通过随机重要性采样的高效设备端分布式 DNN 训练

Mercury: Efficient On-Device Distributed DNN Training via Stochastic Importance Sampling

Introduction

近些年随着并行化深度学习框架的应用，一些深度模型在相关领域中取得了很好的成效。另一方面，智能芯片也在不断的发展，一些终端设备具有训练模型的能力，这使得模型的训练正在向边缘化发展。

在边缘化设备上进行分布式模型训练有很大的价值，但是最大的瓶颈则是无线设备的传输带宽，它大大降低了整个模型的培训过程。

本文提出的Mecury方法主要是通过对本地数据进行采样评估，选取价值最高的一些样本进行训练，从而加快培训效率。

本文的挑战有三：

1. 重要性采样会引入额外的计算开销，如何权衡开销和花费是一个重要问题。
2. 设备采样得到的只是局部重要性分布而不是全局重要性分布，这可能导致模型反复学习无关数据从而降低模型训练效率。
3. 重要性采样的代价不可消除，另外无线网络的带宽变化也是不稳定的。

为了解决这些挑战，文章分别提出了分组重要性计算和采样技术，数据重要性感知重新分配技术，带宽自适应调度器来解决上述挑战。从而在不损失模型精度的同时提高了模型的训练速度。

Background & Motivation

Distributed DNN Training

分布式机器学习采用SGD及其变体来对DL模型进行训练，每次迭代，client将训练好的梯度信息向server进行汇聚。mercury采用multi-PS（*使用一个或多个机器充当server角色*）架构，并且将每个边缘设备都当作parameter server来使用，从而避免单一设备网络带宽成为瓶颈。

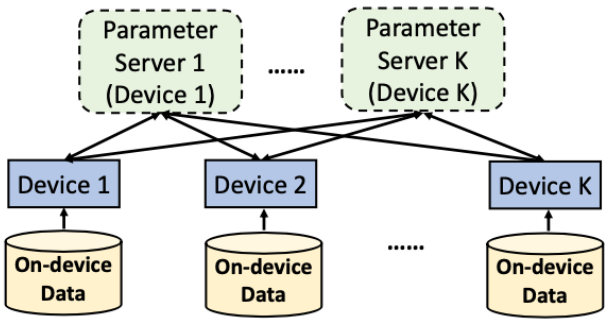


Figure 1: The multi-parameter server (multi-PS) architecture adopted by Mercury for on-device distributed training.

Existing Approaches and Their Limitations

模型的整体训练时间`T`可以表示为：

$$T = E * (T_{cp} + T_{cm})$$

其中`E`是模型收敛所需要的迭代次数，T_{cp}是本地模型的训练时间花费，T_{cm}是通讯开销；

以往的优化方案有：

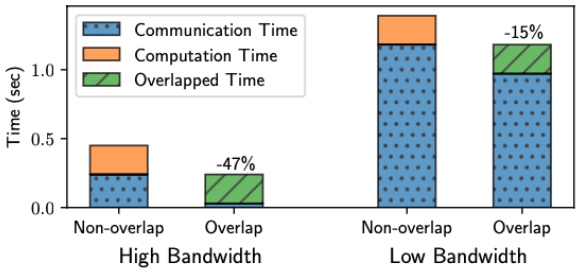
- Gradient Compression

> 梯度压缩方案大致有两个方向：数据压缩——采用更少比特数来进行数据传输；梯度选取——选取更重要的梯度进行传输；它们都是通过降低T_{cm}来加速模型训练速度。
>
> 这种方法在带宽受限的场景下，加速效果是受限的，同时也会影响模型精度。

- Local SGD

> 这种方法让client在本地进行多次迭代后再进行梯度汇聚，通过降低通讯次数来避免通讯瓶颈。但这种方案同样也会对模型精度造成影响。

- Communication-Computation Overlapping

> 这种方案将计算和通讯过程同步进行，从而对外显示的隐藏掉过多的通讯开销。这种方法在带宽受限的情景下，加速效果并不明显。
>
> 
>
>
> **Figure 2: Communication-computation overlapping in high bandwidth and low bandwidth settings.**

