ARTICLE IN PRESS

Neural Networks ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Neural Networks

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

2016 Special Issue

Scalable learning method for feedforward neural networks using minimal-enclosing-ball approximation

Jun Wang [a,*], Zhaohong Deng [a], Xiaoqing Luo [b], Yizhang Jiang [a], Shitong Wang [a]

[a] *School of Digital Media, Jiangnan University, Wuxi, Jiangsu, China*

[b] *School of Internet of Things Engineering, Jiangnan University, Wuxi, Jiangsu, China*

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Feedforward neural networks
Minimal enclosing ball
Scalable learning
Hidden feature space learning

ABSTRACT

Training feedforward neural networks (FNNs) is one of the most critical issues in FNNs studies. However, most FNNs training methods cannot be directly applied for very large datasets because they have high computational and space complexity. In order to tackle this problem, the CCMEB (Center-Constrained Minimum Enclosing Ball) problem in hidden feature space of FNN is discussed and a novel learning algorithm called HFSR-GCVM (hidden-feature-space regression using generalized core vector machine) is developed accordingly. In HFSR-GCVM, a novel learning criterion using L2-norm penalty-based ε-insensitive function is formulated and the parameters in the hidden nodes are generated randomly independent of the training sets. Moreover, the learning of parameters in its output layer is proved equivalent to a special CCMEB problem in FNN hidden feature space. As most CCMEB approximation based machine learning algorithms, the proposed HFSR-GCVM training algorithm has the following merits: The maximal training time of the HFSR-GCVM training is linear with the size of training datasets and the maximal space consumption is independent of the size of training datasets. The experiments on regression tasks confirm the above conclusions.

1. Introduction

Neural networks have strong ability to approximate complex nonlinear functions from the input samples and can provide accurate models for a large class of natural and artificial phenomena. Out of many kinds of neural networks, feedforward neural networks (FNNs) play an important role in practical applications. The standard backpropagation (BP) algorithm (Rumelhart, Hinton, & Williams, 1986) is a typical method for training FNNs. It is based on the gradient descent algorithm well known in optimization theory. However, it is not suitable for large-scale problems since it has a poor convergence rate and depends on user-specified parameters. In order to eliminate these disadvantages, several methods, including conjugate gradient methods (Charalambous, 1992; Möller, 1990), Levenberg–Marquardt (LM) methods (Ampazis & Perantonis, 2002; Hagan & Menhaj, 1994; Moré, 1978) and extreme learning machine (ELM) (Huang & Chen, 2007; Huang, Zhu, & Siew, 2006) have been developed and fully investigated. From these studies, it is evident that these methods are effective in medium to large scale problem since they can train the same network from 10 to 100 times faster than the standard BP.

Recently, hidden-feature-space ridge regression (HFSR) has been proposed for training FNNs. It integrated the idea of ridge regression with hidden-feature-space learning. According to theoretical and experimental studies in Wang, Chung, and Wang (2015), HFSR has demonstrated the following characteristics: (1) The parameters of hidden layer can be generated randomly and the weights connecting hidden layer and output layer can be solved analytically. (2) Compared with the BP based training algorithms, HFSR has shown excellent performance in training time, especially for small and middle size of datasets, which is usually faster than that of BP with tens of times. However, HFSR also confronts some challenges except for its distinctive characteristics. One challenge is that the space complexity is too high for large-scale applications or the network model with large number of hidden nodes due to the necessity of solving the inverse of the matrix in the matrix equation, which makes HFSR infeasible on personal computers with limited memory in these situations. On the other hand, the computational complexity of matrix inversion is between quadratic and cubic with respect to the training size and this still requires plenty of training time for large-scale problems, even though it is much faster than many other traditional methods.

* Corresponding author.
E-mail address: wangjun_sytu@hotmail.com (J. Wang).

http://dx.doi.org/10.1016/j.neunet.2016.02.005
0893-6080/

During the past decade, a variety of learning approaches for very large datasets have been proposed in the context of kernel methods. By applying the criterion of maximizing the separating margins of two different classes, the learning problem can be formulated as a quadratic programming (QP in abbreviation) problem which has the important computational advantage of not suffering from the problem of local minima. However, given training N patterns, a naive implementation of the QP solver takes $O(N^3)$ training time. In order to scale up QP and find QP solutions on very large datasets effectively, many methods have been proposed in recent years (Achlioptas, McSherry, & Schölkopf, 2002; Chu, Ong, & Keerthi, 2005; Fine & Scheinberg, 2001; Smola & Schölkopf, 2000; Tsang, Kwok, & Cheung, 2005; Tsang, Kwok, & Zurada, 2006; Wang, Wang, & Chung, 2014; Williams & Seeger, 2001). These methods included the Nystrom method (Williams & Seeger, 2001), greedy approximation (Smola & Schölkopf, 2000), sampling (Achlioptas et al., 2002; Wang et al., 2014), matrix decompositions (Fine & Scheinberg, 2001) and so on. Among these works, the generalized core vector machine (GCVM) proposed by Tsang et al. (2005, 2006) achieves an asymptotic time complexity that is linear in N and a space complexity that is independent of N by utilizing an approximation algorithm for the CCMEB (Center-Constrained Minimum Enclosing Ball) problem in computational geometry. Experiments on very large datasets for both classification and regression tasks demonstrated that GCVM is as accurate as existing kernel methods implementations, but is much faster and can handle much larger datasets than existing scale-up methods.

