

Caffe 单独测试添加的layer

转载请注明出处，楼燚(yì)航的blog，<http://home.cnblogs.com/louyihang-loves-baiyan/>

一般我们在使用Caffe的时候，可能需要根据自己的任务需求添加一些自定义的Loss，当然这个Loss可以根据自己的需要来添加。比如在这里我添加的是triplet_loss为例

- 首先需要在caffe.proto中添加相应的ID号和MessageParameter
- 建立相应的头文件.hpp和实现文件.cpp
- 如果是某一些特殊的layer，可能对CUDA有要求，不过对于Loss这种层直接用CPP来实现是不太会影响模型训练的速度
- caffe的版本变得比较快，大家在参照网上的一些添加新的层时要注意一些已经改了的接口，某些layer的合并，当时我在添加layer的时候就看到新的一版caffe中，所有的layer都归并到了vision_layer，Loss_layer,common_layer，跟之前网上看到的caffe目录格式已经变了。因此需要灵活变通一下。
- 最后添加的layer 最好是在Caffe\src\test目录下再新建一个test文件，确保网络可以test通过，因为一般代码偶尔还是会带点逻辑问题的，到最后训练的时候出问题得不偿失

在这里如果我们要添加一个新的loss，建立了相应的头文件和实现文件，在书写测试文件。

如果直接运行

```
make all
make test
make runtest
```

虽然make 命令会根据编译文件的时间戳来跳过大部分文件，但是make runtest花的时间会比较长，因为他把所有的test文件都跑了一遍，当我们自己需要添加文件时，并不需要runtest。

这个时候我们需要稍微修改一下makefile，我觉得虽然我们现在不是手写makefile，但是基本的一些makefile的语法规则还是要了解一下的。

根据make命令的格式

```
make [target]
```

我们在make 时执行的就是后面的target，因此打开makefile 文件

下图是我直接make runtest时的报错，一般都不会那么顺利啦。

```
./include/caffe/test/test_gradient_check_util.hpp:174: Failure
The difference between computed_gradient and estimated_gradient is 0.046313442270960702, which exceeds threshold, where
computed_gradient evaluates to -0.026335094624664636,
estimated_gradient evaluates to -0.072648536895625337, and
threshold_ * scale evaluates to 0.01.
debug: (top_id, top_data_id, blob_id, feat_id)=0,0,0,466; feat = -0.7415486709214747; objective+ = 1.0345630058
objective- = 1.0360159765740955
./include/caffe/test/test_gradient_check_util.hpp:174: Failure
The difference between computed_gradient and estimated_gradient is 0.060535164661547201, which exceeds threshold, where
computed_gradient evaluates to -0.034421956376172604,
estimated_gradient evaluates to -0.094957121037719805, and
threshold_ * scale evaluates to 0.01.
debug: (top_id, top_data_id, blob_id, feat_id)=0,0,0,467; feat = -0.68213750468567014; objective+ = 1.034339919
objective- = 1.0362390624155164
[ FAILED ] TripletLossLayerTest/3.TestGradient, where TypeParam = caffe::GPUDevice<double> (225 ms)
[-----] 2 tests from TripletLossLayerTest/3 (230 ms total)

[-----] Global test environment tear-down
[=====] 8 tests from 4 test cases ran. (1108 ms total)
[ PASSED ] 0 tests.
[ FAILED ] 8 tests, listed below:
[ FAILED ] TripletLossLayerTest/0.TestGradient, where TypeParam = caffe::CPUDevice<float>
[ FAILED ] TripletLossLayerTest/0.TestForward, where TypeParam = caffe::CPUDevice<float>
[ FAILED ] TripletLossLayerTest/1.TestForward, where TypeParam = caffe::CPUDevice<double>
[ FAILED ] TripletLossLayerTest/1.TestGradient, where TypeParam = caffe::CPUDevice<double>
[ FAILED ] TripletLossLayerTest/2.TestGradient, where TypeParam = caffe::GPUDevice<float>
[ FAILED ] TripletLossLayerTest/2.TestForward, where TypeParam = caffe::GPUDevice<float>
[ FAILED ] TripletLossLayerTest/3.TestForward, where TypeParam = caffe::GPUDevice<double>
[ FAILED ] TripletLossLayerTest/3.TestGradient, where TypeParam = caffe::GPUDevice<double>
```

公告

昵称：楼燚航的blog

园龄：1年5个月

粉丝：60

关注：1

+加关注

2016年4月						
日	一	二	三	四	五	六
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

搜索

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

更多链接

随笔档案

2016年3月(2)

2016年2月(2)

2016年1月(7)

2015年12月(2)

2015年11月(5)

2015年10月(5)

2015年9月(1)

2015年8月(2)

2015年7月(1)

2015年6月(1)

2015年5月(1)

2015年4月(3)

2014年12月(1)

最新评论

1. Re:opencv 3.0 DPM Cascade 检测（附带TBB和openMP加速）

好了，问题解决了，我用的opencv版本是2412，所以导致那么慢，换成310就没有这问题了

因此为了不不必要的每次make runtest，我们看一下makefile内部这里要说一下makefile 的格式

```
target: $(prerequisite)
[tab] [command]
```

