深度学习GPU选购指南：哪款显卡配得上我的炼丹炉?

机器学习实验室 2023-02-03 16:16 发表于浙江

转自：新智元

众所周知，在处理深度学习和神经网络任务时，最好使用GPU而不是CPU来处理，因为在神经网络方面，即使是一个比较低端的GPU，性能也会胜过CPU。

深度学习是一个对计算有着大量需求的领域，从一定程度上来说，GPU的选择将从根本上决定深度学习的体验。

但问题来了，如何选购合适的GPU也是件头疼烧脑的事。

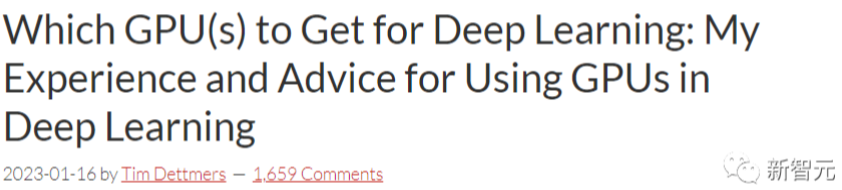
怎么避免踩雷，如何做出性价比高的选择?

曾经拿到过斯坦福、UCL、CMU、NYU、UW 博士 offer、目前在华盛顿大学读博的知名评测博主Tim Dettmers就针对深度学习领域需要怎样的GPU，结合自身经验撰写了万字长文，最后给出了DL领域的推荐GPU。

Tim Dettmers此人的研究方向是表征学习、硬件优化的深度学习，他自己创建的网站在深度学习和计算机硬件领域也是小有名气。

Tim Dettmers此文推荐的GPU全部来自N厂，他显然也认为，搞机器学习，AMD目前还不配拥有姓名。

原文链接小编也贴在下面啦。



原文链接：https://timdettmers.com/2023/01/16/which-gpu-for-deep-learning/#GPU_Deep_Learning_Performance_per_Dollar

RTX40和30系的优缺点

与英伟达图灵架构RTX 20系列相比，新的英伟达安培架构RTX 30系列具有更多优势，如稀疏网络训练和推理。其他功能，如新的数据类型，应更多地被看作是一种易用化功能，因为它们提供了与图灵架构相同的性能提升，但不需要任何额外的编程要求。

Ada RTX 40系列甚至有更多的进步，比如上面介绍的张量内存加速器（TMA）和8位浮点运算（FP8）。与RTX 30相比，RTX 40系列也有类似的电源和温度问题。RTX 40的电源连接器电缆融化的问题可以通过正确连接电源电缆而轻松避免。

稀疏的网络训练

安培允许在密集的速度下进行细粒度结构的自动稀疏矩阵乘法。这是如何做到的？以一个权重矩阵为例，把它切成4个元素的碎片。现在想象这4个元素中的2个元素为零。图1显示了这种情况的样子。

