An Empirical Study of Low-precision Training in Centralized and Decentralized Architecture

Yucheng Lu [1] **Jiaying Wang** [1] **Shangdi Yu** [1]

Abstract

Distributed computing and low-precision numerical representation are two techniques used to speed up computation-intensive machine learning tasks. They have been widely investigated in systemic and algorithmic scope, respectively. However, how we can effectively combine these two techniques has not been well understood in prior arts. In this paper, we aim to empirically investigate the effectiveness of employing low-precision training in both centralized and decentralized architectures, with a combined consideration on both system efficiency and statistical performance. We select two representative machine learning tasks: logistic regression on MNIST by training a linear classifier and image classification on CIFAR-10 by training a 16-layer VGG network. We make the following observations: 1) For both convex and non-convex problems, aggressively quantizing the communication rarely affect the statistical performance while can largely reduce the communication overhead (up to 87.5%). 2) In non-convex problem, training with extremely low precision can no longer achieve state-of-the-art performance. These conclusions can be reference for applying low-precision training in modern systems in the future.

1. Introduction

In the recent decade, the coming of the so called "information age" has provided us unprecedented amount of information available that can be represented in a number form on computer. Not surprisingly, to take advantage of these data, people have been developing machine learning methods and systems that are scalable and can handle large datasets. Due to the large data volume and increasing number of computation-intensive Machine Learning (ML) applications, scaling up ML tasks beyond a single machine has been widely studied. In large-scale distributed ML systems, how workers communicate and coordinate is a crucial design choice. To address this issue, researchers have been investigating new system architectures. State-of-the-art distributed ML systems (e.g. Tensorflow, CNTK, MXNet) are either using synchronous communication via Parameter Servers (PS)/ AllReduce or asynchronous communication via PS (Hsu et al., 2011; Li et al., 2014). However, PS architecture often suffers from congestion in the communication with central node and the straggler issue. To mitigate these problems, researchers have been studying the decentralized distributed training (DC) and found that DC has the same convergence rate compared to PS, and can even outperform PS under certain system configuration (e.g. high-latency or low-bandwidth network). (Lian et al., 2017)

Aside from system design, advanced training techniques on precision variance are also emerging. Researchers have provided sufficient and strong theoretical analysis on this side. Some examples are loss scaling, HALP, and QSGD. (Alistarh et al., 2017; De Sa et al., 2018) However, how we can benefit from these techniques in a distributed system is not well understood.

In this paper, we aim at investigating the effectiveness of employing low-precision training in PS and DC architectures. We select two representative models logistic regression and VGG, former a convex problem and latter a non-convex problem, to demonstrate the performance of our training scheme.

Our contributions in this paper can be summarized as follows:

- Provide an overview of the current distributed machine learning architecture and low-precision techniques.
- Empirically investigate the effectiveness of low-precision training in both centralized and decentralized architectures considering both the system efficiency and statistical performance.

[1]Computer Science Department, Cornell University, Ithaca, NY, USA. Correspondence to: Yucheng Lu <yl2967@cornell.edu>, Jiaying Wang <jw2249@cornell.edu>, Shangdi Yu <sy543@cornell.edu>.

CS6787 www.cs.cornell.edu/courses/cs6787/2018fa/

- Report the statistical results which could serve as a reference for applying low-precision training in modern systems.

2. Related Work

Our paper focuses on the behavior of low-precision training in a distributed computing environment. Low-precision training and inferencing can help limit the model size and energy consumption while retaining the inference accuracy. Such techniques are necessary for special applications of neural network models such as using large Deep Neural Networks (DNNs) on mobile and embedded devices. Using low-precision without compromising accuracy is a challenging task and there have been numerous works on this topic. Researchers have shown that deep networks can be trained with fixed-point number and a stochastic rounding scheme without too much degradation in the classification accuracy.(Gupta et al., 2015) Other advanced training techniques on precision variance, such as loss scaling, HALP, and QSGD are also developed (Alistarh et al., 2017; De Sa et al., 2018). Researchers have also tried to limit the model size and energy consumption from approaches such as weight quantization and exploiting the hashing tricks. Recent works of these other approaches include the deep compression (Han et al., 2015), the HashedNets (Chen et al., 2015), the knowledge distillation (Polino et al., 2018), the UNIQ (Baskin et al., 2018), and the Probabilistic Binary Neural Networks (Peters & Welling, 2018).

