Ring-Star: A Sparse Topology for Faster Model Averaging in Decentralized Parallel SGD

Mohsan Jameel, Josif Grabocka, Mofassir ul Islam Arif, and Lars Schmidt-Thieme

Information Systems and Machine Learning Lab,
University of Hildesheim, Universittsplatz 1, 31141 Hildesheim, Germany
{mohsan.jameel,josif,mofassir,schmidt-thieme}@ismll.uni-hildesheim.de

Abstract. In decentralized distributed systems the data resides on the compute devices, which are connected through a high latency network that can adversely impact the communication cost. In such systems, it is desirable to employ a training regime that is inherently decentralized, where learning algorithms operate on local hosts using only the local data partitions. To ensure their convergence to a joint model, the parameters of the local models have to be regularly averaged. As each averaging operation incurs network communication costs, the right balance has to be found between either communication intensive dense averaging operations or sparse averaging operations which slows down the convergence. We propose a hierarchical two-layer sparse communication topology, a ring of fully-connected meshes of workers that communicate with each other (*Ring-Star*). *Ring-Star* allows a principled trade-off between the convergence speed and communication overhead and is well suited to loosely coupled distributed systems. We demonstrate on an image classification task and a batch stochastic gradient descent learning (SGD) algorithm that our proposed method shows similar convergence behavior as *Allreduce* while having lower communication cost of *Ring*.

Keywords: Decentralize Sparse Topology · Model Averaging · Distributed Stochastic Gradient Descent · Deep Learning

1 Introduction

Mini-batch Stochastic Gradient Descent is often employed for training deep learning models in distributed settings, as each instance of data can be processed in parallel, which is useful in speeding up the learning process. The most widely used distributed learning approaches focus mainly on using centralized training procedures [4,7], which are based on a parameter server (PS) framework. However, the centralized approaches are not suited for the computing environment, where data cannot be centralized and the central server can become a bottleneck due to the underlying network characteristics [8]. Decentralized training procedures [8,9] are proposed to scale on loosely connected, high latency computing systems. In these procedures, workers are sparsely connected to each other forming a *Ring* topology. For synchronization, each worker averages its model with

two neighboring workers. Decentralized approaches are motivated by control systems and wireless sensor network research, which solve a global consensus problem. These procedures show a significant reduction in the communication overhead. However, sparse averaging increases the parameter variance between workers, which is termed as "network error" in the literature [1,13]. The "network error" or variance is large in the early stages of optimizing a non-convex objective and frequent averaging helps to reduce the variance. Despite being communication efficient, decentralized training procedures suffer from high network error, which increases with an increasing number of workers. On the other hand, a grand averaging step, like *Allreduce* [2], incurs zero network error but is a communication inefficient operation, especially in a high latency network.

The competing objectives of reducing communication overhead, while keeping the network error as small as possible is a challenging task, which requires designing a topology that benefits from both worlds. In this paper we analyze different characteristics of decentralized topologies and design a sparse topology that balances trade-off between communication cost and network error. The main contributions of this paper are 1) a new *Ring-Star* topology for a decentralized parallel SGD that balances network error and communication overhead, and 2) detailed analysis of different design choices for designing a sparse topology. The empirical evaluations on an image classification task show, superior convergence behavior of *Ring-Star* as compared to communication efficient *Ring* based topologies. As a result *Ring-Star* achieves better final test accuracy than *Ring* and *RingRandom* in same wall clock time.

2 Decentralized Model Averaging

2.1 Problem Formulation

Decentralized distributed settings consist of a set of distributed workers $\mathcal{V} = \{1, \cdots, V\}$, where each worker $v \in \mathcal{V}$ holds a local model $\hat{y}(\mathbf{x}; \theta_v)$, with model parameters $\theta_v \in \mathbb{R}^K$ and runs a mini-batch SGD to update its model parameters by sampling a mini-batch $\mathcal{B}_v \subset \mathcal{D}_v$ from the local shard of data $\mathcal{D}_v$.