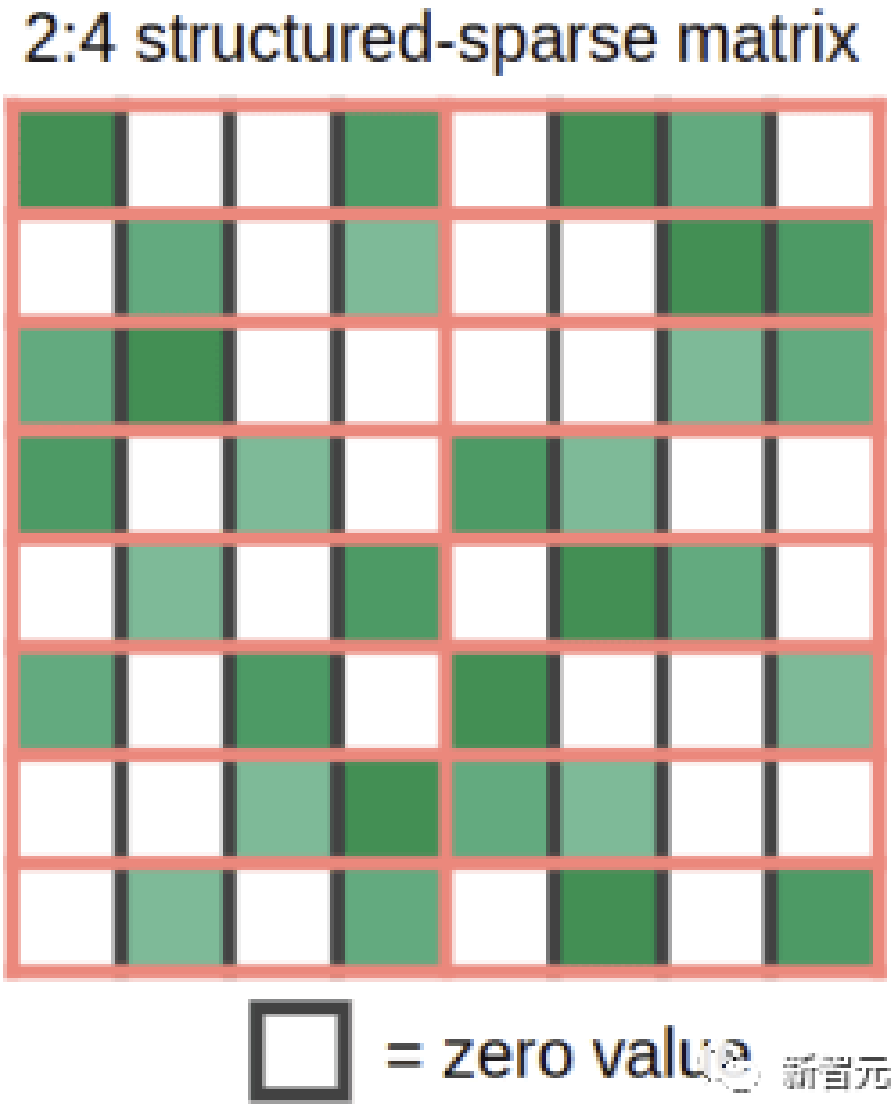


图1：Ampere架构GPU中的稀疏矩阵乘法功能所支持的结构

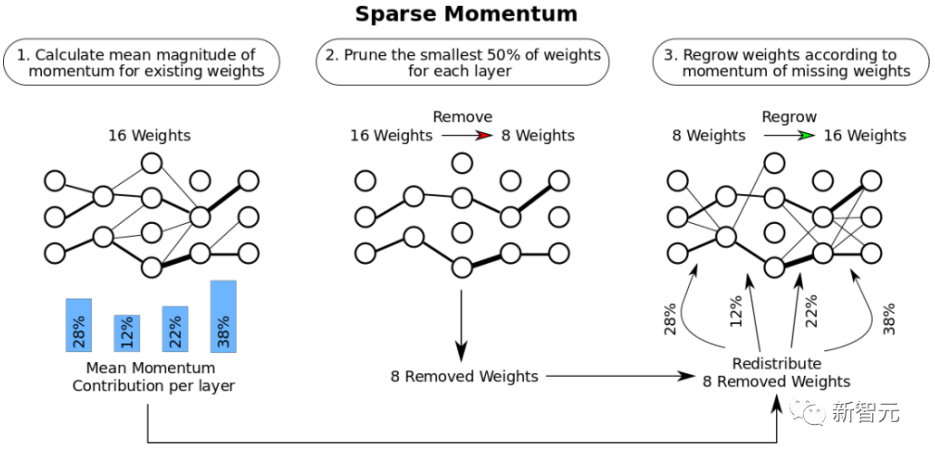
当你将这个稀疏权重矩阵与一些密集输入相乘时，安培的稀疏矩阵张量核心功能会自动将稀疏矩阵压缩为密集表示，其大小为图2所示的一半。

在压缩之后，密集压缩的矩阵瓦片被送入张量核心，张量核心计算的矩阵乘法是通常大小的两倍。这有效地产生了2倍的速度，因为在共享内存的矩阵乘法过程中，带宽要求减半。

图2：在进行矩阵乘法之前，稀疏矩阵被压缩为密集表示。

我在研究中致力于稀疏网络训练，我还写了一篇关于稀疏训练的博文。对我的工作的一个批评是："你减少了网络所需的FLOPS，但并没有产生速度的提升，因为GPU不能进行快速的稀疏矩阵乘法"。

