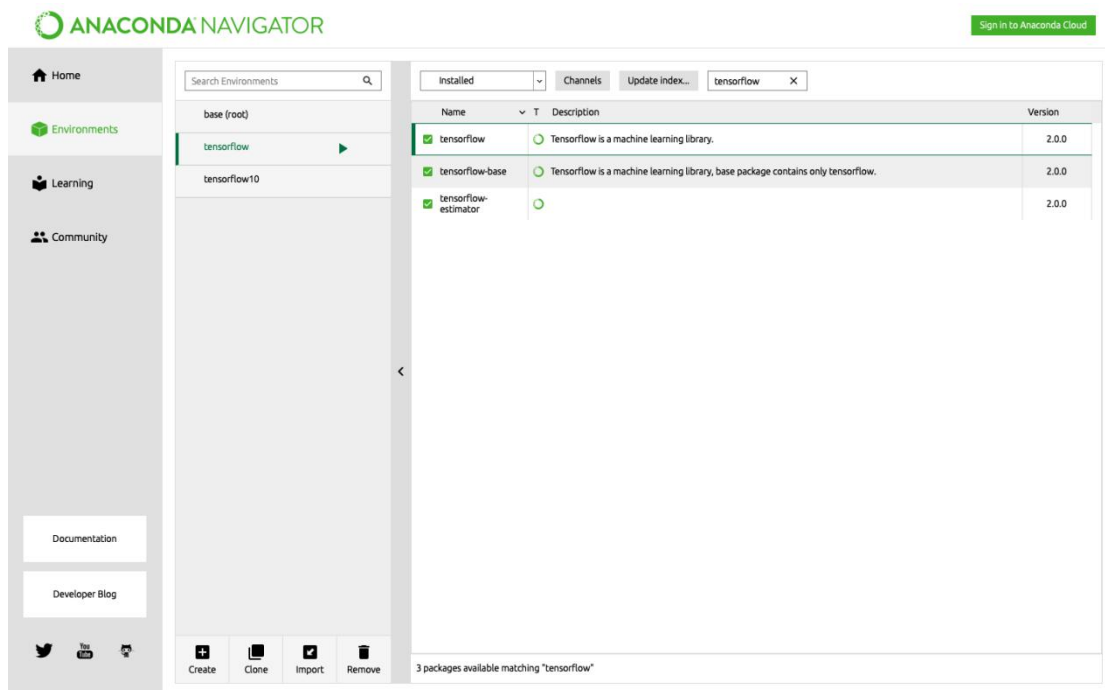
print("重构之前的网络")
model.summary()
# 重建网络
model.load_weights('/Users/yangnuocheng/Desktop/weights/weights.ckpt')
print('loaded weights!')

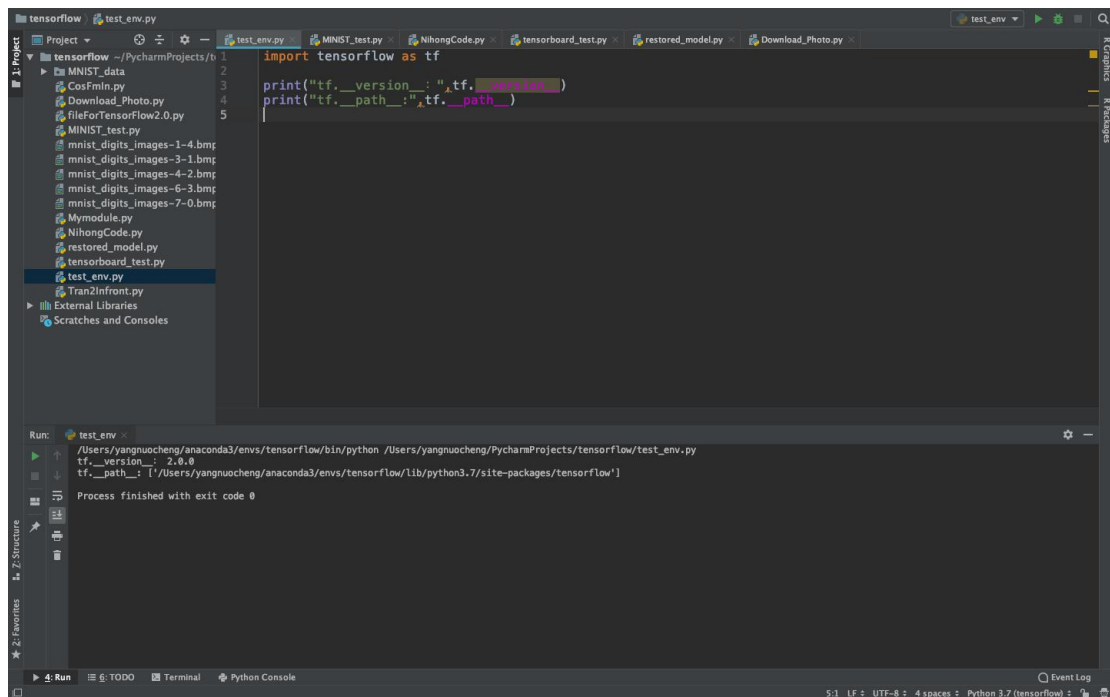
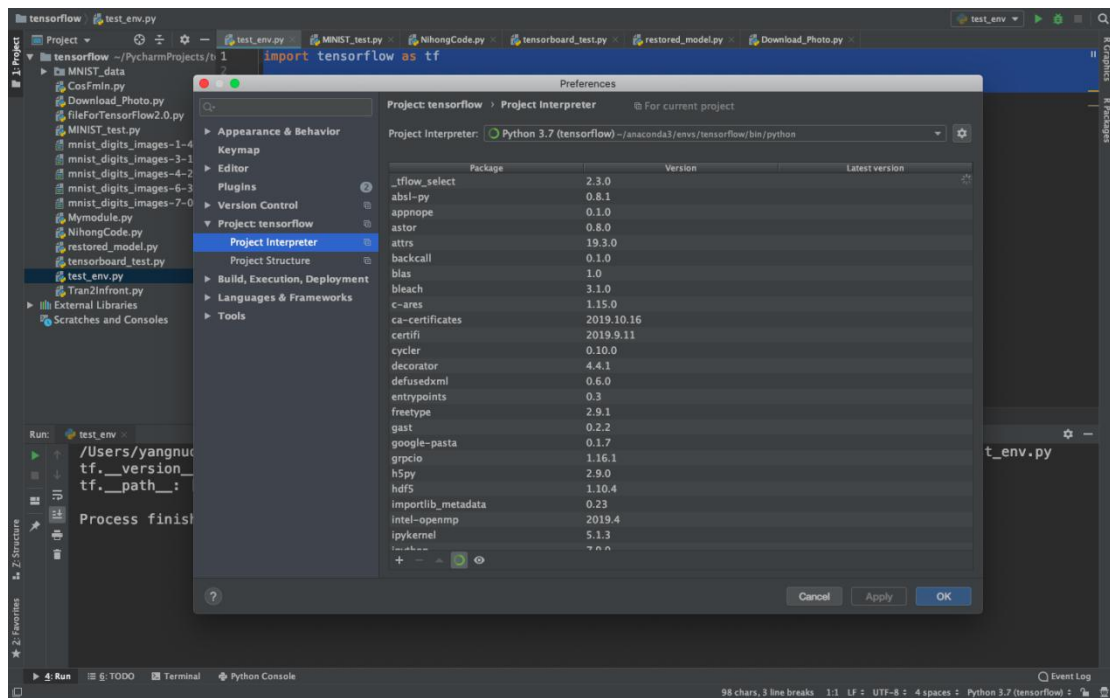
k = getTestPicArray('/Users/yangnuocheng/Desktop/models/sample_8.png')
print(type(k))
out = model.predict(k.reshape(1,28,28))
print(out)
print("模型的预测结果是: ",np.argmax(model.predict(k.reshape(1,28,28)), axis=1))

```

效果图展示：

代码一(test_env.py):





代码二(tensorboard_test.py):

```

predictions = model(x_test)
loss = loss_object(y_test, predictions)

test_loss(loss)
test_accuracy(y_test, predictions)

current_time = datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
train_log_dir = '/Users/yangnuocheng/Desktop/' + current_time + '/train'
test_log_dir = '/Users/yangnuocheng/Desktop/' + current_time + '/test'
train_summary_writer = tf.summary.create_file_writer(train_log_dir)
test_summary_writer = tf.summary.create_file_writer(test_log_dir)

EPOCHS = 5

for epoch in range(EPOCHS):
    for (x_train, y_train) in train_dataset:
        train_step(x_train, y_train)
    with train_summary_writer.as_default():
        tf.summary.scalar('loss', train_loss.result(), step=epoch)
        tf.summary.scalar('accuracy', train_accuracy.result(), step=epoch)

```

Run: tensorboard_test

/Users/yangnuocheng/anaconda3/envs/tensorflow/bin/python /Users/yangnuocheng/PycharmProjects/tensorflow/tensorboard_test.py

2019-11-22 17:32:43.887022: I tensorflow/core/platform/cpu_feature_guard.cc:145] This TensorFlow binary is optimized with Intel(R) MKL-DNN to use the following CPU instructions in performance: AVX, AVX2, FMA, SSE4.1, SSE4.2, SSE4.3. To enable them in non-MKL-DNN operations, rebuild TensorFlow with the appropriate compiler flags.

