

[首页](#)[Web开发](#)[Windows开发](#)[编程语言](#)[数据库技术](#)[移动平台](#)[系统服务](#)[微信](#)[布布扣](#)[其他](#)[数据分析](#)[首页](#) > [其他](#) > [详细](#)

如何在caffe中添加新类型的layer

时间 : 2015-06-16 18:30:04 阅读 : 25 评论 : 0 收藏 : 0 [\[点我收藏+\]](#)标签 : [class](#) [log](#) [com](#) [代码](#) [http](#) [src](#) [si](#) [it](#) [文件](#)

如何在caffe中添加新类型的layer

参考 : <https://github.com/BVLC/caffe/issues/684>

Add a class declaration for your layer to the appropriate one of `common_layers.hpp`, `data_layers.hpp`, `loss_layers.hpp`, `neuron_layers.hpp`, or `vision_layers.hpp`. Include an inline implementation of type and the `*Blobs()` methods to specify blob number requirements. Omit the `*_gpu` declarations if you 'll only be implementing CPU code.

Implement your layer in `layers/your_layer.cpp`.

SetUp for initialization: reading parameters, allocating buffers, etc.

Forward_cpu for the function your layer computes

Backward_cpu for its gradient

(Optional) Implement the GPU versions `Forward_gpu` and `Backward_gpu` in `layers/your_layer.cu`.

Add your layer to `proto/caffe.proto`, updating the next available ID. Also declare parameters, if needed, in this file.

Make your layer createable by adding it to `layer_factory.cpp`.

Write tests in `test/test_your_layer.cpp`. Use `test/test_gradient_check_util.hpp` to check that your Forward and Backward implementations are in numerical agreement.

以上是github上某大神的解答，步骤很清晰，具体说一下，比如现在要添加一个vision layer，名字叫Aaa_Layer：

- 1、属于哪个类型的layer，就打开哪个hpp文件，这里就打开`vision_layers.hpp`，然后自己添加该layer的定义，或者直接复制`Convolution_Layer`的相关代码来修改类名和构造函数名都改为`Aaa_Layer`，如果不用GPU，将`*_gpu`的声明都去掉。
- 2、实现自己的layer，编写`Aaa_Layer.cpp`，加入到`src/caffe/layers`，主要实现`Setup`、`Forward_cpu`、`Backward_cpu`。
- 3、如果需要GPU实现，那么在`Aaa_Layer.cu`中实现`Forward_gpu`和`Backward_gpu`。
- 4、修改`src/caffe/proto/caffe.proto`，好到`LayerType`，添加`Aaa`，并更新ID，如果Layer有参数，添加`AaaParameter`类。

- 5、在src/caffe/layer_factory.cpp中添加响应代码。
- 6、在src/caffe/test中写一个test_Aaa_layer.cpp，用include/caffe/test/test_gradient_check_util.hpp来检查前向后向传播是否正确。

如何在caffe中添加新类型的layer

标签 : class log com 代码 http src si it 文件

赞

(0)

踩

(0)

举报

评论

一句话评论 (0)

共0条

登录后才能评论！ 登录