Our experiments are done using two popular machine learning models: logistic regression and VGG networks. The Very Deep Convolutional Networks developed by the Visual Geometry Group (VGG network) is an effective model in image classification (Simonyan & Zisserman, 2014). There have been works that evaluated the performance of low-precision training (Cai et al., 2017) and mixed-precision training (Das et al., 2018) in variants of VGG networks. Logistic regression has long been a very popular technique in the machine learning community because of its simplicity and effectiveness in many applications. (Komarek & Moore, 2005; McLachlan, 2005; Hosmer Jr et al., 2013) While there have been a lot of works on how L1 and L2 regularization effect logsitic regression, as far as we know people have not extensively studied how low-precision training influences logistic regression. Since logistic regression is equivalent to a linear classifier, its model size is usually not a problem compared to the huge model sizes of deep neural networks. However, we can still benefit from the less communication overhead in distributed training by using a low-precision scheme.

Researchers have investigated how limiting the precision in distributed training can reduce the communication overhead. For example, Wen et al. (2017) has developed the TernGrad that uses ternary gradients and requires only three numerical levels: -1, 0, and 1. land & Raj (2015) showed that reducing the number of bits used to transmit parameters can reduce the communication overhead while having a beneficial effect on the training.

With the increasing need for large-scale machine learning applications, distributed training has become an active area of research. A popular distributed machine learning framework is the parameter server framework where data and workloads are distributed over worker nodes, while the server nodes keep the shared parameters.(Li et al., 2014) Recently, people have also proposed decentralized distributed training algorithms,where workers can directly connected with each other, that can achieve similar performance with the centralized algorithms. (Lian et al., 2017)

As the section has demonstrated, a lot of work has been done to understand training with precision variance and adapt training to distributed environment. However, there has not been much discussion on how low-precision training can be advantageous specifically in the distributed environment. In this paper, we will focus on how exploiting distributed training and low-precision training affect the training accuracy and efficiency.

3. Methodology

Throughout this paper, we adopt stochastic rounding, an unbiased rounding scheme widely used in low-precision training. The quantization function for stochastic rounding is defined as

$$Q(x) = \delta \begin{cases} \lfloor \frac{x}{\delta} \rfloor + 1, & p \leq \frac{x}{\delta} - \lfloor \frac{x}{\delta} \rfloor \\ \lfloor \frac{x}{\delta} \rfloor, & otherwise \end{cases}$$

where $p \in [0, 1]$ is produced by a uniform random number generator. This operator is non-deterministic, and rounds its argument up with probability $\frac{x}{\delta} - \lfloor \frac{x}{\delta} \rfloor$, and down otherwise.

4. System Architecture

In this subsection, we will briefly introduce the state-of-the-art design of architectures used to coordinate the workers. In general, they can be classified into two categories: centralized and decentralized architecture. In this context, a worker can refer to any type of computing unit such as CPU, GPU, sensor, etc.

4.1. Centralized Architecture

In a centralized architecture, all the workers are disconnected from each other, and instead are connected to a centralized server (Figure 1). The centralized server av-

erages or sums up the intermediate results gathered from the workers at the end of each iteration and sends back the updated result to the workers (Hsu et al., 2011; Li et al., 2014). Trained with sufficient iterations, the parameter on the central node can converge to the optimum. Centralized architecture is easy to monitor and implement; however, in practice the centralized server always suffers from communication congestion when scaling up the workers. Besides, the training progress can easily slow down if there is a really slow worker, namely the straggler issue.

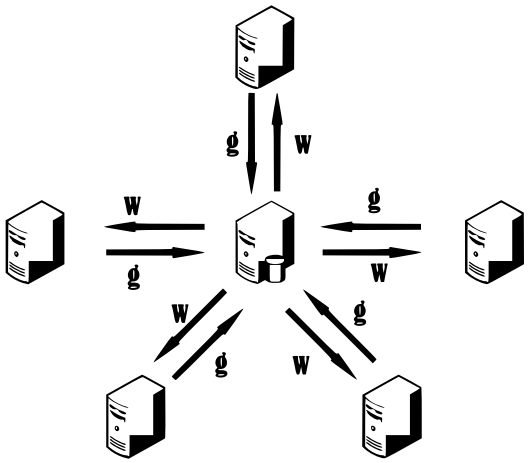


Figure 1. An illustration of centralized architecture, where all the workers push gradient to the centralized node and pull the updated parameters.