In order to solve the learning problems for FNNs on very large datasets, in this work, the connection between FNN training and GCVM is built. A novel learning algorithm called HFSR-GCVM (hidden-feature-space regression using generalized core vector machine) is developed, in which a novel learning criterion using L2-norm penalty-based ε-insensitive function is formulated. In HFSR-GCVM, the merits of GCVM are integrated to solve the QP problem on very large datasets. Generally speaking, the following contributions have been made in this paper:

(1) Existing learning methods for FNNs such as ELM and HFSR were originally developed based on the learning criterion of least square error (Wang et al., 2015). Differentiated from existing works, this work manages to extend the learning criterion of FNNs to a novel learning criterion, in which the flatness of the objective function and the training error are minimized simultaneously.
(2) A novel function approximation algorithm HFSR-GCVM is proposed based on CCMEB approximation. In HFSR-GCVM, the input weights and the hidden layer biases can be randomly assigned and a wide type of feature mappings can be utilized. This is quite different from Tsang's GCVM, in which the kernel parameters should be determined by cross validation and rigorous Mercer condition for kernel functions should be required.
(3) GCVM was originally developed for kernel methods on large datasets. This work manages to extend GCVM to FNNs training on very large datasets. It is shown that a wide type of feature mappings (hidden-layer output functions), including random hidden nodes and kernels, can be utilized. With this extension, the GCVM solution can be obtained for FNNs, RBF network and kernel methods.

2. Feedforward neural networks overview

Fig. 1 shows the structure of FNNs, which includes input layer, hidden layers and output layer. Like traditional FNNs, all the nodes in FNNs are connected to the nodes in the adjacent layers through unidirectional branches and the connection between nodes within one layer is not allowed. The input layer serves to transmit input signal to the first hidden layer and the output nodes in the output layer construct the response vector of FNN. Each node in hidden layers is combined with a linear combiner and an activation function, whose output is the response of the nodes. Notice that in FNNs, the activation function can take any infinite differentiability functions such as sigmoidal functions, decaying RBF functions, Mexican Hat wavelet function, Morlet wavelet function and fuzzy basis functions.

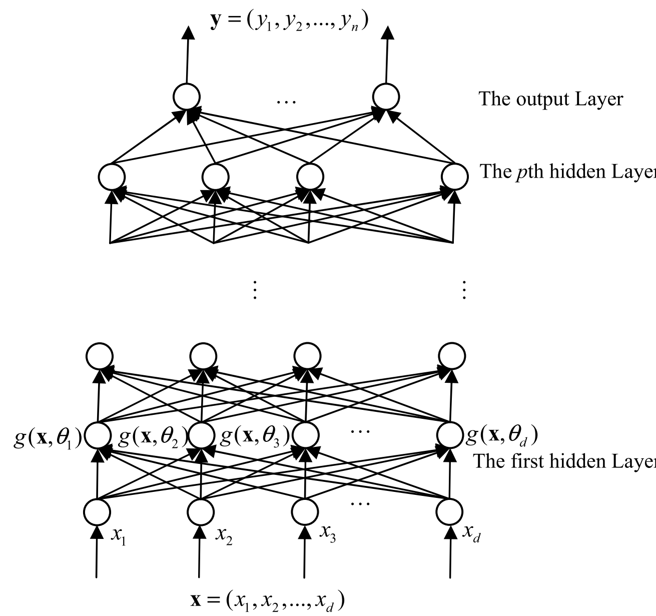

Fig. 1. Structure of FNNs.

Let $\mathbf{x}^{(0)} = [x_1^{(0)}\ x_2^{(0)}, \ldots, x_d^{(0)}]^T$ be the input data and p be the number of hidden layer. The output in $(l+1)$th hidden layer can be calculated as

$$\mathbf{x}^{(l+1)} = g_l(\mathbf{W}^{(l)}\mathbf{x}^{(l)} + \mathbf{b}^{(l)}), \quad l = 0, 1, \ldots, p$$

and the output of the network can be calculated as

$$\mathbf{y}^{(o)} = \mathbf{W}^{(p)}\mathbf{x}^{(p)} + \mathbf{b}^{(p)}$$

where $\mathbf{W}^{(l)}$, $l = 1, 2, \ldots,$ is a weight matrix connecting the lth hidden layer and $(l+1)$th hidden layer, $\mathbf{b}^{(l)}$ is the bias vector of the lth hidden layer and $g_l(\cdot)$ is the activation function of the lth hidden layer. In order to train FNNs, an efficient learning mechanism is needed to adjust all the weights of the connections. Based on the structure in Fig. 1, there are three main approaches in the training of FNNs. (1) gradient-descent based (e.g. backpropagation (BP) method for multi-layer FNNs Rumelhart et al., 1986); (2) least square error based (e.g. extreme learning machines (ELMs) Huang, Zhu, & Siew, 2006 for the single-hidden-layer feedforward networks (SLFNs), hidden-feature-space ridge regression for the multiple-hidden layer feedforward networks (HFSR)) (Wang et al., 2015); (3) standard optimization method based (e.g. support vector machines (SVMs) Cortes & Vapnik, 1995 for the so-called support vector network).

Among these works, one representative approach is hidden-feature-space ridge regression HFSR, which originated from the classical ridge regression (Wang et al., 2015). In HFSR, for arbitrary sample $\mathbf{x}$ in D, an L-dimensional hidden feature space can be constructed from all the d-dimensional input samples in D by presetting L infinitely differential functions $g(\mathbf{x}, \theta_1), g(\mathbf{x}, \theta_2), \ldots, g(\mathbf{x}, \theta_L)$ as the corresponding mapping functions, which can be implemented by the activation functions in the hidden layer of FNNs. The space generated by the activation functions of the hidden layer constructs the hidden feature space