https://www.cnblogs.com/denny402/p/5070928.html

[**Caffe学习系列(2)：数据层及参数**](https://www.cnblogs.com/denny402/p/5070928.html)

要运行caffe，需要先创建一个模型（model)，如比较常用的Lenet,Alex等， 而一个模型由多个屋（layer）构成，每一屋又由许多参数组成。所有的参数都定义在caffe.proto这个文件中。要熟练使用caffe，最重要的就是学会配置文件（prototxt）的编写。

层有很多种类型，比如Data,Convolution,Pooling等，层之间的数据流动是以Blobs的方式进行。

今天我们就先介绍一下数据层.

数据层是每个模型的最底层，是模型的入口，不仅提供数据的输入，也提供数据从Blobs转换成别的格式进行保存输出。通常数据的预处理（如减去均值, 放大缩小, 裁剪和镜像等），也在这一层设置参数实现。

数据来源可以来自高效的数据库（如LevelDB和LMDB），也可以直接来自于内存。如果不是很注重效率的话，数据也可来自磁盘的hdf5文件和图片格式文件。

所有的数据层的都具有的公用参数：先看示例

[复制代码](javascript:void(0);)

layer {

name: "cifar"

type: "Data"

top: "data"

top: "label"

include {

phase: TRAIN

}

transform\_param {

mean\_file: "examples/cifar10/mean.binaryproto"

}

data\_param {

source: "examples/cifar10/cifar10\_train\_lmdb"

batch\_size: 100

backend: LMDB

}

}

[复制代码](javascript:void(0);)

***问题：可以有两个top，输出么？***

**name**: 表示该层的名称，可随意取

**type**: 层类型，如果是Data，表示数据来源于LevelDB或LMDB。根据数据的来源不同，数据层的类型也不同（后面会详细阐述）。一般在练习的时候，我们都是采 用的LevelDB或LMDB数据，因此层类型设置为Data。

**top或bottom**: 每一层用bottom来输入数据，用top来输出数据。如果只有top没有bottom，则此层只有输出，没有输入。反之亦然。如果有多个 top或多个bottom，表示有多个blobs数据的输入和输出。

**data 与 label**: 在数据层中，至少有一个命名为data的top。如果有第二个top，一般命名为label。 这种(data,label)配对是分类模型所必需的。

**include**: 一般**训练**的时候和**测试**的时候，模型的层是不一样的。该层（layer）是属于训练阶段的层，还是属于测试阶段的层，需要用include来指定。如果没有include参数，则表示该层既在训练模型中，又在测试模型中。

**Transformations**: 数据的预处理，可以将数据变换到定义的范围内。如设置scale为0.00390625，实际上就是1/255, 即将输入数据由0-255归一化到0-1之间

其它的数据预处理也在这个地方设置：

[复制代码](javascript:void(0);)

transform\_param {

scale: 0.00390625

mean\_file\_size: "examples/cifar10/mean.binaryproto"

# 用一个配置文件来进行均值操作

mirror: 1 # 1表示开启镜像，0表示关闭，也可用ture和false来表示

# 剪裁一个 227\*227的图块，在训练阶段随机剪裁，在测试阶段从中间裁剪

crop\_size: 227

}

[复制代码](javascript:void(0);)

后面的**data\_param**部分，就是根据数据的来源不同，来进行不同的设置。

**1、数据来自于数据库（如LevelDB和LMDB）**

  层类型（layer type）:Data

**必须设置的参数：**

  source: 包含数据库的目录名称，如examples/mnist/mnist\_train\_lmdb

  batch\_size: 每次处理的数据个数，如64

可选的参数：

  rand\_skip: 在开始的时候，路过某个数据的输入。通常对异步的SGD很有用。

  backend: 选择是采用LevelDB还是LMDB, 默认是LevelDB.

示例：

[复制代码](javascript:void(0);)

layer {

name: "mnist"

type: "Data"

top: "data"

top: "label"

include {

phase: TRAIN

}

transform\_param {

scale: 0.00390625

}

data\_param {

source: "examples/mnist/mnist\_train\_lmdb"

batch\_size: 64

backend: LMDB

}

}

[复制代码](javascript:void(0);)

**2、数据来自于内存**

层类型：MemoryData

必须设置的参数：

 batch\_size：每一次处理的数据个数，比如2

 channels：通道数

  height：高度

  width: 宽度

示例：

[复制代码](javascript:void(0);)

layer {

top: "data"

top: "label"

name: "memory\_data"

type: "MemoryData"

memory\_data\_param{

batch\_size: 2

height: 100

width: 100

channels: 1

}

transform\_param {

scale: 0.0078125

mean\_file: "mean.proto"

mirror: false

}

}

[复制代码](javascript:void(0);)

**3、数据来自于HDF5**

层类型：HDF5Data

必须设置的参数：

source: 读取的文件名称

batch\_size: 每一次处理的数据个数

示例：

[复制代码](javascript:void(0);)

layer {

name: "data"

type: "HDF5Data"

top: "data"

top: "label"

hdf5\_data\_param {

source: "examples/hdf5\_classification/data/train.txt"

batch\_size: 10

}

}

[复制代码](javascript:void(0);)

**4、数据来自于图片**

层类型：ImageData

必须设置的参数：

  source: 一个文本文件的名字，每一行给定一个图片文件的名称和标签（label)

  batch\_size: 每一次处理的数据个数，即图片数

可选参数：

  rand\_skip: 在开始的时候，路过某个数据的输入。通常对异步的SGD很有用。

  shuffle: 随机打乱顺序，默认值为false

  new\_height,new\_width: 如果设置，则将图片进行resize

 示例：

[复制代码](javascript:void(0);)

layer {

name: "data"

type: "ImageData"

top: "data"

top: "label"

transform\_param {

mirror: false

crop\_size: 227

mean\_file: "data/ilsvrc12/imagenet\_mean.binaryproto"

}

image\_data\_param {

source: "examples/\_temp/file\_list.txt"

batch\_size: 50

new\_height: 256

new\_width: 256

}

}

[复制代码](javascript:void(0);)

**5、数据来源于Windows**

层类型：WindowData

必须设置的参数：

  source: 一个文本文件的名字

  batch\_size: 每一次处理的数据个数，即图片数

示例：

[复制代码](javascript:void(0);)

layer {

name: "data"

type: "WindowData"

top: "data"

top: "label"

include {

phase: TRAIN

}

transform\_param {

mirror: true

crop\_size: 227

mean\_file: "data/ilsvrc12/imagenet\_mean.binaryproto"

}

window\_data\_param {

source: "examples/finetune\_pascal\_detection/window\_file\_2007\_trainval.txt"

batch\_size: 128

fg\_threshold: 0.5

bg\_threshold: 0.5

fg\_fraction: 0.25

context\_pad: 16

crop\_mode: "warp"

}

}

[复制代码](javascript:void(0);)