知

首发于
片上神经网络

写文章 登录

给DNN处理器跑个分 - 设计篇

唐杉 · 4 个月前

> 前情提要： 给DNN处理器跑个分 - 指标篇中介绍了哪些指标（Metrics）可以更好的评价不同硬件进行基准测试（Benchmarking）的结果，同时也分析了直接使用几个经典DCNN网

> 络作为“测试程序”的一些弊端。本篇通过百度的DeepBench项目，和大家讨论一下Synthetic Benchmark设计的问题。

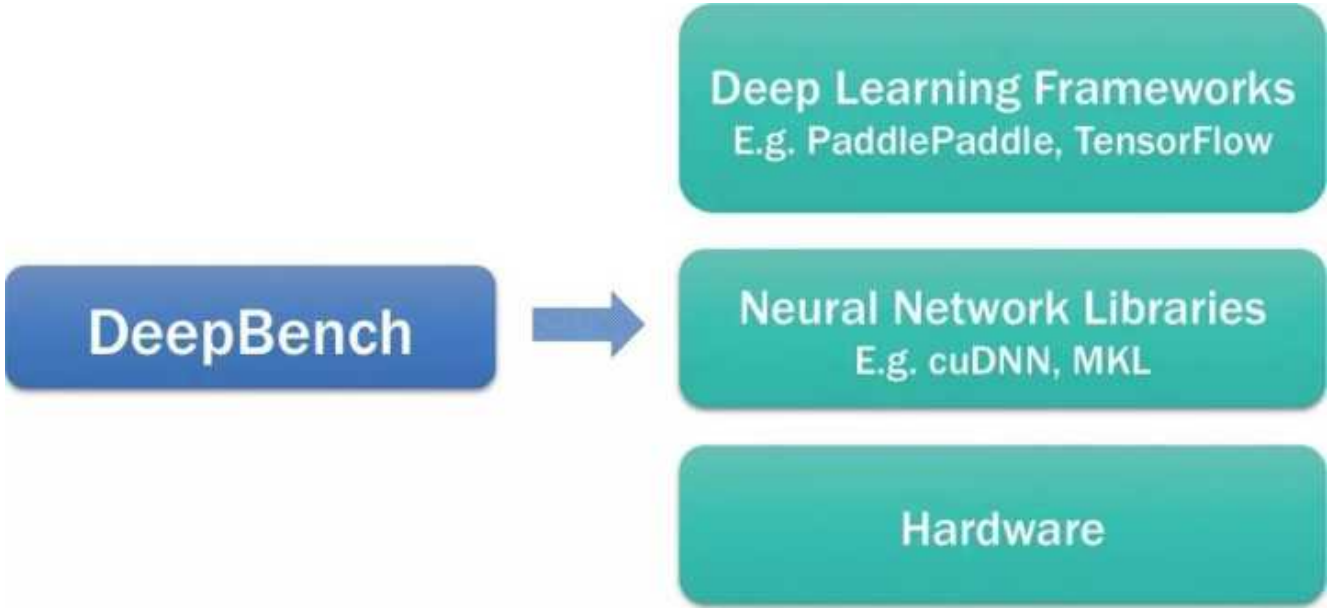
1.

DeepBench项目[1]在百度众多的开源项目中受到的关注相对较少，不过他们对于Deep Learning硬件Benchmarking的讨论和实践，既是非常有意义的尝试，也是我们讨论Benchmarking问题的很好的参考。我们首先看看他们对项目目标的描述：

> *“The primary purpose of DeepBench is to benchmark operations that are important to deep learning on different hardware platforms. Although the fundamental computations behind deep learning are well understood, the way they are used in practice can be surprisingly diverse. For example, a matrix multiplication may be compute-bound, bandwidth-bound, or occupancy-bound, based on the size of the matrices being multiplied and the kernel implementation. Because every deep learning model uses these operations with different parameters, the optimization space for hardware and software targeting deep learning is large and underspecified.” “DeepBench attempts to answer the question, "Which hardware provides the best performance on the basic operations used for deep neural networks?". We specify these operations at a low level, suitable for use in hardware simulators for groups building new processors targeted at deep learning. DeepBench includes operations and workloads that are important to both training and inference.”*

上述文字中的一个关键词就是“basic operation”，和Eyeriss团队直接用经典的DCNN网络作为Benchmark不同[2]，DeepBench尝试使用基本操作而非完整网络模型来作为Benchmark。而另一个差别在于，Eyeriss团队主要工作在于设计Inference专用的硬件加速器，因此他们更关心Inference应用，而DeepBench则同时考虑Training和Inference的硬件评测问题。

下面这幅图是DeepBench在一个Deep Learning生态系统中的定位。



以下是他们对这个定位的说明：

> *DeepBench uses the neural network libraries to benchmark the performance of basic operations on different hardware. It does not work with deep learning frameworks or deep learning models built for applications. We cannot measure the time required to train an entire model using DeepBench. The performance characteristics of models built for different applications are very different from each other. Therefore, we are benchmarking the underlying operations involved in a deep learning model. Benchmarking these operations will help raise awareness amongst hardware vendors and software developers about the bottlenecks in deep learning training and inference.*

