

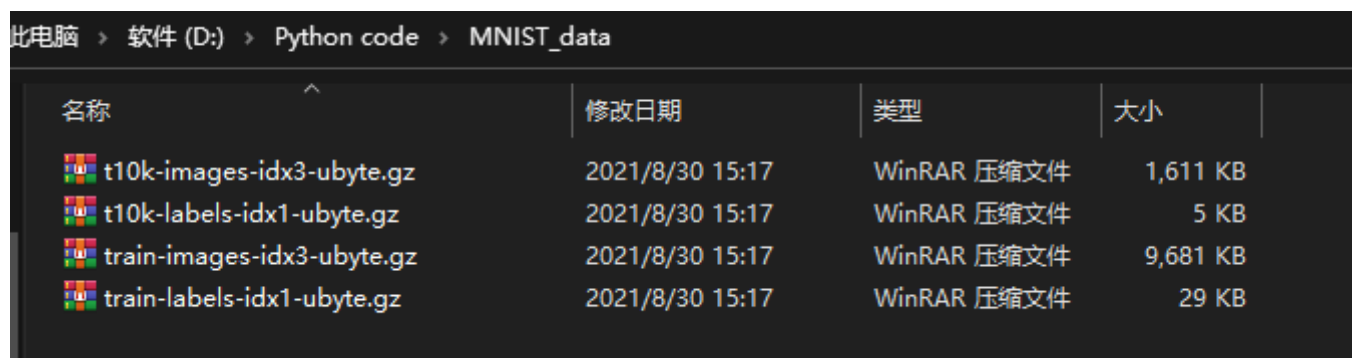
TensorFlow加载MNIST数据集





作者：凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/> (<http://www.cnblogs.com/kailugaji/>).

所用版本：python3.5.2, tensorflow1.8.0, tensorboard1.8.0

首先，在与Python代码相同路径下新建一个文件夹“MNIST_data”。

然后从MNIST数据集官网上<http://yann.lecun.com/exdb/mnist/> (<http://yann.lecun.com/exdb/mnist/>) 下载以下四个文件到“MNIST_data”文件夹中。



名称	修改日期	类型	大小
 t10k-images-idx3-ubyte.gz	2021/8/30 15:17	WinRAR 压缩文件	1,611 KB
 t10k-labels-idx1-ubyte.gz	2021/8/30 15:17	WinRAR 压缩文件	5 KB
 train-images-idx3-ubyte.gz	2021/8/30 15:17	WinRAR 压缩文件	9,681 KB
 train-labels-idx1-ubyte.gz	2021/8/30 15:17	WinRAR 压缩文件	29 KB

注意，不要解压，文件夹只保留这四个文件。

train-images-idx3-ubyte.gz: 训练集图片，包含55000张训练图片与5000张验证图片。

train-labels-idx1-ubyte.gz: 训练集图片对应的数字标签。

t10k-images-idx3-ubyte.gz: 测试集图片，包含10000张测试图片。

t10k-labels-idx1-ubyte.gz: 测试集图片对应的数字标签。

然后运行下面代码即可加载MNIST数据集。

```
In [1]: # 导入TensorFlow中input_data.py文件
```

```
In [2]: from tensorflow.examples.tutorials.mnist import input_data
```

```
In [3]: # 从MNIST_data数据集中读取MNIST数据
```

```
In [4]: mnist = input_data.read_data_sets('MNIST_data', one_hot=True)
```

```
In [5]: # 进一步分析MNIST内容
```

```
In [6]: # 加载数据
train_X = mnist.train.images           #训练集样本
validation_X = mnist.validation.images #验证集样本
test_X = mnist.test.images             #测试集样本
# 加载标签
train_Y = mnist.train.labels           #训练集标签
validation_Y = mnist.validation.labels #验证集标签
test_Y = mnist.test.labels             #测试集标签
```

```
In [7]: print('训练集样本的大小:', train_X.shape)
print('训练集标签的大小:', train_Y.shape)
```

训练集样本的大小: (55000, 784)
训练集标签的大小: (55000, 10)

```
In [8]: print('测试集样本的大小:', test_X.shape)
print('测试集标签的大小:', test_Y.shape)
```

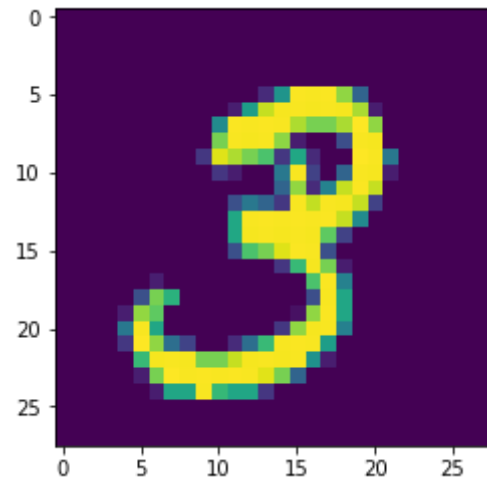
测试集样本的大小: (10000, 784)
测试集标签的大小: (10000, 10)

```
In [9]: print('验证集样本的大小:', validation_X.shape)
print('验证集标签的大小:', validation_Y.shape)
```

