**三 Caffe训练**

**3.1 数据转换工具**

为了解决什么问题?

raw数据是jpg,jpeg,bmp等格式, 尺寸不同, Caffe需要的是LMDB或LiveDB格式. 需要将raw格式转成LMDB,LiveDB格式.

a) convert\_imageset命令(由对应文件编译).

b) 完成raw数据对LMDB,LiveDB的转换.

c) 可选flag有

    i) --gray, 是否以灰度值打开图片

    ii) --shuffle, 是否对raw图片洗牌.

    iii) --backend, 输出的db格式(从LMDB和LiveDB选择,默认LiveDB).

    iv) --resize\_widht/resize\_height, 对图像做resize.

生成图像列表

a) 将路径下(example/images/\*.jpg)写进txt中.

b) txt中每行是(形成,摆成): " 文件名  label值" 的格式.

c) label值从0开始.

**3.2 特征提取工具**

a) extract\_features工具(命令)

b) 基于caffe.model

c) 输出Cafffe model对应的layer

d) 输出对应layer文件的格式(保存格式), LMDB或LiveDB.

e) 缺点是: 这些输出的特征使用起来不是很方便.可以用memorydata来做特征提取.

**3.3 Caffe训练需要的几个部件**

proto文件规定了网络结构.(每层的结构).

a) 前面3.1提到的convert\_imageset是为了准备输入数据.

b) proto文件描述Caffe网络结构(当前层结构)

c) type表示当前层格式(Convolution为卷积层).

d) bottom,该层的输入来源

e) top, 该层的输出去向(输出给哪一层?).

f) convolution\_param, 是可选参数. 不同层参数不一样.

Solver配置文件

a) solver.prototxt是如何训练的文件.

    i) 规定好 学习率策略, 优化策略, 迭代次数等.

b) lr\_policy是学习率下降(decay)策略.

   i) fixed 学习率不下降(不变)

   ii) step, 和step相关的下降(非线性)

   iii) exp, inv, multistep, poly, sigmoid等学习率下降策略(delay policy).

训练命令

a) 采用build/tools/caffe作为训练工具

b) 输入trian, 表示训练阶段

c) 输入solver.prototxt文件, 描述需要如何训练网络.

d) 输入gpu选项, 表示采用的gpu

e) 输入weight选项, 描述是否采用预处理模型做fine-tune.

训练日志(log)

a) 设置GLOG\_logtostderr和GLOG\_log\_dir来使用log.

b) 将上述环境变量输入给caffe工具(命令,程序).

c) 采用extra/parse\_log.py来解析保存的log.

d) 采用extra/plot\_training\_log.py来绘制.

    i) 输入一个整数(id) 描述需要(目标查看)的图.

    ii) 输出一个绘制图, 把这个图保存成输出规定的名字.

    iii) 基于matplotlib实现.

e)