An Improvement of Neural Networks Applied to Pharmaceutical Problems

Kenichi SATO* and Junko NAKAGAWA

Tohoku College of Pharmacy, Komatsushima 4-4-1, Aoba-ku, Sendai 981, Japan.
Received June 5, 1996; accepted October 1, 1996

In applying the neural network to the classification problem in pharmacology, we adopt an extended back-propagation (EBP) learning which adjusts the parameters appearing in an activation function, as well as the weights. The results of simulations show that such an extended learning speeds up the learning process as compared with the conventional basic back-propagation procedure, irrespective of the initial values of the parameters, which is extremely useful in the practical application of the neural network in the pharmaceutical field. We have also found that use of Morita's activation function beyond the sigmoid type further accelerates the EBP learning in some cases.

Key words neural network; extended back-propagation; classification problem; Morita's activation function

For the structure classification of chemical substances based on physical chemistry data, and quantitative structure–activity relationships (QSAR) analysis, multiregression analysis and pattern recognition-related methods such as the adaptive least-squares (ALS)[1] method have been widely applied, but they do not take nonlinearity into account. Most problems in the pharmaceutical field involve marked nonlinearity. To utilize methods such as the advanced nonlinear least-squares method, however, it is necessary to assume a theoretical model, for which the most appropriate values of model parameters are determined by minimizing the least-squares error between the experimental data and the model-predicted values. The difficulty here is that in the pharmaceutical field, it is often still difficult to construct a good model.

Ichikawa and his colleagues have shown in a series of studies that neural networks with the basic back-propagation algorithm (BBP) by Rumelhart *et al.*[2] can improve the fitting, prediction and generalization ability as compared with other conventional methods in the pharmaceutical field.[3-7] Neural networks can be regarded as a natural extension of the pattern recognition-related methods to fully include nonlineality, and the MR-type neural network operates as a nonlinear multiregression analysis.[7] In a neural network approach, we do not assume a specific theoretical model for the problem but learn experimental data according to a back-propagation algorithm and construct the simulator of general applicability on the basis of optimized weight matrices between units. Introducing a forgetting process into the back-propagation learning, often referred to as reconstruction learning,[6,8] allows the network structure to be much simpler, consisting of less important weight matrices, without lowering the simulating ability. In this way we can establish which units play the major role in the classification, and gain insight into a suitable model for the problem. To fully apply such an efficient tool to pharmaceutical problems, speeding up BBP learning is of great practical importance.

In BBP neural networks the continuous analog type unit or neuron, which has an activation function that is differentiable and nonlinear, is very useful. A sigmoid activation function is most frequently used. The ability of neural networks to learn quickly and correctly depends strongly on the values of the parameters appearing in the network. They are not known *a priori*, and so how to find suitable values in particular problems is very important, though it is generally difficult. As for parameters appearing in an activation function, the possibility of quick and correct learning of neural networks depends strongly on their values, as we will show in the later section concerning pharmaceutical problems. Seeking appropriate values of such parameters becomes more difficult if neural networks are larger.

In the present paper we will show that it is possible not to keep the parameters which appear in an activation function fixed, but to adapt them in the course of the learning based on an extended back-propagation algorithm (EBP), which speeds up learning and so greatly helps practical application of neural networks to pharmaceutical problems. A nonmonotone activation function may also be useful. EBP learning using Morita's activation function further facilitates fast learning for some cases.

Theory In a BBP network consisting of m feed-foward layers, the weight $\omega_{ij}^{(k-1,k)}$, which is the strength of the connection between unit i in the $(k-1)$-th layer and unit j in the k-th layer, is adapted so as to minimize the error function $E(\{\omega_{ij}^{(k-1,k)}\}; \{\beta_j^{1(k)}, \beta_j^{2(k)}, \cdots, \beta_j^{n(k)}\})$ through the equation

$$\Delta\omega_{ij}^{(k-1,k)} = -\varepsilon \frac{\partial E}{\partial \omega_{ij}^{(k-1,k)}} \tag{1}$$

Here ε is the learning coefficient and $\beta_j^{l(k)}$ $(l=1,2,\cdots,n)$ are n parameters appearing in an activation function of unit j in the k-th layer. This is further expressible in the form of error back-propagation by

$$\Delta\omega_{ij}^{(k-1,k)} = -\varepsilon o_i^{(k-1)} d_j^{(k)} \tag{2}$$

using the output $o_i^{(k-1)}$ of the unit i in the $(k-1)$-th layer and the quantity

$$d_j^{(k)} = \left(\sum_l \omega_{jl}^{(k,k+1)} d_l^{(k+1)}\right) \frac{\partial f(i_j^k)}{\partial i_j^k} \quad (k<m) \tag{3a}$$

$$= (o_j^{(k)} - t_j) \frac{\partial f(i_j^k)}{\partial i_j^k} \quad (k=m) \tag{3b}$$

where i_j^k is the total net input to unit j in the k-th layer and t_j is the j-th component of the target pattern.

Usually n parameters $\beta_j^{l(k)}$ $(l=1,2,\cdots,n)$ have to be specified prior to learning. Appropriate values, which

* To whom correspondence should be addressed.