随着Tensor Cores的稀疏矩阵乘法功能的增加，我的算法或其他稀疏训练算法，现在实际上在训练期间提供了高达2倍的速度。



开发的稀疏训练算法有三个阶段：（1）确定每层的重要性。(2）删除最不重要的权重。(3）提升与每层的重要性成比例的新权重。

虽然这一功能仍处于实验阶段，而且训练稀疏网络还不普遍，但在你的GPU上拥有这一功能意味着你已经为稀疏训练的未来做好了准备。

低精度计算

在我的工作中，我之前已经表明，新的数据类型可以提高低精度反向传播期间的稳定性。

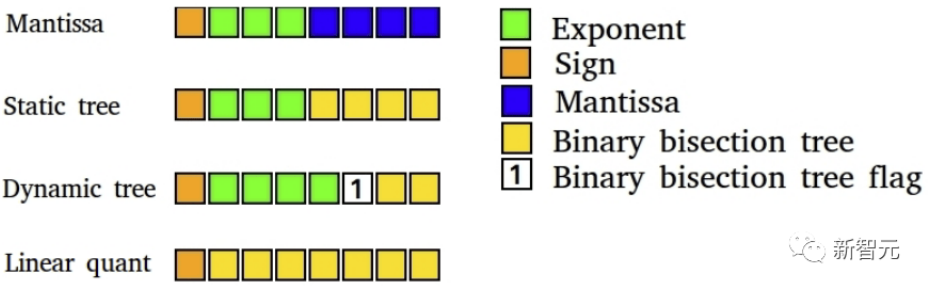


图4：低精度深度学习8位数据类型。深度学习训练得益于高度专业化的数据类型

目前，如果你想用16位浮点数（FP16）进行稳定的反向传播，最大的问题是普通FP16数据类型只支持[-65,504, 65,504]范围内的数字。如果你的梯度滑过这个范围，你的梯度就会爆炸成NaN值。

为了防止在FP16训练中出现这种情况，我们通常会进行损失缩放，即在反向传播之前将损失乘以一个小数字，以防止这种梯度爆炸。

Brain Float 16格式（BF16）对指数使用了更多的比特，这样可能的数字范围与FP32相同，BF16的精度较低，也就是有效数字，但梯度精度对学习来说并不那么重要。

所以BF16所做的是，你不再需要做任何损失缩放，也不需要担心梯度会迅速爆炸。因此，我们应该看到，通过使用BF16格式，训练的稳定性有所提高，因为精度略有损失。

这对你意味着什么。使用BF16精度，训练可能比使用FP16精度更稳定，同时提供相同的速度提升。使用TF32精度，你可以得到接近FP32的稳定性，同时提供接近FP16的速度提升。

好的是，要使用这些数据类型，你只需用TF32取代FP32，用BF16取代FP16--不需要修改代码。

不过总的来说，这些新的数据类型可以被看作是懒惰的数据类型，因为你可以通过一些额外的编程努力（适当的损失缩放、初始化、规范化、使用Apex）来获得旧数据类型的所有好处。

因此，这些数据类型并没有提供速度，而是改善了训练中低精度的使用便利性。

风扇设计和GPU温度

虽然RTX 30系列的新风扇设计在冷却GPU方面表现非常好，但非创始版GPU的不同风扇设计可能会出现更多问题。

如果你的GPU发热超过80C，它就会自我节流，减慢其计算速度/功率。解决这个问题的办法是使用PCIe扩展器，在GPU之间创造空间。

用PCIe扩展器分散GPU对散热非常有效，华盛顿大学的其他博士生和我都使用这种设置，并取得了巨大的成功。它看起来并不漂亮，但它能使你的GPU保持凉爽!

下面这套系统已经运行了4年，完全没有问题。如果你没有足够的空间在PCIe插槽中安装所有的GPU，也可以这么用。

图5: 带PCIE扩展口的4显卡系统，看起来一团乱，但散热效率很高。

优雅地解决功耗限制问题

在你的GPU上设置一个功率限制是可能的。因此，你将能够以编程方式将RTX 3090的功率限制设置为300W，而不是其标准的350W。在4个GPU系统中，这相当于节省了200W，这可能刚好足够用1600W PSU建立一个4x RTX 3090系统的可行性。

这还有助于保持GPU的冷却。因此，设置功率限制可以同时解决4x RTX 3080或4x RTX 3090设置的两个主要问题，冷却和电源。对于4倍的设置，你仍然需要高效散热风扇的GPU，但这解决了电源的问题。