$$\theta_v^{t+1} = \theta_v^t - \eta \frac{1}{|\mathcal{B}_v|} \sum_{(\mathbf{x},y)\in\mathcal{B}_v} \nabla \mathcal{L}(y, \hat{y}(\mathbf{x}; \theta_v^t)) \tag{1}$$

where $\mathcal{L}(\cdot, \cdot)$ is a loss function. Typical loss functions include the cross entropy loss, square loss, hinge loss etc. These workers periodically synchronize their models by averaging over the models learned by other workers. Given a weight matrix $\mathbf{W} \in \mathbb{R}^{V\times V}$, the averaging step at worker v can be defined as:

$$\bar{\theta}_v^t = \sum_{v'\in\mathcal{V}} \mathbf{W}_{v,v'} \theta_{v'}^t \tag{2}$$

The weight matrix (or mixing matrix) $\mathbf{W}$ is symmetric and $\mathbf{W1} = \mathbf{1}$, where $\mathbf{1}$ denotes a vector of all ones. The weight matrix $\mathbf{W}$ defines the influence of

the averaging step in Eq.(2) at each worker. In a dense averaging scheme, such as *Allreduce* [2], each worker gets the model average of all other workers at each averaging step, whereas in a sparse scheme, such as *Ring* [8], each worker averages over two neighboring workers. The weight matrices for *Allreduce* and *Ring* schemes are given as:

$$\mathbf{W}_{Allreduce} = \begin{pmatrix} \frac{1}{V} & \cdots & \frac{1}{V} \\ \vdots & \ddots & \vdots \\ \frac{1}{V} & \cdots & \frac{1}{V} \end{pmatrix}, \mathbf{W}_{Ring} = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & 0 & \cdots & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & \cdots & \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & \cdots \\ & \ddots & \ddots & \ddots & & \\ & \cdots & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 0 & \cdots & 0 & \frac{1}{3} & \frac{1}{3} \end{pmatrix}$$

2.2 Designing a Sparse Topology

To design a sparse topology that incurs a lower communication cost and at the same time has lower network error, we define the average-age matrix and the communication overhead as design characteristics of the topology. The average-age matrix $\mathbf{H}$ holds the information about how old the contribution of a worker u is to a worker v. Given the weight matrix $\mathbf{W}$, the average-age matrix $\mathbf{H}$ can be defined as the shortest path between the workers, which measures the number of averaging steps required to average over the model from any other worker. The second important characteristic of topology is the communication cost, which can be defined, following [11], as $\Upsilon\alpha + \Pi\beta$, where Υ is the number of handshakes, α is the latency, Π is the size of data transferred, and β is the bandwidth. The importance of latency in high latency networks cannot be understated as it can cause a performance bottleneck.

Table 1. Comparison of characteristics of different topologies.

Topology	**Averaging Step**	**Communication Cost**	**Average Age**
Allreduce [2]	Dense	$2(V-1)\alpha + 2K\beta$	$O(1)$
Ring [8]	Sparse	$2\alpha + 2K\beta$	$O(V)$
RingRandom [9]	Sparse	$2\alpha + 2K\beta$	$O(log(V))$
Ring-Star (proposed)	Sparse	$(2(L-1)+2)\alpha + 2K\beta$	$O(G)$ or $O(log(G))$

2.3 Existing Topologies

The *Allreduce* (AR) topology [2] is a dense averaging scheme used for training deep learning models. In this scheme, every worker requires a single averaging step to get the contributions of all other workers, therefore the age matrix $\mathbf{H}$ has one. A disadvantage, however, inherent to this topology is the high communication cost, which for the most optimized implementation still requires $O(V)$ handshakes. The total communication cost incurred by *Allreduce* is $(V-1)\alpha + 2K\beta$, which becomes more pronounced in high latency network as latency grows in V, where V is the number of workers.

The *Ring* (R) topology proposed in [8] has a sparse averaging scheme, where at an averaging step each worker only averages with its two adjacent neighboring workers. This sparse connectivity incurs a very low communication cost of $2\alpha + 2K\beta$ per communication round. However, due to a sparse averaging, a worker on average has to take $O(V)$ averaging steps before it gets the contribution from

its furthest neighbor, which causes high network error, requiring more iterations for a model to converge.

