ハイパフォーマンス
コンピューティング 55－9
（１９９５．３．９）

NONLINEAR SPECTRUM ESTIMATION USING A MODIFIED SELF-ORGANIZED NETWORK

Yoshikazu Miyanaga,*IPSJ member*, Rafiqul Islam *and* Koji Tochinai,*IPSJ member*

Department of Electronic Engineering, Faculty of Engineering
Hokkaido University, Sapporo-shi 060, Japan
Tel. +81-11-706-6534, FAX. +81-11-709-6277 E-mail miyanaga@hudk.hokudai.ac.jp

ABSTRACT

This report proposes a new feedforward neural network which can recognize a set of given data quite correctly. The network has a new training algorithm consisting of a self-organized mechanism and a supervised learning method. The structure of this network consists of a self-organized layer and a supervised layer. The training in the self-organized layer is not only based on the self-organized scheme for input data but also observing the results to the supervised data. The excellent properties of this network are a higher speed training, a better parallel computing and a higher recognition accuracy than any conventional ones. As new application, this network is applied to the nonlinear power spectrum estimation.

KeyWords: self-organized network, high speed learning, nonlinear spectrum estimation, supervied clustering

改良自己組織化ネットワークによる非線形スペクトル推定

宮永　喜一,*IPSJ member*, Rafiqul Islam, 栃内　香次,*IPSJ member*
北海道大学工学部電子工学科、〒０６０札幌市北区北１３条西８丁目

あらまし

本報告では、自己組織化メカニズムと強制学習アルゴリズムの双方を実現した、新しい学習メカニズムをもつフィードフォワードネットワークを提案している。ネットワークの構造は自己組織化を主に行う層と教師付き学習を行う層の２つに分けられる。第１層では入力である学習データの特徴のみで自己組織を行うのではなく、教師データの情報も利用するためネットワークの処理精度が飛躍的に向上する。さらに、自己組織化ネットワークのもつ高速学習のアルゴリズムを用いているため高速学習や並列処理と言った特徴が生かされている。本文ではこれらの特徴を従来の多層パーセプトロンや自己組織化モデルと比較をする事により有効性を示す。特にニューラルネットワークの新しい分野として、音声のARMA分析に見られる非線形スペクトル推定を考え、本手法によって限られた制限内ではあるが非線形のスペクトル推定が可能である事を示す。

キーワード: 自己組織化ネットワーク、高速学習、非線形スペクトル推定、教師付きクラスタリング

1. INTRODUCTION

When the properties of neural networks are considered, some of them, e.g., a multi-layered perceptron [3], [4], [9], can realize good generality and some others, e.g., a self-organized clustering network [1], [2], [9]-[13], have realized high speed learning. The generality of a system means the wide ability for an unknown input data. In order to improve the generality, a function used for data recognition should represent an actual distribution of a given data set, and improve its representation ability in a characteristic space [5],[6],[9],[12]. The perceptron has been recognized to realize such ability and used in many applications. However, the training algorithm, i.e., a back-propagation algorithm, requires a quite large calculation cost, and the network requires both a large size and full-connections to get the above performance.

In order that a network learns within a quite short training time, some self-organized networks have been introduced. If good training rules are implemented in the network, the training time becomes minimum [10]-[13]. However, if the given training rules are not suitable for the observed data set, the recognition ability and its generality cannot be expected high enough [10]-[12]. If such high performance for generality would be kept, all relation among the observed data set should be clearly implemented in this network. Since the self-organization only obtains the characteristics of observed data in the network, the network can hardly know the exact relation among them. Only a supervised training is regarded as a suitable scheme for such learning.

This report introduces a new network in which both high speed training and high recognition accuracy are realized. The network consists of a self-organized module and a supervised module. The self-organized module is trained in part by supervised learning. Compared with the conventional neural networks, i.e., a self-organized neural network and a multi-layered neural network (a perceptron network), the learning ability is higher than them and its convergence speed is greatly improved. The introduced network has two new properties,i.e., the high speed training ability because of a sophisticated self-organization and the high recognition accuracy because of a backward supervised learning. These comparisons are also explored in some experiments in this report.

2. A SELF-ORGANIZED NETWORK WITH A SUPERVISOR

In this section, a self-organized network which is controlled in part by a supervised mechanism for higher generality is introduced. First, the total network is explained and then a training algorithm is described.

A. Two-layer network

The proposed network can be shown in a block diagram of Fig.1. The network consists of two layer network modules. These network modules are called a self-organized module and a supervised module in this report.

