Incremental Consensus based Collaborative Deep Learning

Zhanhong Jiang [1] **Aditya Balu** [1] **Chinmay Hegde** [2] **Soumik Sarkar** [1]

Abstract

Collaborative learning among multiple agents with private datasets over a communication network often involves a tradeoff between communication, consensus and optimality. In this paper, we build on recent algorithmic progresses in distributed deep learning to explore various consensus-optimality trade-offs with more communication over a fixed communication topology. We propose *incremental consensus*-based distributed SGD (i-CDSGD) algorithm and its momentum variant (i-CDMSGD), which involves multiple consensus steps (where each agent communicates information with its neighbors) within each SGD iteration. We support our algorithms via numerical experiments, and demonstrate improvements over existing methods for collaborative deep learning.

1. Introduction

Scaling up deep learning algorithms in a distributed setting (LeCun et al., 2015; Recht et al., 2011; Jin et al., 2016) is becoming increasingly critical, impacting several applications such as learning in robotic networks (Lenz et al., 2015), the Internet of Things (IoT) (Gubbi et al., 2013; Lane et al., 2015), and mobile device networks (Lane & Georgiev, 2015). Several distributed deep learning approaches have been proposed to address issues such as model parallelism (Dean et al., 2012), data parallelism (Dean et al., 2012; Jiang et al., 2017), and the role of communication and computation (Li et al., 2014; Das et al., 2016).

We focus on the constrained communication topology setting where the data is distributed (so that each agent has its own estimate of the deep model) and where information exchange among the learning agents are constrained along the edges of a given communication graph (Jiang et al., 2017; Lian et al., 2017). In this context, two key aspects arise: *consensus* and *optimality*. We refer the reader to Figure 1 for an illustration involving 3 agents. With sufficient information exchange, the learned model parameters corresponding to each agent, $\theta_k^j, j = 1, 2, 3$ could converge to $\hat{\theta}$, in which case they achieve consensus but not optimality (here, θ_* is the optimal model estimate if all the data were centralized). On the other hand, if no communication happens, the agents may approach their individual model estimates (θ_*^i) while being far from consensus. The question is whether this trade-off between consensus and optimality can be balanced so that *all* agents collectively agree upon a model estimate close to θ_*.

Our contributions: In this paper, we propose, a new algorithmic frameworks for distributed deep learning that enable us to explore fundamental trade-offs between consensus and optimality called *incremental consensus*-based distributed SGD (i-CDSGD), which is a stochastic extension of the descent-style algorithm proposed in (Berahas et al., 2017). This involves running multiple consensus steps where each agent exchanges information with its neighbors within each SGD iteration. Specifically, we 1) propose i-CDSGD and show the convergence of i-CDSGD (Theorems 1 & 2) for strongly convex and non-convex objective functions; 2) empirically demonstrate that i-CDMSGD (the momentum variant of i-CDSGD) can achieve similar (global) accuracy as the state-of-the-art with lower fluctuation across epochs as well as better consensus.

2. Problem Formulation

We consider the standard unconstrained empirical risk minimization (ERM) problem typically used in machine learning problems (such as deep learning):

$$\min \frac{1}{n} \sum_{i=1}^{n} f^i(\theta), \tag{1}$$

where $\theta \in \mathbb{R}^d$ denotes the parameter vector of interest, $f : \mathbb{R}^d \to \mathbb{R}$ denotes a given loss function, and f^i is the function value corresponding to a data point i. Our focus is to investigate the case where the ERM problem is solved collaboratively among a number of computational *agents*. In this paper, we are interested in problems where the agents

[1]Department of Mechanical Engineering, Iowa State University, Ames, Iowa, USA [2]Department of Electrical and Computer Engineering, Iowa State University, Ames, Iowa, USA. Correspondence to: Soumik Sarkar <soumiks@iastate.edu>.

