# Feedback GMDH-type neural network algorithm self-selecting optimum neural network architecture

Tadashi Kondo
School of Health Sciences,
the University of Tokushima,
Tokushima, 770-8509, Japan

Junji Ueno
School of Health Sciences,
the University of Tokushima,
Tokushima, 770-8509, Japan

## Abstract

In this study, a feedback Group Method of Data Handling (GMDH)-type neural network algorithm self-selecting the optimum neural network architecture is proposed. In this algorithm, the optimum neural network architecture is automatically selected from three types of neural network architectures such as the sigmoid function type neural network, the radial basis function (RBF) type neural network and the polynomial type neural network. Furthermore, the structural parameters such as the number of layers, the number of neurons in the hidden layers and the useful input variables are automatically selected so as to minimize the prediction error criterion defined as Akaike's Information Criterion (AIC). The feedback GMDH-type neural network has a feedback loop and the complexity of the neural network increases gradually using feedback loop calculations so as to fit the complexity of the nonlinear system. This algorithm is applied to the identification problem of the complex nonlinear system.

**Keywords:** GMDH, Neural network, Identification

## 1 Introduction

The Group Method of Data Handling (GMDH)-type neural networks and their applications have been proposed in our early works [1],[2]. The GMDH-type neural networks can automatically organize the neural network architecture by using the heuristic self-organization method [3],[4]. The GMDH-type neural networks can also determine such structural parameters as the number of layers, the number of neurons in the hidden layers and the useful input variables. In the GMDH-type neural networks, the neural network architecture is organized so as to minimize the prediction error criterion defined as Akaike's Information Criterion (AIC) [5] or Prediction Sum of Squares (PSS) [6].

In this study, the feedback GMDH-type neural network algorithm self-selecting the optimum neural network architecture is proposed. In this algorithm, the optimum neural network architecture is automatically selected from three types of neural network architectures such as the sigmoid function type neural network, the radial basis function (RBF) type neural network and the polynomial type neural network. The feedback GMDH-type neural network has a feedback loop and the complexity of the neural network increases gradually using feedback loop calculations so as to fit the complexity of the nonlinear system.

The feedback GMDH-type neural network algorithm proposed in this paper is applied to the identification problem of the complex nonlinear system. The optimum neural network architecture fitting the complexity of the nonlinear system is selected from three types of the neural network architectures. The identification results of the feedback GMDH-type neural network are compared with those of the GMDH algorithm and the conventional multi-layered neural network trained using the back propagation algorithm. It is shown that the feedback GMDH-type neural network is a very useful identification method of the complex nonlinear system because the optimum neural network architecture is automatically organized so as to minimize AIC.

## 2 Feedback GMDH-type neural network

The architecture of the feedback GMDH-type neural network proposed in this paper has a feedback loop as shown in Fig.1. The feedback GMDH-type neural network algorithm can select the optimum neural network architecture from three types of neural network architectures such as the sigmoid function type neural network, the RBF type neural network and the polynomial type neural network. The feedback GMDH-type neural network algorithm uses three types of neuron architectures which are the sigmoid function type neuron, the RBF type neuron and the polynomial type neuron.

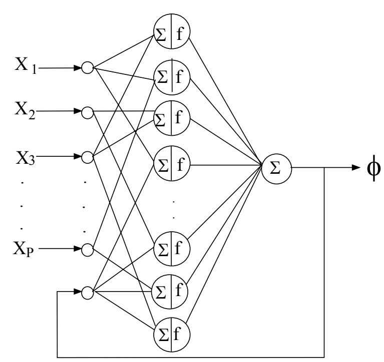


Fig.1 Architecture of the feedback GMDH-type neural network

## 2.1 First loop calculation

First, all data are set to the training data. In this algorithm, it is not necessary to separate the original data into the training and test data because AIC can be used for organizing the network architectures. Then the architecture of the input layer is organized.

**1) Input layer**

$$u_j=x_j \quad (j=1,2,\dots,p) \tag{1}$$

where $x_j$ $(j=1,2,\dots,p)$ are the input variables of the system, and $p$ is the number of input variables. In the first layer, input variables are set to the output variables.

**2) Hidden layer**

All combinations of the $r$ input variables are generated. For each combination, three types of neuron architectures which are the sigmoid function type neuron, the RBF type neuron and the polynomial type neuron, are generated and $L$ neurons which minimize AIC value are selected for each type of neuron architectures.

Furthermore, for each combination, optimum neuron architectures fitting the characteristics of the nonlinear system are automatically selected by using AIC.

