Efficient Decentralized Training of DNNs

Zesen Wang, KTH Royal Institute of Technology
Division of Decision and Control Systems
Supervisor: Prof. Mikael Johansson and Prof. Paris Carbone

Motivation

The motivation for this research field stems from the faster development of computational power over communication bandwidth, which will make the cost of data movement the bottleneck of the distributed training. Big tech companies invest vast resources in dedicated HPC clusters with homogeneous computing units and , which makes AllReduce training feasible in large scale.

Decentralized algorithms, as a technique originally introduced for consensus averaging and privacy preservation, were shown to offer a promising solution up by significantly reducing communication overhead in sub-optimal network conditions.

Background

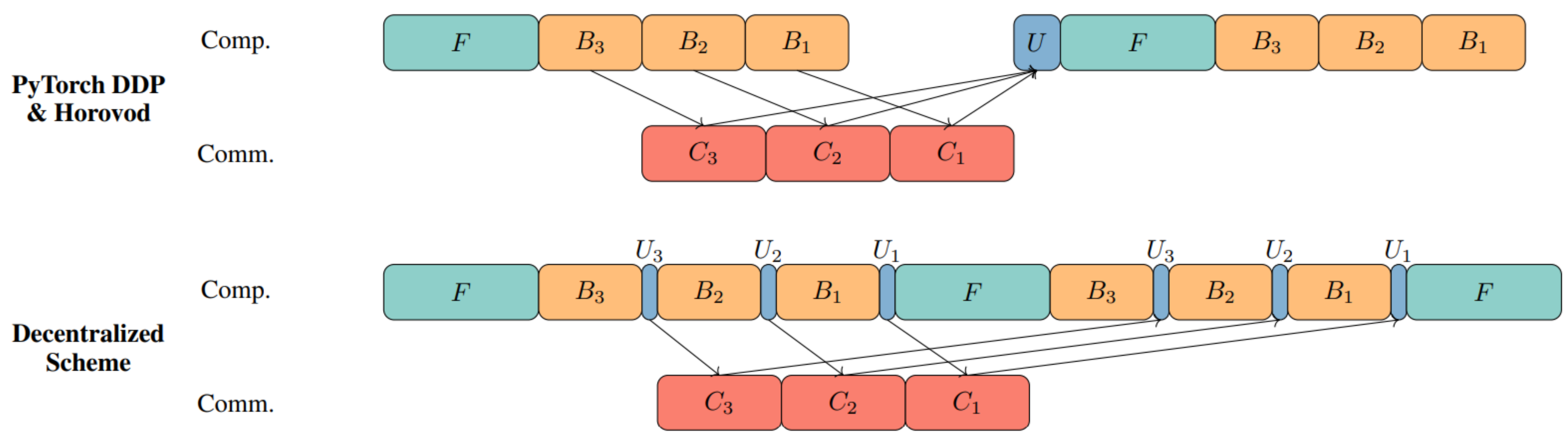
AllReduce Training

$$x_{t+1} = x_t - \alpha D\left(\frac{1}{N}\sum_{n=1}^{N} \nabla f(x_t; \xi_t^i)\right)$$

Decentralized Training

$$x_{t+1}^i = -\alpha D\left[\nabla f_i(x_t^i; \xi_t^i)\right] + \sum_{j=1}^{N} W_{i,j} x_t^j$$

$W \in M_N$ is a doubly stochastic matrix with non-negative entries



F: Forward pass, B_i: Backward pass for i-th bucket, U_i: Parameter update for i-th bucket, C_i: Parameter/Gradient communication for i-th bucket.

Problem

Drawbacks of AllReduce training:

- ❑ Costly global AllReduce Operation
  - ➢ Require strong synchronization across all workers
  - ➢ [DecentDP]: cheap decentralized communication
- ❑ Straggler from system noise or imbalanced workload
  - ➢ Always wait for the slowest workers
  - ➢ [DecentDP]: looser synchronization within neighbours
- ❑ Low utilization of heterogeneous network connections
  - ➢ Ignore the heterogeneous network connections with varying bandwidth and latency (NvLink and Ethernet, for example)
  - ➢ [DecentDP]: flexible decentralized communication patterns

Why not decentralized training?

- Incomplete convergence analysis for non-convex problems with adaptive momentums
- Only focus on the convergence properties and ignore the communication costs
- Algorithm design with good convergence properties may not imply good practical speedups
- Potential loss in generalization performance
- Need decentralized version of Adam(W) for large-scale training like LLMs

Preliminary Results

Convergence bound of decentralized Adam

Theorem 4.1. *Under Assumptions 1–3, if $0 < \beta_1 < \beta_2 < 1$, for Algorithm 1, we have*

