论文分享

CodedReduce: A Fast and Robust Framework for Gradient Aggregation in Distributed Learning

背景知识

1. 参数服务器（Parameter Server, PS）
2. 环式（Ring）架构

《分布式机器学习：算法、理论与实践》第七章

3. Ring - AllReduce)

   "P. Patarasuk and X. Yuan, “Bandwidth optimal all-reduce algorithms for clusters of workstations,” J. Parallel Distrib. Comput., vol. 69, pp. 117–124, Feb. 2009."

4. Gradient Coding (GC) 梯度编码

   "J. Xu, S.-L. Huang, L. Song, and T. Lan, “Live gradient compensation for evading stragglers in distributed learning,” in Proc. IEEE Conf. Comput. Commun., May 2021, pp. 3368–3376."

摘要

在分布式机器学习中，同步梯度下降模式会遇到两个系统瓶颈:通信带宽和掉队延迟。目前已有一些有效且鲁棒性好的策略去克服上述问题，比如RAR (Ring - AllReduce)设计可以避免任何特定节点的带宽瓶颈，它允许每个worker只与其逻辑环上的邻居通信。

其次最近提出的梯度编码 (GC)，它允许将冗余的数据集分配给workers，来减少主从架构中的掉队者(stragglers.)。

此篇论文提出了一种联合通信拓扑设计和数据集分配策略，称为 CodedReduce (CR)，它结合了 RAR 和 GC 的优点。 也就是说，它通过树拓扑并行化通信，从而实现高效的带宽利用，并在节点处精心设计冗余数据集分配和编码策略，使所提出的梯度聚合方案对落后者(stragglers)具有鲁棒性。

实验评估表明CR相比基准GC和RAR快了最高27.2倍和7.0倍

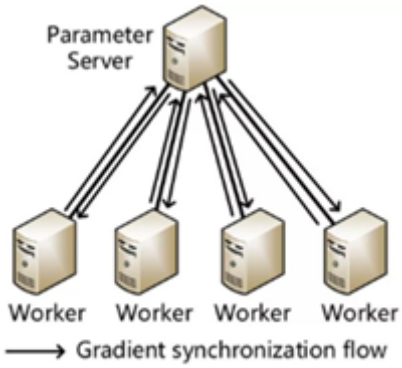
历史问题

主从架构

梯度下降法Gradient Descent (GD)在分布式机器学习中通常采用主从架构(master-worker)。

通过一系列的迭代，主服务器master从worker收到的结果来更新底层模型，worker使用他们的本地数据批计算部分梯度，并在每次迭代时上传到master。

这种方法worker 与 master 的**并发通信**(数据量N*M)会导致 master 的带宽拥塞，以及由缓慢的worker或stragglers造成的延迟，这些延迟会显着增加运行时间。



RAR

Ring-AllReduce可以减轻上述问题带来的影响，数据集 D 均匀分布在 N 个worker之间，每个节点组合为环形，并沿环传递其部分梯度，以便在整体操作结束时，worker拥有完整的副本梯度g，以此来它避免了任何特定节点的带宽瓶颈。RAR 最近已成为用于模型更新的分布式深度学习的核心组件

尽管带宽效率高，AllReduce 类型的算法本质上对stragglers很敏感，这使得它们容易出现显着的性能下降，如果任何一个工作进程变慢，甚至完全失败。随着集群规模的增加，需要更加重视stragglers瓶颈。

