Expert Systems With Applications 106 (2018) 233–243

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Building feedforward neural networks with random weights for large scale datasets

Hailiang Ye[a], Feilong Cao[a,*], Dianhui Wang[b,c], Hong Li[d]

[a] Department of Applied Mathematics, China Jiliang University, Hangzhou 310018, Zhejiang, China
[b] Department of Computer Science and Information Technology, La Trobe University, Melbourne, VIC 3086, Australia
[c] State Key Laboratory of Synthetical Automation for Process Industries, Northeast University, Shenyang 110819, Liaoning, China
[d] School of Mathematics and Statistics, Huazhong University and Science and Technology, Wuhan 430074, Hubei, China

ARTICLE INFO

Article history:
Received 11 August 2017
Revised 27 March 2018
Accepted 5 April 2018
Available online 11 April 2018

Keywords:
Large scale data
Neural networks
Learning
Approximate Newton-type method

ABSTRACT

With the explosive growth in size of datasets, it becomes more significant to develop effective learning schemes for neural networks to deal with large scale data modelling. This paper proposes an iterative approximate Newton-type learning algorithm to build neural networks with random weights (NNRWs) for problem solving, where the whole training samples are divided into some small subsets under certain assumptions, and each subset is employed to construct a local learner model for integrating a unified classifier. The convergence of the output weights of the unified learner model is given. Experimental results on UCI datasets with comparisons demonstrate that the proposed algorithm is promising for large scale datasets.

1. Introduction

Gradient-based optimization methods, such as back-propagation algorithm (BP), usually suffer from local minima, slow convergence, and the sensitive setting of the learning rate. Alternatively, Schmidt, Kraaijveld, and Duin (1992) proposed feedforward neural networks with random weights (NNRWs), where the input weights and biases are assigned randomly with uniform distribution in $[-1, 1]$, and the output weights can be determined analytically by using the well-known least squares method. Such a randomized learner model was not proposed as a working algorithm, but simply as a simple idea to investigate some characteristics of the feedforward neural networks. Theoretically, it is evident that such a way to randomly assign the input weights and biases in $[-1, 1]$ cannot guarantee the universal approximation capability (Li & Wang, 2016; Wang & Li, 2017). Thus, statements on its approximation capability and good generalization in the literature are all misleading and lack scientific justification. A similar idea was proposed by Pao's group (Pao & Takefuji, 1992) and they termed such randomized learning models as random vector functional-link nets (RVFLs). In Igelnik and Pao (1995), theoretical justification on the universal approximation capability of RVFLs was established, where the scope of randomly assigned input weights and biases is specified in a constructive manner. Recently, some advanced randomized learning algorithms have been developed in Cao, Tan, and Cai (2014), Scardapane, Wang, Panella, and Uncini (2015), and Cao, Ye, and Wang (2015). Further, a complete exposition on randomized methods for neural networks has been published in Scardapane and Wang (2017). A substantial advancement of randomized methods for the development of neural networks was presented by Wang and Li (2017), where a constraint on the random weights and biases depending on data was proposed to ensure the universal approximation property. However, it has been aware that the way of computing the output weights in NNRW model is time-consuming and even does not work on desktop computers as the size of data samples or the number of nodes at the hidden layer of neural networks become very large.

Over the past decades, there have been a lot of researches on large scale data modelling problems. Perhaps the simplest strategy for dealing with large scale datasets is to reduce the size of dataset by subsampling. This idea behind this approach is to decompose the problem into a series of smaller tasks. Indeed, this decomposition method was firstly proposed by Osuna, Freund, and Girosi (1997a,b), followed by Lu and Ito (1999) for solving pattern classification problem. Some improved methods on support vector machines (SVMs) with the decomposition scheme were proposed to deal with large scale data and successfully applied for data regression and classification (Collobert & Bengio, 2001; Collobert, Bengio, & Bengio, 2002; Dong, Krzyzak, & Suen, 2005; Flake & Lawrence, 2002; Hsieh, Chang, Lin, Keerthi, & Sundararajan, 2008; Tsang,