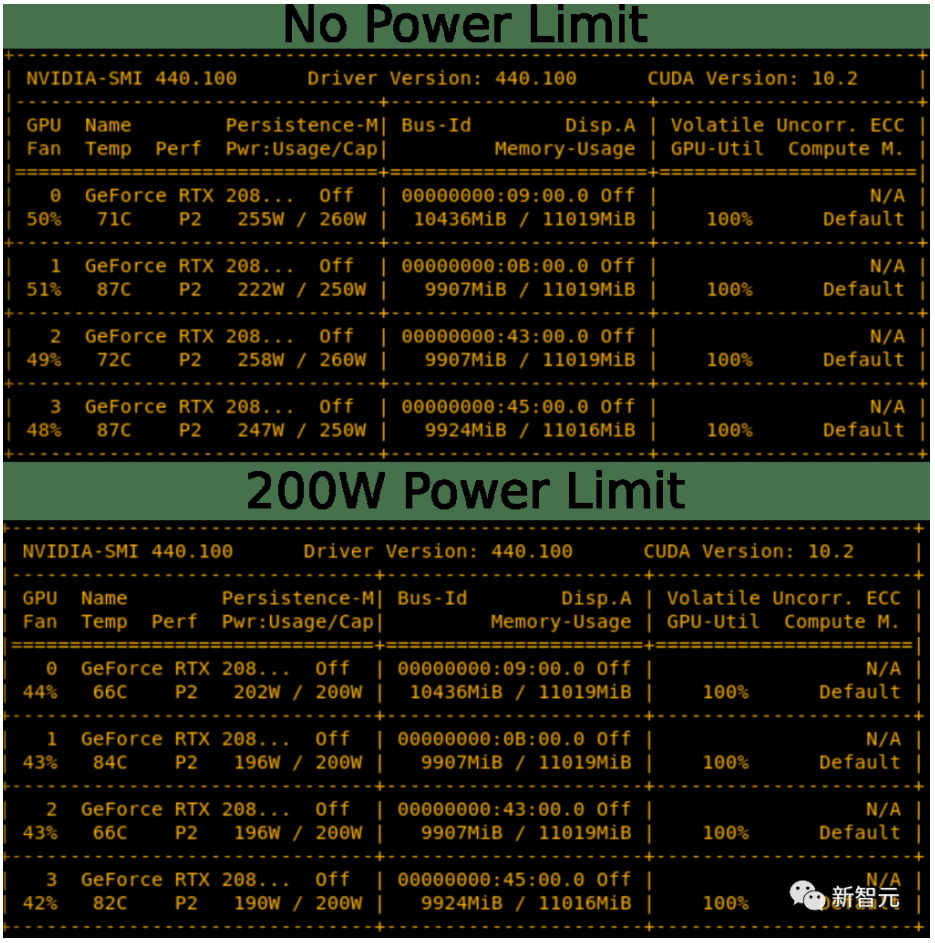


图6：降低功率限制有轻微的冷却效果。将RTX 2080 Ti的功率限制降低50-60W，温度略有下降，风扇运行更加安静

你可能会问，「这不会降低GPU的速度吗？」 是的，确实会降，但问题是降了多少。

我对图5所示的4x RTX 2080 Ti系统在不同功率限制下进行了基准测试。我对推理过程中BERT Large的500个小批次的时间进行了基准测试（不包括softmax层）。选择BERT Large推理，对GPU的压力最大。