4.2. Decentralized Architecture

To mitigate the communication congestion and the straggler issue, researchers also investigated the decentralized architecture. Instead of communicating with a central server, all the workers communicate with adjacent or connected workers to average intermediate results (Figure 2). As such, all the workers will reach consensus and approach the optimum together. Decentralized architecture addresses the two problems brought by centralized architecture, which are mentioned in the previous subsection. However, decentralized architecture makes a trade-off by consuming more bandwidth and handshakes in actual communication.

5. Experiments

In this section, we empirically evaluate the performance of several low-precision training strategies in centralized and decentralized architectures on both convex and non-convex problems. All the experiments are implemented using Pytorch and run on NVIDIA P100 GPUs deployed on the Google cloud platform.

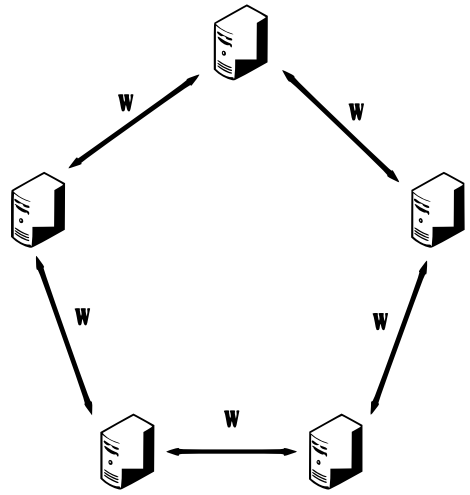


Figure 2. An illustration of decentralized architecture, where all the workers average its intermediate parameters with its adjacent neighbors.

5.1. Experimental Setting

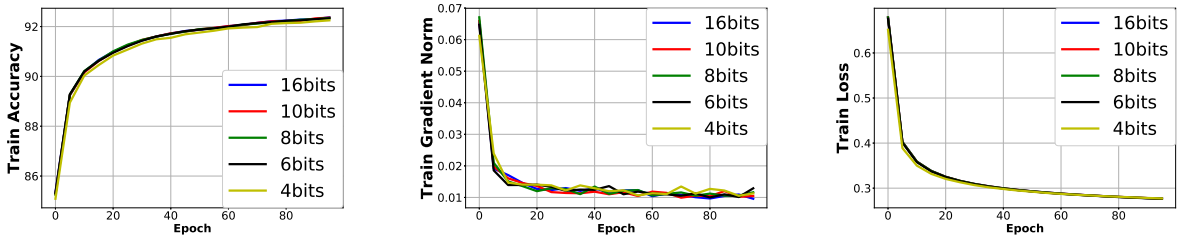
To generalize the results, we evaluate two representative machine learning tasks, one with a convex objective and the other with a non-convex objective.

Logistic Regression We use logistic regression with L2 regularization on MNIST dataset. The objective function is defined as $f(\omega) = -\frac{1}{2}\sum_{i=1}^{n} log(softmax(\omega^\top x_i + b)) + \frac{\lambda}{2}\|\omega\|^2$, with a regularization parameter used in prior work. We choose $\lambda = 10$, which makes the objective a strongly convex function with $M \neq 0$. We use a learning rate of $\alpha = 0.001$ for all settings. Since MNIST is sparse and poorly conditioned, we measure the norm of the gradient at each iteration to illustrate the convergence of the algorithm. It is a metric that has been used for logistic regression on MNIST in previous work.(De Sa et al., 2018)(Johnson & Zhang, 2013)

Image Classification For completeness, we also evaluate the performance on a non-convex problem, namely image classification on CIFAR-10 using a 16-layer VGG network. Following prior works(Izmailov et al., 2018), we apply a channel-wise normalization to make the values in each channel have zero mean and unit variance.