**a) Sigmoid function type neuron:**

**i) The first type neuron**

$\Sigma$: (Nonlinear function)

$$z_k= w_1u_i+w_2u_j+w_3u_iu_j+w_4u_i^2+w_5u_j^2+w_6u_i^3+w_7u_i^2u_j+w_8u_iu_j^2+w_9u_j^3-w_0\theta_1 \tag{2}$$

f : (Nonlinear function)

$$y_k=\frac{1}{1+e^{(-z_k)}} \tag{3}$$

**ii) The second type neuron**

$\Sigma$: (Linear function)

$$z_k= w_1u_1+w_2u_2+w_3u_3+\dots+w_ru_r-w_0\theta_1 \quad (r<p) \tag{4}$$

f : (Nonlinear function)

$$y_k=\frac{1}{1+e^{(-z_k)}} \tag{5}$$

**b) RBF type neuron:**

**i) The first type neuron**

$\Sigma$: (Nonlinear function)

$$z_k= w_1u_i+w_2u_j+w_3u_iu_j+w_4u_i^2+w_5u_j^2+w_6u_i^3+w_7u_i^2u_j+w_8u_iu_j^2+w_9u_j^3-w_0\theta_1 \tag{6}$$

f : (Nonlinear function)

$$y_k=e^{(-z_k^2)} \tag{7}$$

**ii) The second type neuron**

$\Sigma$: (Linear function)

$$z_k= w_1u_1+w_2u_2+w_3u_3+\dots+w_ru_r-w_0\theta_1 \quad (r<p) \tag{8}$$

f : (Nonlinear function)

$$y_k=e^{(-z_k^2)} \tag{9}$$

**c) Polynomial type neuron:**

**i) The first type neuron**

Σ: (Nonlinear function)

$$z_k= w_1u_i+w_2u_j+w_3u_iu_j+w_4u_i^2+w_5u_j^2+w_6u_i^3+w_7u_i^2u_j+w_8u_iu_j^2+w_9u_j^3-w_0\theta_1 \tag{10}$$

f : (Linear function)

$$y_k= z_k \tag{11}$$

**ii) The second type neuron**

Σ: (Linear function)

$$z_k= w_1u_1+w_2u_2+w_3u_3+\dots+w_ru_r-w_0\theta_1 \quad (r<p) \tag{12}$$

f : (Linear function)

$$y_k= z_k \tag{13}$$

Here, $\theta_1=1$ and $w_i$ $(i=0,1,2,\dots)$ are the weights between the first and second layer and estimated by applying the stepwise regression analysis [7] to the training data. Only useful input variables $u_i$ $(i=1,2,\dots)$ are selected by using AIC. In the first type neuron, the value of $r$, which is the number of input variables $u$ in each neuron, is set to two. In the second type neuron, the value of $r$, which is the number of input variables $u$ in each neuron, is set to be greater than two and smaller than $p$. $p$ is the number of input variables $x_i$ $(i=1,2,\dots,p)$. The output variables $y_k$ of the neurons are called as the intermediate variables.

$L$ neurons having the smallest AIC values are selected for three types of neuron architectures which are the sigmoid function type neuron, the RBF type neuron and the polynomial type neuron. The output variables $y_k$ of $L$ selected neurons for three types of neuron architectures are set to the input variables of the neurons in the output layer.

**3) Output layer**

For three types of neural network, the outputs $y_k$ of the neurons in the hidden layer are combined by the following linear function.

$$\phi^* = a_0+\sum_{k=1}^{L}a_k y_k \tag{14}$$

Here, $L$ is the number of combinations of the input variables and $y_k$ is the intermediate variables. The useful intermediate variables $y_k$ are selected by using the stepwise regression analysis in which AIC is used as the variable selection criterion.

Equation (14) is calculated for three types of neural network architectures which are the sigmoid function type neural network, the RBF type neural network and the polynomial type neural network. Then, the neural network architecture which has smallest AIC value is selected as the GMDH-type neural network architecture from three types of neural network architectures

Then, the estimated output values $\phi^*$ which is selected in the output layer is used as the feedback value and it is combined with the input variables in the next loop calculation.

## 2.2 Second and successive loop calculations

The optimum neural network architecture is selected from three types of neural network architectures in the output layer. Therefore, in the second and successive loop calculations, only one type of neuron architecture, which is the sigmoid function type neuron or the RBF type neuron or the polynomial type neuron, is used for the calculation.

First, the estimated output value $\phi^*$ is combined with the input variables and all combinations between the estimated output value $\phi^*$ and the input variables are generated. The same calculation as the first feedback loop is carried out for each combination. Here, only one type of neuron architecture, which is selected in the first loop calculation, is used in the calculation. When AIC value of the linear function in (14) is increased, the loop calculation is terminated and the complete neural network architecture is organized by the $L$ selected neurons in each feedback loop.