© 1997 Pharmaceutical Society of Japan

NII-Electronic Library Service

promote quick and correct learning, are not known *a priori*. It is of interest to extend BBP learning to adjust parameters appearing in an activation function as well as weights, since the error function depends on the network weights and on the parameters $\beta_j^{l(k)}$ ($l=1, 2, \cdots, n$) as well. The error minimization requirement for such parameters is given by the equations

$$\Delta\beta_j^{l(k)} = -\varepsilon' \frac{\partial E}{\partial \beta_j^{l(k)}} = -\varepsilon' \frac{\partial E}{\partial o_j^{(k)}} \frac{\partial o_j^{(k)}}{\partial \beta_j^{l(k)}} = -\varepsilon' d_j^{(k)} \frac{\partial f}{\partial \beta_j^{l(k)}} \left(\frac{\partial f}{\partial i_j^{(k)}}\right)^{-1} \quad (4)$$

which describe the adaptation of $\beta_j^{l(k)}$. Here we introduced a new proportional constant ε'. Considering the change of the error function given by

$$\Delta E = \sum_k \sum_{ij} \frac{\partial E}{\partial \omega_{ij}^{(k-1,k)}} \Delta\omega_{ij}^{(k-1,k)} + \sum_k \sum_{ij} \frac{\partial E}{\partial \beta_j^{l(k)}} \Delta\beta_j^{l(k)} \quad (5)$$

we can show that the change ΔE in EBP with the learning coefficient ε is equivalent to the change ΔE in BBP with the learning coefficients $\varepsilon_{ij}^{(k-1,k)}$ given by

$$\varepsilon_{ij}^{(k-1,k)} = \varepsilon \left[1 + \frac{\varepsilon'}{N_{k-1}\varepsilon} \left(\frac{\dfrac{\partial f}{\partial \beta_j^{l(k)}}}{o_i^{k-1} \dfrac{\partial f}{\partial i_j^{(k)}}} \right)^2 \right] \quad (6)$$

which depend on the neurons i and j coupling with each other and vary in the course of learning.[9]

In the present work we assume that all N_k neurons in the k-th layer take the same value

$$\beta_1^{l(k)} = \beta_2^{l(k)} = \cdots = \beta_{N_k}^{l(k)} = \beta^{l(k)} \quad (7)$$

for simplicity. Then Eq. 4 becomes

$$\Delta\beta^{l(k)} = -\varepsilon' \frac{\partial E}{\partial \beta^{l(k)}} = -\varepsilon' \sum_j d_j^{(k)} \frac{\partial f}{\partial \beta^{l(k)}} \left(\frac{\partial f}{\partial i_j^{(k)}}\right)^{-1} \quad (8)$$

The adaptation equation for the steepness parameter c_k in a sigmoid function

$$f(x; c_k) = \frac{1}{1+e^{-c_k x}} \quad (9)$$

is given by

$$\Delta c_k = -\varepsilon' \sum_j d_j^{(k)} i_j^k (c_k)^{-1} \quad (10)$$

after a little manipulation. Sperduti and Starita have already considered EBP learning for a sigmoid activation function and derived a similar equation to Eq. 10.[10] They successfully applied EBP to simple problems such as the 4 bit parity problem and the encoder problem. It would be interesting to investigate whether EBP works well in treating more complicated pharmaceutical problems. The parameter c_k^{-1} is often treated as a temperature and sometimes varied according to some prescription which assists rapid convergence. Sperduti and Starita also confined the steepness c_k to a positive value. Hereafter we treat c_k purely as a parameter which characterizes the activation function and we do not restrict c_k to be positive. As we will show in the next section, there is the case in which negative steepness ($c_k<0$) makes fast EBP learning possible. This suggests the possibility of fast EBP learning by the use of a nonmonotone type activation function. Furthermore, Morita indicated that a nonmonotone activation function, which expresses the response of an effective neuron consisting of an excitatory type and an inhibitory type rather than of a biological neuron, is useful in giving neural networks better recalling ability.[11] Kotani *et al.* indicated that the use of a nonmonotone activation function in BBP learning generally brings about more rapid convergence than the sigmoid function in their analysis of a practical pattern recognition problem,[12] although they adopted a different type of activation function from Morita's. Thus, it is of interest to examine EBP learning using Morita's activation function

$$f(x; c_k, c_k', h_k, k_k) = \frac{1}{2} \frac{1-e^{-c_k x}}{1+e^{-c_k x}} \cdot \frac{1+k_k e^{c_k'(|x|-h_k)}}{1+e^{c_k'(|x|-h_k)}} + \frac{1}{2} \quad (11)$$

which is nonmonotone for $k_k<1$ and reduces to the sigmoid form

$$f(x; c_k, c_k', h_k, k_k=+1) = \frac{1}{1+e^{-c_k x}} \quad (12)$$

irrespective of c_k' and h_k if k_k is set to be $+1$. In the k-th layer we obtain equations describing the adaptation of parameters c_k, c_k', h_k and k_k respectively, which are more complicated than Eq. 10.

Results and Discussion

Application to the Relationship between ^{13}C-NMR Chemical Shift and the Conformation of Norbornene To investigate the usefulness of an extended back-propagation procedure, we applied a 7-7-2 feed-forward network to the norbornene problem formerly studied by Ichikawa *et al.*[3,6,7] The learning coefficient ε was set to be 0.15. The learning was stopped when the error function decreased to less than 0.001. Table 1 shows the relative ^{13}C-NMR chemical shifts of C_1–C_7 and the conformations of norbornenes first given by Grutzner *et al.*[13] and cited in references 3 and 6. Ichikawa *et al.* took the upper 25 compounds as learning data to fix the weights $\omega_{ij}^{(k-1,k)}$ and predicted the *exo*/*endo* conformation for the lower untrained 13 compounds when their relative ^{13}C-NMR chemical shifts are given, and we also adopt this approach in the present work. The neural network predicted 100% correct conformations while the use of a linear learning machine and cluster analysis both gave only 85% correct (11/13).

Sigmoid Activation Function The initial values of weights $\omega_{ij}^{(k-1,k)}$ are usually set by using small random numbers. Figure 1 shows the number of iterations needed to converge in BBP learning as a function of steepness parameters c_h in the hidden layer and c_o in the output layer, and it can be seen that the learning time depends strongly on these parameters. In Table 2a we show the number of iterations NI in BBP and in EBP learning to attain convergence for 11 sets of steepness parameters (c_h, c_o) of sigmoid activation functions. Fast BBP learning is possible around $(c_h, c_o)=(11, 11)$, but it needs many iterations or does not converge (denoted by symbol x) in general for other values. In EBP learning, values of (c_h, c_o) change according to Eq. 10 starting from those values in the left row, and the final values obtained when the learning has been successfully finished are given in Table 2b.