5.2. Mixed-Precision Training in the Parameter Server

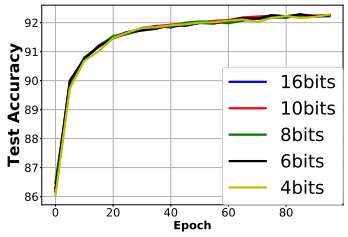
In this experiment, we evaluate performance of mixed-precision training in a Parameter Server architecture. Specifically, we launch 8 workers and 1 parameter server and run the logistic regression. We evenly split the dataset onto all the workers. We let all the workers train with full precision but quantize the intermediate results before pushing them to the centralized node. The parameters stored on



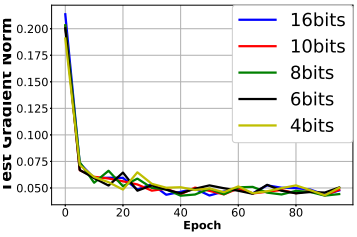
(a) Accuracy on Training Set

(b) Averaged Gradient Norm on Training Set

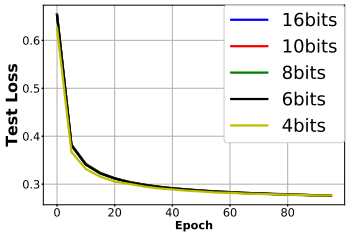
(c) Loss on Training Set



(d) Accuracy on Test Set

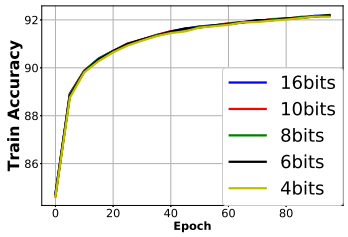


(e) Averaged Gradient Norm on Test Set

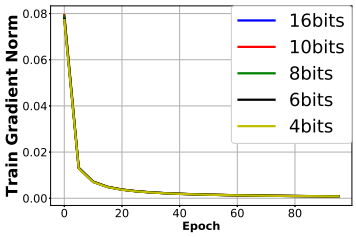


(f) Loss on Test Set

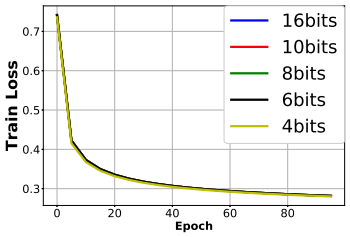
Figure 3. Results of Logistic Regression on MNIST Under Different Quantized Communication Policy in **Parameter Server Architecture**



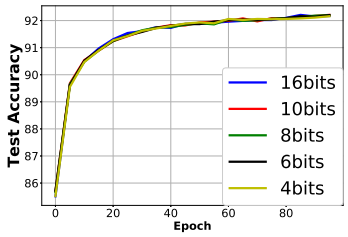
(a) Accuracy on Training Set



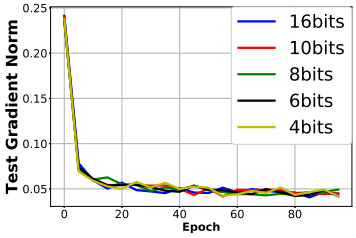
(b) Averaged Gradient Norm on Training Set



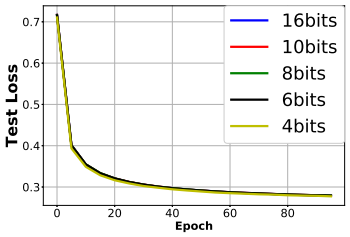
(c) Loss on Training Set



(d) Accuracy on Test Set



(e) Averaged Gradient Norm on Test Set



(f) Loss on Test Set

Figure 4. Results of Logistic Regression on MNIST Under Different Quantized Communication Policy in **Decentralized Ring Architecture**

centralized node is in full precision, i.e. 32 bits. We set communication frequency to be 1 push/pull per iteration. We train the model for 100 epochs.

The main results are shown in Figure 3. As is shown in the figure, we can quantize all the intermediate results to 4 bits and still maintain the statistical results from the baseline. Since with each epoch, we iterate over all the training set on all the workers. Approximately, using 4-bit can save bandwidth up to 87.5%. We summarize the test accuracy and average gradient norm after 100 epochs in Table 2.

5.3. Partially Low-Precision Communication Among Workers

Inspired by QSGD (Alistarh et al., 2017), we employ more relaxations on consistency among workers in a distributed training task. Unlike the widely adopted All-to-All communication paradigm, we let workers communicate to a part of the network with low-precision numbers.

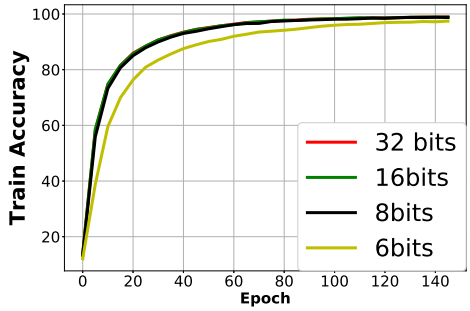
In this experiment, we evaluate the performance of a low-precision communication scheme in a decentralized architecture. Specifically, we launch 8 workers and organize them in a ring network, i.e. all the workers have two neighboring workers. We let all the workers train with full precision but communicate with low-precision numbers. We set the communication frequency to be 1 average with neighbors per iteration.

