# 为什么要开始使用就地激活的BatchNorm

就地激活的BatchNorm（In Place ABN）是BatchNorm+激活步骤的内存高效替换。BN+Relu+Conv2d是现代网络体系结构（UNet、LinkNet、ResNet等）基本构件的组成部分。

DR:InPlace ABN可以节省训练深层神经网络模型所需的GPU内存的50%。

就地ABN的所有信用证归Mapillary:。在这篇文章中，我想鼓励您尝试一下，并在培训您的模型时立即获得减少内存占用的好处。

### 什么是就地ABN？

Inplace ABN是一种新的降低训练深度网络所需内存的方法。我不打算深入讨论实现细节（这可能是Medium上的一篇专门文章的主题；我鼓励您阅读详细解释方法的文章）。非常简单地说，该方法提出了一种将从激活和批处理规范层进行反向传播所需的内存量减少到50%的方法。

### 如何使用就地ABN？

首先，您需要安装CUDA 9.0+库。截至撰写本文时，我还没有遇到TF或MXNet的InPlace ABN的实现。如果你找到了，请在下面留言。

作为一等公民，Linux用户只需输入：

如果您拥有所有依赖项，那么这将编译并安装python绑定到CUDA优化，并将CPU实现用于活动python环境的前向和后向例程。不幸的是，Windows用户需要执行额外的步骤来构建和安装这个包，我将在下一章中对此进行描述。

让我们看看如何在U-Net架构中采用这个模块。U-Net的一个基本构造块是所谓的“双卷积”模块，它是Conv2d+BN+Relu重复两次：

与inpregabn相同的块看起来非常相似：

就这样！训练时间基本保持不变（作者报告约2%的开销），准确度不低于经典的BN+激活方法：

在与检查点比较时，我们观察到一致的速度优势有利于我们的方法，实际的百分比差异取决于块的元参数。正如我们所看到的，INPLACE-ABN导致计算时间比标准增加0.8-2%，而检查点几乎是我们开销的两倍。（源：HTTPS://ARXV.OR/PDF/171202616.PDF）

