

框架使用 (http://www.caffecn.cn/?/topic/%E6%A1%86%E6%9E%B6%E4%BD%BF%E7%94%A8)

caffe的总体流程是怎样的？

关注 | 19 ...

从leveldb的一张图片，到一个blob，到一个层，到一次迭代，到整个训练完。这个总体的思路谁能够详细的讲一下。代码中太多blob，不知道他们的关系，这些blob就不知道干嘛用的。？谢谢

2016-01-21 添加评论 邀请 1

感谢 分享 举报

关注的人 (http://www.caffecn.cn/?/question/123?uid=focus)

时间 ^ (http://www.caffecn.cn/?/question/123?sort_key=add_time&sort=ASC)

票数 (http://www.caffecn.cn/?/question/123&sort_key=agree_count&sort=DESC)



ruirui_ICT (http://www.caffecn.cn/?/people/ruirui_ICT) - 听大大们说，要“左手代码，右手公式”，喵~

赞同来自: 海之蓝 (http://www.caffecn.cn/?/people/%E6%B5%B7%E4%B9%8B%E8%93%9D)、李扬

(http://www.caffecn.cn/?/people/ruirui_ICT)、ZhangXu

(http://www.caffecn.cn/?/people/ZhangXu)、attitude (http://www.caffecn.cn/?/people/attitude)、孙琳钧

(http://www.caffecn.cn/?/people/sunlinjun) 更多 »

我很想详细讲一下的，然而自己才疏学浅，学艺不精，只能大概讲讲，就当抛砖引玉了(○v○)

很多人建议caffe从四个层次来理解：Blob、Layer、Net、Solver，和题主的问题还挺match的

1.Blob

Blob是caffe基本的数据结构，用四维矩阵 $\text{Batch} \times \text{Channel} \times \text{Height} \times \text{Weight}$ 表示，存储了网络的神经元激活值和网络参数，以及相应的梯度（激活值的残差和dW、db）。其中包含有cpu_data、gpu_data、cpu_diff、gpu_diff、mutable_cpu_data、mutable_gpu_data、mutable_cpu_diff、mutable_gpu_diff这一堆很像的东西，分别表示存储在CPU和GPU上的数据（印象中二者的值好像是会自动同步成一致的），其中带data的里面存储的是激活值和W、b，diff中存储的是残差和dW、db，另外带mutable和不带mutable的一对指针所指的位置是相同的，只是不带mutable的只读，而带mutable的可写。

2.Layer

Layer代表了神经网络中各种各样的层，组合成一个网络。一般一个图像或样本会从数据层中读进来，然后一层一层的往后传。除了数据层比较特殊之外，其余大部分层都包含4个函数：

LayerSetUp、Reshape、Forward、Backward。其中LayerSetUp用于初始化层，开辟空间，填充初始值什么的。Reshape是对输入值进行维度变换，比如pooling接全连接层的时候要先拉成一个向量再计算。Forward是前向传播，Backward是后向传播。当然对于我这种喜欢偷懒的童鞋一般学习的时候最喜欢看各种层的Backward函数了，最好是对着公式边推导边看，可以有更直观的理解。

那么数据是如何在层之间传递的呢？每一层都会有一个（或多个）Bottom和top，分别存储输入和输出，比如bottom[0]->cpu_data()存输入的神经元激活值，换成top存输出的，换成cpu_diff()存

的是激活值的残差，换成gpu是存在GPU上的数据，再带上mutable就可写了，这些是神经元激活值相关的，如果这个层前后有多个输入输出层，就会有bottom[1]，比如accuracy_layer就有两个输入，fc8和label。而每层的参数会存在this->blobs_里，一般this->blobs_[0]存W，this->blobs_[1]存b，this->blobs_[0]->cpu_data()存的是W的值，this->blobs_[0]->cpu_diff()存的梯度dW，b和db也类似，然后换成gpu是存在GPU上的数据，再带上mutable就可写了。。（各种变量好多好晕，但愿我说清楚了。。）

哦对了，凡是能在GPU上运算的层都会有名字相同的cpp和cu两个文件，cu文件中运算时基本都调用了cuda核函数，可以在math_function.cu中查看。

3.Net

Net就是把各种层按train_val.prototxt的定义堆叠在一起，首先进行每个层的初始化，然后不断进行Update，每更新一次就进行一次整体的前向传播和反向传播，然后把每层计算得到的梯度计算进去，完成一次更新，这里注意每层在Backward中只是计算dW和db，而W和b的更新是在Net的Update里最后一起更新的。而且在caffe里训练模型的时候一般会有两个Net，一个train一个test。刚开始训练网络时前面的一大堆输出，网络的结构什么的也都是这里输出的。

4.Solver

Solver是按solver.prototxt的参数定义对Net进行训练，首先会初始化一个TrainNet和一个TestNet，然后其中的Step函数会对网络不断进行迭代，主要就是两个步骤反复迭代：①不断利用ComputeUpdateValue计算迭代相关参数，比如计算learning rate，把weight decay加上什么的，②调用Net的Update函数对整个网络进行更新。迭代中的一大堆输出也是在这里输出的，比如当前的loss和learning rate什么的。

综上，为了把整个过程串起来，可以从tools/caffe这个我们最常用的函数入手，训练一个网络然后跟着数据的流动方向看看一个网络是怎么更新的，然后找自己比较感兴趣的地方细看。我本人主要看各种Layer的实现比较多，对其余的部分理解也有限，希望可以一起讨论~

（针对最近大大们头疼的侵权问题，在这里弱弱的加一句：本内容为本人原创，仅授权给CaffeCN使用，如需转载须注明转载来源。（ㄟ_ㄟ））

👍 27 🗨 1 ❌ 没有帮助 ☆ 收藏 ❤ 感谢 ➦ 分享

2016-01-21



那年的融 (<http://www.caffecn.cn/?/people/%E9%82%A3%E5%B9%B4%E7%9A%84%E9%BC%AC>)

写的很好，新手可以了解一个轮廓

(<http://www.caffecn.cn/?/people/%E9%82%A3%E5%B9%B4%E7%9A%84%E9%BC%AC>)

👍 0 🗨 0 ❌ 没有帮助 ☆ 收藏 ❤ 感谢 ➦ 分享

2016-08-28



爱在路上 (<http://www.caffecn.cn/?/people/%E7%88%B1%E5%9C%A8%E8%B7%AF%E4%B8%8A>) - 想成为菜鸟中的战斗机

(<http://www.caffecn.cn/?/people/%E7%88%B1%E5%9C%A8%E8%B7%AF%E4%B8%8A>) 关键是构建网络和调参，这方面的资料基本都没有，全靠自己领悟。

👍 0 🗨 0 ❌ 没有帮助 ☆ 收藏 ❤ 感谢 ➦ 分享

2016-08-30



DonerKiller (<http://www.caffecn.cn/?/integral/rule/>) ☒ 关注问题 ☐ 匿名回复