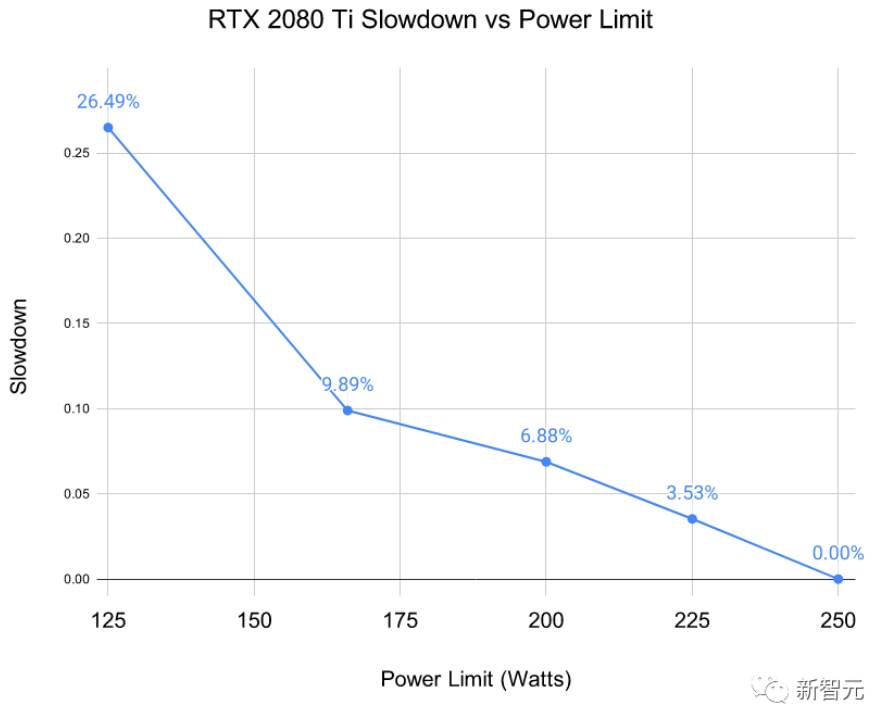


图7：在RTX 2080 Ti上，在给定的功率限制下测得的速度下降

我们可以看到，设置功率限制并不严重影响性能。将功率限制在50W，性能仅下降7%。

RTX 4090接头起火问题

有一种误解，认为RTX 4090电源线起火是因为被弯折过度了。实际上只有0.1%的用户是这个原因，主要问题是电缆没有正确插入。

因此，如果你遵循以下安装说明，使用RTX 4090是完全安全的。

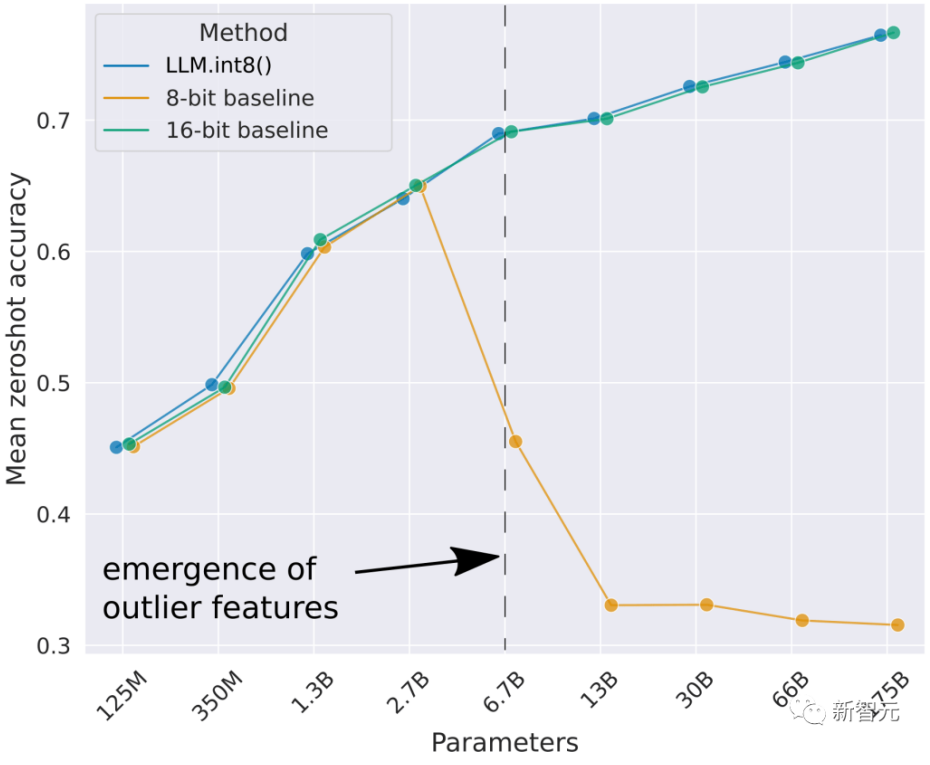
1. 如果你使用旧的电缆或旧的GPU，确保触点没有碎片/灰尘。
2.使用电源连接器，并将其插入插座，直到你听到咔嚓一声--这是最重要的部分。
3. 通过从左到右扭动电源线来测试是否合适。电缆不应该移动。
4.目视检查与插座的接触情况，电缆和插座之间无间隙。

H100和RTX40中的8位浮点支持

对8位浮点（FP8）的支持是RTX 40系列和H100 GPU的一个巨大优势。

有了8位输入，它允许你以两倍的速度加载矩阵乘法的数据，你可以在缓存中存储两倍的矩阵元素，而在Ada和Hopper架构中，缓存是非常大的，现在有了FP8张量核心，你可以为RTX 4090获得0.66 PFLOPS的计算量。

