联邦学习中的通信优化

17363092 叶茂青，17363079 王珺

June 30, 2020

Abstract

联邦学习指众多客户端在中心服务器的协调下进行机器学习任务，联邦学习可以让用户在保证隐私的同时贡献自己的数据，但高昂的通信成本限制了联邦学习的应用。本文总结了联邦学习中针对网络梯度传输所做的优化，并提出这一领域仍需解决的问题。

1 联邦学习的介绍

联邦学习的数学模型描述在 2016 年由 McMahan et al. [1] 提出，区别于分布式学习，联邦学习的数据分散在各个客户端中，且出于对用户隐私的保护，在联邦学习中，数据不能发送给服务器，客户端只能传输对模型的更新信息。联邦学习的提出，有利于解决公司之间的数据壁垒，对个人来说，也可以避免自己的隐私泄露。不过联邦学习的这一特点也使得机器学习模型的训练变得困难，McMahan et al. [1] 提出了联邦学习区别于分布式学习的四大难点：

- Non-IID

- Unbalanced
- Massively distributed
- Limited communication

由于联邦学习的应用场景中，客户端多为个人设备，通信成本高且通信质量也不能保证，直接传输未经处理的梯度是不现实的。面对联邦学习中的这一问题，学者们提出了两种对梯度传输进行优化的方法，一是 Gradient quantization，降低梯度传输的精度，二是 Gradient sparsification，降低梯度传输的数量。2018 年 Google [2] 使用联邦学习训练手机键盘的预测模型中，便使用了一种量化策略，将模型压缩到 1.4MB，使得用户可以快速发送手机上的梯度信息。

2 对于梯度传输的优化策略

2.1 Gradient quantization

Gradient quantization 的思路是将原来 32 位浮点数存储的梯度信息量化为更低准确度的存储表示。这方面的研究有 Seide et al. [3] 提出的 1-bit SGD、Alistarh et al. [4] 提出的 QSGD、Wen et al. [5] 提出的 TernGrad、Zhou et al. [6] 提出的 DoReFa-Net 等。

Gradient quantization 的难点在于如何在压缩梯度后依然能保证网络的收敛，以及如何选择合适的量化程度。

Seide et al. [3] 提出的 1-bit SGD，将梯度量化为 1-bit，为了保证模型的收敛，1-bit SGD 中采用了 Error Feedback 的方法，将每次的量化误差保存下来并再次反馈到量化函数中，这一方法在之后的众多研究中均有体

现，在实验观察下，1-bit SGD 并没有过多的降低模型的精度和收敛速度，但后续的研究表明，对于一些复杂的网络，1-bit SGD 采用的量化策略或许过于激进。

Alistarh et al. [4] 提出的 QSGD 则用数学证明了其在任意函数下的收敛性，在实验的表现中，使用 QSGD 对梯度进行 4bits 或 8bits 量化，无论是在 resnet 还是 lstm 上的精度都和 32-bits SGD 没有太大区别，同时实验也提出了卷积层对于量化的敏感性或许要更高。相比于 1-bit SGD，QSGD 适用于不同的量化程度，允许用户在通信量和训练时间进行权衡。

Wen et al. [5] 提出的 TernGrad 将梯度为 2bits，使用-1,0,1 三个值表示梯度。其数学表示为

$$\tilde{\boldsymbol{g}}_t = \text{ternarize}\left(\boldsymbol{g}_t\right) = s_t \cdot \text{sign}\left(\boldsymbol{g}_t\right) \circ \boldsymbol{b}_t$$

where

$$s_t \triangleq \max\left(abs\left(\boldsymbol{g}_t\right)\right) \triangleq \left\|\boldsymbol{g}_t\right\|_\infty$$

其中 b_t 满足伯努利分布

$$\begin{cases} P\left(b_{tk} = 1 \mid \boldsymbol{g}_t\right) = \left|g_{tk}\right| / s_t \\ P\left(b_{tk} = 0 \mid \boldsymbol{g}_t\right) = 1 - \left|g_{tk}\right| / s_t \end{cases}$$

为了避免某些梯度过大导致 s_t 过大，使得大部分梯度被量化为 0，作者引入 layer-wise ternarizing，逐层的对梯度进行量化，同时对于不同层之间梯度方差的差异，作者引入 gradient clipping，用方差信息对每一层的梯度进行裁剪。作者使用在 MNIST 和 CIFAR-10 上训练的卷积网络进行验证，相比 32-bits SGD，使用 TernGrad 并没有减慢收敛的速度，且精度损失也在 1% 以内。Wen et al. [5] 同时也完善了 Alistarh et al. [4] 对于收敛性的证明，并解释了 QSGD 中的部分机理。

Zhou et al. [6] 提出的 DoReFa-Net 不仅对梯度进行量化，也将网络中的权重进行量化，同时提出 Bit Convolution Kernels 用以训练网络，DoReFa-Net 对权重也同时进行量化的策略，虽然使得网络所需的运算量明显下降，但在实验中使用 AlexNet 进行测试，可以明显观察到精度的下降。

2.2 Gradient sparsification

Gradient sparsification 的想法建立在梯度稀疏性的假设上，只传输部分对权重影响大的梯度，从而减少发送的数据。这方面的研究有 Strom [7]、Dryden et al. [8]、Lin et al. [9]、Aji & Heafield [10]、Chen et al. [11] 等。

Gradient sparsification 的难点主要有两点：

1. 如何选择合适的阈值对梯度进行稀疏？直接设定阈值的方法虽然简单，但可能需要反复调试，且同一网络的各层也可能需要不同的阈值。设定比率的方法则会引入更多额外的计算，会降低模型训练的速度。