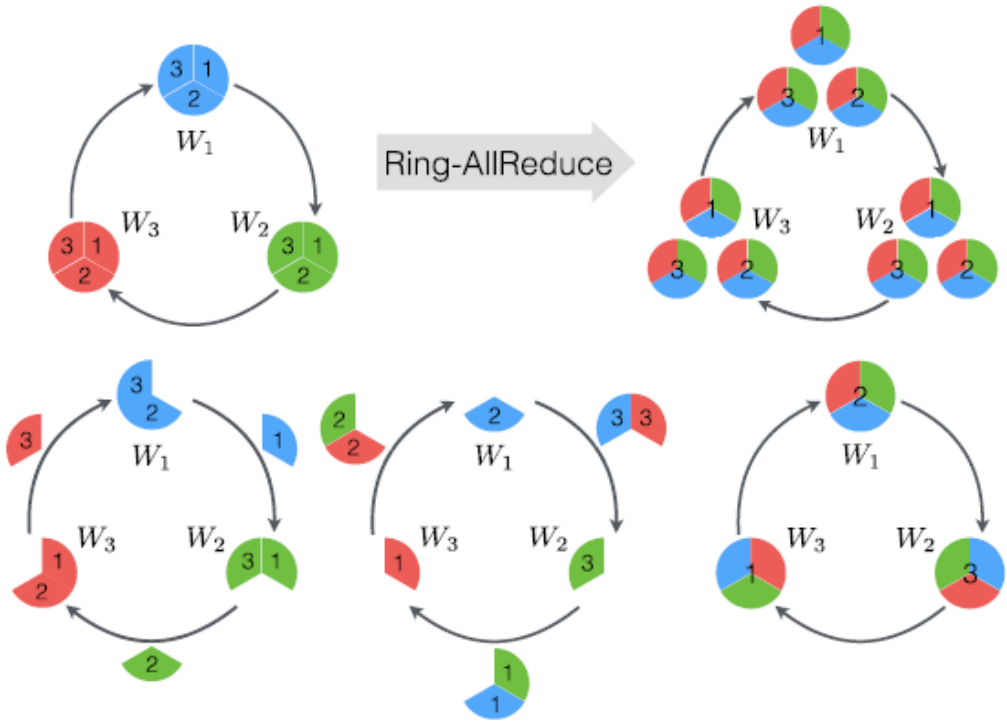


Fig. 3. Illustration of communication strategy in RAR for $N = 3$ workers.

GC

梯度编码Gradient Coding (GC) 来缓解主从拓扑中分布式梯度聚合中的stragglers。在 GC 中，数据集 D 小心地冗余分布在 N 个 worker 中，其中每个 worker 从其本地批次计算编码梯度。 主节点等待任何 N-S 个 worker 的结果并恢复总梯度 g，其中参数 **S 表示可以容忍的最大stragglers数。** 因此，GC 防止主节点等待所有工作节点完成他们的计算，并且与传统未编码主节点模式 相比，GC 已被证明可以实现显着的加速。

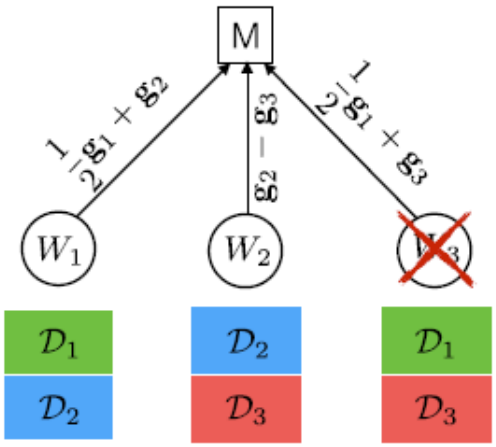


Fig. 4. Illustration of data allocation and communication strategy in GC for $N = 3$ workers.

然而，随着集群规模变大，GC 在 master 上会遭受严重的网络拥塞。 特别是，通信开销增加到 O(N)，因为 master 需要从 O(N) worker 接收消息。 因此必须设计分布式学习策略，以减轻stragglers，同时在整个集群中降低通信开销。

此文目标问题为：能否在分布式梯度聚合中同时实现 RAR 的通信并行化和 GC 的落后容忍度?

设计

此文提出了一种用于同步分布式梯度聚合的可扩展且高鲁棒性的方案，称为 CodedReduce（CR）。

CR 背后有两个关键思想。

首先，我们使用逻辑树拓扑进行通信，该通信由一个主节点、L 层工作节点组成，其中每个父节点有 n 个子节点。在提议的配置中，每个节点仅与其父节点通信以下载更新的模型和上传部分梯度。与经典的 master-worker 设置一样，根节点（master）恢复完整梯度并更新模型。除叶节点外，每个节点从其子节点接收**足够数量的编码部分梯度，将它们与其本地部分梯度组合，并将结果上传到其父节点。**这种分布式通信策略缓解了节点的通信瓶颈，因为多个父节点可以同时从他们的子节点接收。