* Corresponding author.
E-mail addresses: yhl575@163.com (H. Ye), flcao@cjlu.edu.cn (F. Cao), dh.wang@latrobe.edu.au (D. Wang), hongli@hust.edu.cn (H. Li).

https://doi.org/10.1016/j.eswa.2018.04.007
0957-4174/

Kwok, & Cheung, 2005). Tresp et al. (Schwaighofer & Tresp, 2001; Tresp, 2000) proposed the Bayesian committee SVM to process the large scale data, where the dataset was divided into some subsets of the same size and some models were derived from the individual sets. But the condition was that these subsets must be pairwise independent. Furthermore, some other large scale datasets training methods based on the decomposition techniques exist in machine learning such as stochastic gradient methods (Mu, Liu, Liu, & Fan, 2017; Zhang, 2004), a first-order approach (Duchi, Shalev-Shwartz, Singer, & Tewari, 2010) and greedy column subset selection method (Farahat, Elgohary, Ghodsi, & Kamel, 2015). Also, various ensemble learning techniques have been developed, and many interesting ideas and theoretical works, including bagging, boosting and random forests can be found in Breiman (1996), Breiman (2001), Zhang and Ma (2012), Onan, Korukoğlu, and Bulut (2016), Li and Wang (2016), Gonález, Dominguez, Sánchez, and B (2017), Wei et al. (2017), Liu, Ouyang, and Li (2017), and Wang and Cui (2017). These methods are always related to large scale data modeling problems and share some common nature in system design, such as data sampling and the output integration. The basis of ensemble learning theory lies in a rational sampling implementation for building each learner model, which may provide a sound predictability through learning a subset of the whole datasets.

This paper addresses the study on NNRWs models for large scale datasets based on decomposition method. Existing common method to calculate the large scale data is directly dividing the data into a series of small subsets with the same size, and each subset is trained by a local learner independently and average the overall local solution as global solution. From (Zinkevich, Weimer, Li, & Alex J, 2010), we call this approach as One-Shot scheme. Naturally, if the learner is NNRW, then we term the approach as One-Shot NNRW.

This work is built on a framework proposed by Shamir, Srebro, and Zhang (2014). Our goal is to offer an iterative solution for building randomized learner models with large scale datasets. Specifically, we present an iterative method for NNRW based on approximate Newton-type method (ANE-NNRW), and develop an efficient ANE-NNRW iterative algorithm and its convergence. Moreover, we make some comparative studies on several benchmark to explore the usefulness, effectiveness, and efficiency of the proposed algorithm.

The rest of this paper is organized as follows. Section 2 briefly reviews some basic concepts on NNRW and its variant. Section 3 details the proposed ANE-NNRW algorithm. Section 4 provides an analysis on the convergence of ANE-NNRW. Section 5 evaluates the algorithm performance with comparisons. Section 6 concludes this paper with some remarks. Mathematical proofs are given in the appendix.

2. Preliminary

2.1. Revisit of neural networks with random weights

Feedforward neural networks (FNNs) have been widely applied in many fields (Erkaymaz, Ozer, & Perc, 2017; Ozer, Perc, Uzuntarla, & Koklukaya, 2010). FNNs with single hidden layer can be mathematically described as

$$G_N(x) = \sum_{i=1}^{N} \beta_i g(\langle \omega_i, x\rangle + b_i), \tag{1}$$

where N is the number of hidden nodes, $x = [x_1, x_2, \ldots, x_d]^\top \in \mathbf{R}^d$ is the input, g is the activation function, $b_i \in \mathbf{R}$ is the bias, $\omega_i = [\omega_{i1}, \omega_{i2}, \ldots, \omega_{id}] \in \mathbf{R}^d$ and $\beta_i \in \mathbf{R}$ are the input and output weights connecting the ith hidden node and the output node, respectively, and $\langle \omega_i, x\rangle = \sum_{j=1}^{d} \omega_{ij} x_j$ denotes the Euclidean inner product.

