

使用TensorFlow识别数字验证码

19262010009 关晓林

- 一、安装Python和TensorFlow
 - 1.1. 安装Python
 - 1.1.1 下载Python
 - 1.1.2 为Python设置环境变量
 - 1.1.3 验证是否安装成功
 - 1.2. 安装TensorFlow
 - 1.2.1 安装TensorFlow
 - 1.2.2 验证是否安装成功
 - 1.3. 安装其他库
 - 1.3.1 安装imageio和IPython
 - 1.3.2 验证是否安装成功
- 二、复制Mnist（手写数字识别）代码
- 三、运行Mnist程序，完成测试，包括系统自带的测试数据和自己手写数字的数据
 - 3.1. 训练模型
 - 3.2. 准备手写数字
 - 3.3. 验证手写数字

一、安装Python和TensorFlow

1.1. 安装Python

1.1.1 下载Python

从[python官方网站](#)下载与电脑操作系统对应的最新版本的Python。

我下载的是Windows对应的exe文件，运行exe文件安装Python，安装路径为C:\tool\python3.7.3。

1.1.2 为Python设置环境变量

将C:\tool\python3.7.3和C:\tool\python3.7.3\Scripts添加进系统环境变量。

1.1.3 验证是否安装成功

命令行输入python -V，如果安装成功会提示安装的python的版本，如：

```
Python 3.7.3
```

1.2. 安装TensorFlow

1.2.1 安装TensorFlow

使用pip安装，命令行输入pip install tensorflow。

1.2.2 验证是否安装成功

命令行输入`pip show tensorflow`，安装成功会提示TensorFlow的信息，如：

```
Name: tensorflow
Version: 2.0.0
Summary: TensorFlow is an open source machine learning framework for everyone.
Home-page: https://www.tensorflow.org/
Author: Google Inc.
Author-email: packages@tensorflow.org
License: Apache 2.0
Location: c:\tool\python3.7.3\lib\site-packages
Requires: grpcio, wheel, tensorflow-estimator, keras-preprocessing, opt-einsum,
absl-py, wrapt, numpy, protobuf, astor, termcolor, six, tensorboard, google-pasta,
keras-applications, gast
Required-by:
```

1.3. 安装其他库

1.3.1 安装imageio和IPython

使用pip安装，命令行输入`pip install imageio`和`pip install ipython`

1.3.2 验证是否安装成功

命令行输入`pip show imageio`和`pip show ipython`，提示对应信息即表示安装成功。

二、复制Mnist（手写数字识别）代码

从初学者的 [TensorFlow 2.0 教程](#)，复制并整理手写数字识别的代码，代码如下：

```
from __future__ import absolute_import, division, print_function, unicode_literals
import tensorflow as tf

# 载入并准备好 MNIST 数据集。将样本从整数转换为浮点数：
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

# 将模型的各层堆叠起来，以搭建 tf.keras.Sequential 模型。为训练选择优化器和损失函数：
model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation='softmax')
])
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

```
# 训练并验证模型：
model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test, verbose=2)

# 模型保存
model.save('myminst.model')
```

三、运行Mnist程序，完成测试，包括系统自带的测试数据和自己手写数字的数据

3.1. 训练模型

运行上述运行代码，执行结果为：

```
Train on 60000 samples
Epoch 1/5
60000/60000 [=====] - 6s 104us/sample - loss: 0.2971 -
accuracy: 0.9142
Epoch 2/5
60000/60000 [=====] - 6s 96us/sample - loss: 0.1440 -
accuracy: 0.9570
Epoch 3/5
60000/60000 [=====] - 6s 102us/sample - loss: 0.1073 -
accuracy: 0.9671
Epoch 4/5
60000/60000 [=====] - 6s 99us/sample - loss: 0.0887 -
accuracy: 0.9727
Epoch 5/5
60000/60000 [=====] - 8s 138us/sample - loss: 0.0739 -
accuracy: 0.9772
10000/1 - 1s - loss: 0.0389 - accuracy: 0.9771
```

现在，这个照片分类器的准确度已经达到**98%**，并且模型已经保存为**myminst.model**。

3.2. 准备手写数字

准备的数字的图片如下： 、  、 

3.3. 验证手写数字

验证手写数字代码如下：

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

```
# 测试保存的模型
def predict(image_path):
    # 加载保存的模型
    new_model = tf.keras.models.load_model('myminst.model')

    img = Image.open(image_path).convert('L').resize((28, 28))
    flatten_img = np.array(img) / 255
    flatten_img = flatten_img.reshape(28, 28)
    x = np.array([1 - flatten_img])
    y = new_model.predict(x)

    print('image: {}, predict digit: {}'.format(image_path, np.argmax(y[0])))

def main():
    file_dir = 'C:\\tool\\python_tutorial\\tensorflow_tutorial\\images\\'
    list = os.listdir(file_dir)
    for i in range(0, len(list)):
        image_path = file_dir + list[i]
        predict(image_path)

main()
```

验证结果如下：

```
image: C:\tool\python_tutorial\tensorflow_tutorial\images\0.png, predict digit: 0
image: C:\tool\python_tutorial\tensorflow_tutorial\images\1.png, predict digit: 1
image: C:\tool\python_tutorial\tensorflow_tutorial\images\4.png, predict digit: 4
```

结论：成功识别3张图片里的数字，predict digit分别为0、1、4。