Neural Processing Letters **13:** 43–53, 2001.

A Simple Neural Network Pruning Algorithm with Application to Filter Synthesis

KENJI SUZUKI[1], ISAO HORIBA[1] and NOBORU SUGIE[2]
[1]*Faculty of Information Science and Technology, Aichi Prefectural University, Nagakute, Aichi, 480–1198 Japan; e-mail: k-suzuki@ist.aichi-pu.ac.jp*
[2]*Faculty of Science and Technology, Meijo University, Nagoya 468-0073, Japan*

Abstract. This paper describes an approach to synthesizing desired filters using a multilayer neural network (NN). In order to acquire the right function of the object filter, a simple method for reducing the structures of both the input and the hidden layers of the NN is proposed. In the proposed method, the units are removed from the NN on the basis of the influence of removing each unit on the error, and the NN is retrained to recover the damage of the removal. Each process is performed alternately, and then the structure is reduced. Experiments to synthesize a known filter were performed. By the analysis of the NN obtained by the proposed method, it has been shown that it acquires the right function of the object filter. By the experiment to synthesize the filter for solving real signal processing tasks, it has been shown that the NN obtained by the proposed method is superior to that obtained by the conventional method in terms of the filter performance and the computational cost.

Key words: generalization ability, image enhancement, neural filter, optimal structure, redundancy removal, right function, signal processing.

Abbreviations:

NN	neural network
BP	back-propagation algorithm
OBD	optimal brain damage method in [13]
ISNR	improvement in signal-to-noise ratio

1. Introduction

Recently, significant progress has been made in applying neural networks (NNs) to signal processing. Filters based on a multilayer NN model, called the neural filters, have been proposed [1–6]. Through training the NN with the sets of input signals and desired signals, it acquires the function of a desired filter. However, the structure of the NN, i.e. the input region of the NN and the number of units in the hidden layer, can not be determined by the training. Thus, there remain many redundant units in the NN. This prevents the NN from acquiring the right function of the object signal processing and makes the generalization ability be lower, the analysis be difficult, and the computational cost increase. Therefore, how to determine the

structures of the NNs has remained a serious issue. Various methods for determining the structures of the NNs have been proposed so far. They can be put into three broad groups:

1. The methods remove the units in the hidden layer on the basis of the performance indices [7–10].
2. The methods evaluate the NNs trained by varying the number of units in the hidden layer by the information criteria [11, 12].
3. The methods gradually diminish the weights during training or remove the weights on the basis of the performance indices [13–17].

Most studies focus their attention on determining the structure of the hidden layer. Moreover, less attention is given to the problems handling continuous valued inputs and outputs such as signal processing in these studies. Therefore, the conventional methods are not necessarily suitable for determining the structures of both the input and the hidden layers of the NNs for signal processing.

In this paper, a method for reducing the structures of both the input and the hidden layers of the NN is proposed in order to acquire the right function of the object signal processing. In the proposed method, the units are removed from the NN on the basis of the influence of removing each unit on the error, and the NN is retrained to recover the damage of the removal. Each process is performed alternately, and then the structure is reduced. Experiments to synthesize a known filter were performed. The NNs obtained by the proposed method and the conventional method are analyzed in order to verify that each NN acquires the right function of the object filter. Moreover, the experiment to synthesize the filter for reducing the quantum noise in X-ray image sequences was performed. The NNs obtained by the proposed method and the conventional method are evaluated in terms of the filter performance and the computational cost. By the analysis and the evaluation, the effectiveness of the proposed method is shown.

2. Method for Reducing the Structure of a Neural Network

2.1. ARCHITECTURE OF THE NN FOR SIGNAL PROCESSING

The NN for signal processing consists of a multilayer NN in which the activation function of the units in the input, hidden, and output layers are an identity function, a sigmoid function, and a linear function, respectively. The inputs to the NN are an object pixel value $g(x, y)$ and spatially/spatiotemporally neighboring pixel values. We explain a three-layered NN as an example. The numbers of units in the input, hidden, and output layers are N_I, N_H, and 1, respectively (here referred to as $N_I - N_H - 1$). An output of the NN is represented by

$$f(x, y) = NN(\boldsymbol{I}_{x,y}), \tag{1}$$

where

$$\boldsymbol{I}_{x,y} = \{g(x-i, y-j) | i, j \in R\} \tag{2}$$

is the input vector to the NN, $NN(\mathbf{I})$ denotes an output of the multilayer NN, and R denotes the input region of the NN. The input vector is rewritten as

