卷积神经网络 Python 训练过程说明

1. MNIST 数据集相关处理

MNIST数据集下载地址：<http://yann.lecun.com/exdb/mnist/>

下载这四个文件

train-images-idx3-ubyte.gz ： training set images (9912422 bytes)

train-labels-idx1-ubyte.gz ： training set labels (28881 bytes)

t10k-images-idx3-ubyte.gz ： test set images (1648877 bytes)

t10k-labels-idx1-ubyte.gz ： test set labels (4542 bytes)

有images 的是图像数据集，labels是数据集对应的标签。

这四个文件，其实内部都是用二进制进行存储的，如果使用一些常用的编辑软件直接打开，会显示乱码。

下方是MNIST数据集官网对文件格式的介绍。

FILE FORMATS FOR THE MNIST DATABASE

The data is stored in a very simple file format designed for storing vectors and multidimensional matrices. General info on this format is given at the end of this page, but you don't need to read that to use the data files.

All the integers in the files are stored in the MSB first (high endian) format used by most non-Intel processors. Users of Intel processors and other low-endian machines must flip the bytes of the header.

There are 4 files:

train-images-idx3-ubyte: training set images

train-labels-idx1-ubyte: training set labels

t10k-images-idx3-ubyte: test set images

t10k-labels-idx1-ubyte: test set labels

The training set contains 60000 examples, and the test set 10000 examples.

The first 5000 examples of the test set are taken from the original NIST training set. The last 5000 are taken from the original NIST test set. The first 5000 are cleaner and easier than the last 5000.

TRAINING SET LABEL FILE (train-labels-idx1-ubyte):

[offset] [type] [value] [description]

0000 32 bit integer 0x00000801(2049) magic number (MSB first)

0004 32 bit integer 60000 number of items

0008 unsigned byte ?? label

0009 unsigned byte ?? label

........

xxxx unsigned byte ?? label

The labels values are 0 to 9.

TRAINING SET IMAGE FILE (train-images-idx3-ubyte):

[offset] [type] [value] [description]

0000 32 bit integer 0x00000803(2051) magic number

0004 32 bit integer 60000 number of images

0008 32 bit integer 28 number of rows

0012 32 bit integer 28 number of columns

0016 unsigned byte ?? pixel

0017 unsigned byte ?? pixel

........

xxxx unsigned byte ?? pixel

Pixels are organized row-wise. Pixel values are 0 to 255. 0 means background (white), 255 means foreground (black).

TEST SET LABEL FILE (t10k-labels-idx1-ubyte):

[offset] [type] [value] [description]

0000 32 bit integer 0x00000801(2049) magic number (MSB first)

0004 32 bit integer 10000 number of items

0008 unsigned byte ?? label

0009 unsigned byte ?? label

........

xxxx unsigned byte ?? label

The labels values are 0 to 9.

TEST SET IMAGE FILE (t10k-images-idx3-ubyte):

[offset] [type] [value] [description]

0000 32 bit integer 0x00000803(2051) magic number

0004 32 bit integer 10000 number of images

0008 32 bit integer 28 number of rows

0012 32 bit integer 28 number of columns

0016 unsigned byte ?? pixel

0017 unsigned byte ?? pixel

........

xxxx unsigned byte ?? pixel

Pixels are organized row-wise. Pixel values are 0 to 255. 0 means background (white), 255 means foreground (black).

THE IDX FILE FORMAT

the IDX file format is a simple format for vectors and multidimensional matrices of various numerical types.

The basic format is

magic number

size in dimension 0

size in dimension 1

size in dimension 2

.....

size in dimension N

data

The magic number is an integer (MSB first). The first 2 bytes are always 0.

The third byte codes the type of the data:

0x08: unsigned byte

0x09: signed byte

0x0B: short (2 bytes)

0x0C: int (4 bytes)

0x0D: float (4 bytes)

0x0E: double (8 bytes)

The 4-th byte codes the number of dimensions of the vector/matrix: 1 for vectors, 2 for matrices....

The sizes in each dimension are 4-byte integers (MSB first, high endian, like in most non-Intel processors).

The data is stored like in a C array, i.e. the index in the last dimension changes the fastest.

要获得数据集中的有效数据，只需要把文件头信息滤除掉就可以了。

以训练数据train-images-idx3-ubyte 为例，根据它的文件格式介绍：

TRAINING SET IMAGE FILE (train-images-idx3-ubyte):

[offset] [type] [value] [description]