Workshop on Modern Trends in Nonconvex Optimization for Machine Learning. The 35^{th} International Conference on Machine Learning, Stockholm

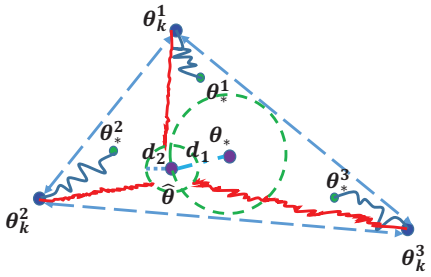

Figure 1. A closer look at the optimization updates in distributed deep learning: Blue dots represent the current states (i.e., learned model parameters) of the agents; green dots represent the individual local optima (θ_*^i), that agents converge to without sufficient consensus; the purple dot (θ_*) represents the ideal optimal point for the entire agent population; another purple dot $\hat{\theta}$ represents a possible consensus point for the agents which is far from optimal; blue and red curves signify the convergence trajectories with different step sizes; the green dashed circles indicate the neighborhoods of θ_* and $\hat{\theta}$, respectively; d_2 represents the consensus bound/error and d_1 represents the optimality bound/error.

exhibit *data parallelism*, i.e., they only have access to their own respective training datasets. However, we assume that the agents can communicate over a static undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a vertex set (with nodes corresponding to agents) and $\mathcal{E}$ is an edge set. Throughout this paper we assume that the graph $\mathcal{G}$ is *connected*.

Let $\mathcal{D}_j,\ j = 1, \ldots, n$ denote the subset of the training data (comprising n_j samples) corresponding to the j^{th} agent such that $\sum_{j=1}^{N} n_j = n$, where N is the total number of agents. With this formulation, and since $f(\theta) = \sum_{j=1}^{N} f_j(\theta)$, we have the following (constrained) reformulation of (1):

$$\min \sum_{j=1}^{N} \sum_{i \in \mathcal{D}_j} f_j^i(\theta^j), \text{ s.t. } \theta^j = \theta^l \ \ \forall (j,l) \in \mathcal{E}, \quad (2)$$

Equivalently, the concatenated form of the above equation is as follows:

$$\min \mathcal{F}(\Theta) := \sum_{j=1}^{N} \sum_{i \in \mathcal{D}_j} f_j^i(\theta^j), \text{ s.t. } (\Pi \otimes I_d)\Theta = \Theta, \quad (3)$$

where $\Theta := [\theta^1; \theta^2; \ldots; \theta^N] \in \mathbb{R}^{dN}$, $\Pi \in \mathbb{R}^{N \times N}$ is the agent interaction matrix with its entries π_{jl} indicating the link between agents j and l, I_d is the identity matrix of dimension $d \times d$, and $\otimes$ represents the Kronecker product.

Definition 1. *A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be H-strongly convex, if for all $x, y \in \mathbb{R}^d$, we have $f(y) \geq f(x) + \nabla f(x)^T(y - x) + \frac{H}{2}\|y - x\|^2$; it is said to be γ-smooth if we have $f(y) \leq f(x) + \nabla f(x)^T(y - x) + \frac{\gamma}{2}\|y - x\|^2$, where $\|\cdot\|$ represents the Euclidean norm; it is said to be coercive if it satisfies: $c(x) \to \infty$ when $\|x\| \to \infty$.*

Assumption 1. *The objective functions $f_j : \mathbb{R}^d \to \mathbb{R}$ are assumed to satisfy the following conditions: a) each f_j is γ_j-smooth; b) each f_j is proper (not everywhere infinite) and coercive.*

Assumption 2. *The interaction matrix Π is normalized to be doubly stochastic; the second largest eigenvalue of Π is strictly less than 1, i.e., $\lambda_2 < 1$, where λ_2 is the second largest eigenvalue of Π. If $(j,l) \notin \mathcal{E}$, then $\pi_{jl} = 0$.*

We will solve (2) in a distributed and stochastic manner. For the formulation in (2), the state-of-the-art algorithm is a method called *consensus distributed SGD*, or CDSGD, recently proposed in (Jiang et al., 2017). This method estimates θ according to the update equation:

$$\theta_{k+1}^j = \sum_{l \in Nb(j)} \pi_{jl}\theta_k^l - \alpha g_j(\theta_k^j) \quad (4)$$

where $Nb(j)$ indicates the neighborhood of agent j, α is the step size, $g_j(\theta_k^j)$ is the (stochastic) gradient of f_j at θ_k^j, implemented by drawing a minibatch of sampled data points. More precisely, $g_j(\theta_k^j) = \frac{1}{b'} \sum_{q' \in \mathcal{D}'} \nabla f_j^{q'}(\theta_k^j)$, where b' is the size of the minibatch $\mathcal{D}'$ selected uniformly at random from the data subset $\mathcal{D}_j$ available to Agent j.

