한국빅데이터학회지
제5권 제1호, 2020, pp. 189-195 https://doi.org/10.36498/kbigdt.2020.5.1.189

A Sequencing Problem with Generalized Due Dates for Distributed Training of Neural Networks

신경망 분산 학습을 위한 일반 납기를 갖는 시퀀싱 문제

Byung-Cheon Choi[1] · Yunhong Min[2†]

School of Business, Chungnam National University[1],
Graduate School of Logistics, Incheon National University[2]

Abstract

We consider the stale problem which makes the training speed slow in the field of deep learning. The problem can be formulated as a single-machine scheduling problem with generalized due dates in which the objective is to minimize the total earliness and tardiness. We show that the problem can be solved in polynomial time if the orders of the small and the large jobs in an optimal schedule are known in advance.

■ Keyword : Scheduling, Generalized Due Date, Distributed Training

요 약

본 논문은 딥러닝을 위한 분산학습에서 학습속도를 저하시키는 stale 문제를 최소화하기 위한 방법으로 데이터 시퀀싱을 제안하였다. 이 데이터 시퀀싱 문제는 일반 납기를 갖는 단일 공정 하에서 일찍 혹은 늦음 정도의 총합을 최소화 하는 스케줄링 문제로 모델링할 수 있다. 만약 최적해에서 크기가 작은 작업과 큰 작업의 순서가 미리 알려져 있다면, 이 스케줄링 문제가 효율적으로 풀린다는 것을 보였다.

■ 중심어 : 스케줄링, 일반 납기, 분산 학습

2020년 08월 05일 접수; 2020년 08월 20일 수정본 접수; 2020년 08월 25일 게재 확정.
† 교신저자 (yunhong.min@inu.ac.kr)

Ⅰ. Introduction

Recently, deep learning [18], a set of techniques for artificial intelligence and machine learning, has received great interest from both the academic and industrial worlds. The name, deep learning, comes from the use of an artificial neural network (ANN) with more than two hidden layers, implemented as parametric functions over the data, where the parameters are either learnable or trainable. Currently, deep learning has achieved remarkable successes in various artificial intelligence tasks, ranging from machine translation [3] to image recognition [17], and is still making advances in these tasks.

There are several factors behind the success of deep learning. Novel architectures [16,19] and algorithms [13,21] have been proposed to handle a huge amount of data, and highly efficient hardwares are developed for efficient data processing. However, the hardware such as graphical processing units (GPUs) designed to implement massively parallel processing of data is still not fast enough to train ANNs in a reasonable time. Thus, to overcome this limitation of the hardware, various forms of distributed processing have been proposed to accelerate the training process [8].

Taking into account the processing and communication aspects of distributed processing, there exist two types of systems, asynchronous and synchronous training systems [22]. In the context of training large-scale ANNs, asynchronous training systems are more preferred due to their fault-tolerant characteristics [4,8]. The basic form of asynchronous training system consists of a single parameter server and multiple data servers, whose operating mechanism is described as follows.

First, the parameter server distributes the entire data to each data servers. Then, the data server carries out the two phases below.

Phase I: The data server shuffles the data received from the parameter server, incrementally constructs chunks one at a time such that all chunks consist of the same number of data, and then sequences the chunks in random order. In the neural networks terminologies, these chunks are typically called mini-batches over which one gradient computation is performed.

Phase II: The data server analyzes the data in the current chunk based on the parameter received from the parameter server, and then transmits the results to the parameter server. Next, the parameter server updates the parameter based on the results received from the data server and transmits the updated parameter to the data server. Note that initially, the parameter server transmits the initial parameter to all data servers, and this process is iterated until all chunks in each data server are processed.

The main weakness of this asynchronous training system is a stale problem which makes the training speed slow [22, 24]. Note that the stale problem refers to the phenomena in which the results of a particular data server happen to be outdated and hence updates by that server impede the entire training procedure. Ideally, the stale problem can be completely resolved through simultaneous updates by the data servers. This requires that the sizes of each chunk are identical, the processing capacities of data servers are identical, and any communication delays are ignorable. However, it is impossible to satisfy these requirements in the real-world situations.

In this paper, we propose the modification of Phase I for the asynchronous training system to mitigate the stale problem, which can be specifically stated as follows:

- All data servers share cycles of the same length, denoted $\triangle$, by which consecutive updates are occurred.
- After constructing the set of chunks, the data server processes the chunks in the appropriate order.

The cycles of a predefined length act as an agreement among data servers under which each data server try to complete the processing of their individual chunk as close as possible within the cycle. Then, the staleness for each data server can be represented as the total deviation between shared cycles and the realized updates.