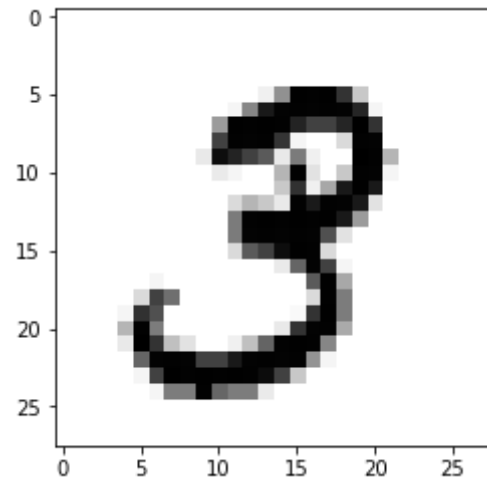
验证集样本的大小: (5000, 784)
验证集标签的大小: (5000, 10)

```
In [10]: import matplotlib.pyplot as plt
```

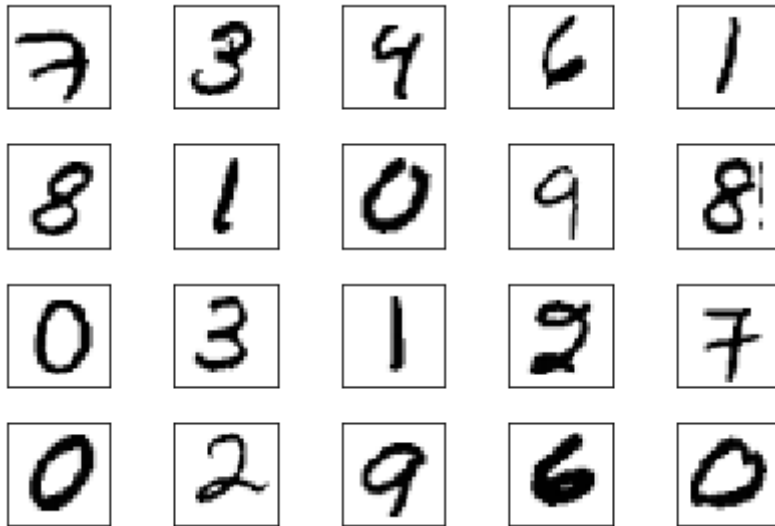
```
In [11]: # 显示出一张RGB图片看看
im = train_X[1]
im = im.reshape(-1, 28)
plt.imshow(im) # RGB图像
plt.show()
```



```
In [12]: # 显示出一张灰度图片看看
im = train_X[1]
im = im.reshape(-1, 28)
plt.imshow(im, cmap='Greys')
plt.show()
```



```
In [13]: #可视化样本，下面是输出了训练集中前20个样本
fig, ax = plt.subplots(nrows=4, ncols=5, sharex='all', sharey='all')
ax = ax.flatten()
for i in range(20):
    img = train_X[i].reshape(28, 28)
    ax[i].imshow(img, cmap='Greys')
ax[0].set_xticks([])
ax[0].set_yticks([])
plt.tight_layout()
plt.show()
```



```
In [14]: #查看数据，例如训练集中第一个样本的内容和标签
print(train_X[0])      #是一个包含784个元素且值在[0, 1]之间的向量
print(train_Y[0])
```

[illegible]

0.	0.	0.	0.	0.	0.
0.	0.	0.54901963	0.9843138	0.9960785	0.9960785
0.9960785	0.9960785	0.9960785	0.9960785	0.9960785	0.9960785
0.9960785	0.9960785	0.9960785	0.9960785	0.9960785	0.9960785
0.7411765	0.09019608	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.8862746	0.9960785	0.81568635	0.7803922	0.7803922	0.7803922
0.7803922	0.54509807	0.2392157	0.2392157	0.2392157	0.2392157
0.2392157	0.5019608	0.8705883	0.9960785	0.9960785	0.7411765
0.08235294	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.14901961	0.32156864
0.0509804	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.13333334	0.8352942	0.9960785	0.9960785	0.45098042	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.32941177
0.9960785	0.9960785	0.9176471	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.32941177	0.9960785	0.9960785
0.9176471	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.4156863	0.6156863	0.9960785	0.9960785	0.95294124	0.20000002
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.09803922
0.45882356	0.8941177	0.8941177	0.8941177	0.9921569	0.9960785
0.9960785	0.9960785	0.9960785	0.94117653	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.26666668	0.4666667	0.86274517	0.9960785	0.9960785
0.9960785	0.9960785	0.9960785	0.9960785	0.9960785	0.9960785
0.9960785	0.5568628	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.14509805	0.73333335	0.9921569
0.9960785	0.9960785	0.9960785	0.8745099	0.8078432	0.8078432
0.29411766	0.26666668	0.8431373	0.9960785	0.9960785	0.45882356

0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.4431373	0.8588236	0.9960785	0.9490197	0.89019614	0.45098042
0.34901962	0.12156864	0.	0.	0.	0.
0.7843138	0.9960785	0.9450981	0.16078432	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.6627451	0.9960785
0.6901961	0.24313727	0.	0.	0.	0.
0.	0.	0.	0.18823531	0.9058824	0.9960785
0.9176471	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.07058824	0.48627454	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.32941177	0.9960785	0.9960785	0.6509804	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.54509807
0.9960785	0.9333334	0.22352943	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.8235295	0.9803922	0.9960785	0.65882355
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.9490197	0.9960785	0.93725497	0.22352943	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.34901962	0.9843138	0.9450981
0.3372549	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.01960784	0.8078432	0.96470594	0.6156863	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.01568628	0.45882356
0.27058825	0.	0.	0.	0.	0.

```
[0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
```