2019-11-22 17:32:43.898369: I tensorflow/core/common_runtime/process_util.cc:115] Creating new thread pool with default inter op settings: 4. Tune using inter_op_parallelism_threads for best performance.

WARNING:tensorflow:Layer flatten is casting an input tensor from dtype float64 to the layer's dtype of float32, which is new behavior in TensorFlow 2. The layer has dtype float32 because it was created by a tf.nn.conv2d layer. To change this layer to have dtype float64 by default, call 'tf.keras.backend.set_floatx('float64')'. To change just this layer, pass dtype='float64' to the layer constructor. If you are the author of the layer, please add an explicit dtype in its constructor. (See https://www.tensorflow.org/api_guides/python/keras_layer for details.)

Epoch 1, Loss: 0.24299387633806507, Accuracy: 92.92333221435547, Test Loss: 0.11904872953801754, Test Accuracy: 96.4800033569336

Epoch 2, Loss: 0.10611638231142044, Accuracy: 96.82499694824219, Test Loss: 0.08959347754716873, Test Accuracy: 97.19999694824219

Epoch 3, Loss: 0.0718577592253685, Accuracy: 97.79999542236328, Test Loss: 0.0663326233625412, Test Accuracy: 98.00999458683594

Epoch 4, Loss: 0.054628562182188034, Accuracy: 98.26333618164862, Test Loss: 0.0679522454738617, Test Accuracy: 97.8899938964844

Epoch 5, Loss: 0.044279791417377, Accuracy: 98.58000185185469, Test Loss: 0.06180759844223404, Test Accuracy: 98.1399938964844

Process finished with exit code 0

test	--	文件夹	昨天 下午 7:24
events.out.tf...7128.186.v2	676 字节	Gen2Letter02	昨天 下午 7:24
train	--	文件夹	昨天 下午 7:24
events.out.tf...17128.178.v2	676 字节	Gen2Letter02	昨天 下午 7:24

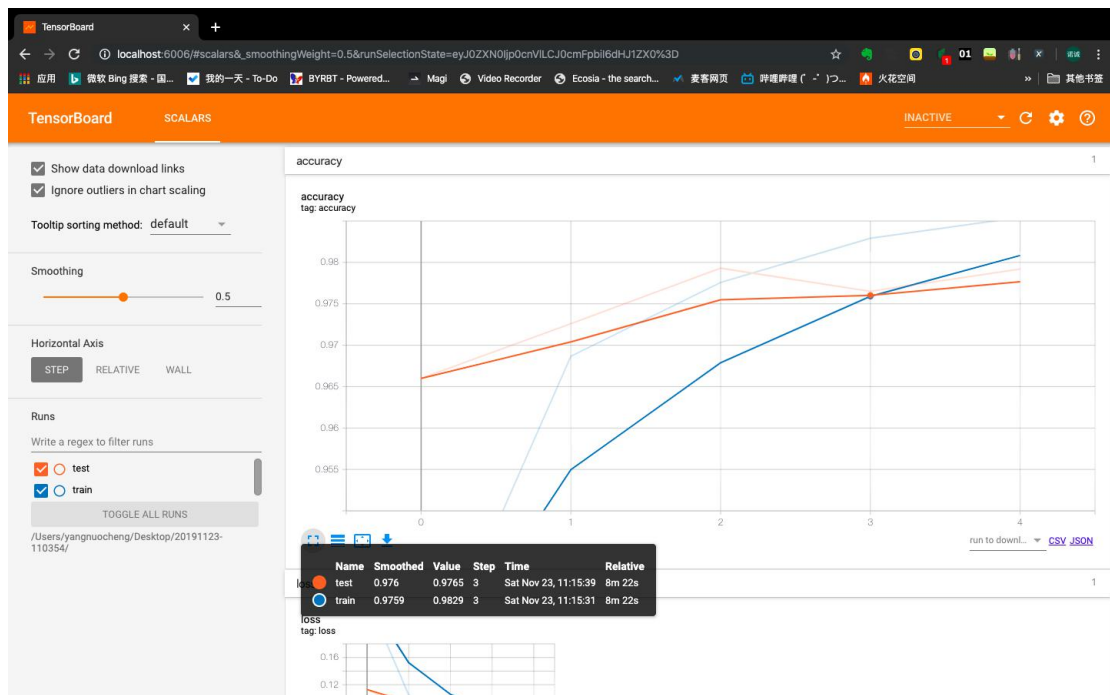
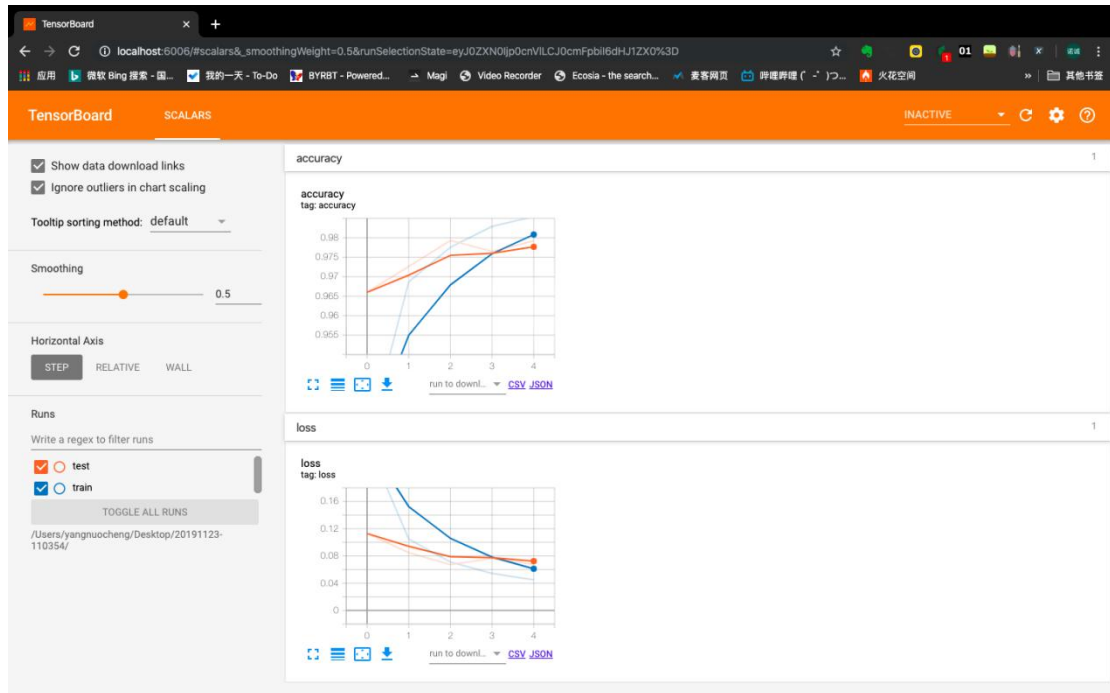
名称	大小	种类	添加日期