0000 32 bit integer 0x00000803(2051) magic number

0004 32 bit integer 60000 number of images

0008 32 bit integer 28 number of rows

0012 32 bit integer 28 number of columns

0016 unsigned byte ?? pixel

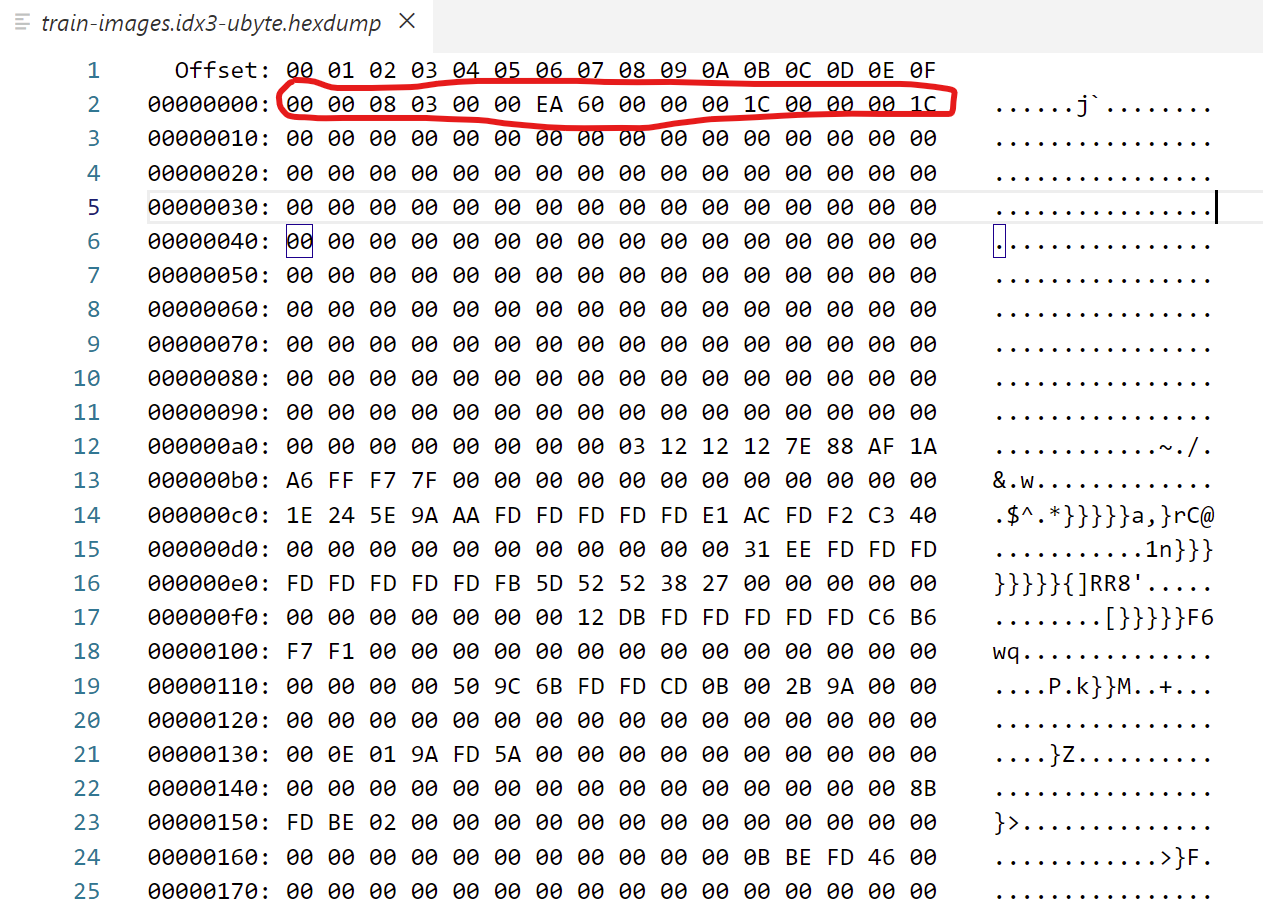
0017 unsigned byte ?? pixel

........

xxxx unsigned byte ?? pixel

Pixels are organized row-wise. Pixel values are 0 to 255. 0 means background (white), 255 means foreground (black).