$$\boldsymbol{I}_{x,y} = \{I_1, I_2, \cdots, I_m, \cdots I_{N_I}\}, \tag{3}$$

where m denotes a unit number in the input layer. An output of the nth unit in the hidden layer is represented by

$$O_n^H = f_S\left\{\sum_{m=1}^{N_I}(W_{mn}^H \cdot I_m) - W_{0n}^H\right\}, \tag{4}$$

where $W_{mn}^H$is a weight between the mth unit in the input layer and the nth unit in the hidden layer, W_{0n}^H is an offset of the nth unit in the hidden layer, and $f_S(\cdot)$ denotes a sigmoid function. An output of a unit in the output layer is represented by

$$NN(\boldsymbol{I}_{x,y}) = f_L\left\{\sum_{n=1}^{N_H}(W_n^O \cdot O_n^H) - W_0^O\right\}, \tag{5}$$

where W_n^O is a weight between the nth unit in the hidden layer and the unit in the output layer, W_0^O is an offset of the unit in the output layer, and $f_L(\cdot)$ is a linear function. The error to be minimized by training is defined as

$$E = \frac{1}{P}\sum_p (T_C^p - O^p)^2, \tag{6}$$

where p is a pattern number, T_C^p is the pth pattern in the teaching image, O^p is the pth pattern in the output image, and P is the number of patterns. The NN is trained by the back-propagation algorithm (BP) [18] until the error E is smaller than or equal to the predetermined error E_P, or the number of training epochs exceeds the predetermined number T_P.

2.2. THE PROPOSED METHOD

The proposed method reduces the structure of both the hidden layer as well as the input layer by removing the units on the basis of the influence of removing each unit on the error. The unit with the smallest influence is removed first. Let r and $E^{(r)}$ represent a unit number, where the units in both the input and hidden layers are numbered sequentially, and the error after removing the rth unit, respectively. The proposed method is composed of the following steps:

Step 1 Train a large enough NN until $E \leqslant E_P$.
Step 2 Remove the rth unit virtually and calculate $E^{(r)}$.

Step 3 If every unit is examined by removing it virtually and calculating $E^{(r)}$, then go to Step 4, else go back to Step 2.

Step 4 Remove the ath unit where $E^{(a)}$ is the minimum among $E^{(r)}$'s.

Step 5 Retrain the NN, which has been removed of the ath unit, by the BP.

Step 6 If $E \leqslant E_P$; then memorize the weights and the structure, and go back to Step 2; else replace the network by the previous one, and finish the steps.

It is easy to remove a unit virtually by setting an output of a certain unit to zero.

3. Experiments to Synthesize a Known Filter

3.1. TRAINING THE NN BY THE SIMPLE BP

In order to perform the experiments to synthesize the Laplacian filter, the following input image and the teaching image are prepared for training: the input image is 512×512 pixels in size with white uniform random noise, containing various intensities and all frequencies; the teaching image is obtained by filtering it using the Laplacian filter defined as

$$L(x, y) = 4 \cdot g(x, y) - g(x, y-1) - g(x-1, y) - g(x+1, y) - g(x, y+1). \quad (7)$$

The input region of the NN consists of 11×11 pixels, which has enough units to acquire the function of the Laplacian filter. The structure of the NN is 121–50–1. The NN was trained 100,000 epochs by the simple BP with training patterns in the rectangular regions of 50×100 pixels in the input and teaching images. The training converged with the error E of 0.017 after 49.0 hr of CPU execution time on a workstation (UltraSPARC II, 250 MHz, made by Sun Microsystems).

3.2. RESULTS AND EVALUATION

3.2.1. *Reducing the structure of the NN by the proposed method*

In order to compare the proposed method with the simple BP, the predetermined error E_P is set to 0.017, which is the error after training with the simple BP, and the initial structure of the NN is set to 121–50–1. The result of reducing the structure by the proposed method is shown in Table I. It removed 116 and 45 units from the input and the hidden layers of the initial NN, respectively, and obtained the structure of 5–5–1. The result of reducing the structure of the input layer

Table I. Comparison of the performance of the proposed method with that of the conventional method.