NII-Electronic Library Service

Table 1. Relative ^{13}C-NMR Chemical Shifts and Conformations in Norbornenes

Compd.	C_1	C_2	C_3	C_4	C_5	C_6	C_7	*exo/endo*
1	6.7	6.7	10.1	0.5	0.2	−1.1	−3.7	*exo*
2	8.9	25.3	12.4	−0.4	−1.2	−3.1	−4.4	*exo*
3	7.7	44.3	12.3	−1.0	−1.3	−5.2	−4.4	*exo*
4	4.6	16.7	4.4	−0.2	−0.3	−1.0	−1.8	*exo*
5	1.8	15.1	4.4	−0.2	0.2	−0.7	−3.3	*exo*
6	5.7	3.0	2.6	−0.5	−0.4	0.7	−3.5	*exo*
7	6.1	5.9	10.6	0.6	0.2	0.2	−3.7	*exo*
8	6.5	6.3	10.4	0.3	−0.8	−0.1	−3.5	*exo*
9	6.5	7.5	9.5	0.5	1.7	0.7	−3.8	*exo*
10	7.8	47.0	11.7	−1.3	3.9	−2.7	−3.2	*exo*
11	6.9	6.4	10.1	0.7	−1.2	0.1	−3.9	*exo*
12	5.6	4.9	7.0	0.2	−1.1	0.2	−3.9	*exo*
13	2.5	42.5	11.9	−0.8	−1.1	−2.4	1.4	*exo*
14	5.4	4.5	10.6	1.4	0.5	−7.7	0.2	*endo*
15	6.8	23.3	10.5	1.2	0.6	−9.5	0.3	*endo*
16	6.3	42.4	9.5	0.9	0.2	−9.7	−0.9	*endo*
17	4.2	16.2	2.1	0.9	−0.6	−4.8	1.9	*endo*
18	1.7	12.8	4.0	0.4	0.2	−7.2	1.4	*endo*
19	4.7	3.1	2.2	0.3	1.3	−6.5	−0.6	*endo*
20	4.7	5.3	9.2	1.3	−0.4	−6.5	1.4	*endo*
21	4.6	11.5	8.9	−0.1	0.8	0.4	1.8	*endo*
22	5.6	7.5	8.7	1.4	1.7	−3.0	1.7	*endo*
23	7.1	47.8	13.3	2.2	3.6	−3.4	0.3	*endo*
24	4.1	4.2	7.0	0.7	0.5	−7.4	0.0	*endo*
25	3.2	40.2	10.4	−0.5	0.0	−10.3	3.1	*endo*
26	5.5	1.0	6.3	−0.3	−1.5	−1.6	−1.3	*exo*
27	3.4	0.1	5.5	0.2	−0.7	−4.9	0.0	*endo*
28	5.1	16.4	4.2	−0.4	−1.1	−1.4	−2.1	*exo*
29	4.0	15.9	2.2	0.7	−0.7	−5.0	1.7	*endo*
30	6.6	7.0	10.1	0.2	−1.2	0.5	−3.7	*exo*
31	6.0	8.4	11.2	−0.1	0.7	−1.5	−1.6	*endo*
32	6.3	7.2	9.8	0.7	−0.1	0.8	−3.5	*exo*
33	5.1	4.8	8.4	1.1	−0.1	−7.3	1.6	*endo*
34	1.9	17.1	5.2	−0.1	0.9	0.9	−3.4	*exo*
35	2.3	18.3	5.0	0.3	1.3	−2.9	1.4	*endo*
36	5.1	4.0	8.4	1.1	0.2	−7.7	1.6	*endo*
37	2.9	30.3	13.4	−0.5	−2.1	−0.7	2.0	*exo*
38	3.7	29.8	10.8	−1.6	−1.1	−9.0	2.2	*endo*

For the cases which need many iterations at $\varepsilon' = \varepsilon$ (the ratio $\gamma = \varepsilon'/\varepsilon = 1$), we tried further simulations at large ε' ($\gamma > 1$). Tables 2a and 2b indicate that the steepness parameters in EBP automatically approach the region of suitable values in BBP and so the learning converges quickly, irrespective of the initial values of such parameters. When derivative $\partial f_k/\partial c_k$ is small, setting a large value of ε'/ε enables the rapid convergence of EBP learning. For large values of initial (c_h, c_o), EBP learning with large ε' converges rapidly while BBP learning does not. These results show that in EBP learning we do not need to know the most appropriate initial values of parameters which appear in the activation function in advance. It should be noted that in EBP learning at $(\gamma_{c_h}, \gamma_{c_o}) = (100, 100)$, the case starting from the initial values $(c_h, c_o) = (1, 1)$ takes a final value of negative sign for c_o when the convergence is attained. The behavior of such an activation function, that is, monotonic increasing in the early stage but monotonic decreasing in the later stage, sometimes resembles the action of an nonmonotone type function. This fact that fast learning is possible even by using a sigmoid activation function with a negative steepness parameter suggests the feasibility of fast learning by the use of a nonmonotone type activation function such as Eq. 11. The values of weight matrices obtained by EBP learning are very different from those by BBP, but we obtain the same outputs in both cases.

In Fig. 2 we show how the steepness parameters c_h and c_o change with the number of iterations for the cases with $(\gamma_{c_h}, \gamma_{c_o}) = (20, 20)$ and with $(\gamma_{c_h}, \gamma_{c_o}) = (100, 100)$. In the former case, both parameters behave similarly. They decrease at first from the initial value of 1.0 and then increase rapidly to a plateau value near 7.0. In the latter case c_o decreases rapidly after 15 iterations and reaches an almost constant value after 30 iterations. To check the generalization ability of the neural network, it is useful to remove n arbitrary pieces of data from the training