The *RingRandom* (RR) topology proposed in [9], improves the averaging steps by averaging randomly with a neighbor that is $2^i + 1$ hops away, where i is an integer between 0 and $log(V) - 1$. They also introduce a bipartite partitioning of the workers, where workers in an *active* group initiate the communication, whereas a *passive* group worker only responds to the request. These random re-links connect any pair of workers in $O(log(V))$ steps. The communication overhead is the same as the *Ring* topology, i.e. $2\alpha + 2K\beta$ per communication round.

2.4 *Ring-Star*: A Sparse Topology

The existing topologies discussed above either incur a high communication overhead or suffer from a low averaging operation which results in a high network error. Keeping in view these characteristics, we propose the *Ring-Star* topology that aims to reduce their disadvantages. In our proposed *Ring-Star* (RS) topology, distributed workers are divided into local groups and a worker from each group is selected as a *Delegate*. The *Delegate* is responsible for averaging models from the local group as well as exchange the group average with two neighboring *Delegates* (similar to a *Ring* topology). After the averaging step, each worker in the connected group gets the average of the two neighboring groups. Let the size of the local group be L then the size of the *Delegates Ring* becomes $G = V/L$. This significantly reduces the average age in $\mathbf{H}$, as each worker requires $O(G)$[1] averaging steps to get contribution from the furtherest worker, and speeds-up the information propagation among workers. *Ring-Star* incurs $((2(L-1)+2)\alpha+2K\beta$ communication cost, where $O(L)$ handshakes are required for local group averaging and two more handshakes are required for averaging between two *Delegates*. *Ring-Star* is a sparse topology, in which, after the averaging step each worker gets the contribution of a subset of workers, and it has a significantly different communication pattern from dense *Allreduce* [6,2], where after the averaging step each worker gets the contribution from all other workers. The characteristics of *Ring-Star* and other topologies are summarized in Table 1.

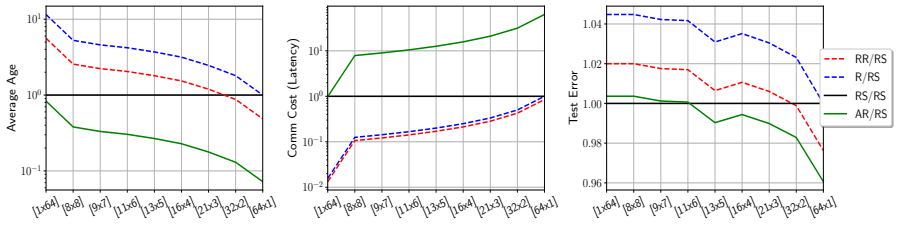


Fig. 1. Analysis of different group configurations for *Ring-Star*.

The group configuration of *Ring-Star*, i.e. $[G \times L]$, controls the sparsity in $\mathbf{W}$ and is a tunable parameter. We designed an experiment by training Resnet20 on the CIFAR10 dataset using 64 workers to analyze the effect of choosing the

[1] replacing *Ring* for *Delegates* with *RingRandom* will give $O(log(G))$.

$[G \times L]$ on average age, communication cost and final test accuracy, and we have used the "Relative Gain"[2] $\in \mathbb{R}^+$ to compare *Ring-Star* and other topologies. Figure 1 shows that *Ring-Star* has better "Relative Gain" in average age over *Ring* and *RingRandom*, whereas it has a lower communication cost as compared to *Allreduce*. The test error of *Ring-Star* is also lower than *Ring* and *RingRandom* across different group configurations. It is also shown that choosing $L = 1$ retrieves the *Ring* topology and $L = V$ retrieves the *Allreduce* topology.