In logistic regression, we train the model for 100 epochs. Results from logistic regression are shown in Figure 4. As shown in the figure, we can quantize all the communication to 4 bits and still maintain the statistical results from the baseline. One thing to notice is that, unlike Parameter Server architecture, where all the workers reach consensus immediately in one iteration by a centralized node, ring network relaxes that requirement and allows workers to have staled consensus. Compared to results from Parameter Server, we obtain that even if consensus among workers is delayed, we still get convergence with quantized intermediate results.

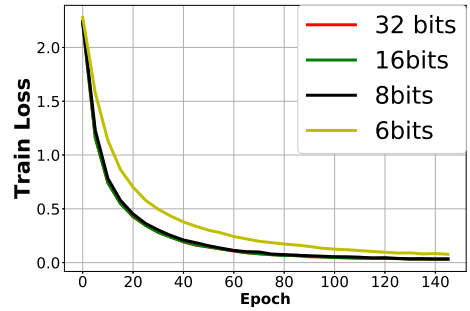
In image classification, we train the model for 150 epochs. Results from image classification are shown in Figure 5. Similar to the results in logistic regression, in image classification we can quantize all the communication to 6 bits and still maintain the statistical results from the baseline.

5.4. Quantized Model

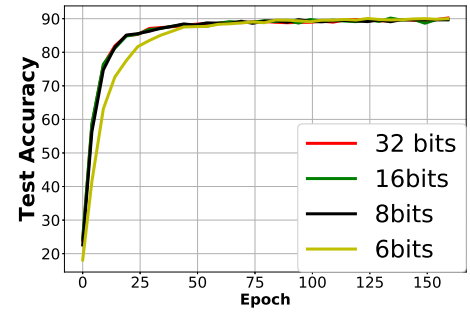
In this experiment, instead of communication, we quantize the deep learning model in a Parameter Server architecture. Specifically, we quantize all the parameters in the VGG network along the training, i.e. quantizing the weights and gradients after each iteration. We plot the test accuracy in Table 1. We train the model for 150 epochs. We can see



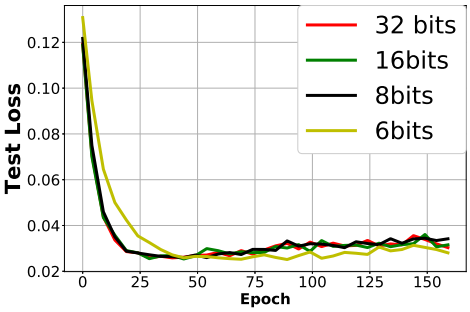
(a) Accuracy on Training Set



(b) Loss on Training Set



(c) Accuracy on Test Set



(d) Loss on Test Set

Figure 5. Results of Image Classification on CIFAR-10 Under Different Quantized Communication Policy in **Decentralized Architecture**

Table 1. Results of Image Classification on CIFAR-10 Under Different Quantized Model in **Parameter Server Architecture**

BITS	EPOCHS	TEST ACCURACY
32	150	89.47%
16	150	89.46%
10	150	89.36%
8	150	89.15%
6	150	78.12%

Table 2. Test Accuracy and Avg. Gradient Norm after 100 epochs (Results of Logistic Regression on MNIST Under Different Quantized Communication Policy in **Parameter Server Architecture**)

BITS	EPOCH	TEST ACCURACY	TEST GRAD NORM
16	100	92.35%	0.04468
10	100	92.34%	0.04538
8	100	92.33%	0.04591
6	100	92.34%	0.04493
4	100	92.33%	0.04518

that quantizing the parameters with 8 bits can still get us the performance training with precision (From 89.47% to 89.15%). However, if we further reduce the precision to 6 bits, the accuracy tends to have a huge degradation (From 89.47% to 78.12%).

The main takeaway in this subsection is for a deep learning problem, quantizing the model to extremely low precision can no longer achieve state-of-the-art performance.