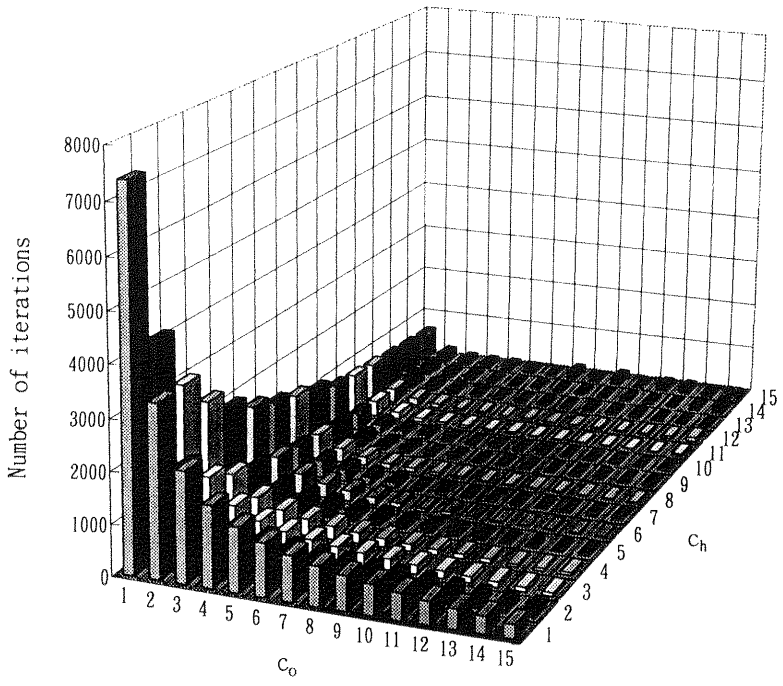

Fig. 1. NI *versus* (c_h, c_o) in BBP Learning for the Norbornenes Problem

NII-Electronic Library Service

Table 2a. The Number of Iterations (NI) in BBP and in EBP Learning (Sigmoid-Type Activation Function)

	BBP	EBP (γ_{c_h}, γ_{c_o})				
		(1, 1)	(10, 10)	(20, 20)	(100, 100)	(500, 500)
(c_h, c_o)	NI	NI	NI	NI	NI	NI
(1, 1)	7369	324	89	66	60	
(2, 2)	1823	291	74	49	29	
(4, 4)	429	227	71	30	25	
(6, 6)	145	116	45	33	31	
(8, 8)	49	53	43	30		
(10, 10)	43	24				
(11, 11)	17	26				
(12, 12)	28	72	28			
(13, 13)	35	26				
(15, 15)	x	x	39			
(20, 20)	x	x	x	x	x	94

Table 2b. Final Values of (c_h, c_o) in EBP Learning Shown in Table 2a

	EBP (γ_{c_h}, γ_{c_o})				
	(1, 1)	(10, 10)	(20, 20)	(100, 100)	(500, 500)
(c_h, c_o)	Final (c_h, c_o)				
(1, 1)	(3.72, 4.01)	(6.02, 6.55)	(6.72, 7.19)	(10.91, −9.85)	
(2, 2)	(3.90, 4.18)	(6.09, 6.57)	(6.82, 7.20)	(9.22, 9.21)	
(4, 4)	(4.75, 4.93)	(6.45, 6.79)	(7.16, 7.09)	(9.31, 9.37)	
(6, 6)	(6.21, 6.25)	(7.14, 7.13)	(7.79, 7.70)	(10.89, 8.42)	
(8, 8)	(8.05, 8.02)	(8.47, 8.19)	(8.69, 8.08)		
(10, 10)	(10.00, 9.98)				
(11, 11)	(11.02, 10.96)				
(12, 12)	(12.01, 11.99)	(11.97, 11.53)			
(13, 13)	(12.99, 12.97)				
(15, 15)		(14.66, 13.45)			
(20, 20)					(12.63, 8.28)

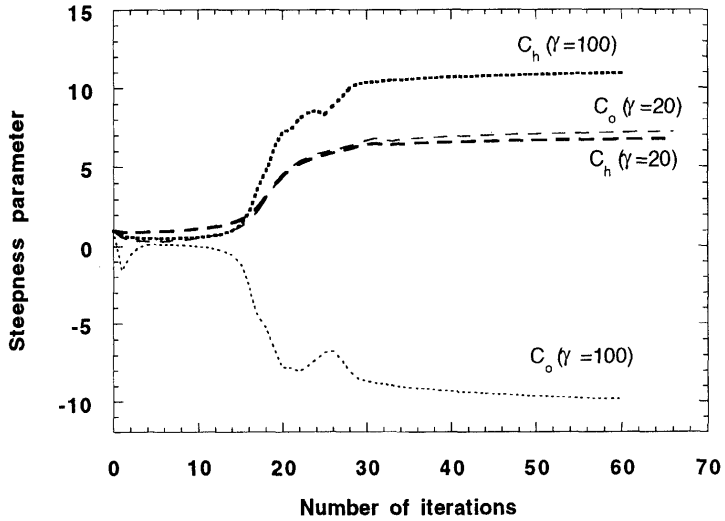

Fig. 2. Change of Steepness Parameters in EBP Learning with $\varepsilon' = 20\varepsilon$ and $\varepsilon' = 100\varepsilon$ for the ^{13}C-NMR Chemical Shifts and Conformations of Norbornenes

patterns to settle the weight matrices and to check the prediction ability of the neural network for the n removed data, though this requires much computation time. We confirmed that the speeding up by EBP makes such investigation easier.