前16个字节是对该文件的说明，包含数据的格式，有多少张图片，以及图片的尺寸。



从文件头可以得知，数据是以unsigned byte格式保存的，总共有60000张图片，一张图片有28行28列。

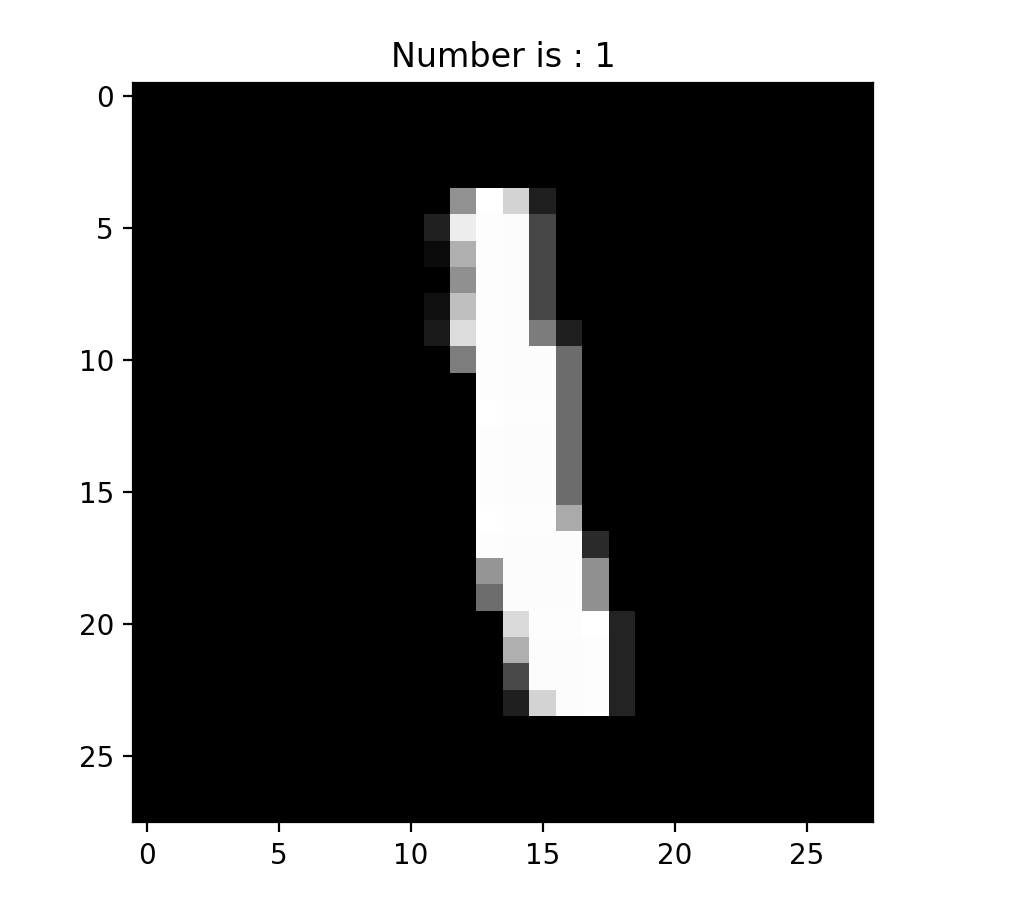
总结一下训练数据集查看图片内容的方法，该文件的前16字节为文件头信息，过掉前16字节后，从第17字节开始属于第一张图片的数据内容，因为图片大小为28\*28，所以第一张图片的数据是第17-第800字节。而字节的顺序，类似于 VGA 画面行扫描的原理。依次是第一行，再第二行，。。。一直是最后一行。

当然，从这个训练数据集里面，你无法直接得出这些数据对应的数字是什么，这需要结合训练数据集的标签train-labels-idx1-ubyte来查看。



现在来看看怎么用 python 代码对数据集进行预处理。当然处理的主要目的，肯定是为了防备后面对数据集进行使用。

|  |
| --- |
| **import** numpy **as** np **import** gzip **import** matplotlib.pyplot **as** plt  img\_size = 784 *# 导入图片* **def** load\_img(filename):  **with** gzip.open(filename, **'rb'**) **as** f:  data = np.frombuffer(f.read(), np.uint8, offset=16)  print(**"Done!"**)  data = data.reshape(-1, img\_size)   **return** data *# 导入标签* **def** load\_label(filename):  **with** gzip.open(filename, **'rb'**) **as** f:  label = np.frombuffer(f.read(), np.uint8, offset=8)   **return** label  **def** load\_mnist():  *# 先下载好 MNIST 数据集到本地，如果使用 python 代码下载，速度太慢了  # 下载地址：http://yann.lecun.com/exdb/mnist/* dataset = {}  dataset[**'train\_img'**] = load\_img(**"./mnist\_dataset/train-images-idx3-ubyte.gz"**)  dataset[**'train\_label'**] = load\_label(**"./mnist\_dataset/train-labels-idx1-ubyte.gz"**)  dataset[**'test\_img'**] = load\_img(**"./mnist\_dataset/t10k-images-idx3-ubyte.gz"**)  dataset[**'test\_label'**] = load\_label(**"./mnist\_dataset/t10k-labels-idx1-ubyte.gz"**)  *# 将图像转换成多维数组* **for** key **in** (**'train\_img'**, **'test\_img'**):  dataset[key] = dataset[key].reshape(-1, 1, 28, 28)   **return** (dataset[**'train\_img'**], dataset[**'train\_label'**]), (dataset[**'test\_img'**], dataset[**'test\_label'**])  (x\_train, t\_train), (x\_test, t\_test) = load\_mnist()  *# print(list(dataset))* plt.imshow(x\_train[6].reshape(28, 28), **'gray'**) plt.title(**'Number is : '** + str(t\_train[6])) plt.show() |



这部分代码的功能，是从已经下载好的MNIST数据集中将数据提取出来，最终保存在(x\_train, t\_train), (x\_test, t\_test)中。

x\_train 表示训练数据集中的图片数据

t\_train 表示训练数据集中的图片标签

x\_test 表示测试数据集中的图片数据

t\_test 表示测试数据集中的图片标签

随机梯度下降