3 Experiments

In this section, we empirically investigate the effect of sparse topologies on the decentralized training of deep convolution neural networks for an image classification task. We selected well-known CIFAR10 and CIFAR100 as evaluation datasets for our experiments, which consists of 32x32 color images with 10 and 100 classes respectively and split into 50K train-set and 10K test-set. The deep learning models and hyperparameters for our experiments are summarized in Table 2. The models are implemented in PyTorch and the distributed framework

Table 2. Hyperparameters for Experiments

Dataset	**Model**	**batch_size**[3]	**lr**	**lr_schedule**	**lr_decay**	**size**
CIFAR10	Resnet20 [3]	32	0.1	{81, 122}	0.1	1MB
	VGG16 [10]	64	0.1	{25, 50, 75, 100}	0.5	60MB
CIFAR100	DensNet-40-12 [5]	64	0.1	{150, 225}	0.1	1MB
	WideResnet-28-10 [12]	64	0.1	{60, 120, 160}	0.2	146MB

is implemented using mpi4py. The experimental setup consists of nodes on the Google Cloud Platform (GCP), where each node is a "n1-standard-64" instance with Intel Xeon E5 v3 (Haswell) 64 vCPUs, 240 GB of memory, 1000GB SSD storage, and 4 Nvidia P100 GPUs. The nodes are connected through a 10Gbit/s Ethernet interconnect.

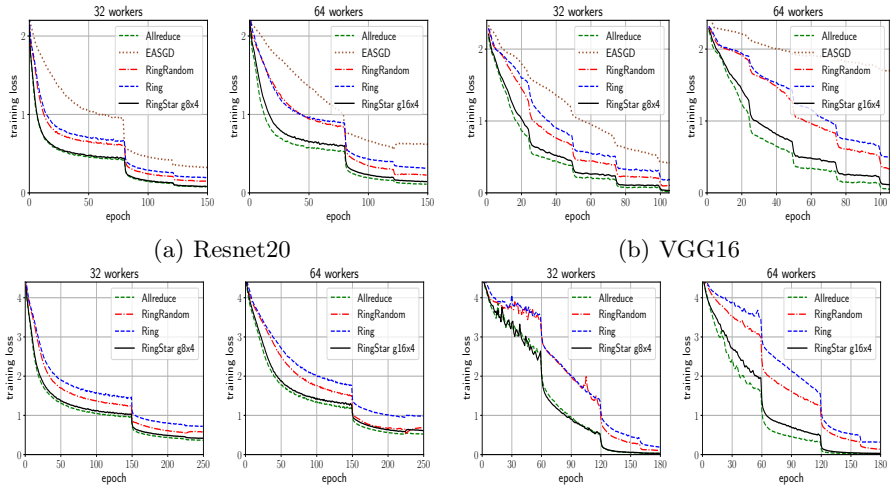
3.1 Convergence behavior of difference topologies with respect to epochs

Experiments on CIFAR10: We looked at the convergence behaviors of different topologies on the CIFAR10 dataset by varying the number of workers. Figures 2(a) and 2(b) summarize the results on the CIFAR10 datasets for Resnet20 and VGG16 respectively. The *Allreduce* and *Ring-Star* consistently show better performance across both the models. It can be seen that *Ring-Star* learning curves follow closely the *Allreduce* learning curves. The impact of fast averaging over all the workers becomes more pronounced as the number of workers is increased. The more sparsely connected workers in *Ring* and *RingRandom* have more divergence among the local models, and they tend to

[2] "Relative Gain" is a ratio between *Ring-Star* and other topologies, i.e. $\frac{AR}{RS} > 1$ indicates *Ring-Star* is better than *Allreduce* and $\frac{AR}{RS} < 1$ indicates otherwise.

[3] the warmup learning rate scaling technique as described in [2] is employed for stabilizing the learning process for large batch sizes i.e $B_{\text{global}} = V \times B$.

converge to the worst local optima. To overcome this issue, Lian et al. [9] decreased the learning rate for *Ring* and *RingRandom* earlier than *Allreduce* in their experiments for the number of workers ≥ 32 to stabilize the optimization. The final test accuracies in Table 3 also show a similar trend, where *Allreduce* and *Ring-Star* achieved the best test accuracy with minimum effect of increasing the number of workers. *EASGD* [14] performs the worst among all methods.