3. Proposed Algorithm

State-of-the-art algorithms such as CDSGD alternate between the *gradient update* and *consensus* steps. We propose a natural extension where one can control the emphasis on *consensus* relative to the *gradient update* and hence, leads to interesting trade-offs between consensus and optimality.

Increasing consensus. Observe that the concatenated form of the CDSGD updates, (4), can be expressed as

$$\Theta_{k+1} = (\Pi \otimes I_d)\Theta_k - \alpha \mathbf{g}(\Theta_k).$$

If we perform τ consensus steps interlaced with each gradient update, we can obtain the following concatenated form of the iterations of the parameter estimates:

$$\Theta_{k+1} = (\Pi^\tau \otimes I_d)\Theta_k - \alpha \mathbf{g}(\Theta_k) \quad (5)$$

where, $\mathbf{g}(\Theta_k) = \left[g_1^T(\theta_k^1),\ g_2^T(\theta_k^2), \ldots, g_N^T(\theta_k^N)\right]^T$. We call this variant *incremental* consensus-based distributed SGD (i-CDSGD) which is detailed in Algorithm 1. Note, in a distributed setting, this algorithm incurs an additional factor τ in communication complexity. We next present the pseudo-code of i-CDSGD shown in Algorithm 1.

Tools for convergence analysis. We now analyze the convergence of the iterates $\{\theta_k^j\}$ generated by our algorithm.. Specifically, we identify an appropriate Lyapunov function

Algorithm 1 i-CDSGD/i-CDMSGD

Initialization: $\theta_0^j, v_0^j,\ j = 1, 2, ..., N$, α, N, τ, m, Π
Distribute the training data set to N agents
for *each agent* **do**
  Randomly shuffle each data subset
  for $k = 0 : m$ **do**
    $t = 0$
    for $j = 1, ..., N$ **do**
      $\theta_t^j = \theta_k^j$
      $v_t^j = v_k^j$ {for i-CDMSGD}
    end for
    while $t \leq \tau - 1$ **do**
      for $j = 1, ..., N$ **do**
        $\theta_{t+1}^j = \sum_{l \in Nb(j)} \pi_{jl} \theta_t^l$ {**Incremental Consensus**}
        $v_{t+1}^j = \sum_{l \in Nb(j)} \pi_{jl} v_t^l$ {for i-CDMSGD}
      end for
      $t = t + 1$
    end while
    $\hat{\theta} = \theta_t^j$
    $\theta_{k+1}^j = \hat{\theta} - \alpha g_j(\theta_k^j)$
    For i-CDMSGD:
    $\hat{v} = v_t^j$
    $v_{k+1}^j = \hat{\theta} - \theta_k^j + \mu \hat{v} - \alpha g_j(\theta_k^j)$
    $\theta_{k+1}^j = \theta_k^j + v_{k+1}^j$
  end for
end for

(that is bounded from below) for each algorithm that decreases with each iteration, thereby establishing convergence. In our analysis, we use the concatenated (Kronecker) form of the updates. For simplicity, let $\mathbf{P} = \Pi \otimes I_d \in \mathbb{R}^{Nd \times Nd}$.

We begin the analysis for i-CDSGD by constructing a Lyapunov function that combines the true objective function with a regularization term involving a quadratic form of consensus as follows:

$$V(\Theta) := \mathcal{F}(\Theta) + \frac{1}{2\alpha} \Theta^T (I_{Nd} - \mathbf{P}^\tau) \Theta \tag{6}$$

It is easy to show that $\sum_{j=1}^N f_j(\theta^j)$ is $\gamma_m := max_j\{\gamma_j\}$-smooth, and that $V(\Theta)$ is $\hat{\gamma}$-smooth with $\hat{\gamma} := \gamma_m + \alpha^{-1}\lambda_{\max}(I_{Nd} - \mathbf{P}^\tau) = \gamma_m + \alpha^{-1}(1 - \lambda_N^\tau)$. Likewise, it is easy to show that $\sum_{j=1}^N f_j(\theta^j)$ is $H_m := min_j\{H_j\}$-strongly convex; therefore $V(\Theta)$ is $\hat{H}$-strongly convex with $\hat{H} := H_m + (2\alpha)^{-1}\lambda_{\min}(I_{Nd} - \mathbf{P}^\tau) = H_m + (2\alpha)^{-1}(1 - \lambda_2^\tau)$. We also assume that there exists a lower bound $V_{\inf}$ for the function value sequence $\{V(\Theta_k)\}, \forall k$. When the objective functions are strongly convex, we have $V_{\inf} = V(\Theta^*)$, where Θ^* is the optimizer.