Mercury

Design Principle

在分布式机器学习中，主要采用SGD算法，这种算法采用随机采样策略，其核心在于认为数据的重要性是平均分配的。

然而，实际上所有数据样本并不是同等重要的。并且随着训练的进行，其重要性也是在不断变化的。

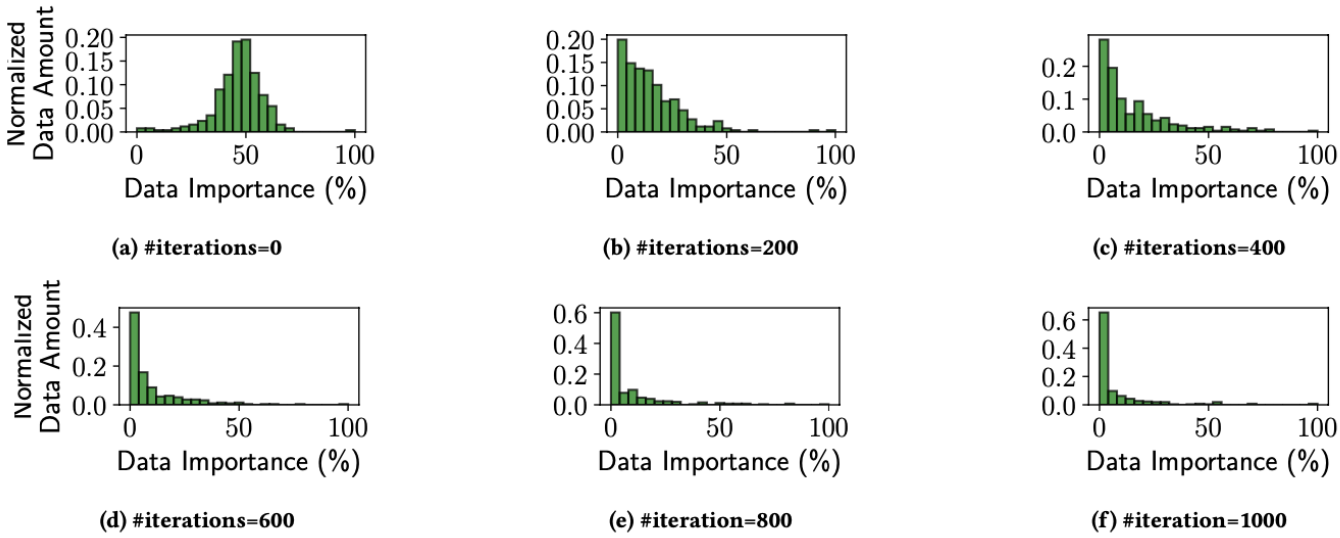


Figure 3: Distributions of data importance as the number of iterations increases during training.

即样本的重要性在时间和空间上的分布是不均匀的。而且重要的数据集中在小样本中，而其他样本提供的贡献则是有限的。

因此采用重要性更高的数据来加速模型收敛是可行的。

对于数据重要性的评估，文章采用前馈损失（feed-forward loss）来代表。

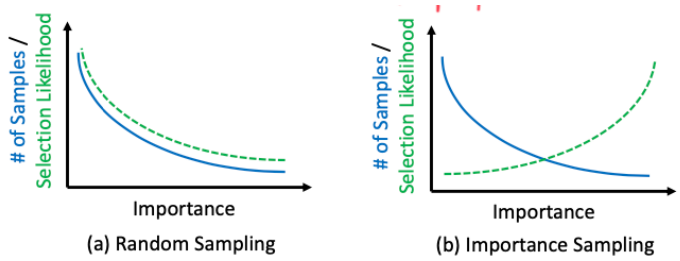


Figure 4: Comparison between (a) random sampling and (b) importance sampling.

