Neurocomputing 79 (2012) 158–163

Contents lists available at SciVerse ScienceDirect

Neurocomputing

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

Letters

Learning algorithm and hidden node selection scheme for local coupled feedforward neural network classifier

Jianye Sun*

Computation Center, Harbin University of Science and Technology, No 52, Xuefu Road, PR China

ARTICLE INFO

Article history:
Received 25 February 2011
Received in revised form
25 September 2011
Accepted 30 September 2011
Communicated by T. Heskes
Available online 11 November 2011

Keywords:
Neural network
Classifier
Gradient
BP algorithm
Hidden node
LCFNN

ABSTRACT

In this paper, a neural classifier based on the newly developed local coupled feedforward neural network, which may improve the convergence of BP learning significantly, is developed. A binary threshold unit is used as the output node of the classifier. A general error gradient of the output node is defined for the BP training of the classifier. And a hidden node selection scheme is developed for the local coupled feedforward neural network. In addition, we derive a result on the "universal approximation" property of the local coupled feedforward neural network with an arbitrary group of window functions, which can cover the region of training samples. Simulation results show that the general error gradient and the hidden node selection scheme work well.

1. Introduction

Data classification is a frequently encountered decision making task of human activity. Many classification techniques have been proposed, such as: decision tree [15], neural network [11], support vector machine (SVM) [3], and rule based classifiers systems [5]. A large number of studies have been devoted to empirical comparisons between neural and conventional classifiers [4,6,13,14,16,19]. As demonstrated by Zhang in [20], their general conclusion is that no single classifier is the best for all data sets although the feedforward neural networks do have good performance over a wide range of problems. Neural network classifier has been successfully applied to a variety of real world tasks ([1,2,7–10,12,17,]).

High efficient learning is essential for neural network classifier. Slow convergence of the Backpropagation (BP) algorithm has blocked the applications of feedforward neural networks for more than 20 years. A local coupled feedforward neural network (LCFNN) has been developed by Sun to treat this problem [18]. In LCFNN, a window function is added to each hidden node. A hidden node need only to remember the learning samples located within its window radium. Therefore, the difficulty of learning tasks for hidden nodes is determined by the window radium selected by user and is not influenced by the size of the learning sample set.

In this paper, a neural classifier based on LCFNN is developed. A binary threshold unit is used as the output node of the classifier. Traditional BP algorithm cannot be applied in this classifier directly because the threshold unit is not differentiable. To solve this problem, a general error gradient of the threshold output node is developed in Section 3.1. And a hidden node selection scheme is developed for LCFNN in Section 3.2. In addition, in Section 2, we derive a result on the "universal approximation" property of the local coupled feedforward neural network with an arbitrary group of window functions, which can cover the region of training samples.

2. Local coupled feedforward neural network

The linking architecture of LCFNN is shown in Fig. 1. In LCFNN, each hidden node is assigned an address in input space, and each input $x \in R^n$ activates hidden nodes with intensity $g(\|x-d_i\|)$, here $d_i \in R^n$ is the address of the hidden node. The output of LCFNN with one output node is:

$$y(x) = \sum_{i \in \sigma} \left(\sum_{k=1}^{m_i} (a_{ik} f(b_{ik} \cdot x + c_{ik}) + p_{ik}) \right) g(\|x-d_i\|) \quad (1)$$

where σ stands for the set of the addresses of hidden nodes, m_i is the number of hidden nodes at the ith hidden node address, $a_{ik} \in R$, $c_{ik} \in R$, $p_{ik} \in R$, and $b_{ik} \in R^n$ are network weights, $x \in R^n$ is an input vector, $y \in R$ is the output, $f(\cdot) \in C^{\infty}$ is a sigmoid function,

* Tel.: +86 13069703961.
E-mail address: jnysun@gmail.com

0925-2312/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2011.09.019

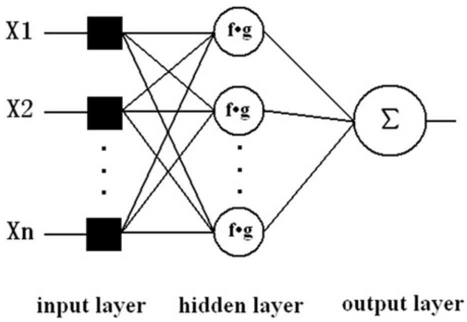

Fig. 1. Structure of LCFNN.

$g(\cdot)$ is a window function, which observes following conditions:

1. $g(\cdot)$ is a continuous function.
2. $g(0)=1$
3. $g(\cdot)=0$ in $[r, +\infty]$
4. $g(\cdot)$ is a monotonic decreasing function in $[0,r]$

where $r>0$ is the window radium. Obviously, one input only activates the hidden nodes located in the sphere whose center is the input point and whose radium is r. The number of hidden nodes activated by one input can be controlled by adjusting r.

Various norms can be used in g(.). In this study, we use ∞-norm for the window function g(.).

Below is a theorem about the "universal approximation" property of LCFNN.

Theorem 1. *For an arbitrary continuous function φ in a compact region D and an arbitrary group of n window function centers $d_1 \sim d_n$, which satisfy:*

$$\left(\sum_{i=1}^{n} g(\|x-d_i\|)\right) > 0, \quad \textit{for all} \quad X \in D,$$

there exists a LCFNN, which uses $d_1 \sim d_n$ as the window function centers of its hidden nodes and can approximate φ in D at any given accuracy.