Due to Assumptions 1 and 2, it is straightforward to obtain an equivalence between the gradient of Eq. 6 and the update law of i-CDSGD. Rewriting (5), we get:

$$\Theta_{k+1} = \mathbf{P}^\tau \Theta_k - \alpha \mathbf{g}(\Theta_k) \tag{7}$$

Therefore, we obtain:

$$\begin{aligned} \Theta_{k+1} &= \Theta_k - \Theta_k + \mathbf{P}\Theta_k + -\alpha \mathbf{g}(\Theta_k) \\ &= \Theta_k - \alpha(\mathbf{g}(\Theta_k) + \frac{1}{\alpha}(I_{Nd} - \mathbf{P}^\tau)\Theta_k) \end{aligned} \tag{8}$$

The last term in (8) is precisely the gradient of $V(\Theta)$. In the stochastic setting, $\mathbf{g}(\Theta_k)$ can be approximated by sampling one data point (or a mini-batch of data points) and the stochastic Lyapunov gradient is denoted by $\mathcal{S}(\Theta_k), \forall k$.

For analysis, we require a bound on the variance of the stochastic Lyapunov gradient $\mathcal{S}(\Theta_k)$ such that the variance of the gradient noise[1] can be bounded from above. The variance of $\mathcal{S}(\Theta_k)$ is defined as: $Var[\mathcal{S}(\Theta_k)] := \mathbb{E}[\|\mathcal{S}(\Theta_k)\|^2] - \|\mathbb{E}[\mathcal{S}(\Theta_k)]\|^2$. The following assumption is standard in SGD convergence analysis, and is based on (Bottou et al., 2016).

Assumption 3. *a) There exist scalars $r_2 \geq r_1 > 0$ such that $\nabla V(\Theta_k)^T \mathbb{E}[\mathcal{S}(\Theta_k)] \geq r_1 \|\nabla V(\Theta_k)\|^2$ and $\|\mathbb{E}[\mathcal{S}(\Theta_k)]\| \leq r_2 \|\nabla V(\Theta_k)\|$ for all $k \in \mathbb{N}$; b) There exist scalars $B \geq 0$ and $B_V \geq 0$ such that $Var[\mathcal{S}(\Theta_k)] \leq B + B_V \|\nabla V(\Theta_k)\|^2$ for all $k \in \mathbb{N}$; c) there exists a constant $G > 0$ such that $\|\nabla V(\Theta_k)\| \leq G, \forall \Theta_k \in \mathbb{R}^{Nd}$.*

Assumption 3 implies that the second moment of $\mathcal{S}(\Theta_k)$ can be bounded above as $\mathbb{E}[\|\mathcal{S}(\Theta_k)\|^2] \leq B + B_m \|\nabla V(\Theta_k)\|^2$, where $B_m := B_V + r_2^2 \geq r_1^2 > 0$.

4. Main Results

This section presents the main results by analyzing the convergence properties of the i-CDSGD for both strongly convex and non-convex objective functions: the *consensus bound* and the *optimality bound*.

Proposition 1. *(Consensus with fixed step size, i-CDSGD) Let Assumptions 1, 2, 3 hold. The iterates of i-CDSGD (Algorithm 1) satisfy the following inequality $\forall k \in \mathbb{N}$, when α satisfies $0 < \alpha \leq \frac{r_1 - (1 - \lambda_N^\tau) B_m}{\gamma_m B_m}$,*

$$\mathbb{E}[\|\theta_k^j - s_k\|] \leq \frac{\alpha \sqrt{B + B_m G^2}}{1 - \lambda_2^\tau} \tag{9}$$

where $s_k = \frac{1}{N} \sum_{j=1}^N \theta_k^j$.

Theorem 1. *(Convergence of i-CDSGD in strongly convex case) Let Assumptions 1, 2 and 3 hold. When the step size*

[1] As our proposed algorithm is a distributed variant of SGD, the noise in the performance is caused by the random sampling (Song et al., 2015).