Morita's Activation Function The number of iterations needed to achieve convergence in BBP learning is shown in Table 3 for Morita's nonmonotone ($k < 1$) activation function given by Eq. 11 as well as for the sigmoid one for a few typical examples. Even if we set the same value

NII-Electronic Library Service

Table 3. NI and Parameter Values in Morita's Nonmonotone Activation Function as well as the Sigmoid One in BBP Learning

BBP							
Sigmoid		Morita		Sigmoid		Morita	
(c_h, c_o)	NI	(c_h, c'_h, h_h, k_h) (c_o, c'_o, h_o, k_o)	NI	(c_h, c_o)	NI	(c_h, c'_h, h_h, k_h) (c_o, c'_o, h_o, k_o)	NI
(1, 1)	7369	(1, 1, 0.9, 0.8) (1, , , +1)	8662	(11,11)	17	(11, 12, 0.9, 0.0) (11, , , +1)	22
(30, 1)	368	(30, 30, 0.9, −1) (1, , , +1)	302			(11, 11, 0.7, −1) (11, , , +1)	15
(4, 4)	429	(4, 4, 0.9, −1) (4, , , +1)	421	(13, 11)	51	(13, 13, 0.9, +0.8) (11, 11, 0.9, −1)	35
(15, 4)	98	(15, 15, 0.7, +0.2) (4, , , +1)	58			(13, 13, 1.2, −0.6) (11, 11, 0.9, −1)	31
		(15, 15, 0.9, −0.8) (4, , , +1)	42			(13, 13, 1.2, −0.8) (11, 11, 0.9, −1) (13, 13, 0.9, −1) (11, 11, 0.9, −1)	22 29

Table 4a. NI in BBP and in EBP Learning (Morita's Activation Function)

	BBP	EBP $(\gamma_{c_h}, \gamma'_{c_h}, \gamma_{h_h}, \gamma_{k_h})/(\gamma_{c_o})$			
		(1, 1, 1, 1)/(1)	(10, 10, 1, 1)/(10)	(20, 20, 1, 1)/(20)	(100, 100, 1, 1)/(100)
$(c_h, c'_h, h_h, k_h)/(c_o)$	NI	NI	NI	NI	NI
(1, 1, 0.5, 1)/(1)	7369	171	64	52	55
(2, 2, 0.5, 1)/(2)	1823	131	49	41	25
(4, 4, 0.5, 1)/(4)	429	88	50	41	25
(6, 6, 0.5, 1)/(6)	145	46	48	33	27
(8, 8, 0.5, 1)/(8)	49	55	35	31	
(10, 10, 0.5, 1)/(10)	43	29			
(11, 11, 0.5, 1)/(11)	17	27			
(12, 12, 0.5, 1)/(12)	28	31	36		
(13, 13, 0.5, 1)/(13)	35	27	29		
(15, 15, 0.5, 1)/(15)	x	x	x	x	46
(20, 20, 0.5, 1)/(20)	x	x	x	x	x

for steepness parameters (c_h, c_o) with the sigmoid activation function, a good choice of other parameters h and k in the nonmonotone function speeds up the learning. But finding optimum values of parameters is generally more difficult for a nonmonotone activation function than for a sigmoid one, since the former has many parameters. Tables 4a—4e show the results of BBP and EBP learning with the use of Morita's activation function. We always fixed the k_o value at 1.0, *i.e.*, sigmoid form, simply to get output in the range [0,1],[14] and only c_o changes in the course of learning in the output layer. Parameters c_k, c'_k, h_k and k_k in the hidden layer change according to equations which are like Eq. 10, but more complicated. For the cases which require many iterations at $\varepsilon' = \varepsilon$ $((\gamma_{c_h}, \gamma'_{c_h}, \gamma_{h_h}, \gamma_{k_h})/(\gamma_{c_o}) = (1,1,1,1)/(1))$, we tried further simulations at large ε' $(\gamma_{c_h}, \gamma'_{c_h}, \gamma_{c_o} > 1$ and $\gamma_{h_h} = \gamma_{k_h} = 1$ due to the relative smallness of the derivatives $\partial f_h/\partial c_h$, $\partial f_h/\partial c'_h$ and $\partial f_o/\partial c_o$ with respect to others). Table 4a gives the number of iterations in BBP and in EBP learning for 11 sets of parameters at initial values $h_h = 0.5$ and $k_h = 1$. Comparing Table 4a with Table 2a shows that Morita's activation function speeds up EBP learning 2—3 times compared with the sigmoid function at $\varepsilon' = \varepsilon$ and is also a little better at $\varepsilon' > \varepsilon$ in the region of small parameters c_h, c'_h and c_o. Table 4b lists the final values of $(c_h, c'_h, h_h, k_h)/(c_o)$ in EBP learning, and Morita's activation function $(k_h > 1)$ behaves as a monotonically increasing one in these cases. Even if we keep the k_h value in the range $|k_h| \leq 1.0$, the use of Morita's activation function gives a good result. As shown in Tables 4a and 4b, however, we can get a much better result if we release this restriction. The activation function having a k value of greater than 1.0 takes a value of more than 1.0, and this can be interpreted as an increase of the effective number of units, which is not necessarily an integer, in the hidden layer. Results for EBP learning starting from initial parameters $h_h = 0.5$ and $k_h = -1$ are shown in Tables 4c and 4d, and those for $h_h = 1$ and $k_h = -1$ are shown in Table 4e. Values of $(h_h = 0.5$ and $k_h = -1)$ and $(h_h = 1$ and $k_h = -1)$ seem not to be suitable for BBP learning, contrary to the former case of $h_h = 0.5$ and $k_h = +1$, and BBP learning in these cases gives a somewhat worse result than the sigmoid activation function. Final values of $(c_h, c_h, h_h, k_h)/(c_o)$, corresponding to Table 4c, are shown in Table 4d and Morita's activation function $(k_h < 1)$ behaves as a nonmonotone one in these cases. Overall features of such final values corresponding to Table 4e are similar to this, and are omitted. The results show that EBP with such a nonmonotone activation function speeds up the learning as compared with BBP, though the sigmoid function seems to be better. To say

NII-Electronic Library Service

Table 4b. Final Values of $(c_h, c'_h, h_h, k_h)/(c_o)$ in EBP Learning of Table 4a