tensorboard.h5	1.6 MB	Gen2Letter01	昨天 下午 7:24

代码三(tensorboard_test.py):

命令行输入:

tensorboard --logdir=/Users/yangnuocheng/Desktop/20200516-214807/

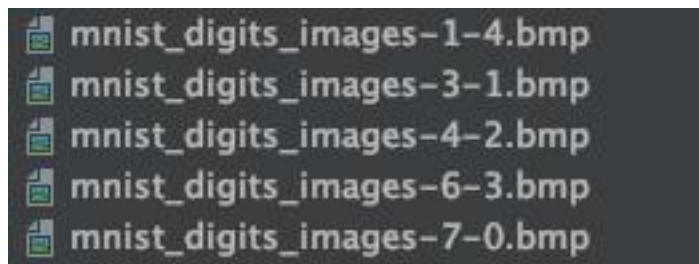
```
Terminal: Local
(tensorflow) yangnuocheng@x86_64-apple-darwin13 scikit-learn-master % tensorboard --logdir=/Users/yangnuocheng/Desktop/20200516-214807/
Serving TensorBoard on localhost; to expose to the network, use a proxy or pass --bind_all
TensorBoard 2.0.0 at http://localhost:6006/ (Press CTRL+C to quit)
```



代码四(Download_Photo.py)



输出的图片命名为 mnist_digits_images-6-3.bmp
说明在 label 中这张图片被标签为 6。



在本文件夹下的目录得到五张图片。

```
10 # 声明图片宽高
11 rows = 28
12 cols = 28
13
14 # 要提取的图片数量
15 images_to_extract = 5
16
17 # 当前路径下的保存目录
18 save_dir = "mnist_digits_images"
19
20 # 读入mnist数据
21 mnist = input_data.read_data_sets("MNIST_data/", one_hot=False)
22 print("mnist.train.images.shape:",mnist.train.images.shape)
23 print("mnist.train.labels.shape:",mnist.train.labels.shape)
24
25 # 通过python图片处理库，生成图片
26
27 labels = mnist.train.labels
28 if mnist.train.images.dtype == "float32":
29     for i in range(0,images_to_extract):
30         for n in range(28*28):
```

Run: Download_Photo

/Users/yanguocheng/anaconda3/envs/tensorflow/bin/python /Users/yanguocheng/PycharmProjects/tensorflow/Download_Photo.py

WARNING:tensorflow:From /Users/yanguocheng/PycharmProjects/tensorflow/Download_Photo.py:21: read_data_sets (from tensorflow.examples.tutorials.mnist.input_data) is deprecated and will be removed in a future version. Use tf.nn.read_file_batches instead.

Instructions for updating:

Please use alternatives such as: tensorflow_datasets.load('mnist')

Extracting MNIST_data/train-images-idx3-ubyte.gz