A self-organizing module can evaluate stochastic characteristics from input data. The input vectors are fed into every node in parallel. In each node, a similarity value is calculated after the evaluation of distance between a cluster vector, i.e., the center of the node, and an input characteristic vector. The similarity value is calculated by using Maharanobis distance.

The input pattern is defined as a vector which represents the characteristics of an observed data set:

$$\mathbf{y}_c(k) = [y_{c,1}(k), y_{c,2}(k), \cdots, y_{c,s}(k)]^T \qquad (1)$$

where k denotes a sample index, s is the dimension of vector, c ($c = 1, 2, ..., m$) is the index of a c-th node, m denotes the total number of nodes and T denotes a transpose. The above vector is assumed

to be the number of the c-th node. If the total number of members in the c-th node is K_c, k in (1) is limited from 1 to K_c and the total number of input vectors is given by $\sum_{n=1}^{m} K_n$.

In each cluster node, the following three informations are memorized:

- Total number of members (Weighted node counter)

  $t_c(k)$

- Node patter(a center of node)

  $\mathbf{x}_c(t_c(k))$

- Erorr Variance Matrix

  $\mathbf{V}_c(t_c(k))$

where they are the memorized data in the c-th node.

The node counter indicates the total number of input vectors which belong to this node. The node pattern is calculated as a center of the node members and its variance is given as the error variance matrix. The error variance matrix is calculated form the following error:

$$\mathbf{e}_c(k) = \mathbf{y}_c(k) - \mathbf{x}_c(k) \tag{2}$$

From (2), the error variance matrix is calculated from

$$\begin{aligned} \mathbf{V}_c(k) &= diag[\sigma_{c,1}^2(k) \cdots \sigma_{c,s}^2(k)\ ] \\ \sigma_{c,j}^2(k) &= \frac{1}{k}\sum_{n=1}^{k}(y_{c,j}(n) - x_{c,j}(k))^2 \end{aligned} \tag{3}$$

where $y_{c,j}(n)$ indicates the j-th elements of a vector $\mathbf{y}_c(n)$ which belongs to the c-th node. When an input vector is fed into the above node, the following similarity value is calculated:

$$d_c(k) = \exp\left(-\frac{D_c}{2s}\right) \tag{4}$$

$$D_c = \alpha \mathbf{e}_c(k)^T \mathbf{V}_c^{-1}(k)\mathbf{e}_c(k) \tag{5}$$

where α is a rectification factor to the calculation of error variance matrix. This factor is introduced for the adjustment of the evaluation of error variances. In other words, since any exact variances of errors cannot be calculated during the self-organization, all variances are estimated as larger values than actual results.

The Maharanobis distance as distance measure can independently represent the variance of each element, and thus provides better evaluation than Euclid distance. Since the similarity value of each node outputs for a given input vector in parallel, the maximum similarity is then selected among them and its corresponding node is recognized as a cluster node to which the given input vector belongs. In addition, once a cluster node is selected for the input vector, all informations of the node is updated according to the input vector. The updating mechanism is shown in the next subsection.

Each node can represent a closed distribution for a certain cluster. Thus, comparing with a multi-layered perceptron using linear hyper plains for clustering, only independent several nodes can describe a specific cluster distribution.

In order to obtain the final recognition result from the proposed network, a supervised module is used. The structure of the module is the same as the output layer in a multi-layered perceptron neural network. The node of the module is called a decision node in this report. The output of this decision node is calculated by

$$s_j(k) = sigmoid\left(\sum_{i=1}^{m} w_{i,j}(k)d_i(k) - \theta_j(k)\right) \tag{6}$$

$$sigmoid(t) = \frac{1}{1+e^{-t}} \tag{7}$$

where m is the total number of cluster nodes, $w_{i,j}(k)$ denotes the weight between each cluster nodes and a decision node and $\theta_j(k)$ is introduced as a threshold. The evaluation algorithm of $w_{i,j}(k)$ and $\theta_j(k)$ are shown in the next subsection.

B. Forward self-organization and backward training

In this network, two different training algorithms are applied. When the training speed in conventional networks is considered, an algorithm which uses many nonlinear functions and a multilayer structure requires quite large amount of calculation cost. The main reason is understood that the optimum result is buried in a complicated space of solutions including many local minimums. This happens on a multi-layered structure. If one or two layer structure is used, the evaluation cost of the optimum solution may be greatly reduced. However, it has also been known that such one or two layer structure can hardly obtain the optimum solution from nonlinearly divided cluster sets, i.e., this structure can only analyze a linearly divided cluster set. As mentioned in the previous subsection, the introduced network is built of two layered structure, and is easily shown that this network can obtain the optimum solutions even from any nonlinearly divided cluster sets. Accordingly, in addition to the suitable ability for analyzing complicated cluster sets, this network holds high speed training property.