	EBP	
	$(\gamma_{c_h}, \gamma'_{c_h}, \gamma_{h_h}, \gamma_{k_h})/(\gamma_{c_o})$	
	(1, 1, 1, 1)/(1)	(10, 10, 1, 1)/(10)
$(c_h, c'_h, k_h, k_h)/(c_o)$	Final $(c_h, c'_h, h_h, k_h)/(c_o)$	
(1, 1, 0.5, 1)/(1)	(3.05, 1.17, −0.21, 2.79)/(3.17)	(5.46, 0.89, 0.34, 1.81)/(5.83)
(2, 2, 0.5, 1)/(2)	(3.23, 2.12, −0.33, 2.59)/(3.35)	(5.53, 1.98, 0.22, 1.77)/(5.86)
(4, 4, 0.5, 1)/(4)	(4.40, 4.02, −0.14, 2.02)/(4.47)	(5.94, 3.98, 0.16, 1.61)/(6.18)
(6, 6, 0.5, 1)/(6)	(6.13, 5.99, 0.15, 1.48)/(6.13)	(7.01, 5.96, 0.27, 1.39)/(6.88)
(8, 8, 0.5, 1)/(8)	(8.05, 7.99, 0.40, 1.20)/(8.02)	(8.39, 7.99, 0.41, 1.22)/(8.01)
(10, 10, 0.5, 1)/(10)	(10.02, 9.99, 0.49, 1.07)/(9.99)	
(11, 11, 0.5, 1)/(11)	(10.999, 10.999, 0.41, 1.15)/(10.968)	
(12, 12, 0.5, 1)/(12)	(11.97, 12.00, 0.39, 1.50)/(11.95)	(11.97, 12.03, 0.48, 1.32)/(11.35)
(13, 13, 0.5, 1)/(13)	(13.007, 13.002, 0.48, 1.40)/(12.98)	(12.999, 13.03, 0.50, 1.39)/(12.70)
(15, 15, 0.5, 1)/(15)		
(20, 20, 0.5, 1)/(20)		
	$(\gamma_{c_h}, \gamma'_{c_h}, \gamma_{h_h}, \gamma_{k_h})/(\gamma_{c_o})$	
	(20, 20, 1, 1)/(20)	(100, 100, 1, 1)/(100)
$(c_h, c'_h, h_h, k_h)/(c_o)$	Final $(c_h, c'_h, h_h, k_h)/(c_o)$	
(1, 1, 0.5, 1)/(1)	(6.30, 0.84, 0.41, 1.62)/(6.59)	(10.41, 0.27, 0.48, 1.30)/(−9.57)
(2, 2, 0.5, 1)/(2)	(6.34, 1.91, 0.33, 1.59)/(6.65)	(9.02, 1.73, 0.45, 1.33)/(9.06)
(4, 4, 0.5, 1)/(4)	(6.72, 3.93, 0.27, 1.50)/(6.83)	(8.72, 3.88, 0.40, 1.32)/(9.30)
(6, 6, 0.5, 1)/(6)	(7.51, 5.94, 0.33, 1.34)/(7.30)	(9.41, 5.91, 0.41, 1.27)/(9.18)
(8, 8, 0.5, 1)/(8)	(8.76, 7.98, 0.41, 1.21)/(8.12)	
(10, 10, 0.5, 1)/(10)		
(11, 11, 0.5, 1)/(11)		
(12, 12, 0.5, 1)/(12)		
(13, 13, 0.5, 1)/(13)		
(15, 15, 0.5, 1)/(15)		(14.64, 15.08, 0.40, 1.45)/(8.32)
(20, 20, 0.5, 1)/(20)		

Table 4c. NI in BBP and in EBP Learning (Morita's Activation Function)

	BBP	EBP $(\gamma_{c_h}, \gamma'_{c_h}, \gamma_{h_h}, \gamma_{k_h})/(\gamma_{c_o})$			
		(1, 1, 1, 1)/(1)	(10, 10, 1, 1)/(10)	(20, 20, 1, 1)/(20)	(100, 100, 1, 1)/(100)
$(c_h, c'_h, h_h, k_h)/(c_o)$	NI	NI	NI	NI	NI
(1, 1, 0.5, −1)/(1)	11129	364	343	576	1192
(2, 2, 0.5, −1)/(2)	2011	161	95	84	56
(4, 4, 0.5, −1)/(4)	448	107	50	34	23
(6, 6, 0.5, −1)/(6)	179	60	48	37	32
(8, 8, 0.5, −1)/(8)	49	49	64	29	
(10, 10, 0.5, −1)/(10)	119	29			
(11, 11, 0.5, −1)/(11)	32	37	26		
(12, 12, 0.5, −1)/(12)	32	31			
(13, 13, 0.5, −1)/(13)	x	x	29		
(15, 15, 0.5, −1)/(15)	228	13807	3143	x	58
(20, 20, 0.5, −1)/(20)	x	x	x	x	x

that the use of a nonmonotone activation function is worse than the use of a sigmoid one may not be correct, however, since another choice of initial parameters h_h and k_h would give better results than the cases of Tables 4c and 4e, as is seen in Table 3 for BBP learning.

Application to the Classification of Activities of Mitomycins Next we applied EBP learning to the classification of the activities of mitomycins by using a 6-12-5 feed-forward neural network with only a sigmoid activation function for simplicity. Using a BBP neural network, Ichikawa *et al.*[4] have already examined this structure–activity relationship (SAR), and found excellent classification and prediction abilities of the neural network. There are 6 structure parameters and 5 classes of observed activities in mitomycins, and they are assigned to 6 neurons in the input layer and 5 neurons in the output layer, respectively. The 16 derivatives are classified into 5 ranks in complete accord with observation, while the classification by the ALS method leads to one case of disagreement, which indicates the excellent classification

NII-Electronic Library Service

Table 4d. Final Values of $(c_h, c'_h, h_h, k_h)/(c_o)$ in EBP Learning of Table 4c