WARNING:tensorflow:From /Users/yanguocheng/anaconda3/envs/tensorflow/lib/python3.7/site-packages/tensorflow_core/examples/tutorials/mnist/input_data.py:297: _maybe_download (from tensorflow.python.util.tf_export) is deprecated and will be removed in a future version. Use tf.keras.utils.get_file instead.

Instructions for updating:

Please write your own downloading logic.

WARNING:tensorflow:From /Users/yanguocheng/anaconda3/envs/tensorflow/lib/python3.7/site-packages/tensorflow_core/examples/tutorials/mnist/input_data.py:299: _extract_images (from tensorflow.python.util.tf_export) is deprecated and will be removed in a future version. Use tf.nn.read_file_batches instead.

Instructions for updating:

Please use tf.data to implement this functionality.

WARNING:tensorflow:From /Users/yanguocheng/anaconda3/envs/tensorflow/lib/python3.7/site-packages/tensorflow_core/examples/tutorials/mnist/input_data.py:304: _extract_labels (from tensorflow.python.util.tf_export) is deprecated and will be removed in a future version. Use tf.nn.read_file_batches instead.

Instructions for updating:

Please use tf.data to implement this functionality.

Extracting MNIST_data/train-labels-idx1-ubyte.gz

Extracting MNIST_data/train-images-idx3-ubyte.gz

Extracting MNIST_data/train-labels-idx1-ubyte.gz

WARNING:tensorflow:From /Users/yanguocheng/anaconda3/envs/tensorflow/lib/python3.7/site-packages/tensorflow_core/examples/tutorials/mnist/input_data.py:328: _DataSet.__init__ (from tensorflow.python.util.tf_export) is deprecated and will be removed in a future version. Use tf.nn.read_file_batches instead.

Instructions for updating:

Please use alternatives such as official/mnist/DataSet.py from tensorflow/models.