(a) Resnet20 (b) VGG16

(c) DensNet-40-12 (d) WideResnet-28-10

Fig. 2. Epoch-wise convergence behavior of different topologies on the CIFAR10 (a-b) and CIFAR100 (c-d) training using 32 and 64 workers.

Experiments on CIFAR100: In this section, we present results on the CIFAR100 dataset. In these experiments, we choose complex workloads i.e. DensNet-40-12 and WideResnet-28-10. Figures 2(c) and 2(d) summarize the results on the CIFAR100 datasets for DensNet-40-12 and WideResnet-28-10 respectively. The results show similar trends in the learning curves as for the CIFAR10 dataset. The hybrid *Ring-Star* is shown to perform at par with *Allreduce* in terms of final test accuracy (Table 4) as well as speed of convergence, whereas *Ring* and *RingRandom* suffers from a slow averaging step, which leads to slower learning.

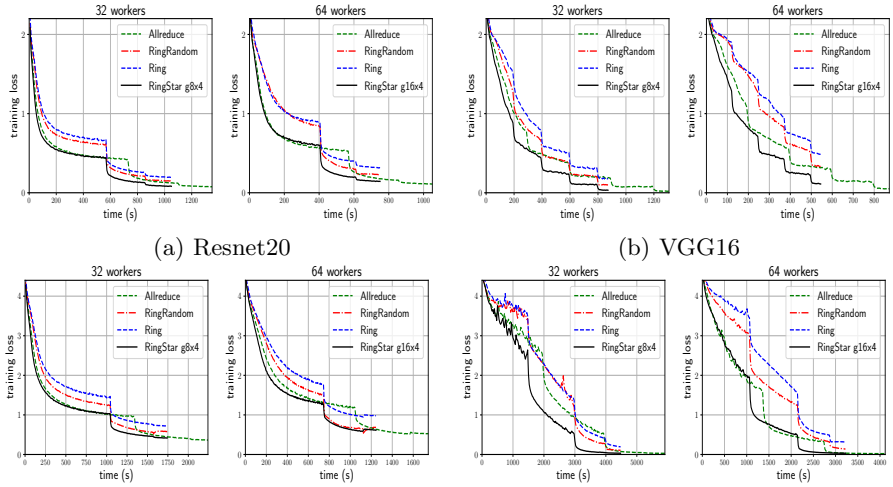
Table 3. Comparison of test accuracy for the CIFAR10 experiments.

Model	**Workers**	***Allreduce***	***Ring-Star***	***RingRandom***	***Ring***	***EASGD* [14]**
Resnet20	16 [4x4]	91.98%	91.93%	91.68%	91.59%	90.76%
	32 [8x4]	91.58%	91.42%	90.82%	90.70%	86.68%
	64 [16x4]	90.90%	90.50%	89.44%	87.32%	81.32%
VGG16	16 [4x4]	91.89%	91.57%	91.61%	91.43%	89.323%
	32 [8x4]	91.77%	91.44%	90.19%	89.71%	83.726%
	64 [16x4]	91.47%	91.25%	88.74%	86.04%	-

3.2 Convergence behavior of difference topologies with respect to time

In the second set of experiments, we analyze the convergence speed with respect to time. The comparisons of convergence with respect to time is presented in

Figures 3(a) and 3(b) for Resnet20 and VGG16 trained on CIFAR10, and Figures 3(c) and 3(d) for DensNet-40-12 and WideResnet-28-10 trained on CIFAR100. The effect of communication is clearly visible, as *Allreduce* requires more time to converge due to high communication overhead. The communication efficient *Ring* and *RingRandom* show better communication behavior and require less amount of time to finish training. However, due to their slow averaging step, they still need more epochs to converge to a similar loss as *Allreduce*. *Ring-Star* on the other hand enjoys superior communication behavior and converges to the lowest loss in less amount of time. *Ring-Star* shows similar communication requirements as *Ring* and *RingRandom*, while achieving a similar solution as a more accurate, but communication inefficient *Allreduce*.