	EBP		
	$(\gamma_{c_h}, \gamma'_{c_h}, \gamma_{h_h}, \gamma_{k_h})/(\gamma_{c_o})$		
	(1, 1, 1, 1)/(1)		(10, 10, 1, 1)/(10)
$(c_h, c'_h, h_h, k_h)/(c_o)$	Final $(c_h, c'_h, h_h, k_h)/(c_o)$		
(1, 1, 0.5, −1)/(1)	(2.52, 2.40, −0.24, −3.39)/(3.24)		(4.38, 5.42, 0.73, −2.03)/(5.87)
(2, 2, 0.5, −1)/(2)	(2.76, 3.39, 1.02, −2.89)/(3.53)		(5.40, 5.86, 0.93, −1.71)/(6.18)
(4, 4, 0.5, −1)/(4)	(4.15, 4.43, 0.58, −2.33)/(4.52)		(5.01, 6.13, 0.60, −1.86)/(6.23)
(6, 6, 0.5, −1)/(6)	(6.04, 6.14, 0.57, −1.88)/(6.14)		(6.27, 6.90, 0.60, −1.75)/(6.81)
(8, 8, 0.5, −1)/(8)	(8.00, 8.06, 0.70, −1.70)/(8.02)		(7.87, 8.14, −0.20, −1.63)/(7.75)
(10, 10, 0.5, −1)/(10)	(10.03, 9.96, 1.27, −0.67)/(9.87)		
(11, 11, 0.5, −1)/(11)	(11.004, 11.003, −0.29, −1.19)/(10.85)		
(12, 12, 0.5, −1)/(12)	(12.004, 11.985, 1.01, −0.79)/(11.90)		(12.62, 12.01, −0.36, −1.24)/(9.02)
(13, 13, 0.5, −1)/(13)			(12.79, 12.88, 0.15, −1.20)/(11.89)
(15, 15, 0.5, −1)/(15)	(15.01, 14.996, 2.47, −0.60)/(14.83)		
(20, 20, 0.5, −1)/(20)			
	$(\gamma_{c_h}, \gamma'_{c_h}, \gamma_{h_h}, \gamma_{k_h})/(\gamma_{c_o})$		
	(20, 20, 1, 1)/(20)		(100, 100, 1, 1)/(100)
$(c_h, c'_h, h_h, k_h)/(c_o)$	Final $(c_h, c'_h, h_h, k_h)/(c_o)$		
(1, 1, 0.5, −1)/(1)	(5.89, 6.66, 0.76, −1.71)/(6.88)		(10.86, 6.59, 0.97, −0.88)/(−9.52)
(2, 2, 0.5, −1)/(2)	(6.77, 6.75, 0.82, −1.47)/(7.02)		(9.48, 9.08, 0.72, −1.20)/(8.21)
(4, 4, 0.5, −1)/(4)	(5.76, 6.99, 0.56, −1.69)/(6.89)		(9.09, 9.34, 0.73, −1.25)/(8.93)
(6, 6, 0.5, −1)/(6)	(6.41, 7.25, 0.55, −1.71)/(7.15)		(7.54, 8.26, 0.63, −1.56)/(8.09)
(8, 8, 0.5, −1)/(8)	(7.83, 8.53, −0.29, −1.48)/(6.80)		
(10, 10, 0.5, −1)/(10)			
(11, 11, 0.5, −1)/(11)			
(12, 12, 0.5, −1)/(12)			
(13, 13, 0.5, −1)/(13)			
(15, 15, 0.5, −1)/(15)			(13.79, 15.29, 1.38, −0.82)/(7.45)
(20, 20, 0.5, −1)/(20)			

Table 4e. NI in BBP and in EBP Learning (Morita's Activation Function)

	BBP	EBP $(\gamma_{c_h}, \gamma'_{c_h}, \gamma_{h_h}, \gamma_{k_h})/(\gamma_{c_o})$			
		(1, 1, 1, 1)/(1)	(10, 10, 1, 1)/(10)	(20, 20, 1, 1)/(20)	(100, 100, 1, 1)/(100)
$(c_h, c'_h, h_h, k_h)/(c_o)$	NI	NI	NI	NI	NI
(1, 1, 1, −1)/(1)	13469	355	167	306	116
(2, 2, 1, −1)/(2)	2966	277	78	35	69
(4, 4, 1, −1)/(4)	485	218	65	41	22
(6, 6, 1, −1)/(6)	135	105	57	36	20
(8, 8, 1, −1)/(8)	47	42	33	29	
(10, 10, 1, −1)/(10)	23	30	24		
(11, 11, 1, −1)/(11)	25	39	25		
(12, 12, 1, −1)/(12)	26	26			
(13, 13, 1, −1)/(13)	43	x	52	93	
(15, 15, 1, −1)/(15)	x	67	39	x	25
(20, 20, 1, −1)/(20)	x	x	x	x	398

ability of the network. They checked the generalization ability by removing 5 arbitrary pieces of data from the 16 training data, and obtained good results. In this case we have to set ε to be 0.03 since a larger value of ε leads to instability in the learning process, and the necessary number of iterations to accomplish the learning is much larger in the case of the norbornene problem.

The number of iterations needed to converge in BBP learning is shown in Fig. 3 as a function of the steepness parameters c_h and c_o. This figure has several sets of (c_h, c_o) marked by x where BBP learning fails and shows more complicated behavior than Fig. 1. The result shown in Table 5 confirms that the overall features of EBP encountered in the norbornene problem are also seen in the present problem. In BBP learning the number of iterations strongly depends on the values of steepness parameters appearing in an activation function and so appropriate setting of those values is very important, despite its difficulty. In EBP, however, fast learning is almost always possible, irrespective of the initial values of steepness

