

DOCTORAL THESIS

Communication optimizations for distributed deep learning

Shi, Shaohuai

Date of Award:
2020

[Link to publication](#)

General rights

Copyright and intellectual property rights for the publications made accessible in HKBU Scholars are retained by the authors and/or other copyright owners. In addition to the restrictions prescribed by the Copyright Ordinance of Hong Kong, all users and readers must also observe the following terms of use:

- Users may download and print one copy of any publication from HKBU Scholars for the purpose of private study or research
- Users cannot further distribute the material or use it for any profit-making activity or commercial gain
- To share publications in HKBU Scholars with others, users are welcome to freely distribute the permanent URL assigned to the publication

Abstract

With the increasing amount of data and the growing computing power, deep learning techniques using deep neural networks (DNNs) have been successfully applied in many practical artificial intelligence applications. The mini-batch stochastic gradient descent (SGD) algorithm and its variants are the most widely used algorithms in training deep models. The SGD algorithm is an iterative algorithm that needs to update the model parameters many times by traversing the training data, which is very time-consuming even using the single powerful GPU or TPU. Therefore, it becomes a common practice to exploit multiple processors (e.g., GPUs or TPUs) to accelerate the training process using distributed SGD. However, the iterative nature of distributed SGD requires multiple processors to iteratively communicate with each other to collaboratively update the model parameters. The intensive communication cost easily becomes the system bottleneck and limits the system scalability.

In this thesis, we study the communication-efficient techniques for distributed SGD to improve the system scalability and thus accelerate the training process. We identify the performance issues in distributed SGD through benchmarking and modeling and then propose several communication optimization algorithms to address the communication issues.

First, we build a performance model with a directed acyclic graph (DAG) to modeling the training process of distributed SGD and verify the model with extensive benchmarks on existing state-of-the-art deep learning frameworks including Caffe, MXNet, TensorFlow, and CNTK. Our benchmarking and modeling point out that existing optimizations for the communication problems are sub-optimal, which we need to address in this thesis.

Second, to address the startup problem (due to the high latency of each communication) of layer-wise communications with wait-free backpropagation (WFBP), we propose an optimal gradient merging solution for WFBP, named MG-WFBP, that exploits the layer-wise property to well overlap the communication tasks with the computing tasks and can be adaptive to the training environments. Experiments are conducted on dense-GPU clusters with Ethernet and InfiniBand, and the results show that MG-WFBP can well address the startup problem in distributed training

of layer-wise structured DNNs.

Third, to make the high computing-intensive training tasks be possible in GPU clusters with low-bandwidth interconnect, we investigate the gradient compression techniques in distributed training. The top- k sparsification can well compress the communication traffic with little impact on the model convergence, but it suffers from a linear communication complexity to the number of workers so that top- k sparsification cannot scale well in large-scale clusters. To address the problem, we propose a global top- k (gTop- k) sparsification algorithm that reduces the communication complexity to be logarithmic to the number of workers. We also provide detailed theoretical analysis for the gTop- k SGD training algorithm, and the theoretical results show that our gTop- k SGD has the same order of convergence rate with SGD. Experiments are conducted on up to 64-GPU cluster to verify that gTop- k SGD significantly improves the system scalability with only a slight impact on the model convergence.

Lastly, to enjoy the both benefits of the pipelining technique and the gradient sparsification algorithm, we propose a new distributed training algorithm, layer-wise adaptive gradient sparsification SGD (LAGS-SGD), which supports layer-wise sparsification and communication, and we theoretically and empirically prove that the LAGS-SGD preserves the convergence properties. To further alliterate the impact of the startup problem of layer-wise communications in LAGS-SGD, we also propose the optimal gradient merging solution for LAGS-SGD, named OMGS-SGD, and theoretical prove its optimality. The experimental results on a 16-node GPU cluster connected 1Gbps Ethernet show that OMGS-SGD can always improve the system scalability while the model convergence properties are not affected.