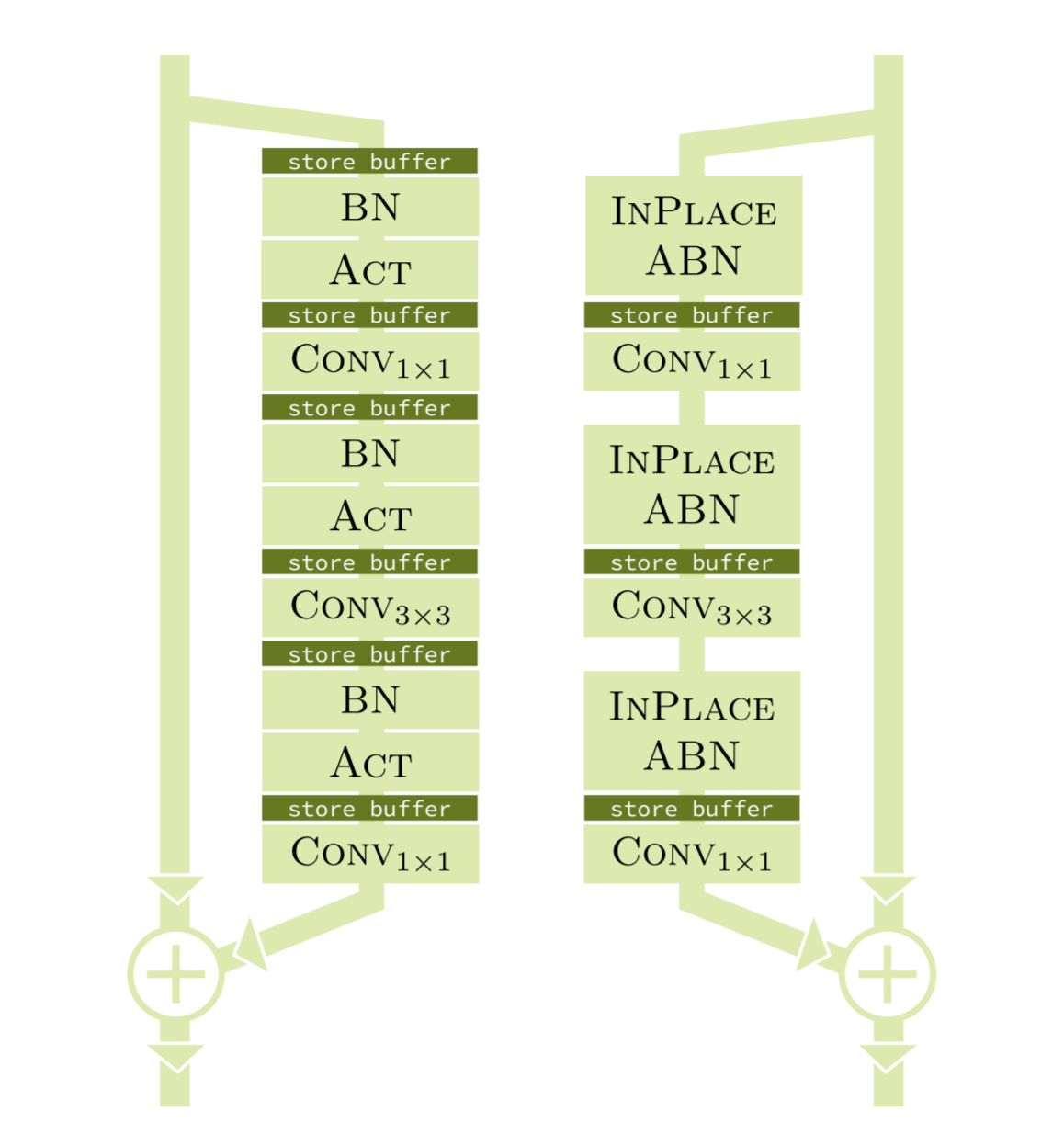
但内存占用量急剧减少：

1080Ti香草U型网（11GB）

* BN+Relu:1024x1024x3的批量大小3
* inpregabn:1024x1024x3的批量4

免费的内存减少了30%。我对U-Net和“U-Net with ABN”都进行了培训，看看他们的表现有没有什么不同：

如您所见，根据验证损失，使用inpregabn的U-Net性能更好一些。两种模型的收敛趋势非常相似，表明inprepabn不引入不稳定或梯度爆炸，可以安全地用作经典BatchNorm+ReLU的替换。



Inplace ABN提供多种激活功能供您选择：

* 漏雷卢
* 埃卢
* 无

正如你可能注意到的，没有ReLU支持。这是故意的，因为所提出的方法要求激活函数是可逆的（例如，具有激活值的人可以还原输入信号）。

### 在Windows上构建

Windows从来都不是深度学习框架的首要平台。甚至连PyTorch也在0.4中添加了官方Windows支持。因此，在构建这样的库时遇到陷阱是很常见的。幸运的是，我已经穿过这个雷区，准备分享一个循序渐进的指南：

#### 先决条件：

* 视觉工作室2017
* 英伟达9.2
* 喷灯0.4
* 您选择的文本编辑器

不幸的是，我无法用VS 2015和CUDA 9.1编译这个库。所以VS 2017和CUDA 9.2是这里的硬性要求。在开始构建之前，我们必须修补CUDA头；）

打开host\_config.h，可以在“C:FilesGPU Computing Toolkit.2\include”中找到，并将第131行更改为如下所示：

这一变化的原因是，对于CUDA作者来说，Microsoft C编译器的版本变化太快。特别是，\_MSC\_VER宏的版本对于CUDA是未知的。修复后，我们可以构建inpregabn：

