



ELSEVIER

10 March 1997

PHYSICS LETTERS A

Physics Letters A 227 (1997) 94–96

A simple way to determine the connecting of a multilinear neural network

Jianwei Shuai, Zhenxiang Chen, Ruitang Liu, Boxi Wu

Physics Department, Xiamen University, Xiamen 361005, China

Received 16 April 1996; revised manuscript received 30 October 1996; accepted for publication 13 December 1996

Communicated by C.R. Doering

Abstract

A simple way is introduced to determine the connecting of a multilinear neural network, in which only matrix division is used. The character of the convergence theorem of the multilinear back-propagation and perceptron algorithms is discussed. For the simplest case, the signum function of the neuron is not needed.

1. Introduction

There is no general convergence theorem for the perceptron [1,2] or the back-propagation [3,4] learning algorithms. This lack is completely overcome by a generalization to multilinear neural networks [5,6]. The neural networks with high-order interaction between neurons are discussed in various models [7–11]. Multilinear neural networks exhibit an unlimited storage capacity [9]. The typical property of the multilinear neural network is precision, not to compress information. In biology, the network can be interpreted as a model for neuropils (sets of multiply connected neurons in the brain), because the presynaptic neurons can act in any functional manner on a postsynaptic neuron. Furthermore, the model is a good candidate for adaptation and performance with incomplete information. There are two learning algorithms known that converge for a mapping task, i.e. the multilinear back-propagation learning algorithm [5] and the multilinear perceptron learning algorithm [6]. In this Letter, it is shown that, to perform a

mapping task, one does not need any iterative learning algorithm. A simple way is introduced to determine the multilinear connecting.

2. The multilinear network

The multilinear network has an input layer and an output layer, and consists of N input neurons $s_i = \pm 1$ and N output neurons $\bar{s}_i = \pm 1$, $i = 1, 2, \dots, N$. It has bilinear couplings J_{ij} , trilinear couplings J_{ijk} , and so forth to $(N+1)$ -linear couplings $J_{i_1 i_2 \dots i_N}$. All couplings act from the input layer to the output one. The state of an out neuron \bar{s}_i is determined by the coupling-weighted states of the input neurons s_i ,

$$\begin{aligned} \bar{s}_i &= \text{sgn}(h_i), \quad i = 1, \dots, N, \\ h_i &= J_i + \sum_{j=1}^N J_{ij} s_j + \sum_{j < k=1}^N J_{ijk} s_j s_k \\ &\quad + J_{i12 \dots N} s_1 s_2 \dots s_N. \end{aligned} \quad (1)$$

Here sgn denotes the signum function, h_i the local field, and $-J_i$ can be taken as the threshold of the out neuron.

A task is to map each of 2^N configurations $\{\xi_i^\mu\}$ of input neurons to one of 2^N configurations of output neurons $\{\bar{\xi}_i^\mu\}$, respectively. It has been proven that, for such a task, the multilinear back-propagation learning algorithm or the multilinear perceptron learning algorithm stops after a finite number of steps [5,6].

3. A simple way to determine the connecting

In this section, it will be shown that, to determine the multilinear connecting of a mapping task, only matrix division is used. In order to express the idea in a coherent and simple manner, one may use the vector notation as follows,

$$\begin{aligned} \xi^\mu &= (1, \xi_1^\mu, \xi_2^\mu, \dots, \xi_N^\mu, \xi_1^\mu \xi_2^\mu, \xi_1^\mu \xi_3^\mu, \dots, \\ &\quad \xi_1^\mu \xi_2^\mu \dots \xi_N^\mu) \\ &\equiv (s_1^\mu, s_2^\mu, s_3^\mu, \dots, s_{N+1}^\mu, s_{N+2}^\mu, \\ &\quad s_{N+3}^\mu, \dots, s_{2^N}^\mu), \end{aligned} \quad (2)$$

in which $s_i^\mu = \pm 1$ and $i, \mu = 1, 2, \dots, 2^N$. An analogous notation J_i is formed for the multilinear coupling from the input layer to each out neuron, e.g., i th neuron ($i = 1, \dots, N$),

$$\begin{aligned} J_i &= (J_i, J_{i1}, J_{i2}, \dots, J_{iN}, J_{i12}, J_{i13}, \dots, J_{i12N}) \\ &\equiv (T_{i,1}, T_{i,2}, T_{i,3}, \dots, T_{i,N+1}, T_{i,N+2}, \\ &\quad T_{i,N+3}, \dots, T_{i,2^N}). \end{aligned} \quad (3)$$