这比2007年世界上最快的超级计算机的全部算力还要高。4倍于FP8计算的RTX 4090，可与2010年世界上最快的超级计算机相媲美。

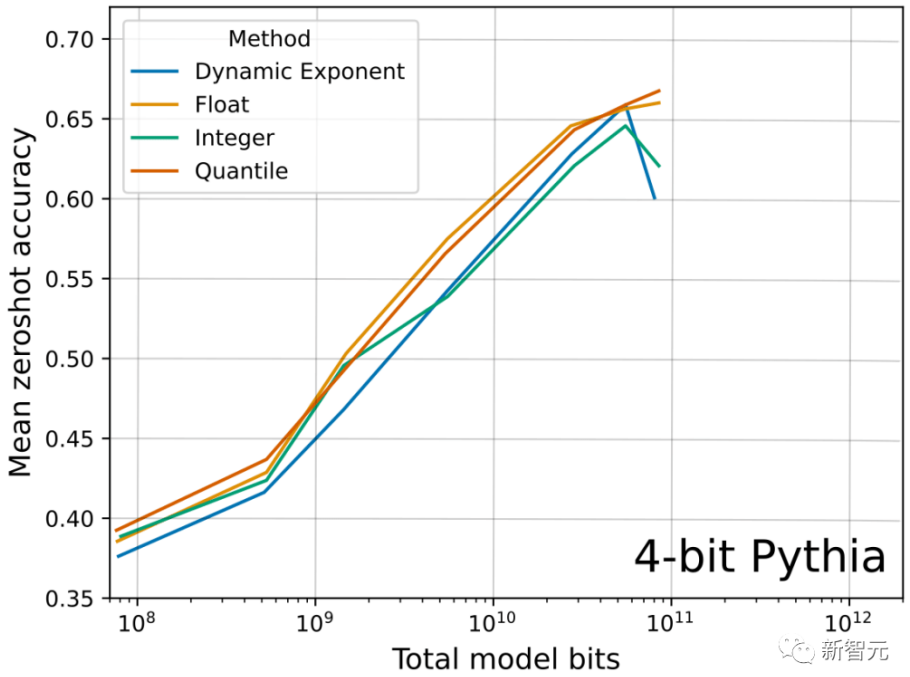


可以看到，最好的8位基线未能提供良好的零点性能。我开发的方法LLM.int8()可以进行Int8矩阵乘法，结果与16位基线相同。

但是Int8已经被RTX 30 / A100 / Ampere这一代GPU所支持，为什么FP8在RTX 40中又是一个大升级呢？FP8数据类型比Int8数据类型要稳定得多，而且很容易在层规范或非线性函数中使用，这在整型数据类型中是很难做到的。

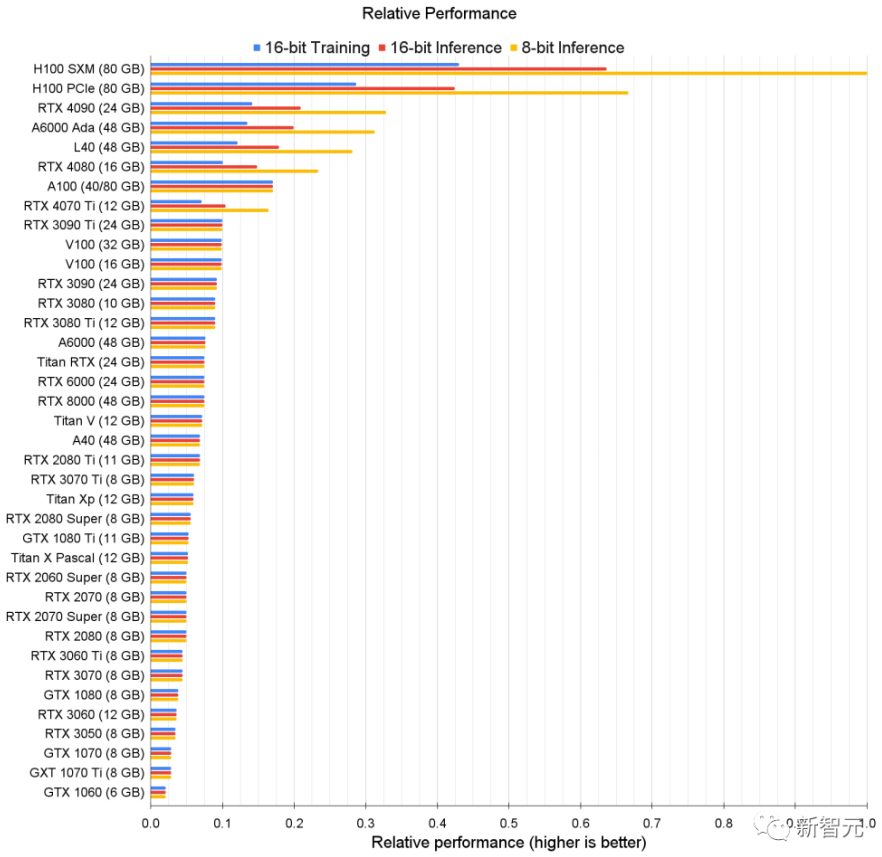
这将使它在训练和推理中的使用变得非常简单明了。我认为这将使FP8的训练和推理在几个月后变得相对普遍。

下面你可以看到这篇论文中关于Float vs Integer数据类型的一个相关主要结果。我们可以看到，逐个比特，FP4数据类型比Int4数据类型保留了更多的信息，从而提高了4个任务的平均LLM零点准确性。



