Addressing Challenges in Large-scale Distributed AI Systems

Zhen Dong (UC Berkeley), Xiaoyong Liu (AliBaba), Kurt Keutzer (UC Berkeley)

1. Efficient Training on Large NLP Models

   In Alibaba's CFP and Brainstorm:

   "Giant DL models training: Lots of giant models training jobs are running in Alibaba's cluster including models for both dense and sparse ones."

2. Dynamic Strategy Adapting to Computation Resources

   "Training and Inference, Multi-strategy for Heterogeneous Targets."

   "For your system, it should be capable enough to select appropriate optimizer and sync/async gradient aggregator for training and inference."

Background and Previous Work:

The delays associated with training NNs are often the main bottleneck in model development and tuning, especially for large NLP models which can have over 175 billion parameters. In the data parallel approach, the model is generally replicated on multiple processors. During training, each process reads a subset of the dataset and performs the forward pass locally. During backpropagation, the gradient updates of the model parameters are then aggregated using an allreduce operation.

The communication overhead of these allreduce collectives has made it difficult to efficiently scale out the training using the synchronous SGD algorithm. As a result, an asynchronous SGD method may be used; however, this may result in suboptimal accuracy, especially when scaled to a large number of processors due to the stale-gradient problem. Another alternative is to quantize the communication to low precision. However, low bit communication often requires hyper-parameter tuning and may even lead to divergence in training. An orthogonal solution is to leverage sparsity in gradients to reduce communication volume. Existing work in the literature compresses communication by only sending gradients with large magnitude. However, those gradients are not necessarily more important than smaller ones, and the magnitude used in previous literature is a local metric, neglecting the difference of sensitivity levels among layers. In addition, a naive implementation can suffer from the overhead to support unstructured sparsity.

Our group has conducted many successful projects in this area. We are the academic pioneers in the acceleration of training by scaling the computations on distributed processors, for both computer vision [1][2][3] and NLP models [4]. For dynamic scheduling with memory constraints, we have demonstrated a general approach (Checkmate) [5] to manage very large models. In terms of model compression techniques, we have developed a systematic sensitivity analysis method that applies for both CV [6][7][8] and NLP [9]. We also demonstrated that, for NLP, large models can actually be easier to train [10], which makes training large models a key capability for the future.

Project Progress:

1. (1) We have developed a topology-aware structured communication (TASC) strategy that utilizes Hessian information from the loss function to determine the importance of gradient values. The core idea behind this approach is that the importance of gradient value is relative to the topology of the loss landscape. That is to say, a small value in a sharp landscape can be more important than a large value in a flat landscape. We conducted thorough experiments on CIFAR10/100 and ImageNet dataset, and we found our method works well for various computer vision models such as VGG16/19, ResNet18/50/101, etc. As an example, for Wide ResNet20 on CIFAR10, we can compress the communication by 130× and achieve higher accuracy of 93.27%, in contrast to the 92.57% achieved by previous state-of-the-art approach SparCML.
   (2) To verify the effectiveness of TASC, we also implemented it on general platforms such as Google Cloud, with 2 settings: (1) 1 node with 8 GPUs, (2) 2 nodes with 16 GPUs. We achieved a real speedup of 2.6× on the communication time cost when training a VGG19 model with the first setting. When we have multiple nodes such as in setting (2), the speedup increases to 11×, because TASC can significantly relieve the bottleneck of low transmission speed among different nodes.
   (3) With the help of Alibaba, we are currently trying to examine the feasibility of TASC on NLP, with much larger models such as Megatron, or potentially, RapidTransformer (internal NLP model from Abibaba). Specifically, the exact form of Hessian information needed, and the frequency of sensitivity calculation will be adjusted accordingly to make TASC computationally tolerable even with a tremendous parameter size.
   (4) We have developed a dynamic strategy of TASC that can adapt to different constraints of computation resources. This is beneficial for implementation on different clusters or platforms.
   (5) Finally, we developed a novel method to efficiently conduct allreduce operation with structured sparsity. We formed the optimization problem as an integer linear programming, and we proposed an efficient solver to dynamically generate communication schedules. Our method integrates the merits of Ring AllReduce and Butterfly AllReduce, which can potentially have up to 30× speedup over dense Ring AllReduce with an efficient implementation.
2. We developed a simple framework to save communication costs for recommendation systems. Specifically, our framework analyses the communication costs of arbitrary distributed systems, and we use it to propose algorithms that maximize throughput subject to memory, computation, and communication constraints. We implemented our framework and algorithms in PyTorch and achieved up to 3× increases in training throughput on GPU systems over baselines, on the Criteo Terabyte and Alibaba User Behavior datasets. We have written the preprint version of paper: SCARS: Stochastic Communication Avoidance for Recommendation Systems.

   Publications describing our progress are in preparation.

References:

1. Iandola, Forrest, Matthew Moskewicz, Khalid Ashraf, and Kurt Keutzer. "Firecaffe: near-linear acceleration of deep neural network training on compute clusters." In CVPR 2016.
2. You, Yang, Zhao Zhang, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. "Imagenet training in minutes." ICPP, 2017.
3. Gholami A, Azad A, Jin P, Keutzer K, Buluc A. Integrated model, batch, and domain parallelism in training neural networks. In Symposium on Parallelism in Algorithms and Architectures 2018.

4. You, Yang, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. "Large batch optimization for deep learning: Training bert in 76 minutes." In ICLR 2019.
5. Jain P, Jain A, Nrusimha A, Gholami A, Abbeel P, Keutzer K, Stoica I, Gonzalez JE. Checkmate: Breaking Memory Wall with Optimal Tensor Rematerialization. MLSys 2020
6. Dong, Zhen, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. "Hawq: Hessian aware quantization of neural networks with mixed-precision." In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019.
7. Dong, Zhen, Zhewei Yao, Yaohui Cai, Daiyaan Arfeen, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. "Hawq-v2: Hessian aware trace-weighted quantization of neural networks." Advances in Neural Information Processing Systems 33 (2020).
8. Cai, Yaohui, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. "Zeroq: A novel zero shot quantization framework." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13169-13178. 2020.
9. Shen, Sheng, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. "Q-bert: Hessian based ultra low precision quantization of bert." In AAAI 2020.
10. Li, Zhuohan, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joey Gonzalez. "Train Big, Then Compress: Rethinking Model Size for Efficient Training and Inference of Transformers." In ICML 2020.