首先我们需要明白make 做的事 构建，对应的是我们IDE中的build，他并不是compile，是准备各种资源配置，为编译做准备，具体的编译还是交给了GCC这样的编译器。

在上面的格式中，首行中的target 目标我们的目标，在我们 make runtest，则会跳转到这个target开始执行，冒号后面跟着的是前置条件，即这个 target 要满足的依赖条件第二行是一个 tab 键，

当我们在 make runtest 之前，我们已经执行了 make test，即会生成test相关的可执行文件，因此在这里我们输入我们相应的 .testbin 文件路径，这个应该是在 .build_release/test/ 目录下，因此我们可以添加一个我们自己的 target 如下

```
1 fi
2
3 runtest: $(TEST_ALL_BIN)
4 $(TOOL_BUILD_DIR)/caffe
5 #$(TEST_ALL_BIN) $(TEST_GPUID) --gtest_shuffle $(TEST_FILTER)
6 echo $(TEST_ALL_BIN)
7 #added by yihanglou
8 TEST_TRIPLET = .build_release/test/test_triplet_loss_layer.testbin
9 runtest_triplet: $(TEST_ALL_BIN)
10 $(TOOL_BUILD_DIR)/caffe
11 echo $(TEST_TRIPLET)
12 $(TEST_TRIPLET) $(TEST_GPUID) --gtest_shuffle $(TEST_FILTER)
```

这样我们在跑的时候每次 make all， make test 之后直接运行 make runtest_triplet，即直接运行我们需要的test文件，方便我们进一步做测试和修改

好文要顶 关注我 收藏该文



楼继航的blog

关注 - 1

粉丝 - 60

+加关注

« 上一篇: [Caffe源码解析7: Pooling_Layer](#)
» 下一篇: [Caffe Python MemoryDataLayer Segmentation Fault](#)

posted @ 2016-03-10 10:33 楼继航的blog 阅读(296) 评论(0) 编辑 收藏

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

- 【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库
【推荐】融云即时通讯云—豆果美食、Faceu等亿级APP都在用
【推荐】百度开放云—三月超低价促销



最新IT新闻:

- LG确认：开发Friends模块设备需要取得授权并协同开发
 - 触角越来越广 华为能成为中国的三星吗？
 - 微软认知服务：人工智能的技术拼图
 - 知己知彼，百战不殆：一篇文章看懂隐藏在阿尔法狗背后的深度学习
 - 女性玩家崛起 研发女性游戏要注意什么
- » 更多新闻...



最新知识库文章:

- 我是一个线程
- 为什么未来是全栈工程师的世界？
- 程序bug导致了天大的损失，要枪毙程序员吗？

--gaosi123

2. Re:opencv 3.0 DPM Cascade 检测（附带TBB和openMP加速）

BTW, 我电脑是i7 4790k + 16GB内存，所以硬件设备应该不会是限制。不知道问题出在哪里

--gaosi123

3. Re:opencv 3.0 DPM Cascade 检测（附带TBB和openMP加速）

如果可以，欢迎留个email

--gaosi123

4. Re:opencv 3.0 DPM Cascade 检测（附带TBB和openMP加速）

你好，我也是直接把DPM代码拷贝到工程里，但是想你这样直接拷进去不会报错吗？我直接拷贝进去按照你的来，报错信息如下：Error 4 error C2039: 'dpm': is not a memb.....

--gaosi123

5. Re:Fast RCNN 训练自己数据集 (2修改数据读取接口)

@楼继航的blog楼主你好！我在EdgeBoxes提取OP的时候也是直接用的默认参数，并且将坐标[x y w h]变成了左上右下的形式，但是发现检测车的时候效果并没有Selective Search好，.....

—JustJay

阅读排行榜

- 1 0
- (请您对文章做出评价)
1. Fast RCNN 训练自己数据集 (2修改数据读取接口)(3946)
 2. Fast RCNN 训练自己数据集 (1编译配置)(3166)
 3. Fast RCNN 训练自己的数据集（3训练和检测）(3097)
 4. RCNN (Regions with CNN) 目标物检测 Fast RCNN的基础(2096)
 5. Hog SVM 车辆 行人检测(979)

评论排行榜

1. Fast RCNN 训练自己数据集 (2修改数据读取接口)(22)
2. Fast RCNN 训练自己数据集 (1编译配置)(21)
3. Fast RCNN 训练自己的数据集（3训练和检测）(5)
4. opencv 3.0 DPM Cascade 检测（附带TBB和openMP加速）(4)
5. DPM检测模型 训练自己的数据集 读取接口修改(2)

推荐排行榜

1. Fast RCNN 训练自己数据集 (2修改数据读取接口)(5)
2. 车脸检测 Adaboost 检测过程(3)
3. Caffe 抽取CNN网络特征 Python(2)
4. DPM检测模型 训练自己的数据集 读取接口修改(2)
5. RCNN (Regions with CNN) 目标物检测 Fast RCNN的基础(2)

- [如何运维千台以上游戏云服务器](#)
- [架构漫谈（一）：什么是架构？](#)
- » [更多知识库文章...](#)