To perform a task correctly, the connecting should be selected as

$$\bar{\xi}_i^\mu = \text{sgn}(h_i^\mu) = \text{sgn}\left(\sum_{j=1}^{2^N} T_{i,j} s_j^\mu\right), \quad (4)$$

in which $i = 1, \dots, N$ and $\mu = 1, \dots, 2^N$. Obviously, for the two layers feed-forward multilinear neural network, each out neuron is independent of the others of the output layer. For a fixed out neuron, e.g., i th neuron, there are a 2^N -dimensional coupling vector and a 2^N -dimensional local field vector corresponding with it,

$$|J_i\rangle = J_i^T = (T_{i,1}, T_{i,2}, \dots, T_{i,2^N})^T,$$

$$|h_i\rangle = (h_i^1, h_i^2, \dots, h_i^\mu, \dots, h_i^{2^N})^T, \quad i = 1, \dots, N. \quad (5)$$

Here, as shown in Ref. [5], the Dirac symbol $|\rangle$ is used to denote the transposed vector. 2^N 2^N -dimensional vectors ξ^μ make up a $2^N \times 2^N$ matrix S , in which $S_{\mu,i} = s_i^\mu$ so that the values are only ± 1 . Then the local field vector of the i th out neuron can be rewritten as

$$|h_i\rangle = S |J_i\rangle. \quad (6)$$

This is actually a set of linear equations with 2^N parameters. As a result, the connecting from the input layer to the i th out neuron can be simply and directly obtained,

$$|J_i\rangle = S^{-1} |h_i\rangle, \quad i = 1, \dots, N. \quad (7)$$

Here we can let

$$h_i^\mu = |\text{random}(i, \mu)| \bar{\xi}_i^\mu, \quad (8)$$

i.e. the multilinear connecting $T_{i,j}$, as noted in Eq. (3), from the input layer to the i th out neuron is

$$T_{i,j} = \sum_{\mu=1}^{2^N} |\text{random}(i, \mu)| S_{j,\mu}^{-1} \bar{\xi}_i^\mu, \quad (9)$$

in which $i = 1, \dots, N$ and $j = 1, \dots, 2^N$.

4. Discussion

From Eqs. (6) and (8), one can see we have actually N sets of linear equations with 2^N parameters. This is the character of the convergence theorem of the multilinear back-propagation algorithm [5] and multilinear perceptron algorithm [6]. With the coefficients $s_i^\mu = \pm 1$ shown in Eq. (2), the linear equations are simple and can be resolved directly and accurately.

With various $\text{random}(i, \mu)$ for Eq. (8), difference solutions of multilinear connecting can be obtained. The simplest case is to let $\text{random}(i, \mu) \equiv 1$, then

$$|J_i\rangle = S^{-1} |\bar{\xi}_i\rangle, \quad i = 1, \dots, N. \quad (10)$$

For this case, the structure of the neuron is rather simple. The neuron does not need the signum function.

In some cases, the network is asked to map a part of a task, i.e. to map a part of 2^N input configurations to the output layer. To do this with the method discussed above, one can only complement the rest configurations with a random mapping relationship.

As mentioned above, for the neurons in the output layer, each one is independent of the others. One can have any number of neurons in the output layer, e.g. M neurons. Then a task is to map 2^N N -dimensional input configurations to 2^N M -dimensional output ones. To perform it, we need only let the subscript i of Eq. (9) change from 1 to M .

Besides the application discussed in Refs. [5] and [6], a possible application is that we can use the multilinear neural network to store the dot-matrix of Chinese words. There are about 10000 Chinese words that are usually used. The normal method is to use fourteen bits ASCII codes to store the hardware addresses. In each address, the information of M dots of each Chinese word is stored. Using the multilinear neural network, it needs the network to map 16384 fourteen-dimensional input addresses to the 16384 M -dimensional dots information of Chinese words in the output layer.

Acknowledgement

Funding for this work was provided in part by the National Natural Science Important Foundation with No. 19334032, and in part by the National Natural Science High Technology Research Foundation with No. 69481003.

References

- [1] F. Rosenblatt, *Principles of Neurodynamics* (Spartan, New York, 1962).
- [2] S.I. Gallant, *IEEE Trans. on Neural Networks* 1 (1990) 179.
- [3] P.J. Werbos, *Beyond regression: New tools for prediction and analysis in the behavioral sciences* (Harvard Univ., 1974).
- [4] D.E. Rumelhart, G.E. Hinton and R.J. Williams, *Nature* 323 (1986) 533.
- [5] H.-O. Carmesin, *Phys. Lett. A* 188 (1994) 27.
- [6] H.-O. Carmesin, *Phys. Rev. E* 50 (1994) 622.
- [7] Y.C. Lee, G. Doolen, H.H. Chen, G.Z. Sun, T. Maxwell, H.Y. Lee and C.L. Giles, *Physica D* 22 (1986) 276.
- [8] E. Gardner, *J. Phys. A* 20 (1987) 3453.
- [9] H.-O. Carmesin, *Phys. Lett. A* 156 (1991) 183.
- [10] W.A.T. Wan Abdullah, *Phys. Lett. A* 176 (1993) 202.
- [11] N. Lemke, J.J. Arenzon and F.A. Tamarit, *J. Stat. Phys.* 79 (1995) 415.