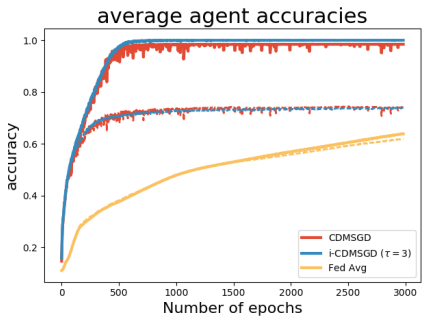

Figure 2. Performance of different algorithms with unbalanced sample distribution among agents. (Dashed lines represent test accuracy & solid lines represent training accuracy.)

satisfies $0 < \alpha \leq \frac{r_1-(1-\lambda_N^\tau)B_m}{\gamma_m B_m}$*, the iterates of i-CDSGD (Algorithm 1) satisfy the following inequality* $\forall k \in \mathbb{N}$*:*

$$\mathbb{E}[D_k] \leq C_1^{k-1} D_1 + C_2 \sum_{q=0}^{k-1} C_1^q \tag{10}$$

where $D_k = V(\Theta_k) - V^*$, $C_1 = 1-(\alpha H_m + \frac{1}{2}(1-\lambda_2^\tau))r_1$, $C_2 = \frac{(\alpha^2\gamma_m + \alpha(1-\lambda_N^\tau))B}{2}$.

In a conventional manner, to eliminate the negative effect of "noise" caused by the stochastic gradients, a diminishing step size is used. However, in this context, it can be observed from Theorem 1 that a constant step size results in the convergence to a neighborhood of the local minimum. We claim that using a constant step size can lead to a linear convergence rate instead of a sublinear convergence rate. Although we show the convergence for strongly convex objectives, we note that objective functions are highly non-convex for most deep learning applications. While convergence to a global minimum in such cases is extremely difficult to establish, we prove that i-CDSGD still exhibits weaker (but meaningful) notions of convergence.

Theorem 2. *(Convergence to the first-order stationary point for non-convex case of i-CDSGD) Let Assumptions 1, 2, and 3 hold. When the step size satisfies* $0 < \alpha \leq \frac{r_1-(1-\lambda_N^\tau)B_m}{\gamma_m B_m}$*, the iterates of i-CDSGD (Algorithm 1) satisfy the following inequality* $\forall K \in \mathbb{N}$*:*

$$\mathbb{E}[\frac{1}{K}\sum_{k=1}^{K}\|\nabla V(\Theta_k)\|^2] \leq \frac{(\gamma_m\alpha + (1-\lambda_N^\tau))B}{r_1} + \frac{2(V(\Theta_1) - V_{inf})}{Kr_1\alpha}. \tag{11}$$

5. Experimental Results

We validate our algorithms via several experimental results using the CIFAR-10 image recognition dataset (with standard training and testing sets) with a deep convolutional neural network (CNN) model The mini-batch size is set to 512, and step size is set to 0.01 in all experiments. We use a sparse network topology with 5 agents. We use both balanced and unbalanced data sets for our experiments. In the balanced case, agents have an equal share of the entire training set. However, in the unbalanced case, agents have (randomly selected) unequal parts of the training set while making sure that each agent has at least half of the equal share amount of examples.

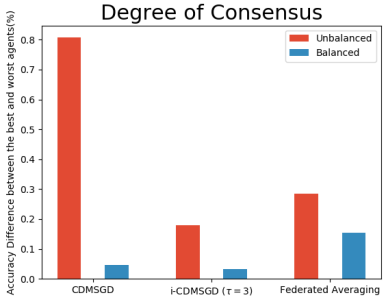

Figure 3. The accuracy percentage difference between the best and the worst agents for different algorithms with unbalanced and balanced sample distribution among agents.

Performance of algorithms. While the experiments were performed for both momentum based and non-momentum based methods, due to space constraints we show the results of the momentum variant, which provides key insights to the analysis performed in the earlier section. In Figure 2, we compare the performance of i-CDMSGD with state-of-the art techniques such as CDMSGD and Federated Averaging using an unbalanced data set. All algorithms were run for 3000 epochs. Observing the average accuracy over all the agents for both training and test data, we note that i-CDMSGD can converge as fast as CDMSGD with lesser fluctuation in the performance across epochs and with slightly better performance eventually. Both the algorithms significantly outperform Federated Averaging in terms of average accuracy.