In this paper, we propose a machine scheduling formulation for the modification of Phase I above. This formalism helps one handle the proposed modification by using the previous results in the filed of the machine scheduling.

The remainder of the paper is organized as follows. In Section II, we introduce machine scheduling formulation of the proposed modification. Relevant literature in the context of machine scheduling is reviewed in Section III. Section IV presents a polynomially solvable case. Finally, we complete the paper with concluding remarks.

Ⅱ. Machine scheduling formulation

In this paper, the scheduling problem is to find the sequence of chunks for a data server minimizing the total staleness.

Our problem can be formally stated as follows. Let N be the dataset received from the parameter server and q_i be the processing time of datum $i \in N$. Assume that the data server has constructed an n-partition $(N_1,\cdots,N_n)$ of N, where the sizes of each partition j, $P_j = \sum_{i \in N_j} q_i$ are as close to each other as possible, that is, $\frac{p_{\max}}{p_{\min}} \le \tau$, where $p_{\min}$ and $p_{\max}$ are the minimum and the maximum lengths of chunks, respectively and $\tau > 1$ is a sufficiently small threshold. Henceforth, let a chunk be referred to as a job, and $J = \{1,\cdots,n\}$ be the job set. Let $J_\alpha = \{j : p_j \le \Delta \text{ for } j \in J\}$ and $J_\beta = J \setminus J_\alpha$. Let the jobs in J_α and J_β be referred to as small and large jobs, respectively. Let n_α and n_β be the cardinality of J_α and J_β, respectively. Note that the hth due date is calculated as $h\Delta$, $h = 1,2,...,n$. Let $\pi = (\pi(1), \pi(2), \pi(n))$ be a sequence, where $\pi(j)$ is the jth job in π. Let $\pi^{-1}(j)$ and $C_j(\pi)$ be the completion order and the completion time of job j under π. The objective is to find a sequence π that minimizes

$$z(\pi) = \sum_{h=1}^{n} (E_{\pi(h)}(\pi) + T_{\pi(h)}(\pi)),$$

where for each $h = 1,2,\cdots,n$,

$$E_{\pi(h)}(\pi) = \max\{0, h\Delta - C_{\pi(h)}(\pi)\}$$

and

$$T_{\pi(h)}(\pi) = \max\{0, C_{\pi(h)}(\pi) - h\Delta\}.$$

Note that $z(\pi)$ means the total staleness. Since the data server should not remain idle due to the expensive operating cost, the first job should be started at time 0 and no idle time between the consecutive jobs exists. Let the problem above be referred to as Problem P.

Ⅲ. Literature review

Problem P belongs to the class of the single-machine scheduling problems with generalized due dates (GDD) such that each due date is given not for a specific job, but for a specific position. The scheduling problem with GDD was introduced by Hall [11]. Hall [11] and Hall et al. [12] established the computational complexities for the single-machine cases with various objective function (e.g., maximum lateness, total weighted completion time, total weighted tardiness and weighted number of tardy jobs). Since the case to minimize the total weighted tardiness is related with our problem, we focus on that case. It was known from Hall [11] to be polynomiall solvable if the weights are identical. Sriskandarajah [20] and Gao and Yuan [10] proved the NP-hardness and the strong NP-hardness, respectively.

The single-machine scheduling problem to minimize the total weighted earliness and tardiness has been initiated from Hall et al. [14] and Hall et al. [15]. They considered the case with a common due date. It was known from [14,15] that it is weakly NP-hard, and remains so even if the weights are identical. Wan and Yuan [23] considered the case with different due dates, and showed that it is strongly NP-hard even for the case with identical weights. Choi et al. [5] showed that the problem of Wan and Yuan [23] remains strongly NP-hard, even if the intervals between the consecutive due dates are identical. Recently, Choi et al. [6] considered the problem of Choi et al. [5], and analyzed how the computational complexity changes depending on the number of the large jobs. Choi and Park [7] considered a single-machine case with the identical intervals between the consecutive due dates, in which the objective is to minimize the total weighted number of early and tardy jobs. They analyzed the computational complexity for various cases.

Ⅳ. Computational complexities

In this section, we introduce the previous results for the computational complexity of Problem P, and develop a polynomial-time approach to solve the case with the predetermined precedence constraints among small and large jobs, respectively.

Proposition 1 [6] i) Problem P is NP-hard if the number of the large jobs exists between one and $n-1$.

ii) Problem P is polynomially solvable if the number of the large jobs is equal to zero or n.