6. Conclusion

In this paper, we empirically evaluate the performance of low-precision training in both centralized and decentralized architecture. We investigate both the convex and non-convex problems. We observe that, 1) For both convex and non-convex problems, aggressively quantizing the communication rarely affect the statistical performance while can largely reduce the communication overhead (up to 87.5%). 2) In non-convex problem, training with extremely low precision can no longer achieve state-of-the-art performance. For future work, one can theoretical investigate how low-precision computation and low-communciation affect each other in distributed training.

References

Alistarh, Dan, Grubic, Demjan, Li, Jerry, Tomioka, Ryota, and Vojnovic, Milan. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 1709–1720. Curran Associates, Inc., 2017.

Baskin, C., Schwartz, E., Zheltonozhskii, E., Liss, N., Giryes, R., Bronstein, A. M., and Mendelson, A. UNIQ: Uniform Noise Injection for Non-Uniform Quantization of Neural Networks. *ArXiv e-prints*, April 2018.

Cai, Z., He, X., Sun, J., and Vasconcelos, N. Deep Learning with Low Precision by Half-wave Gaussian Quantization. *ArXiv e-prints*, February 2017.

Chen, W., Wilson, J. T., Tyree, S., Weinberger, K. Q., and Chen, Y. Compressing Neural Networks with the Hashing Trick. *ArXiv e-prints*, April 2015.

Das, D., Mellempudi, N., Mudigere, D., Kalamkar, D., Avancha, S., Banerjee, K., Sridharan, S., Vaidyanathan, K., Kaul, B., Georganas, E., Heinecke, A., Dubey, P., Corbal, J., Shustrov, N., Dubtsov, R., Fomenko, E., and Pirogov, V. Mixed Precision Training of Convolutional Neural Networks using Integer Operations. *ArXiv e-prints*, February 2018.

De Sa, C., Leszczynski, M., Zhang, J., Marzoev, A., Aberger, C. R., Olukotun, K., and Ré, C. High-Accuracy Low-Precision Training. *ArXiv e-prints*, March 2018.

Gupta, S., Agrawal, A., Gopalakrishnan, K., and Narayanan, P. Deep Learning with Limited Numerical Precision. *ArXiv e-prints*, February 2015.

Han, S., Mao, H., and Dally, W. J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *ArXiv e-prints*, October 2015.

Hosmer Jr, David W, Lemeshow, Stanley, and Sturdivant, Rodney X. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

Hsu, D., Karampatziakis, N., Langford, J., and Smola, A. Parallel Online Learning. *ArXiv e-prints*, March 2011.

Izmailov, Pavel, Podoprikhin, Dmitrii, Garipov, Timur, Vetrov, Dmitry, and Wilson, Andrew Gordon. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Johnson, Rie and Zhang, Tong. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pp. 315–323, 2013.

Komarek, Paul and Moore, Andrew. Making logistic regression a core data mining tool: A practical investigation of accuracy, speed, and simplicity. Technical Report CMU-RI-TR-05-27, Carnegie Mellon University, Pittsburgh, PA, May 2005.

Li, Mu, Andersen, David G., Park, Jun Woo, Smola, Alexander J., Ahmed, Amr, Josifovski, Vanja, Long, James, Shekita, Eugene J., and Su, Bor-Yiing. Scaling distributed machine learning with the parameter server. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pp. 583–598, Broomfield, CO, 2014. USENIX Association. ISBN 978-1-931971-16-4.

Lian, Xiangru, Zhang, Ce, Zhang, Huan, Hsieh, Cho-Jui, Zhang, Wei, and Liu, Ji. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5330–5340. Curran Associates, Inc., 2017.

McLachlan, Geoffrey J. *Discriminant Analysis and Statistical Pattern Recognition*. Wiley, Newark, NJ, 2005. URL https://cds.cern.ch/record/997160.

Peters, J. W. T. and Welling, M. Probabilistic Binary Neural Networks. *ArXiv e-prints*, September 2018.

Polino, A., Pascanu, R., and Alistarh, D. Model compression via distillation and quantization. *ArXiv e-prints*, February 2018.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

Wen, Wei, Xu, Cong, Yan, Feng, Wu, Chunpeng, Wang, Yandan, Chen, Yiran, and Li, Hai. Terngrad: Ternary gradients to reduce communication in distributed deep learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 1509–1519. Curran Associates, Inc., 2017.

land, A. and Raj, B. Reducing communication overhead in distributed learning by an order of magnitude (almost). In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2219–2223, April 2015. doi: 10.1109/ICASSP.2015.7178365.