

基于 PyTorch 实现 MNIST 手写数字识别

1. 手写数字数据集 MNIST 简介

MNIST 是一个非常有名的手写体数字识别数据集(手写数字灰度图像数据集)，在很多资料中，这个数据集都会被用作深度学习的入门样例。

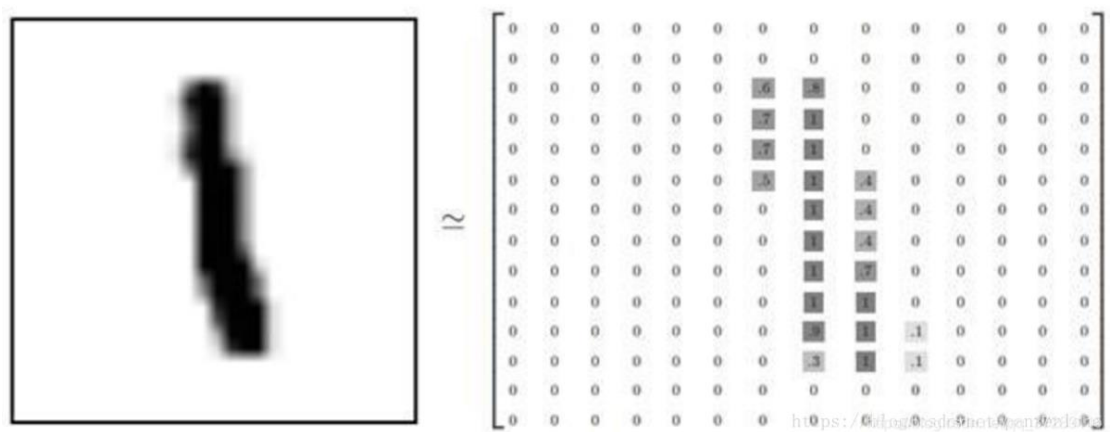
MNIST 数据集是由 0 到 9 的数字图像构成的。训练图像有 6 万张，测试图像有 1 万张。MNIST 数据集是 NIST 数据集的一个子集，它包含了 60000 张图片作为训练数据，10000 张图片作为测试数据。由 250 个不同的人手写而成。

每一张图片都有对应的标签数字，训练图像一共有 60000 张，供研究人员训练出合适的模型。测试图像一共有 10000 张，供研究人员测试训练的模型的性能。

这些手写图片都是 2828 像素，灰度图像。



但是它们并不是作为图像文件存储的，而是作为 28x28 的二维数组保存起来的。数组中的每个元素对应数组中的每个像素。



MNIST 数据集包含四个文件，如下

| 数据集 | MNIST中的文件名 | 下载地址 | 文件大小 |
|-------|----------------------------|---|-----------|
| 训练集图像 | train-images-idx3-ubyte.gz | http://yann.lecun.com/exdb/mnist/train-images-idx3-ubyte.gz | 9912422字节 |
| 训练集标签 | train-labels-idx1-ubyte.gz | http://yann.lecun.com/exdb/mnist/train-labels-idx1-ubyte.gz | 28881字节 |
| 测试集图像 | t10k-images-idx3-ubyte.gz | http://yann.lecun.com/exdb/mnist/t10k-images-idx3-ubyte.gz | 1648877字节 |
| 测试集标签 | t10k-labels-idx1-ubyte.gz | http://yann.lecun.com/exdb/mnist/t10k-labels-idx1-ubyte.gz | 4542字节 |

MNIST 数据集的官方网站：<http://yann.lecun.com/exdb/mnist/>

下载方法主要有两种，官网下载到本地或者 python 下载（dataloaders 方法也可以）。

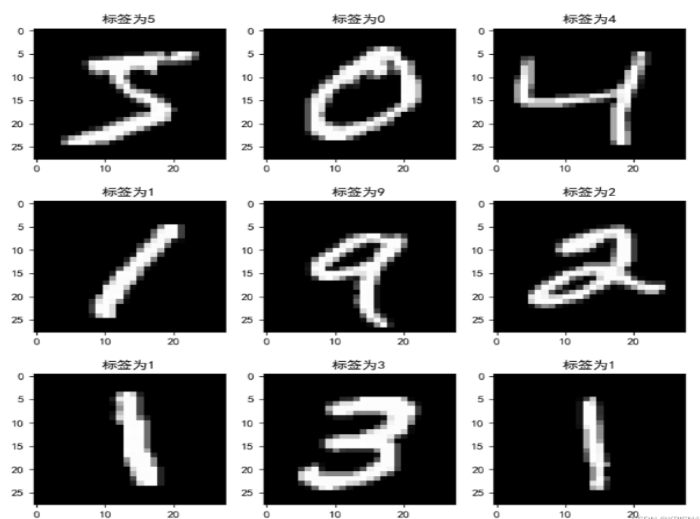
| | | | |
|----------------------------|------------------|--------------|-----------|
| .ipynb_checkpoints | 2023/6/26 14:17 | 文件夹 | |
| displaydata.ipynb | 2023/6/26 14:18 | Jupyter 源文件 | 309 KB |
| mnist.rar | 2023/6/26 14:24 | WinRAR 压缩文件 | 21,503 KB |
| t10k-images.idx3-ubyte | 1998/1/26 23:07 | ID3-UBYTE 文件 | 7,657 KB |
| t10k-images.idx3-ubyte.gz | 2023/6/26 14:14 | WinRAR 压缩文件 | 1,611 KB |
| t10k-labels.idx1-ubyte | 1998/1/26 23:07 | ID3-UBYTE 文件 | 10 KB |
| t10k-labels.idx1-ubyte.gz | 2023/6/26 14:14 | WinRAR 压缩文件 | 5 KB |
| train-images.idx3-ubyte | 1996/11/18 23:36 | ID3-UBYTE 文件 | 45,938 KB |
| train-images.idx3-ubyte.gz | 2023/6/26 14:14 | WinRAR 压缩文件 | 9,681 KB |
| train-labels.idx1-ubyte | 1996/11/18 23:36 | ID3-UBYTE 文件 | 59 KB |
| train-labels.idx1-ubyte.gz | 2023/6/26 14:13 | WinRAR 压缩文件 | 29 KB |