## 3 An application to nonlinear system identification

To verify the performance of the feedback GMDH-type neural network, it is applied to the nonlinear system identification problem.

### 3.1 Nonlinear system identification problem

The nonlinear system is assumed to be described by the following equations:

$$\phi_1 = (1.0+1.1x_1+1.2x_2+1.3x_3)^4 + \varepsilon_1 \quad (15)$$

$$\phi_2 = (1.0+1.4x_1+1.5x_2+1.6x_3)^4 + \varepsilon_2 \quad (16)$$

$$\phi_3 = (1.0+1.7x_1+1.8x_2+1.9x_3)^4 + \varepsilon_3 \quad (17)$$

$$\phi_4 = (1.0+2.0x_1+2.1x_2+2.2x_3)^4 + \varepsilon_4 \quad (18)$$

Here, $\phi_1 \sim \phi_4$ show output variables and $x_1 \sim x_3$ show input variables. $\varepsilon_1 \sim \varepsilon_4$ show noises. Furthermore, $x_4$ is added as the input variable of the neural network in order to check that the feedback GMDH-type neural network can eliminate the useless input variables. The neural network is organized by using twenty training data. The prediction is carried out by using twenty testing data so as to check the generalization ability.

### 3.2 Identification results obtained by using the GMDH-type neural network

**(1) Input variables**

Four input variables were used but the useless input variables $x_4$ was automatically eliminated.

**(2) Number of selected neurons**

Four neurons were selected in the hidden layer.

**(3) Selection of the neural network architecture**

Polynomial type neural network architecture was selected as the GMDH-type neural network architecture in the first feedback loop calculation.

**(4) Structure of the neural network**

The calculation of the GMDH-type neural network was terminated at the seventh feedback loop calculation.

**(5) Estimation accuracy**

The estimation accuracy was evaluated by using the following equation:

$$J_1 = \frac{\sum_{i=1}^{20} |\phi_i - \phi_i^*|}{\sum_{i=1}^{20} |\phi_i|} \quad (19)$$

where $\phi_i$ $(i = 1,2,\cdots,20)$ were the actual values and $\phi_i^*$ $(i = 1,2,\cdots,20)$ were the estimated values by the feedback GMDH-type neural network. The values of $J_1$ for four output variables are shown in Table1.

**(6) The prediction accuracy**

The prediction accuracy was evaluated by using the following equation:

$$J_2 = \frac{\sum_{i=21}^{40} |\phi_i - \phi_i^*|}{\sum_{i=21}^{40} |\phi_i|} \quad (20)$$

where $\phi_i$ $(i = 21,22,\cdots,40)$ were the actual values and $\phi_i^*$ $(i = 21,22,\cdots,40)$ were the predicted values by the feedback GMDH-type neural network. The values of $J_2$ for four output variables are shown in Table1.

**Table 1 Prediction and estimation accuracy**

| Models | J | Φ1 | Φ2 | Φ3 | Φ4 |
|---|---|---|---|---|---|
| GMDH-NN | J1 | 0.013 | 0.022 | 0.023 | 0.024 |
| | J2 | 0.025 | 0.028 | 0.029 | 0.029 |
| GMDH | J1 | 0.056 | 0.058 | 0.038 | 0.039 |
| | J2 | 0.055 | 0.058 | 0.044 | 0.044 |
| NN | J1 | 0.119 | 0.133 | 0.108 | 0.11 |
| | J2 | 0.114 | 0.133 | 0.102 | 0.109 |

**(7) Variation of AIC and estimated values**

The variation of AIC in Eq.(14) of the output variables $\phi_1$ is shown in Fig.2. It decreased gradually by the feedback loop calculations and converged at the seventh feedback loop calculation. The estimated values of $\phi_1$ by the feedback GMDH-type neural network is shown in Fig.3. We can see that the estimated values are very accurate.

### 3.3 Comparison of the feedback GMDH-type neural network and other models

The identification results were compared with those of the GMDH algorithm and the conventional multilayered neural network trained using the back propagation algorithm.

**(1) The GMDH algorithm**

The identification results were referred from [8]. Four input variables were used but the useless input variable $x_4$ was automatically eliminated. Four intermediate variables were selected. The calculation was terminated at the fourth layer.

The values of $J_1$ and $J_2$ are shown in Table1.

**(2) The conventional multilayered neural network**

The neural network had three layered structures. Four input variables were used in the input layer and eight neurons were used in the hidden layer. The neurons in the output layer were described by the linear function. The values of $J_1$ and $J_2$ are shown in Table1.