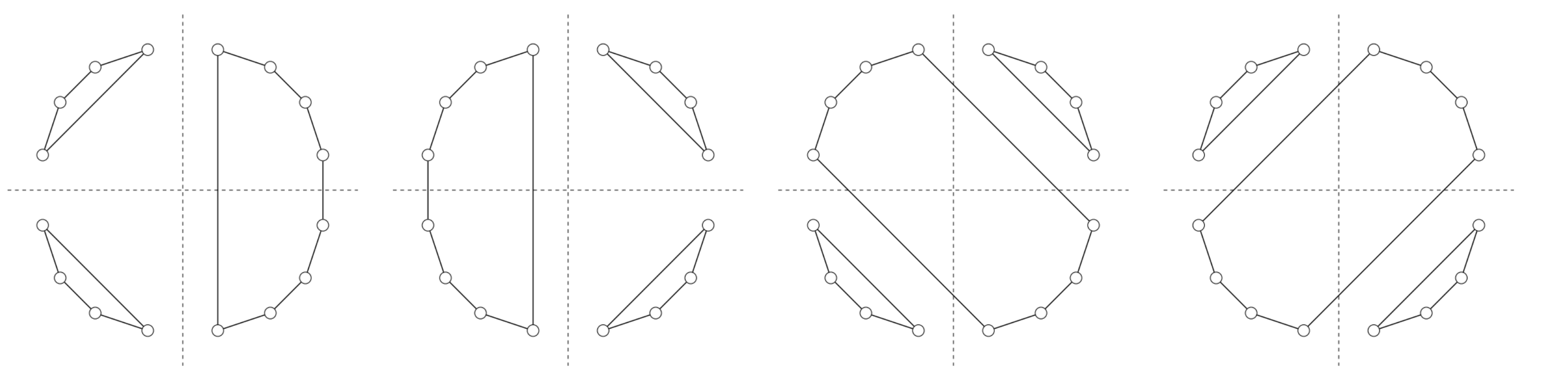
$$\mathbb{E}\left[\left\|\nabla F(\bar{x}^{(\tau)})\right\|_2^2\right] \le \frac{4R}{\alpha\tilde{T}}\left(F(\bar{x}^{(0)}) - F_*\right) + E\left[\frac{1}{\tilde{T}}\ln\left(1 + \frac{R}{\epsilon(1-\beta_2)}\right) - \frac{T}{\tilde{T}}\ln(\beta_2)\right], \qquad (5)$$

where $\bar{x}^{(t)} = \frac{1}{N}\sum_{i=1}^{N} x_i^{(t)}$, τ is defined in (4), *$\tilde{T} = T - \frac{\beta_1}{1-\beta_1}$, F_* is the optimal value of* (1), *and* $E = \frac{24DR^2\sqrt{1-\beta_1}}{\sqrt{1-\beta_2}(1-\beta_1/\beta_2)^{3/2}} + \frac{2\alpha DLR(1-\beta_1)}{(1-\beta_2)(1-\beta_1/\beta_2)} + \frac{4\alpha^2L^2D\beta_1}{(1-\beta_1/\beta_2)(1-\beta_2)^{3/2}} + \frac{8\alpha^2(1+\lambda^2)RL^2D\sqrt{1-\beta_1}}{(1-\lambda^2)^2(1-\beta_2)(1-\beta_1/\beta_2)\sqrt{\epsilon}}$.

Decentralized Adam (with the accumulation trick)

```
s ← # of iterations in one accumulation loop (for example, 4);
$\hat{m}_i^{(0)}, \hat{v}_i^{(0)} \leftarrow 0; x_i^{(0)} \leftarrow x^0; b_i \leftarrow 0$
for $t = 1, 2, \ldots, T$ do
    $\hat{t} \leftarrow \lceil t/s \rceil$
    $g_i^{(t)} \leftarrow \nabla \ell(x_i^{(t-1)}; \xi_i^{(t)})$
    $m_i^{(t)} \leftarrow \beta_1 \hat{m}_i^{(\hat{t}-1)} + (1-\beta_1) g_i^{(t)}$
    $v_i^{(t)} \leftarrow \beta_2 \hat{v}_i^{(\hat{t}-1)} + (1-\beta_2)\left[g_i^{(t)}\right]^2$
    $x_i^{(t)} \leftarrow -\alpha \frac{m_i^{(t)}/(1-\beta_1^{\hat{t}})}{\sqrt{v_i^{(t)}/(1-\beta_2^{\hat{t}})+\epsilon}} + \sum_{j \in \mathcal{N}_i^{(t)}} w_{ij}^{(t)} x_j^{(t-1)}$
    $b_i \leftarrow b_i + \frac{1}{s} g_i^{(t)}$
    if $t \mod s == 0$ then
        $\hat{m}_i^{(\hat{t})} \leftarrow \beta_1 \hat{m}_i^{(\hat{t}-1)} + (1-\beta_1) b_i$
        $\hat{v}_i^{(\hat{t})} \leftarrow \beta_1 \hat{v}_i^{(\hat{t}-1)} + (1-\beta_1)[b_i]^2$
        $b_i \leftarrow 0$
    end if
end for
```

Alternating Exponential Graph



Speedup

Task / Model	Node Setup / Inter-connection	Topology	Total Runtime / Relative Speedup	Feature
Translation En-De / Transformer-base	4×4×A40 / 25Gbps Ether.	Complete Ring AER	5.80 hrs. / 22.03% 3.76 hrs. / 49.35% 2.90 hrs. / 61.05%	Comm-bound / High Comp. Var.
Translation En-Fr / Transformer-big	2×4×A100 / 100Gbps Infini.	Complete Ring AER	11.45 hrs. / 10.61% 10.92 hrs. / 14.82% 10.89 hrs. / 15.01%	Comp-bound / High Comp. Var.
Image Classification / ResNet-50	4×8×T4 / 100Gbps Infini.	Complete AER	9.80 hrs / 5.41% 9.70 hrs / 6.46%	Comp-bound / Low Comp. Var.

Generalization Performance (NMT with transformer model)

Methods	BLEU Score / Val Loss / Train Loss		
	Topology		
	Ring	Alternating Exp Ring	Complete
AllReduce		27.66 / 3.042 / 3.063	
DAdam	27.63 / 3.043 / 3.063	27.57 / 3.038 / 3.057	27.61 / 3.039 / 3.059
AccumAdam	**27.76 / 3.035 / 3.048**	**27.80 / 3.032 / 3.043**	**27.62 / 3.031 / 3.044**