NII-Electronic Library Service

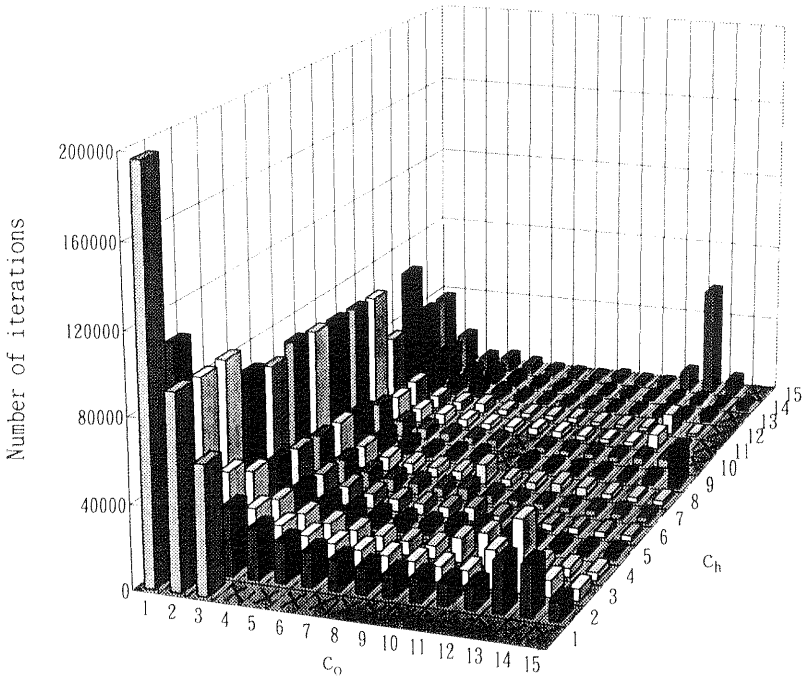

Fig. 3. NI *versus* (c_h, c_o) in BBP Learning for the Mitomycins Problem

Table 5. NI in BBP and in EBP Learning (Sigmoid Activation Function) in the Mitomycins Problem

BBP		EBP (γ_h, γ_o)=(1, 1)	
(c_h, c_o)	NI	Final (c_h, c_o)	NI
(1, 1)	196167	(7.15, 7.50)	6331
(2, 2)	51008	(7.12, 7.35)	3776
(3, 3)	23255	(7.23, 7.34)	3165
(4, 4)	13312	(7.47, 7.52)	3002
(5, 5)	8651	(7.80, 7.82)	2922
(6, 6)	6109	(8.29, 8.32)	2838
(7, 7)	4884	(8.82, 8.98)	4049
(8, 8)	9303		x
(9, 9)	4052	(10.15, 10.07)	3268
(10, 10)	2633	(10.88, 10.75)	2299
(11, 11)	2224	(11.64, 11.55)	2026
(12, 12)	1834	(12.45, 12.38)	1688
(13, 13)	2397	(13.36, 13.32)	2933
(14, 14)	1970	(14.26, 14.19)	2283
(15, 15)	x	(15.31, 15.45)	77910
(20, 20)	x		x

parameters. An exceptional case occurs at initial values of (c_h, c_o)=(8, 8), which reflects the fact that BBP learning itself is unstable around there, as is shown in Fig. 3. This exceptional failure is not a serious problem in EBP learning since we have only to choose any initial values rather than the special value of 8.0.

Concluding Remarks

To speed up the basic back-propagation learning in feed-forward neural networks, which is important for its practical application to pharmaceutical problems, we investigated EBP which adapts parameters appearing in an activation function as well as synaptic weights. Its application to classification problems such as conformations of norbornenes and activities of mitomycins indicated that EBP speeds up the learning and improves the convergence ability beyond the bench mark examples of Sperduti and Starita. In fact, EBP enables the same fast learning, irrespective of the initial values of parameters appearing in an activation function, as BBP does when the optimum values of such parameters are given. Considering the change of the error function in EBP learning, EBP learning can be viewed as equivalent to BBP learning having large effective learning coefficients that depend on units i and j coupling with each other and vary in the course of learning, which explains why EBP speeds up the learning.

We studied whether the generalized activation function given by Morita, which is nonmonotonic or monotonic according to parameter values, is more effective in EBP learning than the usual sigmoid type or not. The result indicated the superiority of Morita's activation function for some initial values of parameters.

It would be interesting to extend the present work to large networks consisting of many units, and to important QSAR problems, where EBP would be much more effective. In principle, EBP would be applicable together with other accelerating convergence methods of BBP, such as the momentum method,[2,15] or the kick out method,[16] or together with a pruning method such as reconstruction learning,[6,8] though the validity of such approaches must be checked by practical application to problems in pharmacology and in other fields.

References and Notes

1) Moriguchi I., Komatsu K., *Chem. Pharm. Bull.*, **25**, 2800—2802 (1977).
2) Rumelhart D. E., Hinton G. E., Williams R. J., "Parallel Distributed Processing," Vol. 1, ed. by Rumelhart D. E., McClelland J. L., MIT Press, 1986, Chapter 8.

NII-Electronic Library Service

3) Aoyama T., Suzuki Y., Ichikawa H., *Chem. Pharm. Bull.*, **37**, 2558—2560 (1989).
4) Aoyama T., Suzuki Y., Ichikawa H., *J. Med. Chem.*, **33**, 905—908 (1990).
5) Aoyama T., Suzuki Y., Ichikawa H., *J. Med. Chem.*, **33**, 2583—2590 (1990).
6) Aoyama T., Ichikawa H., *Chem. Pharm. Bull.*, **39**, 1222—1228 (1991).
7) Ichikawa H., "Multi-Layer Neural Networks," Kyoritsu Press, Tokyo, 1993.
8) Ishikawa M., *Neural Networks*, **9**, 509—521 (1996).
9) Sato K., presented at the 6th Annual Conference of the Japanese Neural Network Society, Sendai, October 1995. The details will be reported elsewhere.
10) Sperduti A., Starita A., *Neural Networks*, **6**, 365—383 (1993).
11) Morita M., *Neural Networks*, **6**, 115—126 (1993).
12) Kotani M., Matsumoto H., Kanagawa T., *Keisoku Jidōseigyo Gakkai Ronbunshu*, **29**, 1465—1473 (1993).
13) Grutzner J. B., Jautelat M., Dence J. B., Smith R. A., Roberts J. D., *J. Am. Chem. Soc.*, **92**, 7107—7120 (1970).
14) The restriction $|k| \leq 1.0$ is needed to keep the output of Morita's activation function in the range [0,1].
15) Hagiwara M., Sato A., *IEICE Trans. Inf. Syst.*, **E78-D**, 1080—1086 (1995).
16) Ochiai K., Toda N., Usui S., *Neural Networks*, **7**, 797—807 (1994).

NII-Electronic Library Service