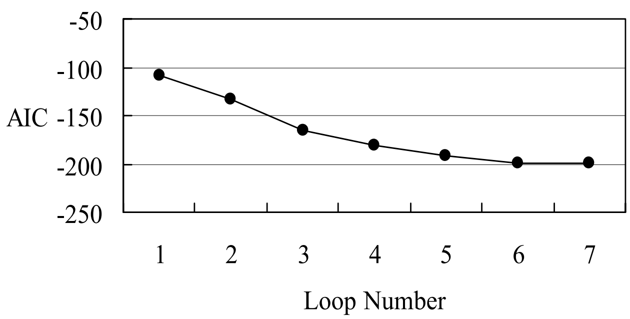


Fig.2 AIC variation

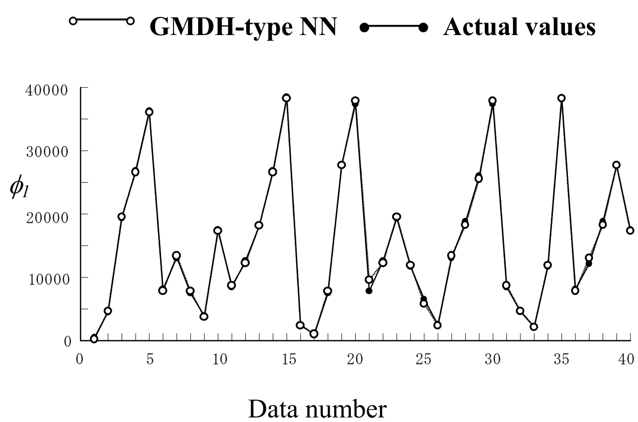


Fig.3 Estimated values of $\phi_1$ by feedback GMDH-type neural network

## 4 Discussion

From the identification results, we can say the following:

**(1)** Both estimation and prediction errors of the feedback GMDH-type neural network were smallest in the three identified models. We can see that the feedback GMDH-type neural network was a very accurate identification method for the nonlinear system.

**(2)** In the feedback GMDH-type neural network, AIC value at the first loop calculation was not small but it was gradually decreased by the feedback loop calculations. So we can see that the feedback loop calculation plays a very important role in the feedback GMDH-type neural network.

**(3)** In the conventional neural network, the effects of high order terms of the input variables are not considered. Furthermore, it does not have the ability of self-selecting useful input variables and the optimum neural network architecture. So the accuracy of the neural network was not good.

The feedback GMDH-type neural network can organize the conventional neural network architecture (sigmoid function type architecture) and the GMDH architecture (polynomial type architecture). This algorithm contains the characteristics of the conventional neural network and the GMDH algorithm and it is a very flexible method for the identification problem of the complex nonlinear system.

## 5 Conclusion

In this study, the feedback GMDH-type neural network self-selecting optimum neural network architecture was proposed. This algorithm can automatically organize a multilayered neural network architecture fitting the complexity of the nonlinear system by using the heuristic self-organization method. It is very easy to apply this algorithm to the identification problem of the practical complex system because the optimum neural network architecture fitting the complexity of the nonlinear system is automatically organized. This algorithm was applied to the nonlinear system identification problem and it was shown that this algorithm was a very useful method.

## References


[1] T.Kondo, "GMDH neural network algorithm using the heuristic self-organization method and its application to the pattern identification problem", *Proc. of the 37th SICE Annual Conference*, pp.1143-1148, 1998.

[2] T.Kondo, "Revised GMDH-type neural networks using prediction error criterions AIC and PSS", *Proc. of SCIS & ISIS 2004*, *WP-6-4*, pp.1-6, 2004.

[3] S.J.Farlow ed, *Self-organizing methods in modeling, GMDH-type algorithm*, Marcel Dekker, Inc., New York, 1984.

[4] A.G.Ivakhnenko, "Heuristic self-organization in problems of engineering cybernetics", *Automatica*, Vo.6, No.2, pp.207-219, 1970.

[5] H.Akaike, "A new look at the statistical model identification", *IEEE Trans. Automatic Control,* Vol.AC-19, No.6, pp.716-723, 1974.

[6] H.Tamura and T.Kondo, "Heuristics free group method of data handling algorithm of generating optimum partial polynomials with application to air pollution prediction", *Int. J. System Sci.,* Vol.11, No.9, pp.1095-1111, 1980.

[7] N.R.Draper and H.Smith, *Applied Regression Analysis*, John Wiley and Sons, New York, 1981.

[8] T.Kondo, "GMDH type Neural Network Algorithm Identifying a Network Structure with the Heuristic Self-Organization Method", *Trans. ISCIE*, Vol.11, No.4, pp.198-207, 1998.