2. 解析 MNIST 数据集

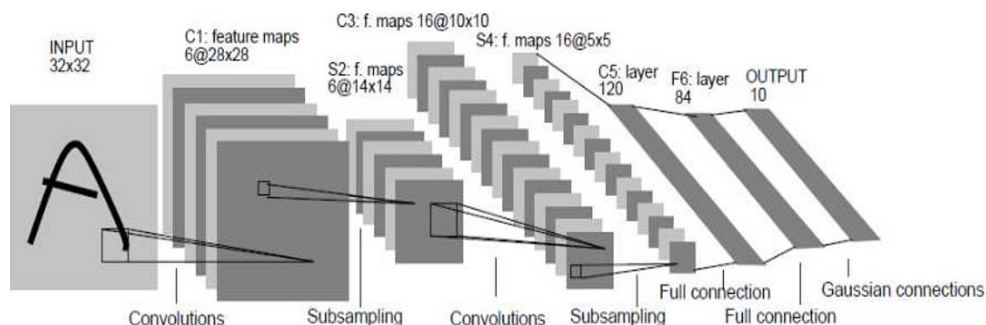
不同于我们常见到的数据集图片通过 jpg 这样的图片格式保存，标签通过 txt、xml、json 等常规文本文件保存。MNIST 的图片和标签均通过二进制文件进行保存，也就是我们无法直接在 Windows 中查看手写数字的图片和标签，必须先解码。

也可查看文件 display.ipynb （anaconda jupyter notebook 打开）

https://blog.csdn.net/KRISNAT/article/details/130670596?ops_request_misc=&request_id=&biz_id=102&utm_term=mnist%E6%95%B0%E6%8D%AE%E9%9B%86%E4%B8%8B%E8%BD%BD&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-1-130670596.nonecase&spm=1018.2226.3001.4187



3. 基于 PyTorch 深度学习框架，搭建 LeNet 实现 MNIST 手写数字识别



大致结构:

输入-> 卷积层+激活函数+池化层 -> 卷积层+激活函数+池化层
-> 线性层+激活函数-> 线性层+激活函数->线性层 (分类器, 输出节点 10, 对应 10 种手写数字)

```

53 model = LeNet()
54 print(model)

```

问题 输出 调试控制台 终端

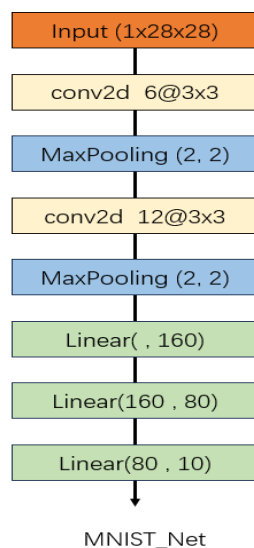
```

PS D:\Pycharm\zzz> D:/Anaconda3/anaconda3/Scripts/activate
PS D:\Pycharm\zzz> conda activate zzz
PS D:\Pycharm\zzz> & D:/Anaconda3/anaconda3/envs/zzz/python.exe d:/Pycharm/zzz/bahe_go/model_zoo/lenet.py
LeNet(
  (conv1): Sequential(
    (0): Conv2d(1, 6, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (conv2): Sequential(
    (0): Conv2d(6, 16, kernel_size=(5, 5), stride=(1, 1))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (fc1): Sequential(
    (0): Linear(in_features=400, out_features=120, bias=True)
    (1): ReLU()
  )
  (fc2): Sequential(
    (0): Linear(in_features=120, out_features=84, bias=True)
    (1): ReLU()
  )
  (fc3): Linear(in_features=84, out_features=10, bias=True)
)

```

4.实验内容:

基于 PyTorch 深度学习框架编程实现如下图所示的网络 (MNIST_Net):



具体的要求如下:

- 1.实现 MNIST_Net 的结构。
- 2.导入 MNIST 数据集的训练子集进行训练, 输出训练过程中的 Loss 变化曲线图以及测试集正确率变化曲线图。
- 3.保存训练模型, 截屏显示最后测试的模型精度。
- 4.将代码、训练好的模型、两个曲线图和截屏的测试精度图放到文件夹, 待教辅老师检查。