可以看出，DeepBench试图设计更具普遍性的Benchmark，因此选择使用底层的操作（underlying operations）来评测在Training和Inference中的瓶颈问题。这实际上也是试图解

决我们在上篇文章中提到的，“使用几个DCNN网络作为Benchmark是否具有普遍性？”这一问题。到这里，我们可以比较清楚的看到DeepBench的基本思路是设计类似Dhrystone这样的“Synthetic Benchmark”思路。

需要指出的是，DeepBench使用了硬件厂家提供的Library，比如Nvidia的cuDNN，Intel的MKL，还有ARM的Compute Library。也就意味着它的“基本操作”是比这些Library封装的“基本操作”（比如矩阵乘法）的粒度要粗的，并且需要Library的支持。如果其它硬件厂商也想使用DeepBench，则也需要提供类似的Library（当然还包括一些porting的工作）。另外，它评估的结果实际是包括了目标硬件加Library的整体性能，而非单纯硬件本身。由于实际的应用者一般也是基于Library进行开发，所以这种方式也是合理的。但是，对于需要定位硬件设计问题或者瓶颈的人来说，使用这样的方法还需要考虑到Library对结果的影响。比如在DeepBench最新发布的时候，ARM的Compute Library还只支持单精度浮点的卷积操作，不支持8比特定点；同时它也没有对RNN对的支持，这些功能限制对于测试效果的评估还是有比较大的影响的。

和Eyeriss提出了很多有特色的“Benchmarking Metrics”不同，DeepBench的测试指标是比较简单的，主要就采用运行时间“Time（msec）”进行性能的评价。所以，我们对DeepBench的讨论主要还是集中于它设计的“测试程序”。

2.

下面我们就具体看看DeepBench设计的用于Training和Inference的Benchmark。

对于Training，DeepBench从不同的应用模型（Source列）中总结出了一些特定的操作（Type of Operations）。其中最基本的操作包括三类：Dense Matrix Multiplication，Convolution，Recurrent Layers。这些选择是比较自然的，基本可以覆盖各种DNN的情况。最后一类All-Reduce是专门针对多GPU并行Training的场景。对于每一种Operation，都有不同的参数。比如Converlution操作，就有如下的参数：W (input - time), H (input), C (channels), N (batch

size), K (number of filters), R (filter width), S (filter height), pad_h pad_w, Vertical Stride, Horizontal Stride, Platform (server or device), Forward, wrt Inputs, wrt Parameters。DeepBench对每种Operation都定义并且实现了多种不同的参数组合，反映各种应用中比较常见的配置（可以看作是一个测试内核Kernel）。如下表所示：

Type of Operations	Source	Num of Kernels
Dense Matrix Multiplication	DeepSpeech	44
	DeepSpeech, output layer	16
	DeepSpeech	18
	Language Modelling	8
	Speaker ID	2
	DeepSpeech	72
Convolution	DeepSpeech	8
	OCR	4
	Face Recognition	5
	Vision	19
	Face Recognition	18
	Speaker ID	8
	Resnet	32
Recurrent Layers - Vanilla	DeepSpeech	12
Recurrent Layers - LSTM	Machine Translation	12
	Language Modelling	7
	Character Language Modelling	3
Recurrent Layers - GRU	DeepSpeech	17
	Speaker ID	2
All-Reduce	General	5
	DeepSpeech	20
Total		332

表中包括的各种Kernel（按照对应的典型应用）的数量，目前的版本总共有332个。从Kernel的分布来看Dense Matrix Multiplication，Convolution这两类数量最多；而从Source来看，最多的模型是DeepSpeech，ResNet。具体的各种case大家可以看网站提供的Excel表格，以及

Kernel的具体实现。针对这些Kernel的直接问题就是，这些Kernel是否是合理的选择，是否有足够的代表性。对此，也欢迎各位读者发表自己的看法。

以下是针对Inference的Kernel的信息：

Type of Operations	Source	Number of Kernels
Dense Matrix Multiplication	DeepSpeech, output layer	8
	DeepSpeech	6
	KWS	1
	Language Modelling	6
	Speaker ID	1
	DeepSpeech, gru affine	15
	DeepSpeech, gru	9
	DeepSpeech, gru low rank	8
	DeepSpeech	20
Sparse Matrix Multiplication	DeepSpeech	24
Convolution	DeepSpeech	6
	OCR	4
	Face Recognition	5
	Vision	26
	KWS	1
	Face Recognition, ID	18
	Speaker ID	16
	Resnet	32
Recurrent Layers - Vanilla	KWS	1
Recurrent Layers - LSTM	Machine Translation	9
	Language Modelling	3
	Character Language Modelling	3
Recurrent Layers - GRU	Deep Speech	51
	Speaker ID	3
Total		276