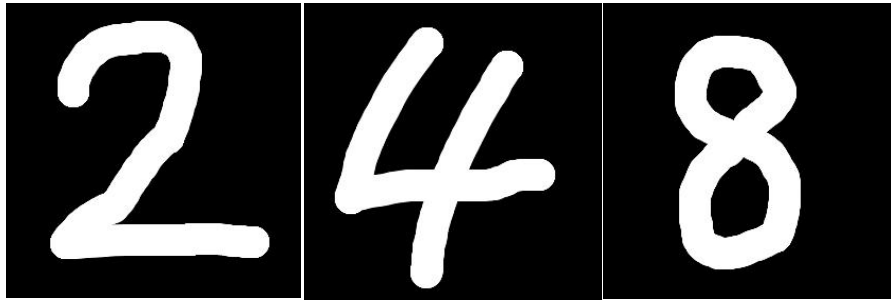
mnist.train.images.shape: (55000, 784)

mnist.train.labels.shape: (55000,)

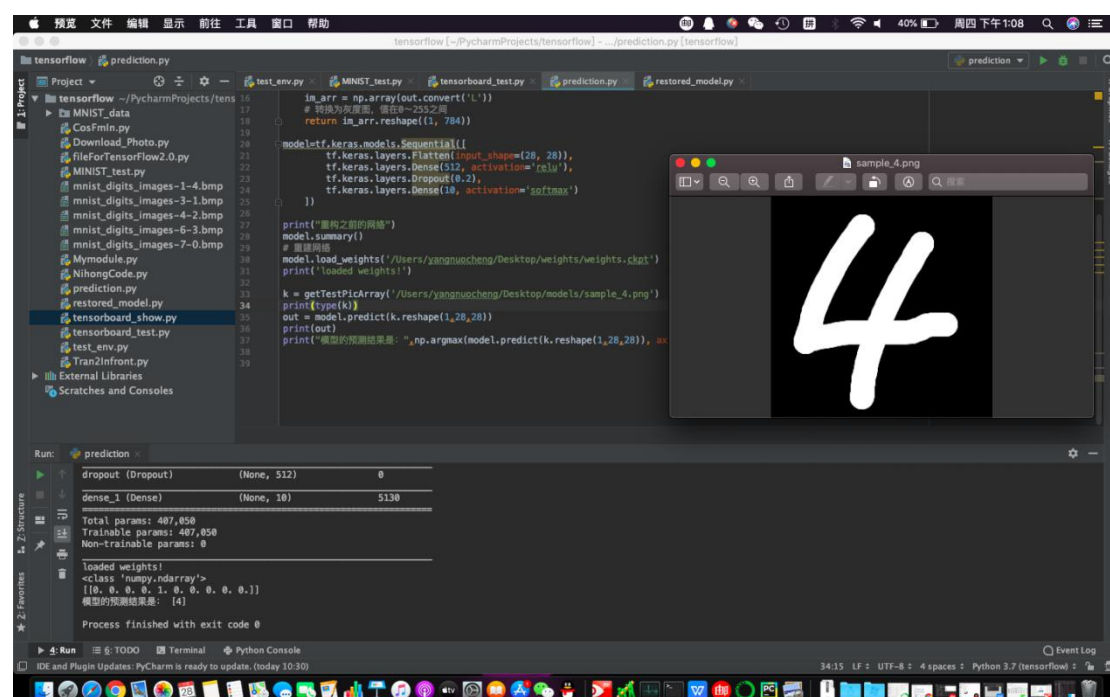
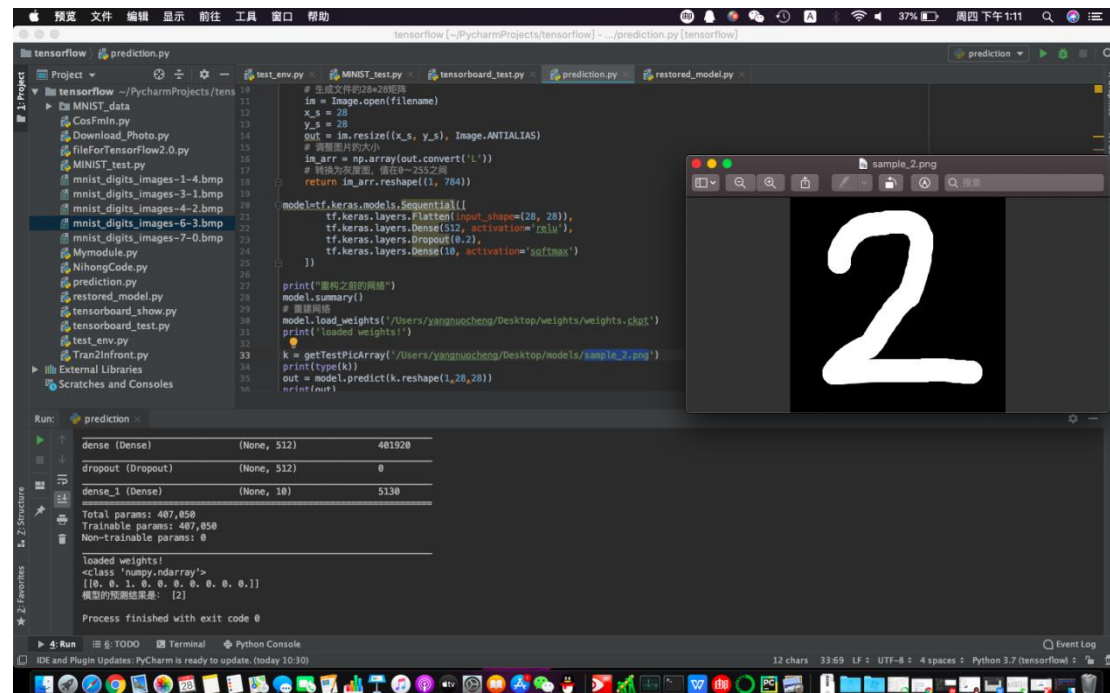
Process finished with exit code 0