Performance Model

文章提出的mercury架构在传统SGD架构上增加了重要性采样和重要性计算两个操作。

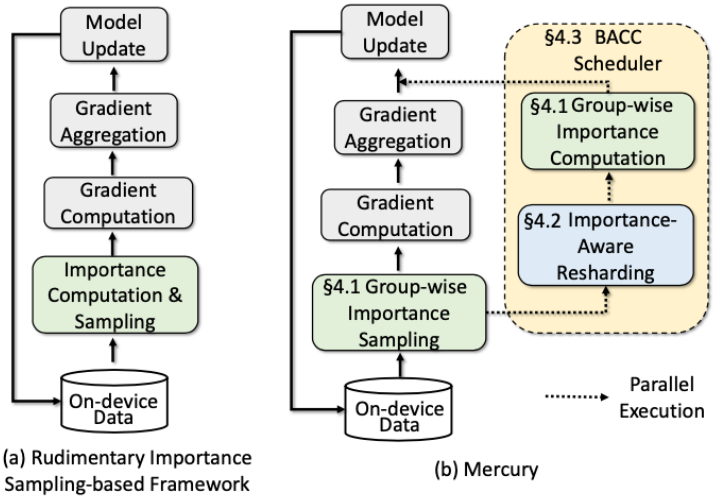


Figure 5: Comparison between (a) rudimentary importance sampling-based framework and (b) Mercury.

优化后的模型框架的加速比可以表示为：

$$
\begin{aligned}
Speedup &= \frac{E \cdot (T_{cp} + T_{cm})}{T_{is} \cdot (T_{cp} + T_{cm} + T_{is})} \\
&= \frac{1}{\frac{E_{is}}{E} \cdot (1 + \frac{T_{is}}{T_{cp}+T_{cm}})}
\end{aligned}
$$

其中，E_{is}是优化后模型训练需要迭代的次数，T_{is}代表样本重要性计算所引入的额外开销。公式表示，该方案也需要进行优化，不然也会存在加速比小于1的情况在。

Overall Design

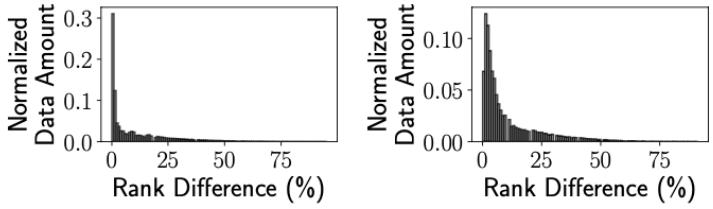
文章提出的mercury提出了三个方向的技术从而确保能够获得理想的加速比：1). 重要性计算和采样；2). 数据重分布；3). 带宽通讯调度器

Details

Group-wise Importance Computation and Sampling

文章提出，每次迭代对本地所有数据进行重要性计算和采样所产生的计算代价是非常昂贵的，因此文章提出采用组采样的方法，即将本地数据随机分为多个组，并以其中某一个组的重要性分布来代表本地数据的重要性分布，并以此构建迷你批数据。

同时，文章发现，数据的重要性分布一般不会突然变化，因此可以减少重要性分布的计算次数。



(a) Distribution of importance rank changes after 1 iteration. (b) Distribution of importance rank changes after 10 iterations.

Figure 6: Distribution of importance rank changes across iterations.

Importance-aware Data Resharding

文章指出，在联邦学习当中，本地数据训练的模型并不能代表全局模型，尤其是当模型数据的分布不是non-IID (*identically and independently distributed*)时，这种场景下，学习零碎数据可能会阻碍模型的快速收敛。因此文章提出数据重分布过程，将重要数据重新分配到各个边缘设备当中，从而保证模型训练梯度下降方向。