GPU深度学习性能排行

先上一张图来看GPU的原始性能排行，看看谁最能打。



我们可以看到H100 GPU的8位性能与针对16位性能优化的旧卡存在巨大差距。

上图显示的是GPU的原始相对性能，比如对于8位推理，RTX 4090的性能大约是 H100 SMX 的 0.33 倍。

换句话说，与RTX 4090相比，H100 SMX的8位推理速度快三倍。

对于此数据，他没有为旧GPU建模8位计算。

因为8位推理和训练在Ada/Hopper GPU上更有效，而张量内存加速器 (TMA) 节省了大量寄存器，这些寄存器在 8 位矩阵乘法中非常精确。

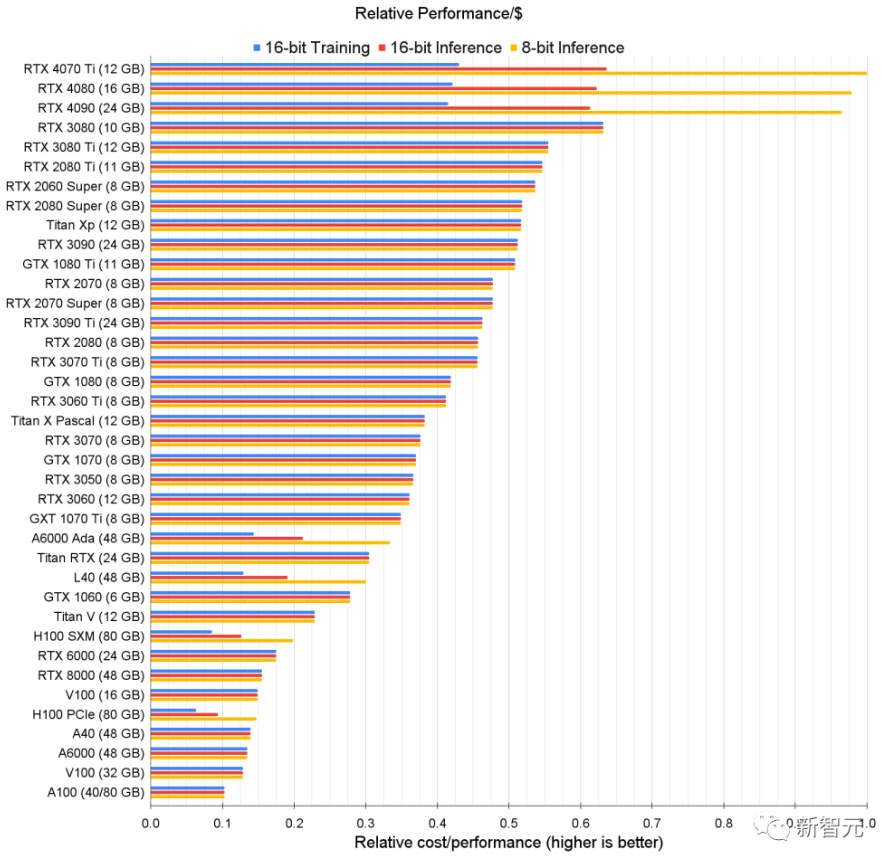
Ada/Hopper 也有 FP8 支持，这使得特别是 8 位训练更加有效，在Hopper/Ada上，8位训练性能很可能是16位训练性能的3-4倍。

对于旧GPU，旧GPU的Int8推理性能则接近16位推理性能。

每一美元能买到多少算力

那么问题来了，GPU性能强可是我买不起啊……

针对预算不充足的小伙伴，接下来的图表是他根据各个GPU的价格和性能统计的每美元性能排名（Performance per Dollar），侧面反映了GPU性价比。



选择一个完成深度学习任务并且符合预算的GPU，可分为以下几个步骤：

- **首先确定你需要多大的显存（至少12GB用于图像生成，至少24GB用于处理Transformer）；**
- **针对选8位还是16位（8-bit or 16-bit），建议是能上16位就上，8位在处理复杂编码任务时还是会有困难；**
- **根据上图中的指标，找到具有最高相对性能/成本的GPU。**

