Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER Applied Mathematics and Computation 174 (2006) 34–50
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

Improved constrained learning algorithms by incorporating additional functional constraints into neural networks

Fei Han [a,b,*], De-Shuang Huang [a]

[a] *Intelligent Computing Lab., Hefei Institute of Intelligent Machines, Chinese Academy of Sciences, P.O. Box 1130, Hefei Anhui 230031, China*
[b] *Department of Automation, University of Science and Technology of China, Hefei 230027, China*

Abstract

In this paper, two improved constrained learning algorithms that are able to guarantee to obtain better generalization performance are proposed. These two algorithms are substantially on-line learning ones. The cost term for the additional functionality of the first improved algorithm is selected based on the first-order derivatives of the neural activation at hidden layers, while the one of the second improved algorithm is selected based on second-order derivatives of the neural activation at hidden layers and output layer. In the course of training, the cost terms selected from these additional cost functions can penalize the input-to-output mapping sensitivity or high-frequency components included in training data. Finally, theoretical justifications and simulation results are given to verify the efficiency and effectiveness of the two proposed learning algorithms.
© 2005 Elsevier Inc. All rights reserved.

* Corresponding author. Intelligent Computing Lab., Hefei Institute of Intelligent Machines, Chinese Academy of Sciences, P.O. Box 1130, Hefei Anhui 230031, China.
E-mail address: hanfei1976@iim.ac.cn (F. Han).

0096-3003/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2005.04.073

Keywords: Feedforward neural network; On-line constrained learning algorithm, Generalization; Mapping sensitivity; High-frequency components, Time series prediction

1. Introduction

Good generalization performance is one of most important tasks in designing neural network model. However, a proper matching of the underlying problem complexity and the network complexity is crucial for improving the network generalization capability [1]. It is well known that although the network with too many synaptic weights can solve exactly the involved problems, it may suffer from overfitting, thus result in poor generalization performance. On the other hand, while a smaller network cannot solve the problem accurately, it can obtain a better generalization capability [2,3]. So, in practical applications, we should make a proper compromise between the generalization capability and the network complexity. In order to avoid occurring the overfitting case, usually, ones have to reduce the network complexity by pruning synaptic weights or increase the problem complexity by incorporating additional functional constraints into the network [4–10]. In literature [11], a learning algorithm was proposed that supplemented the training phase in feedforward neural networks (FNN) with the architectural constraints and additional conditions representing desirable information about the learning process. This method can improve the convergence and generalization properties and efficiently avoid local minima through prompt activation of the hidden units, optimal alignment of successive weight vector offsets, elimination of excessive hidden nodes, and regulation of the magnitude of search steps in the weight space. In literatures [12–15], several new constrained learning algorithms (CLA) were proposed by coupling the a priori information from problems into the cost functions defined at the network output layer. As a result, the solutions for the involved problems finding the roots of polynomials can be very easily obtained. In addition, in literature [16], the authors proposed a method of noise injection into training data. This method can reduce the occurrence of the overfitting case. Although these approaches above can alleviate the overfitting case to some extent, the additional computational requirements are increased greatly.

On the other hand, several pruning algorithms that can reduce the network complexity were also proposed. These approaches include synaptic weight elimination strategy [17], weight decay method [18], and weight sharing technique [7,19]. Although these algorithms can also avoid the overfitting case to some extent, they were not specifically designed for the purpose performing better generalization ability.

Download English Version:

https://daneshyari.com/en/article/4637396

Download Persian Version:

https://daneshyari.com/article/4637396

Daneshyari.com