The updating mechanism for a cluster node is given by the simple time averaging as follows:

$$\begin{aligned} \mathbf{x}_c(k+1) &= \Gamma_0(t_c(k+1))\mathbf{x}_c(k) \\ &- \Gamma_1(t_c(k+1))\mathbf{y}_c(k) \end{aligned} \tag{8}$$

$$\Gamma_0(t_c(k+1)) = \frac{t_c(k)}{t_c(k+1)} \tag{9}$$

$$\Gamma_1(t_c(k+1)) = \frac{1}{t_c(k+1)} \tag{10}$$

$$\begin{aligned} \sigma^2_{c,j}(k+1) &= \Gamma_0(t_c(k+1))\sigma^2_{c,j}(k) \\ &+\Gamma_1(t_c(k+1))((y_{c,j}(k) - x_{c,j}(k+1))^2 \\ &-\sigma^2_{c,j}(k)) \end{aligned} \tag{11}$$

$$t_c(k+1) = \lambda t_c(k) + 1 \tag{12}$$

where (8)-(12) is applied only when an input vector is recognized as a member of the c-th node. The value λ is introduced for a forgetting factor of the past data.

In addition, the following initial conditions are applied for the control of a network structure: The total number of node, i.e., m, should be given as a prior information. In addition, the information of node patterns and error variances are given as random values. The value $t_c(k)$ is set to be 1 at the first time.

The self-organization is only applied to the first layer module. Since this network assumes all supervised training data have been prepared, the supervised module does not need the mechanism of self-organization. This module learns the supervised data and the self-organized module is also trained by the same supervised data. Although the self-organized module has been already updated by using an input data, this module is trained again in order to output a better result. Compared with conventional self-organization, all information of a cluster node are controlled twise by using input data and reference output data. Thus, it is considered that the exact information can be realized in every cluster node.

Let us consider the supervised learning algorithm. The algorithm is based on a stochastic approximation method. Since the network consists of two layers and the structure of nodes in each layer are different from that in the other layer, different learning algorithms are applied to them. In the supervised module, the following adaptive algorithm is used:

Define the reference output be

$$\boldsymbol{z}(k) = [\, z_1(k)\ z_2(k) \cdots\ z_n(k)\,] \tag{13}$$

where n denotes the total number of nodes in the supervised module. The input data of this layer consists of the similarities given from the super-

vised module:

$$\boldsymbol{d}(k) = [\, d_1(k)\ d_2(k) \cdots\ d_m(k)\,] \quad (14)$$

If all connection weights have been initially given as random values, the supervised module outputs the results for the input $\boldsymbol{d}(k)$ as

$$\hat{\boldsymbol{z}}(k) = [\, \hat{z}_1(k)\ \hat{z}_2(k) \cdots\ \hat{z}_n(k)\,] \quad (15)$$

All weights are adjusted by minimizing the following criterion:

$$V(k) = (\boldsymbol{z}(k) - \hat{\boldsymbol{z}}(k))^T(\boldsymbol{z}(k) - \hat{\boldsymbol{z}}(k)) \quad (16)$$

If each connection weight is independent of others, the criterion is the same as that in a perceptron neural network. Thus the algorithm in the supervised module is also the same as the back-propagation for the output layer:

$$\Delta w_{ij}(k) = \eta \xi_i d_j(k) + \epsilon \Delta w_{ij}(k-1) \quad (17)$$

$$\xi_i(k) = 2(z_i(k) - \hat{z}_i(k))\hat{z}_i(k)(1 - \hat{z}_i(k)) \quad (18)$$

where η and ϵ are arbitrary positive values and Δw_{ij} stands for the updating value of the connection weight between the i-th node in the supervised module and the j-the node in the self-organized module at time k.