我们可以看到，RTX4070Ti 对于8位和16位推理的成本效益最高，而RTX3080对于16位训练的成本效益最高。

虽然这些GPU最具成本效益，但他们的内存也是个短板，10GB和12GB的内存可能无法满足所有需求。

但对于刚入坑深度学习的新手来说可能是理想GPU。

其中一些GPU非常适合Kaggle竞赛，在Kaggle比赛中取得好成绩，工作方法比模型大小更重要，因此许多较小的 GPU非常适合。

Kaggle号称是全球最大的数据科学家汇聚的平台，高手云集，同时对萌新也很友好。

如果用作学术研究和服务器运营的最佳GPU似乎是 A6000 Ada GPU。

同时H100 SXM的性价比也很高，内存大性能强。

个人经验来说，如果我要为公司/学术实验室构建一个小型集群，我推荐66-80%的A6000 GPU 和20-33%的 H100 SXM GPU。

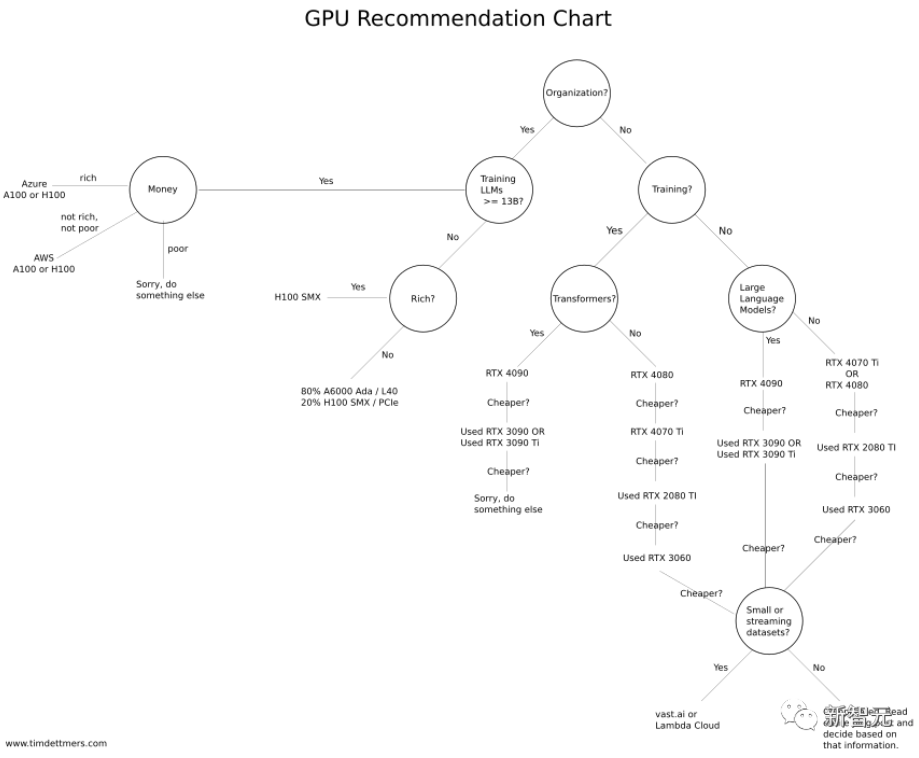
综合推荐

说了这么多，终于到了GPU安利环节。

Tim Dettmers专门制作了一个「GPU选购流程图」，预算充足就可以上更高配置，预算不足请参考性价比之选。

这里首先强调一点：无论你选哪款 GPU，首先要确保它的内存能满足你的需求。为此，你要问自己几个问题：

我要拿GPU做什么？是拿来参加 Kaggle 比赛、学深度学习、做CV/NLP研究还是玩小项目？



预算充足的情况下，可以查看上面的基准测试并选择适合自己的最佳GPU。

还可以通过在vast.ai或Lambda Cloud中运行您的问题一段时间来估算所需的GPU内存，以便了解它是否能满足你的需求。

如果只是偶尔需要一个GPU（每隔几天持续几个小时）并且不需要下载和处理大型数据集，那么vast.ai或 Lambda Cloud也能很好地工作。

但是，如果一个月每天都使用GPU且使用频率很高（每天12小时），云GPU通常不是一个好的选择。

参考资料：

https://timdettmers.com/2023/01/16/which-gpu-for-deep-learning/#more-6

https://timdettmers.com/

往期精彩：

深度学习论文精读[14]：Vision Transformer
深度学习论文精读[13]：Deeplab v3+
深度学习论文精读[12]：Deeplab v3
深度学习论文精读[11]：Deeplab v2
深度学习论文精读[10]：Deeplab v1
深度学习论文精读[9]：PSPNet
深度学习论文精读[8]：ParseNet
深度学习论文精读[7]：nnUNet
深度学习论文精读[6]：UNet++
深度学习论文精读[5]：Attention UNet
深度学习论文精读[4]：RefineNet
深度学习论文精读[3]：SegNet
深度学习论文精读[2]：UNet网络
深度学习论文精读[1]：FCN全卷积网络

阅读原文