Method	Execution time (hours)	Structure of the obtained NN
Simple BP	49.0	121–50–1
OBD	81.5	5–43–1
Proposed method	74.3	5–5–1

corresponds to the input region of the Laplacian filter. This suggests that the proposed method removes only redundant units effectively. In order to verify the optimality of the number of units in the hidden layer, the NNs were trained by varying the number of units in the hidden layer. Consequently, the NN did not finish training where the number of units in the hidden layer is smaller than five. This result demonstrates that the number of units in the hidden layer, obtained by the proposed method, corresponds to the optimal size of the hidden layer.

3.2.2. *Evaluation of the generalization ability*

The error maps, subtractions of the output images of the NNs from the teaching image, are shown in Figure 1. In the error map of the NN obtained by the simple BP (a), the error is small only in the trained region. This effect is due to the over-training. In contrast, the error is small over the whole error map of the NN obtained by the proposed method. The error over the whole error map of the NN obtained by the proposed method is about three times smaller than that of the NN obtained by the simple BP. This means that the NN obtained by the proposed method has high generalization ability.

3.2.3. *Comparison with the conventional method*

The experiments to synthesize the Laplacian filter with several conventional methods were performed. Since the performance of the optimal brain damage method (OBD)

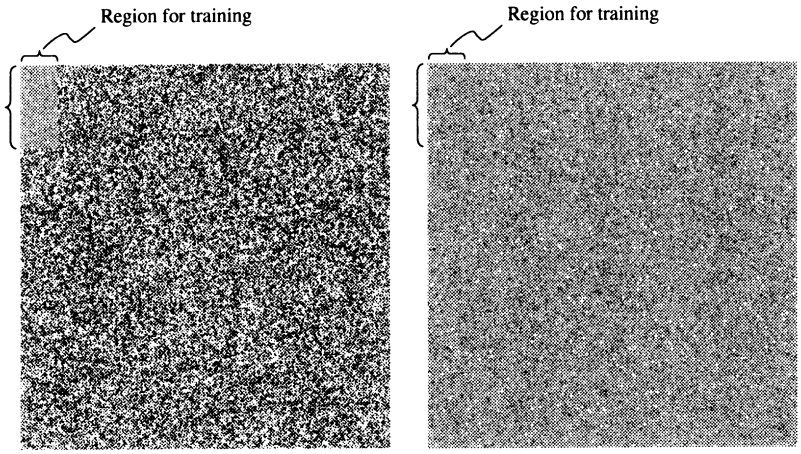

Figure 1. Comparison of the generalization ability of the NN obtained by the proposed method with that obtained by the simple BP. These images are error maps: subtractions of output images of the NNs from the teaching image. The mean absolute errors over the whole error maps (a) and (b) are 5.31% and 1.68%, respectively.

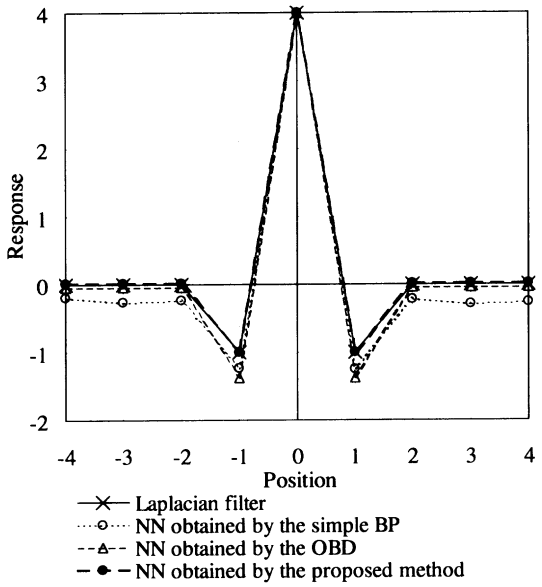

Figure 2. Comparison of the impulse response of the Laplacian filter with that of each NN.

in [13], which is the well-known representative, was the best of all, the result of the OBD is shown on behalf of them. The OBD removes the weights on the basis of the damage estimated from the second derivative of the error with respect to the weights. When all output weights from a certain unit are removed, the unit itself can be removed. The result is shown in Table I. The proposed method obtained the smaller size of the network than the OBD did. This shows that the performance of the proposed method is superior to that of the OBD. In order to verify that each NN acquires the right function of the object filter, its impulse response was measured. The results are shown in Figure 2. These results indicate that no NNs except one obtained by the proposed method acquire the right function of the object filter.