Once all weights are calculated in the supervised module, the information of nodes in the self-organized layer are also updated by using the reference outputs. The main purpose in the training for this layer is the design of higher recognition accuracy than no training case. Thus, the information of node patterns, i.e., $\boldsymbol{x}_i(k)$, are adjusted in this training. The same criterion is applied in this training. Then we get

$$\Delta x_{jt}(k+1) = \epsilon' \Delta x_{jt}(k) + \eta' (\sum_{s=1}^{m} \xi_s w_{sj} \times d_j(k) \alpha (x_{jt}(k) - y_t(k))/(s\sigma_{c,t}^2(k))) \quad (19)$$

The value $\Delta x_{jt}(k)$ is defined as an updating value of a t-th element of the node pattern in the j-th node of the self-organized module.

3. COMPARISONS ON NETWORK SIZES AND TRAINING ABILITY

In this section, two different kinds of results are introduced, i.e., data clustering and spectrum analysis of speech waves.

A. Experiments of Clustering

In this section, some results in comparisons with conventional networks are shown. As conventional networks, a multi-layered perceptron network [3], [4] and a self-organized clustering network based on Maharanobis distance [10]-[13] are used. The perceptron network is trained by using a back-propagation.

The first experiment is the design of an exclusive-OR function. Two kinds of data are yielded. The label /a/ is located in both upper left and lower right, and the label /b/ is in both upper right and lower left. When the 3 layer perceptron with 4 nodes in the input layer, 4 nodes in the hidden layer and 2 nodes in the output layer is applied, the total power of output errors is shown in Fig.2(a). The 3000 sampled data is used for training. On the other hand, when the proposed network is applied with 4 nodes in the self-organized module and 2 ndoes in the supervised module, the total power of output errors is shown in Fig.2(b). From Fig.2, it turns out that the designed network realizes higher convergence speed than the 3 layered perceptron. In addition, the network can correctly recognize the given data even in a nonlinearly divided cluster set.

In the second experiment, more complicated 4 cluster data are used where the input dimension is 3. The convergence speeds are explored to the 3 layer perceptron with 20 nodes in the input layer, 10 nodes in the hidden layer and 4 nodes in the output layer, the self-organized network with a single perceptron, and the proposed network. The self-organized with a single perceptron is designed with a self-organized module without any super-

vised training and a single perceptron as the output layer. The results are shown in Fig.3. According to them, the proposed network can show the highest convergence speed. In addition, Fig.4 shows the error properties for unknown data set. In this figure, the horizontal axis indicates the total number of training times. The vertical axis indicates the 1st and 2nd error counts for 3000 unknown data where the 1st error counts show that the maximum output of the network is not correct to the reference, and the 2nd error counts show that both the maximum output and the next maximum output are not correct. As to the recognition accuracy, the proposed network shows the excellent performance among them.

B. Experiments of Nonlinear Spectrum Estimation

In this subsection, some results on nonlinear spectrum estimation using our network are explained. As spectrum estimation methods, the parametric methods of AR modeling [14] and ARMA modeling [16] have been quite known. If ARMA modeling is used for the estimation of speech spectra, some nonlinear processing techniques are required. Even if a cepstrum analysis [15] as nonparametric method is also applied to that application, they require complicated processing in order to obtain good accuracy.

This report applies the designed network to the nonlinear spectrum estimation in order to explore the possibility of new parallel processing on spectrum analysis using a neural network. In this subsection, 5 Japanese vowels are examined. Those are sampled at 10 kHz frequency by using 12 bit A/D converter. Fig.5 shows the results on spectrum analyses. In these figures, 128 point FFT spectra, 64 point ARMA spectra using Model Identification System [16] and 64 point spectra obtained using the proposed network are presented. The network has been trained with the network structure of 30 input nodes and 64 output nodes by using 25 supervised data. i.e., 5 supervised data for each vowel. Other initial values except the numbers of nodes are the same as the previous. All figures in Fig.5 are given for the training data. Accordingly, this result only shows that the network can memorize the given data set of estimated spectra.

In order to explore the ability of spectrum estimation, the estimates of spectrum envelopes on vowel /o/ are considered in the next experiment. The input dimension of the network is 256 dimensional vector which is given from FFT and the output dimension is set as 64. The other initial conditions are the same as the previous. First 25 training data are used for data learning in the network. Then some FFT spectra which are different from the training spectra are fed into the network as input data. Fig.6(a) and (b) show the training results on vowel /o/. In addition, fig.6(c) and (d) show the estimated results for unknown data. From the results, it is shown that the spectrum envelops can be estimated by using the proposed network although some small parts of the estimates are influenced by the original FFTs.