BACC Scheduler

文章指出，边缘设备的带宽波动很大，所有需要一个自适应带宽调度算法，来充分利用边缘设备的带宽资源。文章提出采用双线程的思量来灵活利用边缘设备的带宽资源和计算资源。

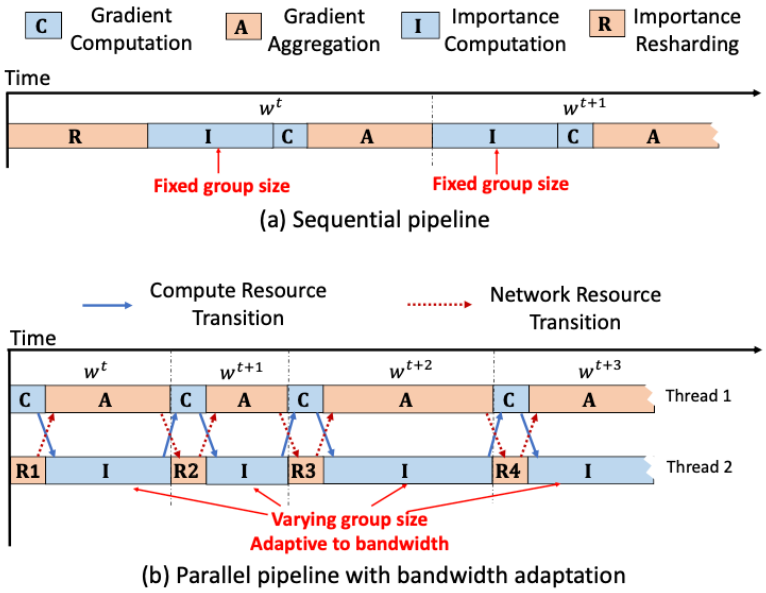


Figure 9: Comparison between (a) sequential pipeline and (b) parallel pipeline with bandwidth adaptation. w^t represents the model weight at iteration t. R = R1 + R2 + R3 + R4 in length.

Evaluation

文章在六个公共数据集上进行了测试，应用场景包含图像识别，语音识别，自然语言处理。

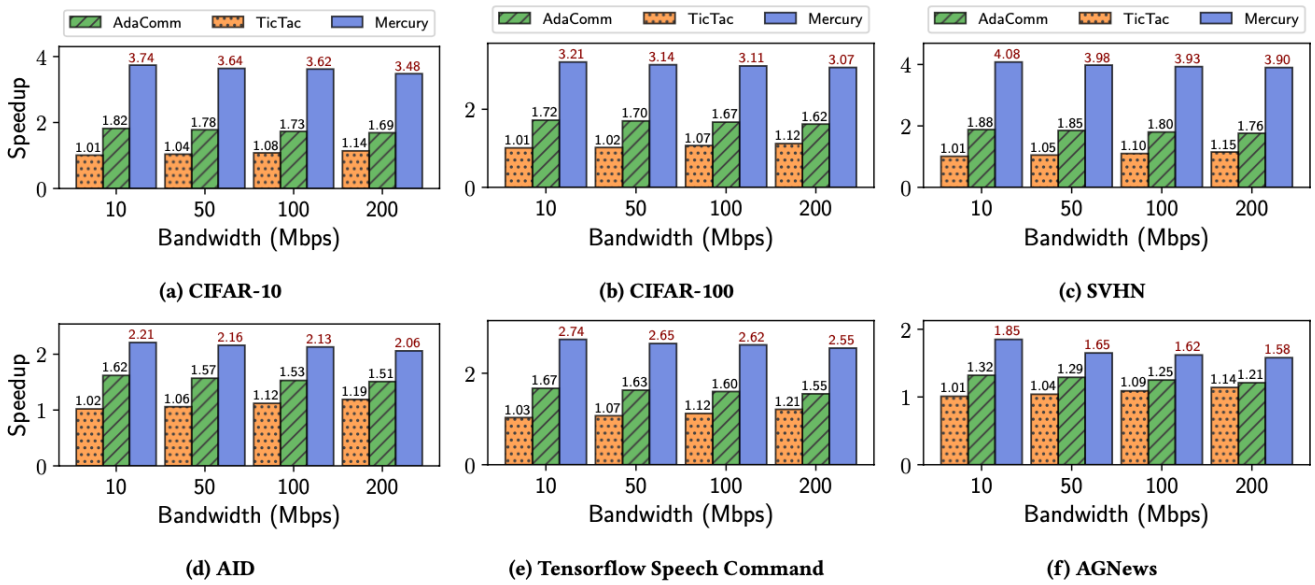


Figure 11: Overall performance comparison between Mercury, TicTac and AdaComm. Each bar represents the training speedup over standard distributed SGD in total training time.

实验发现，相较于同类分布式训练框架，mercury明显优于TicTac和AdaComm。

Thinking

advantage

文章通过使用重要性数据进行模型训练的方法来加速模型训练的方法有其独特的特点。

disadvantage

文章所进行的重要性采样算法其实是一个估计指标，重要性数据和收敛模型之间的关系需要确保相关性。

文章中提出的数据重分布过程打破了联邦学习中本地数据私有化的原则。