Keywords: Deep Learning, Distributed Deep Learning, Communication Optimizations, SGD, Distributed SGD, Gradient Sparsification

Table of Contents

Declaration	i
Abstract	ii
Acknowledgements	iv
Table of Contents	vi
List of Tables	x
List of Figures	xi
Chapter 1 Introduction	1
1.1 Deep Learning	1
1.2 Parallelisms of Distributed Deep Learning	2
1.3 Thesis Goals and Contributions	4
1.4 Organization	5
Chapter 2 Related Work	7
2.1 Optimization Algorithms	8
2.1.1 Large-Batch Training with BSP	8
2.1.2 Lossy Algorithms: Communication Synchronization	9
2.1.3 Lossy Algorithms: Communication Compression	10
2.1.4 Lossless Algorithms: Scheduling	11
2.2 System Architectures	12
2.2.1 Parameter Server	13
2.2.2 All-to-all	13
2.3 Communication Infrastructures	14
2.3.1 Communication Protocols	14
2.3.2 Network Topologies	14
2.4 Summary	15

Chapter 3	Benchmarking and Modeling	16
3.1	Introduction	16
3.1.1	Single-GPU Training	16
3.1.2	Multi-GPU Training	18
3.2	Preliminaries	19
3.2.1	Mini-batch SGD	19
3.2.2	S-SGD	19
3.3	Performance Modeling	20
3.3.1	A DAG Model	20
3.3.2	Optimization opportunities	22
3.4	Experimental Methodology	24
3.5	Experimental Results and Analysis	26
3.5.1	Results on Single GPU and CPU	27
3.5.2	Single GPU	27
3.5.3	Multiple GPUs	29
3.5.4	Multiple machines	31
3.6	Summary	34
Chapter 4	Pipelining Communications with Computations	35
4.1	Introduction	35
4.2	Preliminaries	36
4.2.1	Mini-batch SGD	36
4.2.2	Synchronized SGD	38
4.2.3	WFBP-SGD	39
4.2.4	Single-Layer S-SGD	40
4.2.5	Communication Model	41
4.3	Problem Formulation	42
4.4	Solution: MG-WFBP	45
4.4.1	Theoretical Analysis	46
4.4.2	Algorithms	49
4.5	System Implementation	51
4.5.1	Time Measurement of Backward Propagation	52
4.5.2	Parallelism between Gradient Computation and Aggregation	52
4.5.3	Efficient Gradient Merging	53
4.6	Experimental Studies	53
4.6.1	Experimental Settings	53
4.6.2	Measurement of All-reduce Communication	54
4.6.3	Real-world Experiments	55

4.6.4	Simulation	59
4.7	Summary	60
Chapter 5 Gradient Sparsification		62
5.1	Introduction	62
5.2	Preliminaries	65
5.2.1	DenseAllReduce	65
5.3	The Algorithm of gTop- k S-SGD	67
5.3.1	Observations from Top- k sparsification	68
5.3.2	The key idea of gTop- k	68
5.3.3	Communication-Efficient Collective	68
5.3.4	Communication Complexity Analysis	71
5.3.5	The Algorithm of gTop- k S-SGD	72
5.4	Convergence Analysis	72
5.4.1	Notations and Assumptions	73
5.4.2	Main Theoretical Results	74
5.4.3	Main Proofs	75
5.5	Experimental Results	79
5.5.1	Experimental Settings	79
5.5.2	Verification of Assumption 5.4.1 and Convergence	81
5.5.3	Scalability	82
5.5.4	Discussion	83
5.6	Summary	84
Chapter 6 Layer-wise Adaptive Gradient Sparsification		85
6.1	Introduction	85
6.2	Preliminaries	87
6.2.1	Pipelining on distributed SGD	87
6.2.2	TopK-SGD	89
6.2.3	LAGS-SGD: Pipelining on TopK-SGD	89
6.2.4	Top- k Selection on GPUs	90
6.2.5	Communication Model	90
6.3	Convergence Analysis of Layer-wise Sparsification	91
6.3.1	Algorithm	91
6.3.2	Convergence Analysis	92
6.4	Problem Formulation	98
6.5	Solution	100
6.5.1	Theoretical Analysis	100

6.5.2	Algorithm	104
6.6	Evaluation	105
6.6.1	Experimental Settings	105
6.6.2	Performance Models	106
6.6.3	Convergence Performance	107
6.6.4	Iteration Time	108
6.6.5	Time Breakdown	109
6.7	Summary	110
Chapter 7 Conclusion and Future Work		111
7.1	Conclusion	111
7.2	Future Research Directions	112
7.2.1	Synchronization Mechanism	112
7.2.2	Relaxed Synchronization and Communication Compression . .	113
7.2.3	Compression Level	113
7.2.4	Generic Scheduling	113
7.2.5	More Efficient System Architecture	114
7.2.6	Network Topology for More Efficient Collectives	114
Bibliography		115
Curriculum Vitae		128