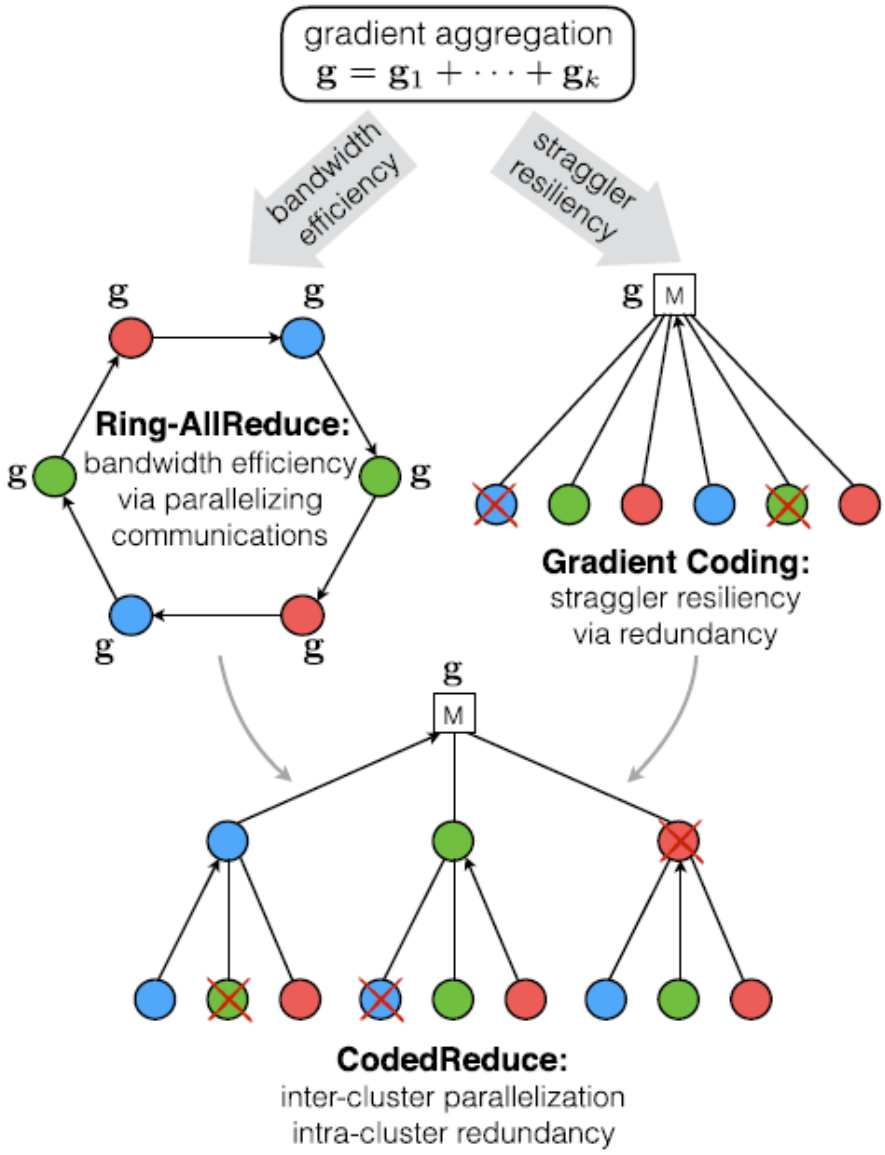


Fig. 1. Illustration of RAR, GC and CR: In RAR, workers communicate only with their neighbors on a ring, which results in high bandwidth utilization; however, RAR is prone to stragglers. GC is robust to stragglers by doing redundant computations at workers; however, GC imposes bandwidth bottleneck at the master. CR achieves the benefits of both worlds, providing high bandwidth efficiency along with straggler resiliency.