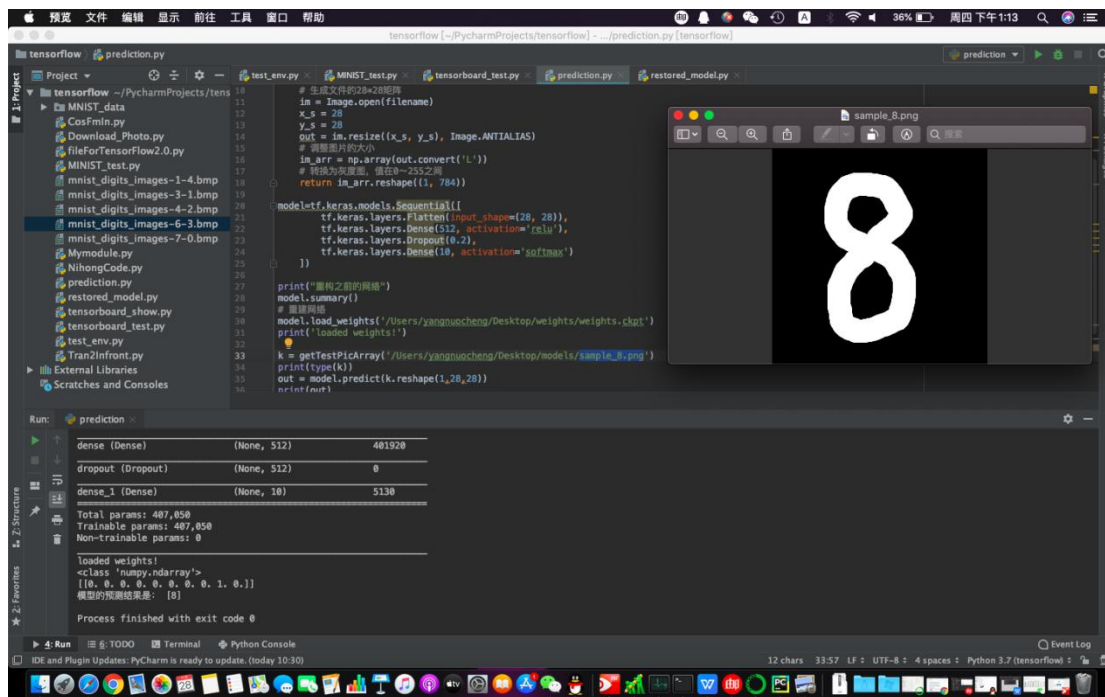
代码五(prediction.py):

分别读取制作好的图片[sample_2,sample_4,sample_8]



在模型中可以获得预测结果展示图如下：





说明

张闯老师：

非常抱歉未能及时提交作业。我在之前的学习过程中没有及时调整学习节奏、掌控学习进度，导致作业未能按时完成提交，非常抱歉。我在未来的学习过程中一定紧跟老师步伐，坚持学习，克服在家的懒惰。

学生
杨诺诚