(a) Resnet20 (b) VGG16

(c) DensNet-40-12 (d) WideResnet-28-10

Fig. 3. Time-wise convergence behavior of different topologies on the CIFAR10 (a-b), and CIFAR100 (c-d) training using 32 and 64 workers.

Table 4. Comparison of test accuracy for the CIFAR100 experiments.

Model	**Workers**	***Allreduce***	***Ring-Star***	***RingRandom***	***Ring***
DensNet-40-12	16 [4x4]	71.59%	71.61%	71.11%	71.09%
	32 [8x4]	71.31%	71.24%	69.37%	67.91%
	64 [16x4]	71.25%	71.19%	68.70%	66.01%
WideResnet-28-10	16 [4x4]	78.86%	78.73%	78.49%	78.10%
	32 [8x4]	78.26%	78.31%	77.18%	76.37%
	64 [16x4]	78.15%	78.20%	76.23%	74.77%

4 Conclusion

In this paper we address the design choices for a sparse model averaging strategy in a decentralized parallel SGD. The detailed analysis of different topologies show the importance of averaging age, communication overhead and variance among workers, and how it could effect the overall learning behavior of the deep learning

model. We propose a hierarchical two-layer sparse communication topology, a ring of fully-connected meshes of workers that communicate with each other (*Ring-Star*). *Ring-Star* allows a principled trade-off between convergence speed and communication overhead and is well suited to loosely coupled distributed systems. We demonstrate on an image classification task and a batch stochastic gradient descent learning (SGD) algorithm that our proposed method shows similar convergence behavior as *Allreduce* while having lower communication cost of *Ring*.

References

1. Bijral, A.S., Sarwate, A.D., Srebro, N.: Data-dependent convergence for consensus stochastic optimization. IEEE Transactions on Automatic Control **62**(9), 4483–4498 (2017)
2. Goyal, P., Dollár, P., Girshick, R.B., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., He, K.: Accurate, large minibatch SGD: training imagenet in 1 hour. CoRR **abs/1706.02677** (2017)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR'2016. pp. 770–778 (2016)
4. Ho, Q., Cipar, J., Cui, H., Lee, S., Kim, J.K., Gibbons, P.B., Gibson, G.A., Ganger, G., Xing, E.P.: More effective distributed ml via a stale synchronous parallel parameter server. In: NIPS'2013. pp. 1223–1231 (2013)
5. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR'2017. pp. 2261–2269 (2017)
6. Jia, X., Song, S., He, W., Wang, Y., Rong, H., Zhou, F., Xie, L., Guo, Z., Yang, Y., Yu, L., Chen, T., Hu, G., Shi, S., Chu, X.: Highly scalable deep learning training system with mixed-precision: Training imagenet in four minutes. CoRR **abs/1807.11205** (2018)
7. Li, M., Andersen, D.G., Smola, A.J., Yu, K.: Communication Efficient Distributed Machine Learning with the Parameter Server. In: NIPS'2014, pp. 19–27 (2014)
8. Lian, X., Zhang, C., Zhang, H., Hsieh, C.J., Zhang, W., Liu, J.: Can Decentralized Algorithms Outperform Centralized Algorithms? A Case Study for Decentralized Parallel Stochastic Gradient Descent. In: NIPS'2017, pp. 5330–5340 (2017)
9. Lian, X., Zhang, W., Zhang, C., Liu, J.: Asynchronous decentralized parallel stochastic gradient descent. In: ICML'2018. pp. 3049–3058 (2018)
10. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)
11. Thakur, R., Rabenseifner, R., Gropp, W.: Optimization of collective communication operations in mpich. Int. J. High Perform. Comput. Appl. **19**(1), 49–66 (2005)
12. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: Proceedings of the British Machine Vision Conference (BMVC). pp. 87.1–87.12 (2016)
13. Zhang, J., Sa, C.D., Mitliagkas, I., Ré , C.: Parallel sgd: When does averaging help. In: Optimization in Machine Learning Workshop ICML (2016)
14. Zhang, S., Choromanska, A.E., LeCun, Y.: Deep learning with Elastic Averaging SGD. In: NIPS'2015, pp. 685–693 (2015)