3.3. ANALYSIS OF THE NN OBTAINED BY THE PROPOSED METHOD

The NN obtained by the proposed method is analyzed by modifying the network. The modification for the analysis is shown in Figure 3. The weights of the NN obtained by the proposed method are shown in Table II. First, to modify the NN shown in Figure 3(a) into the NN shown in Figure 3(b), we define the weights as follows:

$$\alpha_n = sgn(W_{3n}^H)\frac{\sum_{m=1}^{5}|W_{mn}^H|}{8}, \tag{8}$$

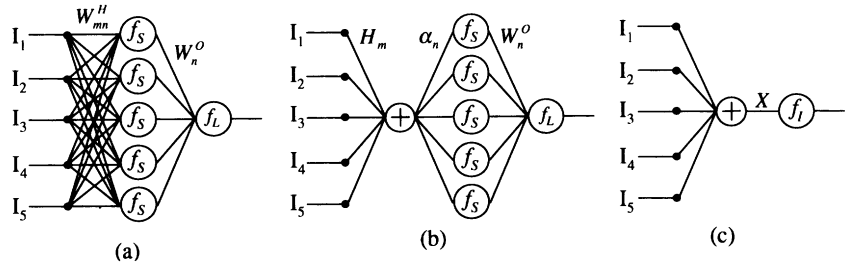

Figure 3. Modification of the network of the NN for analysis. (a) The original NN obtained by the proposed method. (b) The connections between the input and the hidden layers in (a) is modified using the weights H_m and α_n. (c) The network from the weights α_n through the linear function f_L in (b) is modified into a synthesized function f_I.

Table II. Weights of the NN obtained by the proposed method.

Weights between the input and the hidden layers W^H_{mn}						
	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	Offset W^H_{0n}
$n=1$	–3.57	–3.58	14.30	–3.59	–3.61	8.07
$n=2$	3.54	3.56	–14.24	3.55	3.56	7.93
$n=3$	2.89	2.89	–11.56	2.90	2.89	3.21
$n=4$	–2.89	–2.89	11.55	–2.89	–2.88	3.29
$n=5$	2.72	2.72	–10.90	2.72	2.73	–0.03
Weights between the hidden and output layers W^O_n						
	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	Offset W^O_0
	1.12	–1.14	–1.16	1.16	–1.16	0.08

where $sgn(\cdot)$ denotes a sign function. By setting

$$\frac{W^H_{mn}}{\alpha_n} \equiv H_m, \tag{9}$$

an output of the unit in the hidden layer in Equation (4) is rewritten as

$$O^H_n = f_S\left\{\alpha_n \sum_{m=1}^{5} (H_m \cdot I_m) - W^H_{0n}\right\}. \tag{10}$$

By the modification mentioned above, the NN shown in Figure 3(a) is modified into the NN shown in Figure 3(b). By setting

$$\sum_{m=1}^{5} (H_m \cdot I_m) \equiv X, \tag{11}$$

and synthesizing the sigmoid functions $f_S(\cdot)$ and the linear function $f_L(\cdot)$, the NN shown in Figure 3(b) is modified into the NN shown in Figure 3(c) using a

synthesized function as follows:

$$f_I(X) = f_L\left\{\sum_{n=1}^{5} W_n^O \cdot f_S(\alpha_n \cdot X - W_{0n}^H) - W_0^O\right\}. \tag{12}$$

The results of calculation of H_m and α_n are shown in Table III. The mean absolute error between H_m and the coefficients of the Laplacian filter is 0.00185. This shows that H_m's are good approximations of them. The result of the synthesized function is shown in Figure 4. The synthesized function is nearly equal to an identity function; the mean absolute error between the synthesized function and an identity function is 0.0262. This result leads to a conclusion that the NN obtained by the proposed method realizes the Laplcacian filter approximately. The impulsive errors in the Figure 1(b) is rationalized by the characteristics with the saturation of the synthesized function in Figure 4.

