Distributed learning for Random Vector Functional-Link networks

Simone Scardapane [a],*, Dianhui Wang [b], Massimo Panella [a], Aurelio Uncini [a]

[a] Department of Information Engineering, Electronics and Telecommunications (DIET), "Sapienza" University of Rome, Via Eudossiana 18, 00184 Rome, Italy
[b] Department of Computer Science and Computer Engineering, La Trobe University, Melbourne, VIC 3086, Australia

ARTICLE INFO

ABSTRACT

This paper aims to develop distributed learning algorithms for Random Vector Functional-Link (RVFL) networks, where training data is distributed under a decentralized information structure. Two algorithms are proposed by using Decentralized Average Consensus (DAC) and Alternating Direction Method of Multipliers (ADMM) strategies, respectively. These algorithms work in a fully distributed fashion and have no requirement on coordination from a central agent during the learning process. For distributed learning, the goal is to build a common learner model which optimizes the system performance over the whole set of local data. In this work, it is assumed that all stations know the initial weights of the input layer, the output weights of local RVFL networks can be shared through communication channels among neighboring nodes only, and local datasets are blocked strictly. The proposed learning algorithms are evaluated over five benchmark datasets. Experimental results with comparisons show that the DAC-based learning algorithm performs favorably in terms of effectiveness, efficiency and computational complexity, followed by the ADMM-based learning algorithm with promising accuracy but higher computational burden.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Over the past decades, supervised learning techniques have been well developed with theoretical analyses and empirical studies [15]. The ICT world, however, is being rapidly reshaped by emerging trends such as big data [22], pervasive computing [33], commodity computing [11], Internet of things [2], and several others. All these frameworks have a similar common theme underlying them: computing power is now a widespread feature surrounding us, and the same can be said about data. Consequently, supervised learning is expected to face major technological and theoretical challenges, since in many situations the overall training data cannot be assumed to lie at a single location, nor is it realistic to have a centralized authority for collecting and processing it. The previous trends also put forth the challenge of analyzing *structured* and *heterogeneous* data [4], however, we are not concerned with this issue in this paper.

As a prototypical example, consider solving a music classification task (e.g., genre classification [35]) over a peer-to-peer (P2P) network of computers, each node possessing its own labeled database of songs. It is reasonable to assume that, to obtain good performance, no single database may be sufficient, and there is the need of leveraging over the data of *all* users.

* Corresponding author. Tel.: +39 06 44585495; fax: +39 06 4873300.
  E-mail addresses: simone.scardapane@uniroma1.it (S. Scardapane), dh.wang@latrobe.edu.au (D. Wang), massimo.panella@uniroma1.it (M. Panella), aurelio.uncini@uniroma1.it (A. Uncini).
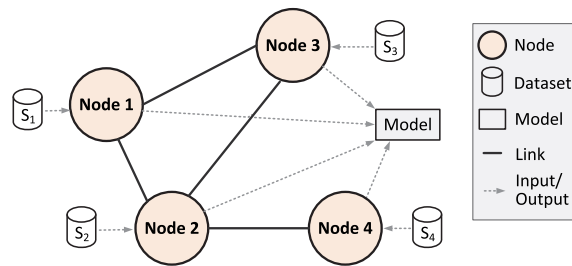
Fig. 1. Supervised learning in a network of agents: data is distributed throughout the nodes, and all of them must converge to a single learning model. For readability, we assume undirected connections between agents.

However, in a P2P network no centralized authority exists, hence the nodes need a *distributed* training protocol to solve the classification task. In fact, it is known that a fully decentralized training algorithm can be useful even in situations where having a master node is technologically feasible [16]. In particular, such a distributed algorithm would remove the risks of having a single point of failure, or a communication bottleneck towards the central node. Similar situations are also widespread in Wireless Sensor Networks (WSN), where additional power concerns arise [5]. Finally, it may happen that data simply cannot be moved across the network: either for being large (in term of number of examples or dimensionality of each pattern), or because fundamental privacy concerns are present [40]. The general setting, which we will call 'data-distributed learning', is graphically depicted in Fig. 1.

So far, a large body of research has gone into developing fully distributed, decentralized learning algorithms, including works on diffusion adaptation [21,34], learning by consensus [16], distributed learning on commodity clusters architectures [8], adaptation on WSNs [5,32], distributed online learning [13], distributed optimization [6,14,18,38], ad-hoc learning algorithms for specific architectures [12,26], distributed databases [20], and others. Despite this, many important research questions remain open [31], and in particular several well-known learning models, originally formulated in the centralized setting, have not yet been generalized to the fully decentralized setting.

In this paper, we propose two distributed learning algorithms for a yet-unexplored model, that is Random Vector Functional-Link (RVFL) networks [1,10,30,37]. As illustrated successively, RVFLs can be viewed as feedforward neural networks with a single hidden layer, resulting in a linear combination of a (fixed) number of non-linear expansions of the original input. A remarkable characteristics of such a learner model lies in the way of parameter assignment, that is, the input weights and biases are randomly chosen and fixed in advance before training. Despite this simplification, RVFLs can be shown to possess universal approximation capabilities, provided a sufficiently large set of expansions [17]. This grant them with a number of peculiar characteristics, making them particularly suited in a distributed environment. In particular, RVFL models are linear in the parameters, thus optimal parameters can be found with a standard linear regression routine, which can be implemented efficiently even in low-cost hardware, such as sensors or mobile devices [30]. In fact, the optimum of the training problem can be formulated in a closed form, involving only matrix inversions and multiplications, making the model efficient even when confronted with large amounts of data. Finally, the same formulation can be used equivalently in the classification and in the regression setting. In this paper, we focus on *batch* learning scheme development, however the proposed algorithms can be further extended for sequential learning with the use of standard gradient-descent procedures [10], whose decentralized formulation have been only partially investigated in the literature [34].

The key idea behind the proposed algorithms is to let all nodes train a local model (simultaneously) using the subset of training data, followed by finding the common output weights of the master learner model. Two effective approaches for defining the common output weights are adopted in this study. One is the Decentralized Average Consensus (DAC) strategy [28], and another is the well-known Alternating Direction Method of Multipliers (ADMM) algorithm [6]. DAC is an efficient protocol to compute averages over very general networks, with two main characteristics. Firstly, it does not require a centralized authority coordinating the overall process, and secondly, it can be easily implemented even on the most simple networks [16]. These characteristics have made DAC an attractive method in many distributed learning algorithms, particularly in the 'learning by consensus' theory outlined in [16]. From a theoretical viewpoint, the DAC-based algorithm is similar to a bagged ensemble of linear predictors [7], and despite its simplicity and non-optimal nature, our experimental simulations show that it results in highly competitive performance. The second strategy (ADMM) is the most widely employed distributed optimization algorithm in machine learning (e.g. for LASSO [6] and Support Vector Machines [14]), making it a natural candidate for the current research. This second strategy is more computational demanding than the DAC-based one, but it has high theoretical guarantees in term of convergence, speed and accuracy. Our simulation results obtained from both algorithms are quite promising and comparable to a centralized model exploiting the overall dataset. Moreover, the consensus strategy is extremely competitive on a large number of realistic network topologies.

The remainder of the paper is organized as follows. Section 2 briefly reviews RVFLs and its learning algorithm, and introduces the DAC algorithm. Section 3 describes the data-distributed learning framework, and proposes two training algorithms for RVFLs models. Sections 4 and 5 detail the experimental setup and the numerical results on four realistic datasets, plus an additional experiment on a large-scale image classification task, respectively. Section 6 concludes this paper with some discussions and future possible researches.