4. CONCLUSIONS

This report has introduced a new network with a self-organized mechanism and a supervied mechanism for data clustering and recognition. The network consists of a self-organized module and a supervised module. The self-organized module has been mainly controlled by the information of input data. However, this module has been also trained by a supervisor. Compared with the conventional neural networks, i.e., a self-organized neural network and a multi-layer perceptron network, the learning ability of the designed network is quite higher than them and its convergence speed is greatly improved. This results are recognized as new properties that the proposed network can realize the high speed training ability because of a sophisticated self-organization and the high accu-

racy because of a backward supervised learning. These comparisons are also explored in the design experiment of exclusive-OR function and the estimation of speech envelops.

5. ACKNOWLEDGMENT

The authors would like to deeply thank Dr.Masaaki Tougo, Dr.Hirohisa Innami and Dr.Hirohiko Kazato of Yamatake Honeywell Co.Ltd. for fruitful discussions. This study is supported in part by Grant-in-Aid for General Scientific Research (05808031) administered by The Ministry of Education, Science and Culture in Japan.

6. REFERENCES

[1] Teuvo Kohonen:"The neural phonetic typewriter", IEEE Computer, 21, 3, pp.25-40 (March 1988)

[2] T.Kohonen,"Self-organization and associative memory", Springer-Verlag, London, 1987.

[3] D.E.Rumelhart, J.L.McClelland et al:"Parallel distributed processing-Explorations in the Microstructure of Cognition", The MIT press, Cambridge (1986)

[4] Stephen Grossberg (ed.):"Neural networks and Natural intelligence", The MIT Press, London (1988)

[5] Shunichi Amari:"Mathematics of Neural Networks", Sangyo-Tosyo Publisher, Tokyo (1978)

[6] Kaoru Nakano (ed.):"An introduction to Neuro-Computer", Korona Publisher, Tokyo (1990)

[7] O.K. Ersoy, and D.Hong:"Parallel, self-organizing, hierarchical neural networks", IEEE Neural Networks, 1, 2, pp.690-695 (July 1990)

[8] Hidemitsu Ogawa:"Neural network theory as an inverse problem", IEICE, 73, 7, pp.690-695 (July 1990)

[9] "Artificial neural systems", IEEE Computer, Mar. 1988.

[10] Yoshikazu Miyanaga, and Koji Tochinai:"High efficient neural network for parallel signal processing", Tech. Report of IEICE, ICD89-148, pp.31-37, (November 1989)

[11] Y.Miyanaga and K.Tochinai, "Parallel and Adaptive Clustering Method Suitable for a VLSI System", IEEE Proceedings of International Symposium on Circuits and Systems '91, 1,pp.356-369, (August 1991)

[12] Y.Miyanaga, K.Tochinai, "An adaptive recognition using self-organized network",IEEE International Symposium on Circuits and Systems, 1, pp.332-335, (May 1992)

[13] Y.Miyanaga, K.Tochinai, "Parallel VLSI architecture for multi-layer self-organizing cellular network" IEICE Trans. Electron., E76-C, 7, pp.1174-1181, (July 1993)

[14] F.Itakura and S.Saito,"A statistical method for estimation of speech spectral density and formant frequencies", Trans. IECE Japan, 53-A, 1, pp.35-42, (Jan. 1970)

[15] L.R.Rabiner and R.W.Schaber, "Digital processing of speech signals", Prentice Hall Inc (1978)

[16] Y.Miyanaga, N.Miki and N.Nagai, "Adaptive identification of a time-varying ARMA speech model", IEEE Trans. Acoust., Speech, and Sig. Pro., ASSP-34, 3, pp.423-433 (June 1986)

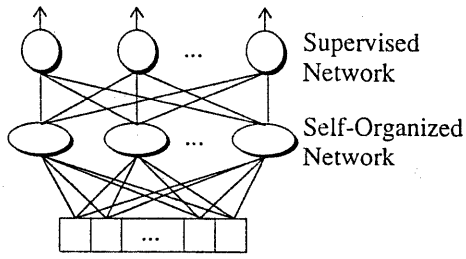

Fig.1 Block Diagram of a Total System.

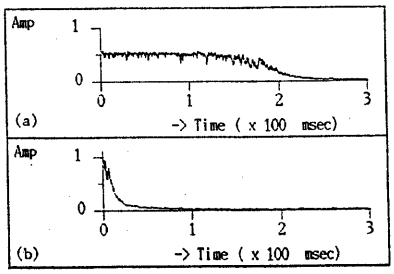

Fig.2 Error curves on Training
Training property of (a) a multi-layered perceptron and (b) the proposed network.