4. Experiments to Synthesize an Unknown Filter

In order to synthesize the filter for reducing the quantum noise in X-ray image sequences, the following images are prepared for training: the input image sequence $g_L(x, y, t)$ is made from the high-dose X-ray image sequence, taken at high X-ray

Table III. Results of calculation of H_m and α_n.

n	H_1	H_2	H_3	H_4	H_5	α_n
1	–0.996	–1.001	3.995	–1.002	–1.007	3.58
2	–0.996	–1.002	4.003	–0.997	–1.002	–3.56
3	–1.000	–1.000	4.000	–1.002	–0.998	–2.89
4	–1.001	–1.001	3.998	–1.001	–0.999	2.89
5	–1.000	–0.999	4.001	–0.999	–1.001	–2.72

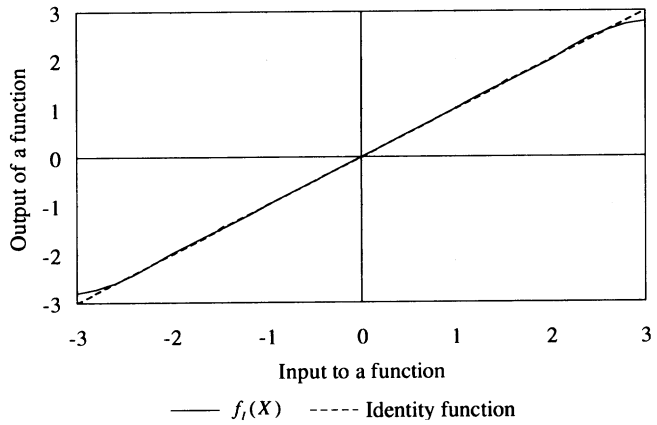

Figure 4. Synthesized function $f_I(X)$.

exposure level, by adding the quantum noise [6]; the high-dose X-ray image is used as the teaching image sequence $T_C(x, y, t)$. In order to handle time-varying images, the spatiotemporal input region of the NN consists of 5×5 pixels in each of 5 consecutive frames. The initial structure of the NN is 125–50–1. The NN was trained 80,000 epochs by each method in the same manner in the previous section. The results are shown in Table IV. The proposed method achieved the smallest size of the network of all.

In order to evaluate the filter performance, the improvement in signal-to-noise ratio (ISNR) [19] is adopted. The ISNR is defined as:

$$ISNR(t) = 10 \log_{10} \left[\frac{\sum_{x,y \in R_E} \{T_C(x, y, t) - g_L(x, y, t)\}^2}{\sum_{x,y \in R_E} \{T_C(x, y, t) - f(x, y, t)\}^2} \right], \tag{13}$$

where R_E is a region for evaluation. The result is shown in Figure 5. The frame number 15 was used in training. It is shown that the NN obtained by the proposed method achieves relatively higher generalization ability.

Table IV. Comparison of the performance of the proposed method with that of the conventional method for the NNs in solving a real signal processing task.

Method	Execution time (hours)	Structure of the obtained NN
Simple BP	49.0	125–50–1
OBD	112.5	58–16–1
Proposed method	91.5	40–9–1

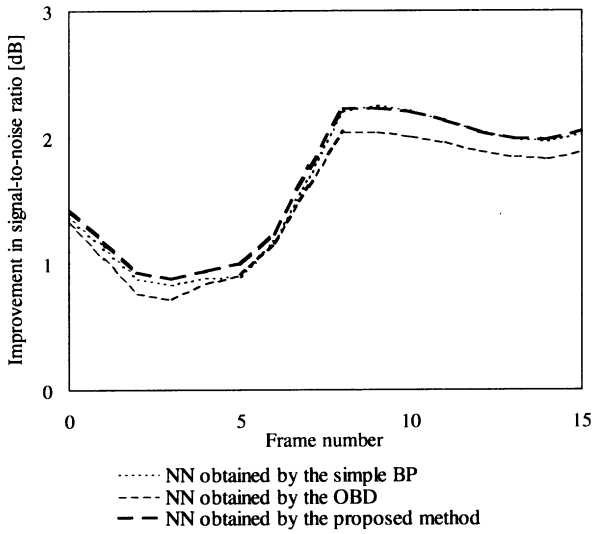

Figure 5. Comparison of the filter performance of the NN obtained by the proposed method with that of the NN obtained by the conventional method.

5. Conclusions

An approach to synthesize desired filters using a NN is described in this paper. In order to acquire the right function of the object filter, a new method, a simple and yet effective method, for reducing the structures of the input and the hidden layers of the NN is proposed. Through the experiments to synthesize the Laplacian filter, it has been shown that the performance of the proposed method is superior to that of the conventional method. By the analysis of the NN obtained by the proposed method, it has been shown that it acquires the right function of the object filter. By the experiment to synthesize the filter for solving the real signal processing task, it has been shown that the NN obtained by the proposed method is superior to that obtained by the conventional method in terms of the filter performance and the computational cost.

