Information Sciences 373 (2016) 404–418

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A zero-gradient-sum algorithm for distributed cooperative learning using a feedforward neural network with random weights☆

Wu Ai[a,c], Weisheng Chen[b,*], Jin Xie[a]

[a] School of Mathematics and Statistics, Xidian University, Xi'an 710071, PR China
[b] School of Aerospace Science and Technology, Xidian University, Xi'an 710071, PR China
[c] College of Science, Guilin University of Technology, Guilin 541004, PR China

ARTICLE INFO

Article history:
Received 30 January 2016
Revised 28 August 2016
Accepted 5 September 2016
Available online 7 September 2016

Keywords:
Distributed cooperative learning
Feedforward neural network with random weights (FNNRW)
Consensus
Distributed optimization

ABSTRACT

This paper focuses on developing new algorithms for distributed cooperative learning based on zero-gradient-sum (ZGS) optimization in a network setting. Specifically, the feedforward neural network with random weights (FNNRW) is introduced to train on data distributed across multiple learning agents, and each agent runs the program on a subset of the entire data. In this scheme, there is no requirement for a fusion center, due to, e.g., practical limitations, security, or privacy reasons. The centralized FNNRW problem is reformulated into an equivalent separable form with consensus constraints among nodes and is solved by the ZGS-based distributed optimization strategy, which theoretically guarantees convergence to the optimal solution. The proposed method is more effective than the existing methods using the decentralized average consensus (DAC) and alternating direction method of multipliers (ADMM) strategies. It is simple and requires less computational and communication resources, which is well suited for potential applications, such as wireless sensor networks, artificial intelligence, and computational biology, involving datasets that are often extremely large, high-dimensional and located on distributed data sources. We show simulation results on both synthetic and real-world datasets.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

The topic of learning in distributed environments has received increased attention in recent years [1,3,8,28,43]. Research in this area has focused on the aspect where the learning behavior is geographically or logically scattered among a network of learning agents, in which all these agents perform their tasks cooperatively. Particularly, the distributed learning problem arises when training data are gathered from a set of agents connected in a network, but their communication to a fusion center is costly or even unavailable. Three basic reasons for such a situation to occur can be identified: (i) Increasingly, big-data applications in machine learning often require dealing with large-scale datasets or datasets available at different

☆ This work was supported in part by the National Natural Science Foundation of China (Nos. 61673308, 61673014, 61503292, 61203074 and 61174213), the Natural Science Foundation of Guangxi (Nos. 2014GXNSFBA118023 and 2015GXNSFBA139259) and the Guangxi Universities Scientific Research Project (Nos. 2013LX043, 2013YB102, and KY2015ZL095).

* Corresponding author.
E-mail addresses: aiwu818@gmail.com (W. Ai), wshchen@126.com (W. Chen), Static_6417@163.com (J. Xie).

http://dx.doi.org/10.1016/j.ins.2016.09.016
0020-0255/© 2016 Elsevier Inc. All rights reserved.

locations, but these applications are too difficult to implement by centralized methods. (ii) As the diversity of data sources and formats increases, it is essential to build distributed learning systems with scalability and robustness. (iii) For some practical cases (e.g., knowledge discovery from clinical or financial records), the agents need a system that does not ask them to directly share their data for privacy and security reasons.

Since distributed algorithms are generally preferred over centralized solutions in large-scale systems, many research efforts have been devoted to devising strategies for distributed learning systems [6,14,26,37,42,43,49]. Several approaches have been developed for this problem, including the use of incremental strategies [26], diffusion strategies [5,7,9,27,50], consensus strategies [13,16,49], and alternating direction method of multipliers (ADMM) strategies [3,31,32,42,43]. In particular, Predd et al. [37] discuss the problems and challenges of distributed learning in wireless sensor networks (WSNs). In [26], the cooperative linear estimation problem is addressed using incremental techniques, where information flows in a sequential manner from one agent to the adjacent agent. This mode of operation requires a cyclic pattern of collaboration among the agents, and it tends to require the least amount of communications and power [38]. In [6], an adaptive diffusion mechanism is introduced to optimize a sum of cost functions corresponding to multiple agents. The work [49] proposed a scheme for distributed sensor fusion based on average consensus, which is framed for the classical weighted least-square estimation in the network setting. In [14], a special consensus algorithm, called gossip [2], is applied to develop distributed support vector machines (SVMs). Recently, the alternating direction method of multipliers (ADMM) [3] has become very popular for solving optimization and learning problems in a distributed fashion. It has been used for developing distributed versions of the least-absolute shrinkage and selection operator (LASSO) [31], SVM [15], basis pursuit [32], random vector functional-link (RVFL) networks [43], and many others.

