

使用 Caffe 进行图像分类的流程

Wu Bin

July 13, 2015

1. 创建训练集

在测试阶段我们选择的训练集是 CIFAR-10，共有 60000 张图像，其中有 10000 张为测试图像，这个分为 10 类，详细的介绍在戴嘉伦的 github 上有详细的介绍。

其实在拿到一个数据集后，一般是包含两部分的，一是图像本身，二是图像中包含 label 数据。有了训练集以后我们要把它转换成 Caffe 可以读取的数据格式，主要有 Lmdb 和 LevelDB 两种形式。

先说 LevelDB，他是一个由 Google 实现的非常高效的 Key-Value 数据库，支持十亿级别的数据量。主要归功于 LSM 算法使得它在非常大的数量级别下还有非常高的性能。

LMDB 的全称是 Lightning Memory-Mapped Database，闪电般的内存映射数据库。它文件结构简单，一个文件夹，里面一个数据文件，一个锁文件。数据随意复制，随意传输。它的访问简单，不需要运行单独的数据库管理进程，只要在访问数据的代码里引用 LMDB 库，访问时给文件路径即可。

它们都是键/值对 (Key/Value Pair) 嵌入式数据库管理系统编程库。虽然 lmdb 的内存消耗是 leveldb 的 1.1 倍，但是 lmdb 的速度比 leveldb 快 10% 至 15%，更重要的是 lmdb 允许多种训练模型同时读取同一组数据集。因此 lmdb 取代了 leveldb 成为 Caffe 默认的数据集生成格式。

对于 CIFAR-10 数据集，Caffe 通过其 example 下的 “create_cifar10.sh” 脚本对 data 目录下的二进制的 Cifar 数据集转换成 lmdb 格式。其中主要用到了 “convert_cifar_data.cpp” 这个程序，这个程序目前还没有看明白。当然如果数据集是自己的数据集的话，可以通过 caffe_root/examples/imagenet 下的 “create_imagenet.sh” 脚本将自有的图片和 Label 文件转化为 lmdb 格式，不过这个目前还在测试中。

2. 对训练集进行训练

在选定 CIFAR-10 作为训练集的情况下，可以直接使用 Caffe 提供的 model 进行训练，CIFAR-10 模型是一个由卷积层、池层、修正线性单元和一个在最顶层的局部对比度归一化的线性分类器组成的卷积神经网络。并且这个模型已经在 CAFFE_ROOT/examples/cifar10 下的 “cifar10_quick_train_test.prototxt” 和 “cifar10_full_train_test.prototxt” 给定了。

在训练的过程中，只要使用对应的 model 对测试集进行训练就好了。

```
1 cd $CAFFE_ROOT
2 ./examples/cifar10/train_full.sh
```

通过阅读“train_full.sh”，可以知道脚本主要使用了 `caffe train --solver=examples/cifar10/cifar10_full`

2.1 MNIST 介绍

MNIST 数据集是手写数字图像，其训练集有 60000 张图像，测试集有 10000 张图像。这只是一个更大的数据集 NIST 的一个子集。这些数字图像已经经过尺寸的规范化与在图像的置中心处理。

这个数据集适合一些人想用学习技术以及模式识别方法来处理真实世界的的数据，而且不需要花太多功夫在预处理和格式化方面的工作上。

这个数据集总共有 4 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)

在 NIST 中的手写数字图像，都为灰度保真图，而且经过规范化处理，图像大小为 28×28 。

MNIST 数据集是由 NIST 的 Special Database 3 (SD3) 和 Special Database 1 (SD1) 所组成，二者包含了手写数字的二值图像。NIST 中最初将 SD-3 作为训练集，SD-1 作为测试集。然而，SD-3 比 SD-1 更清晰，更容易识别。主要是因为 SD-3 是从 Census Bureau 的员工处收集的，而 SD-1 是从高中生处收集的。从学习实验中得出可靠的结论要求结果与训练集的选择无关，并且测试样本是完整数据集。因此，必须通过混合 NIST 数据集来建立一个全新的数据集。

MNIST 训练集是由 SD-3 的 30000 张图像和 SD-1 的 300000 张图像组成。测试集是由 SD-3 的 5000 张图像和 SD-1 的 5000 张图像组成。训练集中所包含的图像是从大约 250 个人中收集到的。基本可以判定训练集与测试集的数字是不同的人所写的。

SD-1 是由 500 个不同的人所写的 58527 张数字图像。在 SD-1 中，数据是杂乱无章地排列的。这个排列方式与 SD-3 不同，在 SD-3 中每个人所写的数字图像是按序列排列的。我们可用人工方法来分辨 SD-1 的数字图像由哪个人缩写。我们将 SD-1 分为两部分：由前 250 个人所写的数字图像放在训

练集中，剩下的 250 个人所写数字放在测试集中。因此，我们有了两个子集，每个子集有大约 30000 张图像。新的训练集从 SD-3 中的 #0 开始提取图像 30000 张图像，与之前 SD-1 中一个子集的 30000 张图像合并，组成一个 60000 张的训练集。与其相似，新的测试集是由 SD-3 中的 #35000 开始提取图像，与之前 SD-1 中另一个子集的图像合并，组成一个大的测试图像。但是在这里只需要测试集中的 10000 张测试图像（从 SD-1 中的 5000 张图像和从 SD-3 中的 5000 张图像）。当然，60000 张的训练图像也是完全可用的。

2.2 MNIST 数据库格式

数据存储在一个用来存放向量和多维矩阵的文件格式中。MNIST 数据库主要包含四个文件：train-images-idx3-ubyte.gz, train-labels-idx1-ubyte.gz, t10k-images-idx3-ubyte.gz, t10k-labels-idx1-ubyte.gz。在 Caffe 中，用 get_mnist.sh 自动获取的数据集就是上述形式。训练集包含了 60000 张图像，测试集包含了 10000 张图像。

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.			

(a)

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

(b)

Figure 1: MNIST 数据库训练集内容

在 MNIST 数据库中，图像数据以上述形式存储。如果要查看图像，需要经过程序的变换，转换成可打开与显示的图像格式。如图 Figure ??。

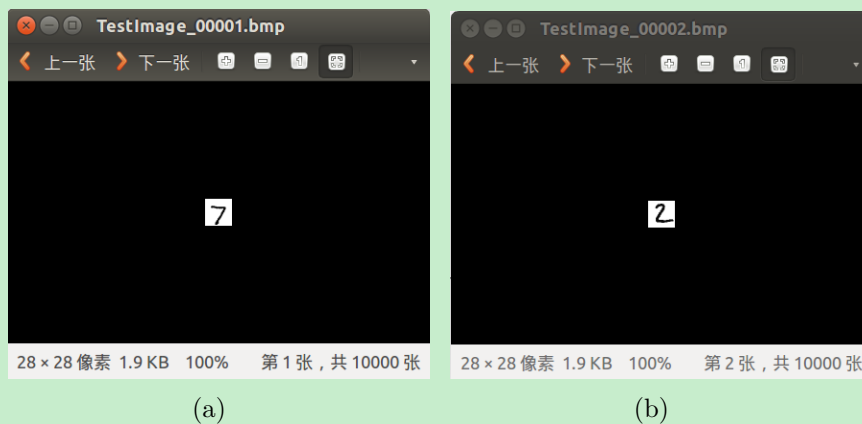


Figure 2: One image in MNIST