Degree of Consensus. One of the main contribution of our paper is to show that one can control the degree of consensus while maintaining average accuracy in distributed deep learning. We demonstrate this by observing the accuracy difference between the best and the worst performing agents (identified by computing the mean accuracy for the last 100 epochs). As shown in Figure 3, the degree of consensus is similar for all three algorithms for balanced data set, with i-CDMSGD performing slightly better than the rest and Federated Averaging performing the worst. However, for an unbalanced set, i-CDMSGD performs significantly better compared to CDMSGD and CDMSGD is the worst among all. We do not compare these results by comparing τ as the doubly stochastic agent interaction matrix for the small agent population becomes stationary very quickly with a very small value of τ. However, this will be explored in our future work with significantly bigger networks.

6. Conclusion and Future Work

For investigating the trade-off between consensus and optimality in distributed deep learning with constrained communication topology, this paper presents a new algorithm, called i-CDSGD. We show the convergence properties for the proposed algorithm and the relationships between the hyperparameters and the consensus & optimality bounds. Theoretical and experimental comparison with the state-of-the-art algorithm called CDSGD, shows that i-CDMSGD can improve the degree of consensus among the agents while maintaining the average accuracy especially when there is data imbalance among the agents. Future research directions include learning with non-uniform data distributions among agents and time-varying networks.

Acknowledgments

This work was partly supported by USDA NIFA No. 2017-67007-26151, U.S. Air Force Office of Scientific Research under the YIP grant FA9550-17-1-0220 and National Science Foundation under the grant CCF 1750920.

References

Berahas, Albert S, Bollapragada, Raghu, Keskar, Nitish Shirish, and Wei, Ermin. Balancing communication and computation in distributed optimization. *arXiv preprint arXiv:1709.02999*, 2017.

Bottou, Léon, Curtis, Frank E, and Nocedal, Jorge. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*, 2016.

Das, Dipankar, Avancha, Sasikanth, Mudigere, Dheevatsa, Vaidynathan, Karthikeyan, Sridharan, Srinivas, Kalamkar, Dhiraj, Kaul, Bharat, and Dubey, Pradeep. Distributed deep learning using synchronous stochastic gradient descent. *arXiv preprint arXiv:1602.06709*, 2016.

Dean, Jeffrey, Corrado, Greg, Monga, Rajat, Chen, Kai, Devin, Matthieu, Mao, Mark, Senior, Andrew, Tucker, Paul, Yang, Ke, Le, Quoc V, et al. Large scale distributed deep networks. In *Advances in neural information processing systems*, pp. 1223–1231, 2012.

Gubbi, Jayavardhana, Buyya, Rajkumar, Marusic, Slaven, and Palaniswami, Marimuthu. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.

Jiang, Zhanhong, Balu, Aditya, Hegde, Chinmay, and Sarkar, Soumik. Collaborative deep learning in fixed topology networks. *Neural Information Processing Systems (NIPS)*, 2017.

Jin, Peter H, Yuan, Qiaochu, Iandola, Forrest, and Keutzer, Kurt. How to scale distributed deep learning? *arXiv preprint arXiv:1611.04581*, 2016.

Lane, Nicholas D and Georgiev, Petko. Can deep learning revolutionize mobile sensing? In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pp. 117–122. ACM, 2015.

Lane, Nicholas D, Bhattacharya, Sourav, Georgiev, Petko, Forlivesi, Claudio, and Kawsar, Fahim. An early resource characterization of deep learning on wearables, smartphones and internet-of-things devices. In *Proceedings of the 2015 International Workshop on Internet of Things towards Applications*, pp. 7–12. ACM, 2015.

LeCun, Yann, Bengio, Yoshua, and Hinton, Geoffrey. Deep learning. *Nature*, 521(7553):436–444, 2015.

Lenz, Ian, Lee, Honglak, and Saxena, Ashutosh. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015.

Li, Mu, Andersen, David G, Smola, Alexander J, and Yu, Kai. Communication efficient distributed machine learning with the parameter server. In *Advances in Neural Information Processing Systems*, pp. 19–27, 2014.

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 *Advances in Neural Information Processing Systems*, pp. 5336–5346, 2017.

Recht, Benjamin, Re, Christopher, Wright, Stephen, and Niu, Feng. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pp. 693–701, 2011.

Song, Shuang, Chaudhuri, Kamalika, and Sarwate, Anand. Learning from data with heterogeneous noise using sgd. pp. 894–902, 2015.