In the context of distributed training of neural networks or other kinds of learning tasks in machine learning, the special case can be observed when curriculum learning [2] is considered. In the setting of curriculum learning, training data is fed into the neural network in a meaningful order which illustrates gradually more complex ones. Thus, jobs can be ordered in advance depending on the level of “difficulty”. Motivated from this, we further assume that the orders of small and large jobs are separately determined in advance. Formally, let

$$\theta_1(\pi)=(\theta_1(\pi,1),\theta_1(\pi,2),\cdots,\theta_1(\pi,n_\alpha))$$

and

$$\theta_2(\pi) = (\theta_2(\pi,1), \theta_2(\pi,2), \cdots, \theta_2(\pi,n_\beta))$$

be the precedence relations among the jobs in J_α and J_β under π, respectively, where job $\theta_1(\pi,j)$ ($\theta_2(\pi,j)$) is the job sequenced jth among all small (large) jobs in π. The next theorem claims that when the orders of small and large jobs are determined in advance, Problem P can be solved in polynomial time.

Theorem 1 If $\theta_1(\pi^*)$ and $\theta_2(\pi^*)$ are known in an optimal schedule π^*, then Problem P is polynomially solvable.

Proof. We prove the polynomiality by a reduction to the shortest path problem. Let $N(0,0)$ and $N(n_\alpha, n_\beta)$ be the source and the sink nodes, respectively. Let $N(a,b)$ be the node indicating that the jobs in

$$\{\theta_1(\pi,1), \theta_1(\pi,2), \cdots, \theta_1(\pi,n_a)\}$$

and

$$\{\theta_2(\pi,1), \theta_2(\pi,2), \cdots, \theta_2(\pi,n_b)\}$$

have been sequenced. Let $N(a,b)$ be connected to:

- if $a+1 \le n_\alpha$, $N(a+1,b)$ with length

$$\left| \sum_{j=1}^{a+1} p_{\theta_1(\pi^*,j)} + \sum_{j=1}^{b} p_{\theta_2(\pi^*,j)} - (a+b+1)\Delta \right|$$

- if $b+1 \le n_\beta$, $N(a,b+1)$ with length

$$\left| \sum_{j=1}^{a} p_{\theta_1(\pi^*,j)} + \sum_{j=1}^{b+1} p_{\theta_2(\pi^*,j)} - (a+b+1)\Delta \right|$$

The objective is to find the shortest path from the source to sink node in the reduced graph. Note that is reduction can be in $O(n^2)$. Note that the total number of edges is at most $O(n^2)$. Thus, since the reduced graph is acyclic, the shortest path problem can be solved in $O(n^2)$ by the algorithm in [1]. It is observed that the optimal schedule σ^* can be constructed immediately from the shortest path.

From the result of the above theorem, we can also deduce the complexity result of another interesting case. In Problem P, we allow each chunk to have different sizes. This situation arises when we train neural networks to process the data with different sizes such as audio, video and so on. In other circumstance, however, all data has the same size, e.g., the image data of a fixed resolution. In this case, we can assume that every chunk has the same size.

Corollary 2 If the processing times of the small (large) jobs are identical, Problem P is polynomially solvable.

Proof. It holds immediately from Theorem 1.

Ⅴ. Conclusions

We considered a scheme for training neural networks in asynchronous distributed setting. We formulated it as a single-machine scheduling problem with generalized due dates in which the objective is to minimize the total earliness and tardiness. This objective means the total staleness that makes the training speed slow. We developed an approach to solve the problem in polynomial time if the orders of the small and the large jobs in an optimal schedule are known in advance.

For future research, it would be interesting to investigate different settings of asynchronous training and their machine scheduling formulations. In practical senses, it is also interesting to verify the effectiveness of the proposed scheme by experimenting to train large-scale neural networks over massive training data.

References

[1] R.A. Ahuja, K. Mehlhorn, J.B. Orlin. “Faster algorithm for the shortest path problem,” Journal of the Association for Computing Machinery, Vol. 37, pp. 213-223, 1990.

[2] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. “Curriculum learning,” Proceedings of the 26th International Conference on Machine Learning, pp. 41-48, Montreal, Canada, June 2009.

[3] B. Bahdanau, K. Cho, and Y. Bengio. “Neural machine translation by jointly learning to align and translate.” Proceedings of 3rd International Conference on Learning Representations, San Diego, USA, May, 2015.

[4] T. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman. “Project ADAM: Building an efficient and scalable deep learning training system.” Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation, pp. 571-582, CO, USA, October. 2014.

[5] B.C. Choi, Y.H. Min, and M.J. Park. “Strong NP-hardness of minimizing total deviation with generalized and periodic due dates,” Operations Research Letters, Vol. 47, pp. 433-437, Sep. 2019.