其次，CR 中使用的编码策略为stragglers提供了鲁棒性。利用 GC 的思想并提出数据分配和通信策略，使每个节点只需等待其任何 n - s个子节点的 返回其结果。s为可容忍的straggler的个数，每层可容忍s个。

CR总共可容忍S=s^L个straggler。

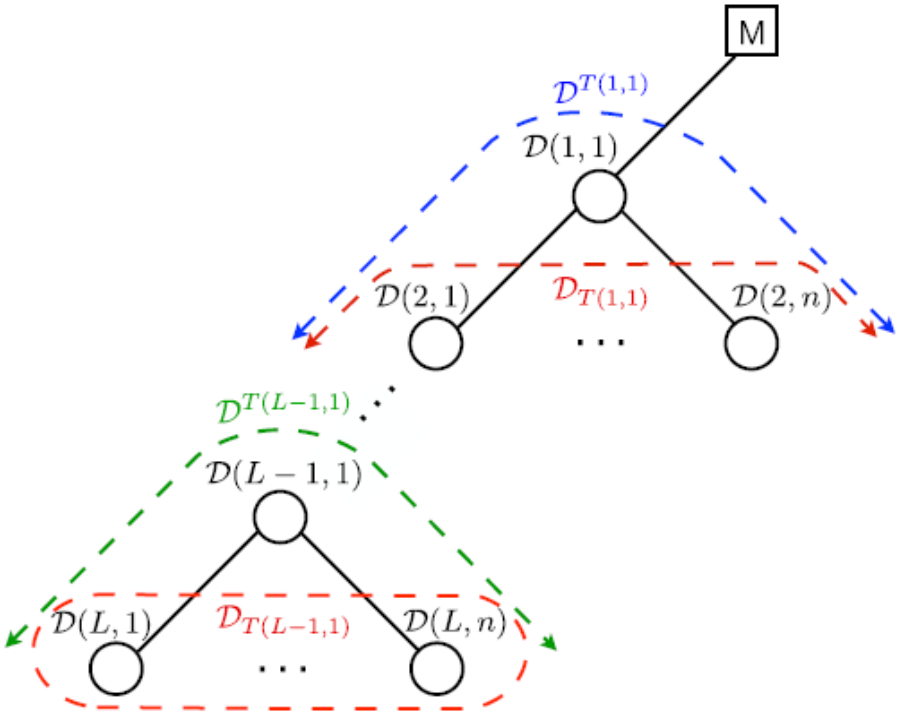


Fig. 6. Illustration of task allocation in CR.

结论

除了可证明的理论保证外，所提出的 CR 方案在实践中提供了实质性的改进。 作为一个代表性案例，图 2 提供了在 Amazon EC2 集群上实施的许多梯度下降迭代的平均梯度聚合时间。 与经典的 Uncoded Master-Worker (UMW)、GC、RAR 三个基准相比，所提出的 CR 方案分别获得了 22.5 倍、6.4 倍和 4.3 倍的加速。

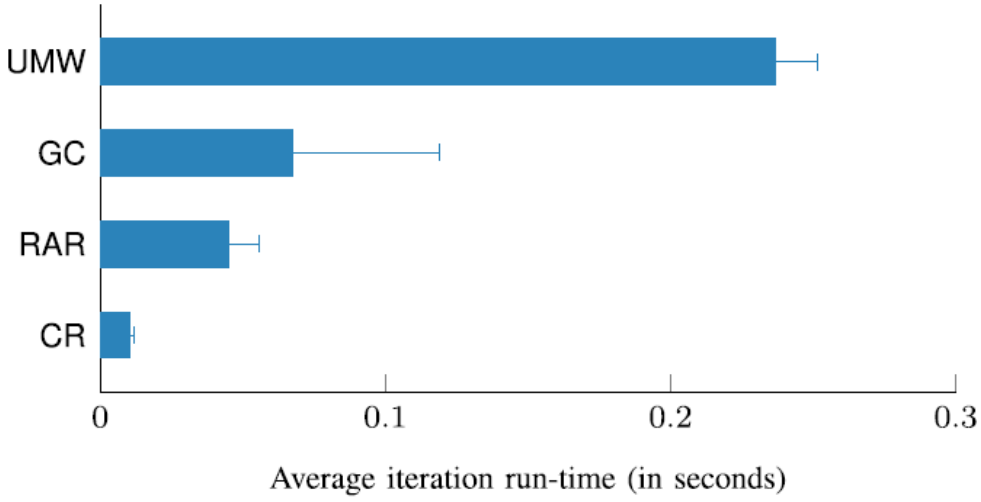


Fig. 2. Average iteration time for gradient aggregation in different schemes CR, RAR, GC and UMW: Training a linear model is implemented on a cluster of $N = 84$ `t2.micro` instances.

##