Proof. Because the upper bound of g(.) is one, we have:

$$n \geq \left(\sum_{i=1}^{n} g(\|x-d_i\|)\right) > 0 \quad \textit{for all} \quad X \in D$$

According to the wide recognized "universal approximation" property of multilayer perceptron (MLP), for an arbitrary $\delta>0$, there exists a MLP, $\sum_{k=1}^{m}(a_k f(b_k \cdot x+c_k)+p_k)$, satisfying:

$$\left| \frac{\varphi(x)}{\sum_{i=1}^{n} g(\|x-d_i\|} - \sum_{k=1}^{m}(a_k f(b_k \cdot x+c_k)+p_k) \right| < \frac{\delta}{n} \textit{ for all} \quad X \in D$$

Here m is the number of hidden nodes, $f(.)$ is the sigmoid activation function of hidden nodes. Then multiply the two sides of the formula with $\sum_{i=1}^{n} g(\|x-d_i\|)$,

$$\left| \left[\sum_{i=1}^{n}\left(\frac{\varphi(x)}{\sum\limits_{i=1}^{n} g(\|x-d_i\|}\right) g(\|x-d_i\|) - \left[\sum_{i=1}^{n}\left(\sum_{k=1}^{m}(a_k f(b_k \cdot x+c_k)+p_k\right) g(\|x-d_i\|)\right]\right| < \frac{\delta}{n}\sum_{i=1}^{n} g(\|x-d_i\|)) \leq \delta \quad \textit{for all} \quad x \in D$$

This formula can be transformed into:

$\left|\varphi(x) - \sum_{i=1}^{n}(\sum_{k=1}^{m}(a_k f(b_k \cdot x+c_k)+p_k))g(\|x-d_i\|)\right| < \delta$ for all $x \in D$

Then we have the following conclusion:

For an arbitrary $\delta>0$, there exists a LCFNN, $\sum_{i=1}^{n}(\sum_{k=1}^{m}(a_k f(b_k \cdot x+c_k)+p_k))g(\|x-d_i\|)$, satisfying:

$\left|\varphi(x) - \sum_{i=1}^{n}(\sum_{k=1}^{m}(a_k f(b_k \cdot x+c_k)+p_k))g(\|x-d_i\|)\right| < \delta$ for all $x \in D$

End of proof.

3. Local coupled feedforward neural; network classifier

3.1. Local coupled feedforward neural; network classifier

The linking architecture of LCFNN classifier is shown in Fig. 2. The activation function of the output node is.

$$z = \begin{cases} 1 & u \geq 0 \\ -1 & u < 0 \end{cases} \tag{2}$$

where u is the input of the output node, z is the output of the output node. The classification error e is defined as:

$$e = |z-t| \tag{3}$$

where $t \in \{-1, 1\}$ is the known target value of the training sample.

The Traditional BP algorithm cannot be applied in this classifier directly because the threshold unit is not differentiable. The present solution to this problem is to use a LCFNN to approximate the classifier. Then put a threshold node behind the LCFNN. The objection function J for the BP training of the LCFNN is:

$$\mathrm{J} = \frac{1}{2}\sum_{i \in \sigma}(t_i - f_o(u_i))^2 \tag{4}$$

where σ is the set of training samples, t_i is the target value of the ith sample, $f_o(u_i)$ is the output of the LCFNN for the ith sample, and $f_o(.)$ is the activation function of the output node of the LCFNN.

The network weight updating formula for the BP training is:

$$w = w - k \cdot \sum_{i \in \sigma}(f_o(u_i) - t_i)\frac{\partial f_o(u_i)}{\partial u_i}\frac{\partial u_i}{\partial w} \tag{5}$$

where w is the vector of the network weights of the LCFNN, k is the learning rate.

As shown in Fig.3(a), when a sigmoid function is used as $f_o(.)$, the derivatives of $f_o(.)$ for those samples, which are classified wrongly and whose inputs of the output node are far from zero, are near zero (the shaded region in Fig. 3(a)). The information backpropagation of these samples are blocked by the sigmoid $f_o(.)$. This problem may cause the BP training to be trapped in local minimum.

A linear $f_o(.)$ can solve this problem. However, as shown in Fig. 3(b), for a linear $f_o(.)$, the samples, which are classified correctly and whose classification margins are larger than one will produce gradients to reduce their classification margins (the shaded region in Fig. 3(b)). These requirements for reducing classification margins are unreasonable for a classifier and will disturb the training of the classifier.

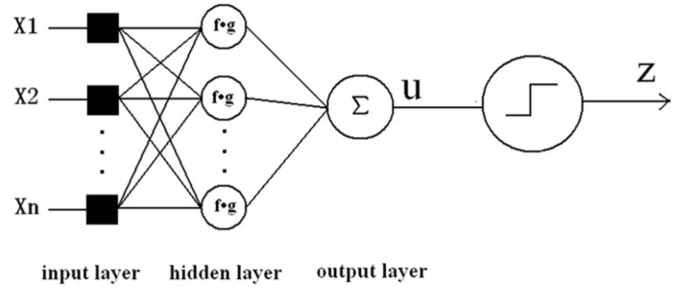

Fig. 2. Structure of LCFNN classifier.