#### 台阶

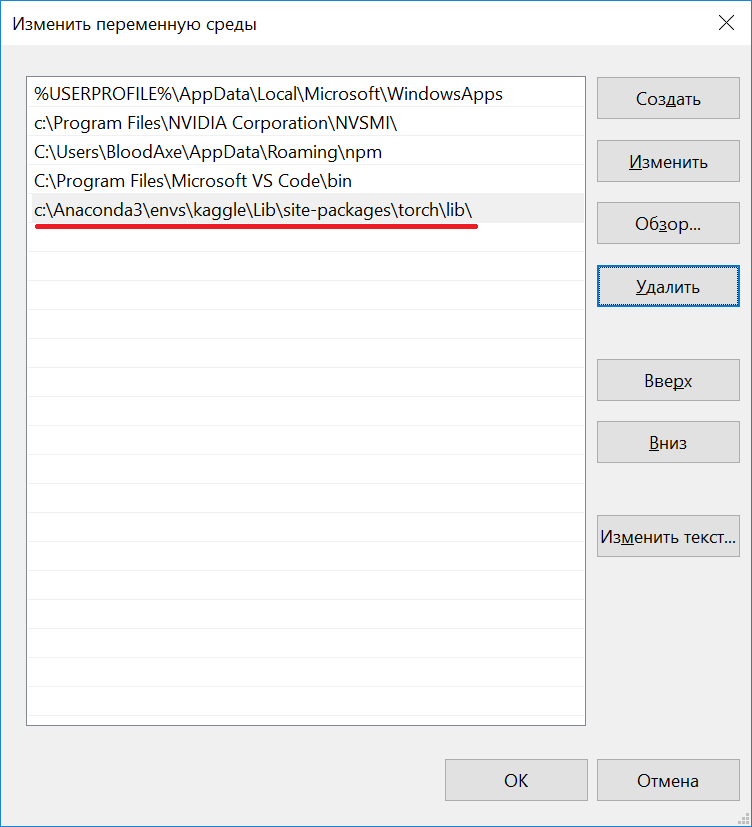
1. 克隆
2. 打开“VS 2017的x64本机工具命令提示符”。这将在VS环境中启动一个新的cmd解释器。
3. 导航到在其中克隆inplace\_abn repository的目录
4. python setup.py安装

最后一步是最关键的一步，它可能由于许多原因（错误的VS版本、错误的CUDA版本、错误的Pytorch版本）而失败，因此请仔细阅读错误消息。

我们快到了！如您所知，在Windows上，Python扩展是常规的DLL。我们的inplace\_abn包依赖于Pytorch.DLL和ATen.DLL，因此如果您尝试导入它，它将失败，并显示神秘的“DLL load failed”消息。

1. 将Pytorch.dll的位置添加到%PATH%

幸运的是，这是可以解决的，但有点“老套”的解决方案。如果你知道如何使它更优雅，请在评论中发表你的解决方案。要修复“Dll load failed”，必须修改PATH环境变量并添加Pytorch.Dll和ATen.Dll的位置。对于水蟒，可以在“c:3包”中找到：



就这样！现在应该解析所有运行时依赖项。

### 好吧，接下来呢？

开始黑客攻击！有了这个模块，即使在1050这样的内存不足的GPU上，也可以适应现代架构，或者在1080上有更多的批处理，或者在ABN块中利用高效的GPU间同步。

我想借此机会强调一下Github中已经使用inpregabn的几个项目。第一个是我的个人项目：研究和评估二进制分割问题的网络模型：

1. [https://github.com/BloodAxe/segmentation-networks-benchmark](https://github.com/BloodAxe/segmentation-networks-benchmark/blob/master/lib/models/unet_abn.py)

第二个项目有第二个解决方案的网络定义和权重，其中还使用了inpregabn：

[2. https://github.com/ternaus/TernausNetV2](https://github.com/ternaus/TernausNetV2)

### 结论

由Apple研究的ABN提供了一个BN+激活的内存高效模块，可以用于现有神经网络模型中的替换，并将内存占用减少到50%。

* Paper: <https://arxiv.org/pdf/1712.02616.pdf>
* Official implementation: <https://github.com/mapillary/inplace_abn>

Author would like to thank Open Data Science community (ods.ai) for many valuable discussions and educational help in the growing field of machine/deep learning.