对于Inference的BenchMark，除了基本操作以外DeepBench还讨论了以下几个问题。

Deployment Platform：分析了DNN模型部署在Server端（我们之前经常说的Cloud/Datacenter）和Device端（Edge/Embedded）的特点，也说明了DeepBench对不同的部署平台做了有针对性的测试。

Inference Batch Size：分析Batch Size对于效率和延时的影响（Batch Size越大，并行处理的可能性阅读，实现效率越高，但处理延时也越大），以及在两者之间如何平衡。从DeepBench测试的结果来看，“*在Server端采用4~5的batch size，而在Device端采用1的batch size是一个比较好的选择*”。

Inference Precision：讨论Inference精度的问题，以及DeepBech对精度选择的原因。

Sparse Operations：分析稀疏性对inference性能的影响。一个有趣的结论是“*基于我们的研究，与其密集模型基线相比，具有90至95％稀疏度的神经网络可以实现相对较好的性能。然而，当前实现的稀疏矩阵乘法主要是对更高的稀疏度（大约99％或更高）而优化的。通过包含稀疏内核（我们在上面的表格中看到的“Sparse Matrix Multiplication” operation），我们希望激励硬件供应商和软件开发人员构建为90％至95％的稀疏性提供更好性能的硬件和库。*”

Measuring Latency：“*许多Inference应用具有实时延迟要求。DeepBench定义的内核可以作为衡量单个操作的最佳延迟的起点。但是，考虑到基础操作不是完整的应用程序，测量全系统延迟不在本版本DeepBench的范围之内。*” 这里反映了使用基本操作作为测试程序的问题：无法统计整个实际应用程序（比如AlexNet）所花费完整时间。这是目前DeepBench面临的一个主要困难。在未来能否设计出更好的基本操作，并基于这些基本操作设计一些组合操作来模拟实际的应用程序，是值得研究的课题。

> 对于最后这几个问题的讨论，也反映出Synthetic Benchmark设计的难度：如何通过“设计”，让不是真实应用的测试程序来准确“模仿”真实的应用，并得到能够反映现实世界情况

> 的结果。从更大范围来看，随着DNN软硬件实现的不断优化，如何对其进行评价，验证和测试将会是我们一直要思考的问题。DeepBench是一个非常有益的尝试，希望未来还能看到更多更好的解决方案。

T.S.

> Reference: [1] “DeepBench”, github.com/baidu-resear... [2] "Benchmarking DNN Processors" , eyeriss.mit.edu/benchma...

推荐阅读

给DNN处理器跑个分 - 指标篇

Deep Learning Hardware - 我的文章

智慧云中的FPGA

自己动手设计专用处理器！

当我们设计一个专用处理器的时候我们在干什么？（指令集）

当我们设计一个专用处理器的时候我们在干什么？（微结构）

深度神经网络的模型·硬件联合优化

追求极限性能的芯片设计方法（一）

追求极限性能的芯片设计方法（二）

追求极限性能的芯片设计方法（三）

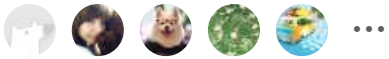
追求极限性能的芯片设计方法（四）

题图来自网络，版权归原作者所有

欢迎关注我的微信公众号:StarryHeavensAbove

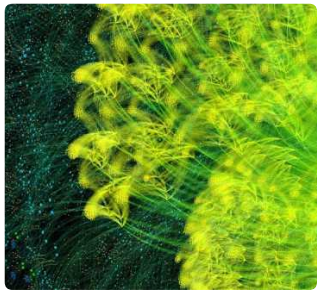
深度学习（Deep Learning） 神经网络

芯片设计



写下你的评论...

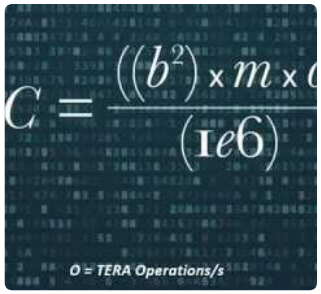
推荐阅读



Training or Inference?

Graphcore’s machine intelligence processors support both training and inference. If y… 查看全文 >

唐杉 · 4 个月前



AI硬件的Computational Capacity详解

之前文章“给DNN处理器跑个分 - 指标篇 - 知乎专栏”，提到Intel/Nervana的Naveen Rao提出的… 查看全文 >

唐杉 · 4 个月前

最高法：2017年夫妻共同债务最新认定标准——赌博、高利贷算不算？

【按语】日前，最高人民法院审判委员会副部级专职委员杜万华法官接受记者采访，就以夫妻一方名义所负债务是否及如何认定为夫妻共同债务问题作出最新解答。记者：近年来妇联系统陆续收到投诉… 查看全文 >

杨锡锋律师 · 23 天前 · 编辑精选

西安 | 盛唐的长安是什么样？不如穿越去看看

听见第一声沉郁的击鼓，而不是滴滴答答的手机闹钟电子音时，你懵了三秒——原来这就穿越到唐… 查看全文 ›

穷游锦囊 · 11 天前 · 编辑精选