In the field of neural network research, single-layer feedforward neural networks (SLFNs) have been widely applied to solve problems such as classification and regression because of their universal approximation capability [20,21,24,36]. Due to its applicability for real-time data processing, a type of learning model with random parameters has attracted a surge of interest in recent years. In [45], the feedforward neural network with random weights (FNNRW) was introduced to analyze the functional behavior of SLFNs with respect to learning. Almost simultaneously, the effort in [35] established the foundation of a type of random-weights neural networks known as RVFL networks. They show that the actual values of the weights from the input layer to the hidden layer can be randomly generated in a suitable domain and kept fixed in the learning stage. Further investigations were presented in [21], where some theoretical results were established. Readers may refer to a recently published editorial for a more in-depth discussion of the randomized learner models and their relevant learning algorithms [48]. However, these centralized learning algorithms depend critically on the assumption that the entire training data set is available to a single processor, which is equipped with significantly more computation and computational resources in a distributed environment.

This paper aims to develop and analyze, in terms of convergence, a distributed FNNRW learning algorithm based on the zero-gradient-sum (ZGS) strategy (cf. Section 3.2), which relies on in-network data processing across a network. Although the use of incremental solvers (see, e.g., [38,39]) may be possible, they make communication inefficient by requiring a large number of information transmissions. In addition, incremental algorithms generally require the nodes of the network to construct a Hamiltonian cycle, i.e., a closed path that passes through every node once, which may be very difficult to carry out. The present work is motivated by the distributed cooperative learning framework developed in [8,16,43] and recent studies of distributed optimization (see, e.g., [29,30]). Firstly, the centralized FNNRW training problem is recast into an equivalent separable form with consensus constraints imposed on the output weights. Secondly, the ZGS-based distributed convex optimization tool is used to model the decentralized problem. Finally, the construction of the learning algorithm is obtained, based solely on information exchanges among neighboring nodes, which is provably convergent to the centralized FNNRW.

Our contribution is closest to the recent paper [43], which provides two distributed learning algorithms for RVFL networks using the decentralized average consensus (DAC) and ADMM strategies. These algorithms allow all the nodes to agree on a single model, whose testing performance is similar to that obtained by a centralized model. However, there are also some disadvantages to be considered: (i) Although the implementation of DAC-based RVFL is simple, the result of the averaging step is not theoretically equivalent to that of the centralized model (see Eq. (19) in [43]). The global output weight vector is just an arithmetic average of the local output weight vectors obtained at the nodes of the network. (ii) For a smaller sized model of DAC-based RVFL, the equivalent solution (see (20) in [43]) is obtained by executing two averaging steps on two groups of high-dimensional matrices, which is impractical to implement in situations with limited bandwidth. (iii) Although the ADMM-based strategy ensures convergence to the global optimum, the solution of the algorithm still introduces two averaging steps (see (27) in [43]), which demand more computational time than the DAC-based strategy. In contrast with the type of algorithms designed in [43], our method is more effective because it theoretically converges to a solution of the equivalent form using the whole set of training data, which is essential for distributed application to big data problems. Furthermore, our method runs through only one consensus step, with the initialization obtained independently using local centralized FNNRW on subsets of the training data. Our strategy is simpler and requires fewer computational and communication resources.

The ZGS-based FNNRW (ZGS-FNNRW), proposed in this paper, also has no notion of a centralized fusion center or a master coordinator, making it well suited for distributed applications. Moreover, the proposed strategy exhibits the following features.

Download English Version:

https://daneshyari.com/en/article/4944913

Download Persian Version:

https://daneshyari.com/article/4944913

Daneshyari.com