Fast Distributed Deep Learning via Worker-adaptive Batch Sizing

Chen Chen[1]*, Qizhen Weng[1], Wei Wang[1], Baochun Li[2], Bo Li[1]
[1] Hong Kong University of Science and Technology [2] University of Toronto
*

ABSTRACT

In heterogeneous or shared clusters, distributed learning processes are slowed down by straggling workers. In this work, we propose LB-BSP, a new synchronization scheme that eliminates stragglers by adapting each worker's training load (batch size) to its processing capability. For training in shared production clusters, a prerequisite for deciding the workers' batch sizes is to know their processing speeds before each iteration starts. To this end, we adopt NARX, an extended recurrent neural network that accounts for both the historical speeds and the driving factors such as CPU and memory in prediction.

CCS CONCEPTS

• **Computing methodologies** → *Distributed algorithms*; **Cooperation and coordination**; *Distributed artificial intelligence*;

KEYWORDS

Distributed deep learning, load balancing, batch size

ACM Reference Format:
Chen Chen, Qizhen Weng, Wei Wang, Baochun Li, Bo Li. 2018. Fast Distributed Deep Learning, via Worker-adaptive Batch Sizing. In *ACM Symposium on Cloud Computing (SoCC '18), October 11–13, 2018, Carlsbad, CA, USA*, 1 page. https://doi.org/10.1145/3267809.3275463

1 MOTIVATION

Deep learning (DL) models are often trained under the Parameter Server architecture [4], with multiple workers iteratively refining the model parameters using subsets of training data, i.e., *sample batches*. With the prevalence of DL, there emerge many circumstances to train DL models with *heterogeneous* or *shared* clusters [3], where workers running on less capable computing resources would progress slower and become *stragglers*.

Stragglers slow down the distributed learning process, by either prolonging the duration of each iteration (low *hardware efficiency*), or alternatively depending on the mechanism used for worker coordination, requiring more iterations for DL models to converge (low *statistical efficiency*). For example, a typical coordination scheme, Bulk Synchronous Parallel (BSP), enforces workers to synchronize at the end of each iteration, and this suffers from low hardware efficiency because fast workers have to wait for the slower ones.

For fast distributed DL in heterogeneous scenarios, we need to eliminate stragglers by balancing workers' loads based on their computing capabilities. Nonetheless, existing load balancing solutions [2] bring non-negligible computation/communication overheads, and are too time-consuming for typical DL workloads, whose iterations are quite *short*.

In this work, we propose *Load-Balanced Bulk Synchronous Parallel* (LB-BSP), which adaptively adjusts workers' batch size based on their processing capabilities, i.e., slower workers have a smaller batch to process, so that all workers can finish each iteration simultaneously. This mechanism is referred to as *worker-adaptive batch sizing*, and can be easily implemented in modern DL frameworks.

2 LB-BSP DESIGN

While each worker's processing time *increases monotonically* with its workload (represented by its batch size), it remains a challenge how each worker's workload can be accurately apportioned to equalize their iteration durations. To design LB-BSP, we mathematically formulate that problem, and further work out the optimal batch size setup for CPU- and GPU-clusters, respectively.

In particular, for shared production clusters with legacy CPU machines, a worker's computing capability varies with its temporal resources. One prerequisite for solving our optimization problem is to know each worker's sample processing speed before each iteration runs. Such processing speed may be affected by the worker's available resources, such as CPU and memory. We use *Nonlinear AutoRegressive eXogenous model* (NARX) [1], an extended recurrent neural network that makes speed predictions with those driving resources accounted. Meanwhile, for GPU-clusters, we have identified the pairwise-linear relationship between a GPU's batch processing time and its batch size, with which LB-BSP can accurately set each GPU's batch size for the best load-balancing performance.

We have implemented LB-BSP as a Python library that can be integrated in both TensorFlow and MXNet. Our experimental evaluation using benchmark deep learning workloads demonstrates that LB-BSP can effectively accelerate the training of deep models in heterogeneous environments, with up to 2× speedup.

*Chen now works at Huawei Future Network Theory Lab.

SoCC '18, October 11–13, 2018, Carlsbad, CA, USA

ACM ISBN 978-1-4503-6011-1/18/10.
https://doi.org/10.1145/3267809.3275463

REFERENCES

[1] Eugen Diaconescu. 2008. The use of NARX neural networks to predict chaotic time series. *Wseas Transactions on computer research* 3, 3 (2008), 182–191.
[2] Aaron Harlap, Henggang Cui, Wei Dai, Jinliang Wei, Gregory R Ganger, Phillip B Gibbons, Garth A Gibson, and Eric P Xing. 2016. Addressing the straggler problem for iterative convergent parallel ML. In *ACM SoCC*.
[3] Jiawei Jiang, Bin Cui, Ce Zhang, and Lele Yu. 2017. Heterogeneity-aware distributed parameter servers. In *ACM SIGMOD*.
[4] Mu Li, David G Andersen, Jun Woo Park, Alexander J Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J Shekita, and Bor-Yiing Su. 2014. Scaling Distributed Machine Learning with the Parameter Server.. In *USENIX OSDI*.