2. 稀疏之后如何保证网络仍能收敛？对此，几乎所有的研究都会保存下未发送的梯度，并以不同的方式反映到下一次的迭代中，以尽可能的保留梯度信息。

Strom [7] 同时运用了量化和稀疏的方法，只传送大于某一阈值的梯度，同时将梯度量化为 1bit，为了存储需要更新的梯度索引，用 31bit 传输梯度的索引信息，同时使用了 1-bit SGD 中提到的 Error Feedback 方法。

Dryden et al. [8] 直接选取阈值对梯度进行选择时可能需要多次实验才能找到适合的阈值，Dryden et al. 提出使用固定比例来选择梯度，发送网络中梯度绝对值前 π 大的正负梯度，但相比直接使用阈值进行稀疏，这样的方法会引入额外的计算时间。

Lin et al. [9] 使用的稀疏方法与之前类似，为了避免稀疏过程丢失大量梯度信息，会将没有超过阈值的梯度保存下来，直至其累加超过阈值才

会被发送出去，同时也提出了一种对动量进行修正的方法，使得网络使用 Momentum SGD 进行训练时，更新的路径大致相同。

Aji & Heafield [10] 提出的 Gradient Dropping 通过设定 dropping rate 进行梯度稀疏，为了尽可能多的保存信息依旧采用 Error Feedback 的思路，将上一轮迭代的残差加入到计算当中。使用比例的一大缺点是会引入额外的计算误差，为此 Gradient Dropping 使用 0.1% 的梯度来计算近似阈值，Gradient Dropping 还引入了 Layer normalization [12] 规范各层的梯度范围，从而可以使用全局阈值对梯度进行稀疏。实验中使用卷积网络在 MNIST 上训练发现，在丢弃掉 99% 的梯度后，模型的收敛速度和精度基本不受影响，继续提高丢弃率则会明显延长网络的训练时间。Layer normalization [12] 在卷积网络上没有什么作用，但在翻译任务上使用的 LSTM 则加快了收敛速度。实验也尝试了与梯度量化技术的结合，发现在更为复杂的 LSTM 模型上，1-bit Quantization 会过于激进，而 2-bits quantization 对模型的影响较小。

Chen et al. [11] 提出的 AdaComp，使用自适应的稀疏调节方法，将梯度根据网络的层级结构分入若干个桶中，统计每个桶中的最大值，如果上一轮梯度的残差加上当前梯度的两倍大于该最大值，则将该梯度量化后发送。在多个 CNN 网络和 LSTM 网络的实验下，AdaComp 均保证了模型的精度和收敛速度。作者也尝试了将 AdaComp 和 Adam 优化器结合起来，在 CNN 网络训练的结果表明对模型并没有太大影响。

3 未来方向

对于梯度量化的研究，Alistarh et al. [4]、Wen et al. [5] 已经从数学层面上证明了量化后的梯度依然可以使得网络收敛，但如何选取合适的量

化程度仍是一个值得研究的问题，Aji & Heafield [10] 在实验中表明，在 LSTM 模型中使用 1-bit Quantization 会影响模型的收敛速度，但至今也没有研究指出应该如何设定量化的程度，以及网络量化的极限是多少。

对于梯度稀疏化的研究表明在丢弃网络 99% 以上的梯度后，模型依然能保持原有的精度和收敛速度，但目前在阈值选择上依旧有很多待研究的问题，直接选取阈值需要进行多次实验，且不同层间也可能需要选择不同阈值，Aji & Heafield [10] 引入 Layer normalization [12] 以保证使用全局阈值时模型的收敛速度，但同时也使得模型更为复杂。Dryden et al. [8] 采用比例的方法则会引入很多额外的运算，如何平衡梯度稀疏后的模型的复杂度与精度依然是一个值得研究的问题。

References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017*, feb 2016. [Online]. Available: http://arxiv.org/abs/1602.05629

[2] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated Learning for Mobile Keyboard Prediction," nov 2018. [Online]. Available: http://arxiv.org/abs/1811.03604

[3] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, "1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs," Tech. Rep., 2014. [Online]. Available: https://www.isca-speech.org/archive/interspeech_2014/i14_1058.html

[4] D. Alistarh, D. Grubic, J. Z. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," Tech. Rep., 2017. [Online]. Available: http://papers.nips.cc/paper/6768-qsgd-communication-efficient-sgd-via-gradient-quantization-and-encoding

[5] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, "TernGrad: Ternary gradients to reduce communication in distributed deep learning," Tech. Rep., 2017. [Online]. Available: https://github.com/wenwei202/terngrad

[6] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "DoReFa-Net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients," Tech. Rep., 2016. [Online]. Available: http://arxiv.org/abs/1606.06160

[7] N. S. S. A. C. of the International and undefined 2015, "Scalable distributed DNN training using commodity GPU cloud computing," *isca-speech.org*. [Online]. Available: https://www.isca-speech.org/archive/interspeech_2015/i15_1488.html

[8] N. Dryden, T. Moon, S. A. Jacobs, and B. Van Essen, "Communication quantization for data-parallel training of deep neural networks," *Proceedings of MLHPC 2016: Machine Learning in HPC Environments - Held in conjunction with SC 2016: The International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–8, 2017. [Online]. Available: https://github.com/LLNL/lbann.

[9] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training," dec 2017. [Online]. Available: http://arxiv.org/abs/1712.01887

[10] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*. Association for Computational Linguistics (ACL), 2017, pp. 440–445.

[11] C. Y. Chen, J. Choi, D. Brand, A. Agrawal, W. Zhang, and K. Gopalakrishnan, "ADaComP: Adaptive residual gradient compression for data-parallel distributed training," *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp. 2827–2835, 2018. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/viewPaper/16859

[12] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," jul 2016. [Online]. Available: http://arxiv.org/abs/1607.06450