[6] B.C. Choi, K.M Kim, Y.H. Min, and M.J. Park. “A single machine scheduling with generalized and periodic due dates,” Submitted to Discrete Applied Mathematics.

[7] B.C. Choi, and M.J. Park. “Just-in-time scheduling with generalized due dates and identical due date intervals,” Asian-Pacific Journal of Operational Research, Vol. 6, pp. 1-13, Nov. 2018.

[8] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A.Y. Ng. “Large scale distributed deep networks,” Proceedings of the 25th International Conference on Advances in Neural Information Processing Systems, pp. 1223-1231, Nevada, USA. December, 2012.

[9] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. “Learning hierarchical features for scene labeling,” IEEE Transactions in Pattern Aanlysis and Machine Intelligence, Vol. 35, pp. 1915-1929, 2013.

[10] Y. Gao and J. Yuan. “Unary NP-hardness of minimizing total weighted tardiness with generalized due dates,” Operations Research Letters, Vol. 44, pp. 92-95, 2006.

[11] N.G. Hall. “Scheduling problems with generalized due dates,” IIE Transactions, Vol. 18, pp. 220-222, 1986.

[12] N.G. Hall, W. Kubiak, S.P. Sethi. “Earliness-tardiness scheduling problems II: Deviation of completion times about a restrictive common due date,” Operations Research, Vol. 39, pp. 847-856, 1991.

[13] G.E. Hinton, S. Osindero, and Y.W. The. “A fast learning algorithm for deep belief nets,” Neural Computation, Vol. 18, pp. 1527-1554, 2006.

[14] N.G. Hall and M.E. Posner. “Earliness-tardiness scheduling problems I: Weighted deviation of completion times about a common due date,” Operations Research, Vol. 39, pp. 836-846, 1991.

[15] N.G. Hall, S.P. Sethi, and S. Srikandarajah. “On the complexity of generalized due date scheduling problems,” European Journal of Operational Research, Vol. 51, pp. 100-109, 1991.

[16] K. He, X. Zhang, S. Ren, and J. Sun. "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," Proceedings of the 2015 International Conference on Computer Vision, pp. 1026-1034, Santiago, Chile, December, 2015.
[17] A. Krizhevsky, I. Sutskever, and G. Hinton. "ImageNet classification with deep convolutional neural networks," Proceedings of 25th International Conference on Advances in Neural Information Processing Systems, pp. 1090-1098, Nevada, USA. December, 2012.
[18] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning," Nature, Vol. 521, pp. 436-444, 2015.
[19] V. Nair and G. Hinton. "Rectified linear units improve restricted Boltzmann machines," Proceedings of the 27th International Conference on Machine Learning, pp. 807-814, Haifa, Israel, June. 2010.
[20] S. Srikandarajah. "A note on the generalized due dates scheduling problem," Naval Research Logistics, Vol. 37, pp. 587-597, 1990.
[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: A simply way to prevent neural networks from overfitting," Journal of Machine Learning Research, Vol. 15, pp. 1929-1958, 2014.
[22] J. Tsitsiklis, D. Bertsekas, M. Athans. "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," IEEE Transactions on Automatic Control, Vol. 31, pp. 803-812, 1986.
[23] L. Wa and J. Yuan. "Single-machine scheduling to minimize the total earliness and tardiness is strongly NP-hard," Operations Research Letters, Vol. 41, pp. 363-365, 2013.
[24] W. Zhang, S. Gupta, X. Lian, and J. Liu. "Staleness-aware async-SGD for distributed deep learning," Proceedings of the 25th International Joint Conference on Artificial Intelligence, pp. 2350-2356, New York, USA, July. 2016.

Acknowledgement

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2019040403) .

About the Author

최 병 천(Byung-Cheon Choi)
·Dr. Choi received the B.S. and Ph.D. in Industrial Engineering from Seoul National University in 2000 and 2006, respectively. He has joined the faculty of School of Business at Chungnam National University in 2009, and is currently an professor. He is interested in combinatorial optimization, scheduling theory, and operations management.

민 윤 홍(Yunhong Min)
·Dr. Min received the B.S. degree in Industrial Engineering &Management Science from POSTECH, and Ph.D. in Industrial Engineering from Seoul National University in 2006 and 2012, respectively. He has joined the faculty of Graduate School of Logistics at Incheon National University in 2017. Before joining Incheon National University, he was a research staff member of Samsung Advanced Institute of Technology. He is interested in mathematical optimization, machine learning and their applications to logistics and supply chain management.