Acknowledgements

This work was partially supported by the Ministry of Education, Science, Sports and Culture of Japan under grant-in-aid for quantum information theoretical approach to life science and grant-in-aid for encouragement of young scientists; Kayamori Foundation of Information Science Advancement; and Hibi Foundation. The authors thank Naoko Hara of Meijo University for her helpful cooperation, Prof. Shigeru Okabayashi and Prof. Shin Yamamoto of Meijo University, and Ken Ishikawa and Shigeyuki Ikeda of Hitachi Medical Corporation for their valuable suggestions.

References

1. Yin, L., Astola, J. and Neuvo, Y.: A new class of nonlinear filters – neural filters, *IEEE Trans. Signal Processing* **41**(3) (1993), 1201–1222.
2. Zhang, Z. Z. and Ansari, N.: Structure and properties of generalized adaptive neural filters for signal enhancement, *IEEE Trans. Neural Networks* **7**(4) (1996), 857–868.
3. Yin, L., Astola, J. and Neuvo, Y.: Adaptive multistage weighted order statistic filters based on the back propagation algorithm, *IEEE Trans. Signal Processing* **42** (1994), 419–422.
4. Hanek, H., Ansari, N. and Zhang, Z. Z.: Comparative study on the generalized adaptive neural filter with the other nonlinear filter, *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Vol. I, Minneapolis, MN, April (1993), pp. 649–652.
5. Arakawa, K. and Harashima, H.: A nonlinear digital filter using multi-layered neural networks, *Proc. IEEE Int. Conf. Communications* **2** (1990), 424–428.
6. Suzuki, K., Horiba, I., Sugie, N. and Nanki, M.: Noise reduction of medical x-ray image sequences using a neural filter with spatiotemporal inputs, *Proc. Int. Sympo. Noise Reduction for Imaging & Commu. Systems*, Tokyo, Japan, Nov. (1998), pp. 85–90.
7. Sietsma, J. and Dow, R. J. F.: Creating artificial neural networks that generalize, *Neural Networks* **4**(1) (1991), 67–69.
8. Kameyama, K. and Kosugi, Y.: Neural network pruning by fusing hidden layer units, *Trans. IEICE* **E 74**(12) (1991), 4198–4204.

9. Castellano, G., Fanelli, A. M. and Pelillo, M.: An iterative pruning algorithm for feedforward neural networks, *IEEE Trans. Neural Networks* **8**(3) (1997), 519–531.
10. Hagiwara, M.: Novel back propagation algorithm for reduction of hidden units and acceleration of convergence using artificial selection, *Proc. Int. Joint Conf. Neural Networks* **II** (1990), 625–630.
11. Murata, N., Yoshizawa, S. and Amari, S.: Network information criterion – determining the number of hidden units for an artificial neural network model, *IEEE Trans. Neural Networks* **5**(6) (1994), 865–872.
12. Kurita, T.: A method to determine the number of hidden units of three layered neural networks by information criteria, *Trans. IEICE D-II* **J73-D-II**(11) (1990), 1872–1878 (in Japanese).
13. Cun, Y. L., Denker, J. S. and Solla, S. A.: Optimal brain damage, *Advances in Neural Information Processing*, D.S. Touretzky (ed.), Vol. 2, (1990), pp. 598–605.
14. Weigend, A. S., Rumelhart, D. E. and Huberman, B. A.: Generalization by weight-elimination applied to currency exchange rate prediction, *Proc. Int. Joint Conf. Neural Networks*, Vol. 1, Seattle, USA, pp. 837–841, 1991.
15. M. Ishikawa, Structural learning with forgetting, *Neural Networks* **9**(3) (1996), 509–521.
16. Ji, C., Snapp, R. R. and Psaltis, D.: Generalizing smoothness constraints from discrete samples, *Neural Computation* **2**(1) (1990), 188–197.
17. Nowlan, S. J. and Hinton, G. E.: Simplifying neural networks by soft weight-sharing, *Neural Computation* **4**(4) (1992), 473–493.
18. Rumelhart, D. E., Hinton, G. E. and Williams, R. J.: Learning internal representations by error propagation, *Parallel Distributed Processing*, Vol. 1, Chap. 8, M.I.T. Press, MA (1986), pp. 318–362.
19. Banham, M. R. and Katsaggelos, A. K.: Digital image restoration, *IEEE Signal Processing Magazine* **14**(2) (1997), 24–41.