Chapter 10
Weights Direct Determination of Feedforward Neural Networks without Iterative BP-Training

Yunong Zhang
Sun Yat-Sen University (SYSU), China

Ning Tan
Sun Yat-Sen University (SYSU), China

ABSTRACT

Artificial neural networks (ANN), especially with error back-propagation (BP) training algorithms, have been widely investigated and applied in various science and engineering fields. However, the BP algorithms are essentially gradient-based iterative methods, which adjust the neural-network weights to bring the network input/output behavior into a desired mapping by taking a gradient-based descent direction. This kind of iterative neural-network (NN) methods has shown some inherent weaknesses, such as, 1) the possibility of being trapped into local minima, 2) the difficulty in choosing appropriate learning rates, and 3) the inability to design the optimal or smallest NN-structure. To resolve such weaknesses of BP neural networks, we have asked ourselves a special question: Could neural-network weights be determined directly without iterative BP-training? The answer appears to be YES, which is demonstrated in this chapter with three positive but different examples. In other words, a new type of artificial neural networks with linearly-independent or orthogonal activation functions, is being presented, analyzed, simulated and verified by us, of which the neural-network weights and structure could be decided directly and more deterministically as well (in comparison with usual conventional BP neural networks).

INTRODUCTION

Benefiting from parallel-processing nature, distributed storage, self-adaptive and self-learning abilities, artificial neural networks (ANN) have been investigated and applied widely in many scientific, engineering and practical fields, such as, classification and diagnosis (Hong & Tseng, 1991; Jia & Chong, 1995; Sadeghi, 2000; Wang & Li, 1991), image and signal processing (Steriti & Fiddy, 1993), control system design (Zhang & Wang, 2001, 2002), equations solving (Zhang, Jiang & Wang, 2002; Zhang & Ge, 2005; Zhang

DOI: 10.4018/978-1-61520-757-2.ch010

& Chen, 2008), robot inverse kinematics (Zhang, Ge & Lee, 2004), regression and identification (Zhang *et al*, 2008).

As we may realize, the feedforward neural network (FNN) based on the error back-propagation (BP) training algorithm or its variants is one of the most popular and important neural-network (NN) models, which has been involved in many theoretical analyses and real-world applications (Hong & Tseng, 1991; Jia & Chong, 1995; Rumelhart, McClelland & PDP Research Group 1986; Wang & Li, 1991; Yu, Chen & Cheng, 1993; Zhang *et al*, 2008). In particular, BP neural networks proposed in mid 1980s (or even earlier, in 1974) is a kind of multilayer feedforward neural network (Rumelhart, McClelland & PDP Research Group 1986; Zhang *et al*, 2008; Zhou & Kang, 2005), of which the error back-propagation algorithm could be summarized simply as

$$\begin{aligned} w(k+1) &= w(k) + \Delta w(k) \\ &= w(k) - \eta(\partial E \,/\, \partial w)\,\big|_{w=w(k)} \end{aligned} \tag{1}$$

where w denotes a vector or matrix of neural weights (and/or thresholds), $k = 0, 1, 2, \ldots$ denotes the iteration number during the training procedure, $\Delta w(k)$ denotes the weights-updating value at the kth iteration of the training procedure with η denoting the learning rate (or termed, learning step-size) which should be small enough, and finally we use E to denote the error function that monitors and control such a BP-training procedure.

The above conventional BP neural network and algorithm are essentially a gradient-descent based error-minimization method, which adjusts the neural-network weights (and/or thresholds) to bring the neural-network input/output behavior into a desired mapping as of some specific application task or environment. For better performance (e.g., in terms of the training efficiency and the generalization accuracy), many improved BP algorithms have been proposed since mid 1980s; see Corwin, Logar & Oldham (1994), Goodman & Zeng (1994), Hong & Tseng (1991), Jenkins (2006), Jia & Chong (1995), Pai (2004), Ren & Zhou (2008), Yu, Chen & Cheng (1993), Zhang, Lin & Tang (2008), Zhang *et al* (2008), Zhou & Kang (2005) and the references therein. Generally speaking, there are two widely-adopted types of improvements. On one hand, BP algorithms could be improved based on standard gradient-descent method (e.g., introducing momentum). On the other hand, numerical minimization techniques could be employed for network training. It is worth pointing out that people usually pay more attention to the iterative learning procedure and algorithms so as to ameliorate the performance of such BP neural networks. However, as researchers (including us) realize and experience quite frequently, the inherent weaknesses of BP-type neural networks are still there! Specifically, BP-type neural networks appear to have the following weaknesses (Hong & Tseng, 1991; Jenkins, 2006; Jia & Chong, 1995; Lippmann, 1987; Looney, 1996; Miller, Arguello & Greenwood, 2004; Wen *et al*, 2000; Wilson & Martinez, 2001; Yeung & Zeng, 2002; Yu, Chen & Cheng, 1993; Yu & Chen, 1995; Zhang, 1999; Zhang *et al*, 2008; Zhou & Kang, 2005):

1) possibility of being trapped into some local minima,
2) difficulty in choosing appropriate learning rates, and
3) inability to design the optimal or smallest NN-structure in a deterministic way.

Since the late 1990s (e.g., Zhang, 1999, 2002), we have asked ourselves the question: Could the above NN problem be solved radically? Based on our ten-year research experience on the neural network topic, as reported here in this chapter, we are now trying to resolve these neural-network weaknesses (i.e., local minima, slow convergence of training, and network-structure uncertainty) while maximizing the neural-network perfor-

Related Content