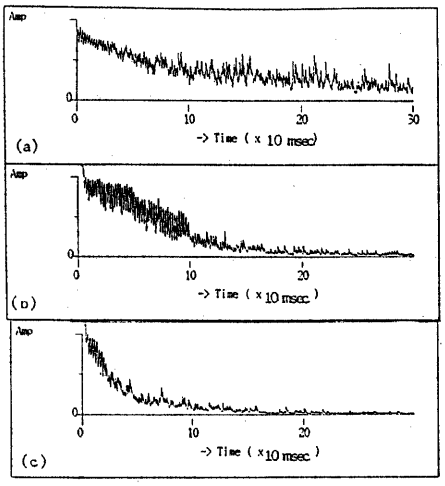

Fig.3 Error curves on Training
Training property of (a) a multi-layered perceptron, (b) a network consisting of a self-organized network and a simple perceptron, and (b) the proposed network.

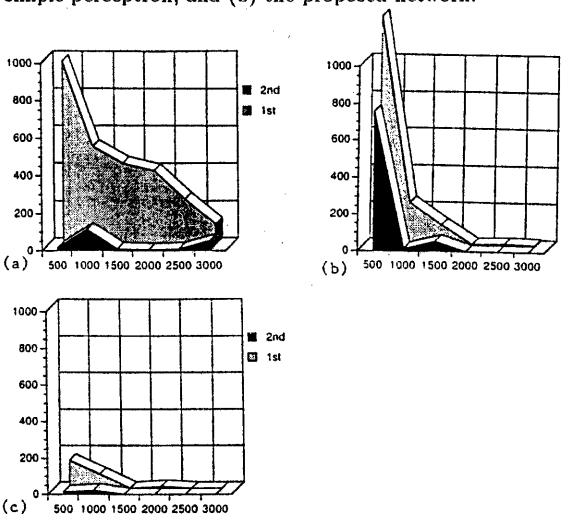

Fig.4 Comparisons of Recognition Ability.
(a) Total number of misrecognition using a multi-layered perceptron where **the 1st** stands for the case when the first candidate is only recognized as the final result and **the second** stands for the case when the first and the second candidates are recognized as the final result. The total number of misrecognition using (b) a self-organized model and a simple perceptron network and (c) the proposed network.

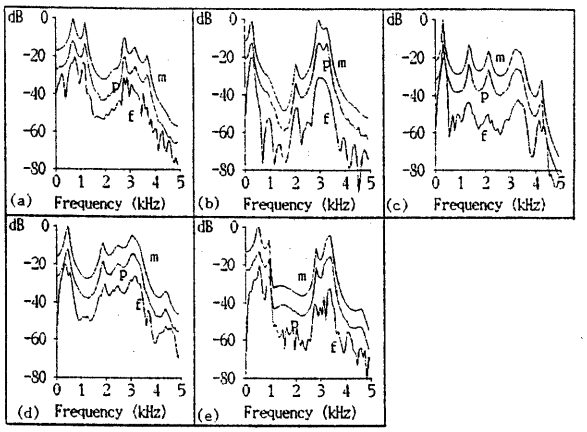

Fig.5 Examples of nonlinear spectrum learning for Japanese 5 vowels using the proposed network.
Each fig. shows (m)MIS analysis spectrum, (p) the proposed network and (f) FFT spectrum of Japanese vowels (a) /a/, (b) /i/, (c) /u/, (d) /e/ and (e) /o/.

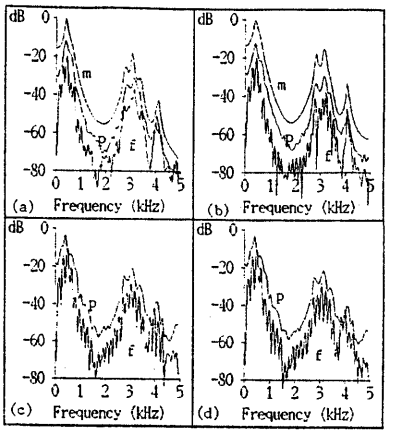

Fig.6 Examples of nonlinear spectrum estimation for Japanese /o/ using the proposed network.
Figs.(a) and (b) show (m)MIS analysis spectrum, (p) the proposed network and (f) FFT spectrum of Japanese vowels /o/ which are given as training data. Figs.(c) and (d) show (p) The proposed network and (f) FFT spectrum of Japanese vowels /o/ which are unknown data.