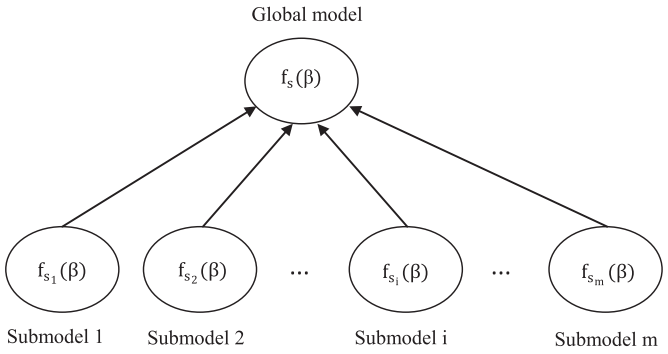

Fig. 1. The relationship between local models $f_{s_i}(\beta)(i = 1, 2, \ldots, m)$ and global model $f_s(\beta)$. Actually, for a large scale dataset, it can be separated into m small subsets given by $s = \{s_1, s_2, \ldots, s_m\}$ and m submodels are derived from the individual subsets.

For a set of training samples $s = \{(x_j, t_j) : x_j \in \mathbf{R}^d, t_j \in \mathbf{R}, j = 1, 2, \ldots, M\}$, let $\beta = [\beta_1, \beta_2, \ldots, \beta_N]^\top$, $\mathbf{T} = [t_1, t_2, \ldots, t_M]^\top$, and

$$\mathbf{H} = \begin{bmatrix} g(\langle \omega_1, x_1\rangle + b_1) & \ldots & g(\langle \omega_N, x_1\rangle + b_N) \\ \vdots & \ldots & \vdots \\ g(\langle \omega_1, x_M\rangle + b_1) & \ldots & g(\langle \omega_N, x_M\rangle + b_N) \end{bmatrix}. \tag{2}$$

If the input weights and biases are assigned randomly with uniform distribution, then the output weights can be determined analytically by using the well-known least-squares method:

$$\min_{\beta \in \mathbf{R}^N} \{\|\mathbf{H}\beta - \mathbf{T}\|_2^2\}. \tag{3}$$

which gives $\beta = \mathbf{H}^\dagger \mathbf{T}$, where $\mathbf{H}^\dagger = (\mathbf{H}^\top \mathbf{H})^{-1} \mathbf{H}^\top$ is the Moore–Penrose generalized inverse of $\mathbf{H}$ (Rao & Mitra, 1971).

Nevertheless, the least squares problem is usually ill-posed. So one can employ the following ℓ_2 regularization method (Tikhonov, 1963) to find the solution, i.e.,

$$\min_{\beta \in \mathbf{R}^N} \{\|\mathbf{H}\beta - \mathbf{T}\|_2^2 + \mu \|\beta\|_2^2\}, \tag{4}$$

where $\mu > 0$ is a positive constant called the regularizing factor. This model is referred as ℓ_2-NNRW and can improve the stability on the solution of NNRW (Cao, Wang, Zhu, & Wang, 2016; Cao et al., 2015). If μ is given such that $\mathbf{H}^\top \mathbf{H} + \mu \mathbf{I}$ is invertible, then the minimizer of (4) is easily described as

$$\beta = (\mathbf{H}^\top \mathbf{H} + \mu \mathbf{I})^{-1} \mathbf{H}^\top \mathbf{T}, \tag{5}$$

where $\mathbf{I}$ denotes the identity matrix.

Unfortunately, when the size of data samples or the number of hidden nodes becomes rather large, the computation of the inverse matrix in (3) and (4) is time-consuming and even ineffective. Accordingly, it is essential to develop fast and effective schemes for large scale datasets.

2.2. Decomposition model of neural networks with random weights

As mentioned before, it is not suitable to train large scale data for traditional NNRW method. One common approach is data decomposition, which can facilitate the difficulty resolved by breaking the data up into smaller ones and solving each of the smaller ones separately (see Fig. 1).

It is observed from Fig. 1 that the large scale dataset with M samples will be decomposed into m small subsets denoted by $s = \{s_1, s_2, \ldots, s_m\}$ and m submodels are derived from the individual subsets. Indeed, this data decomposition method should have some prior assumptions before learning. Firstly, each small subset should