Journal of the Korean Institute
of Industrial Engineers
Vol.20, No.4, December., 1994

Forecasting Using Interval Neural Networks: Application to Demand Forecasting

Kitaek Kwon*, Hisao Ishibuchi* and Hideo Tanaka*

Abstract

Demand forecasting is to estimate the demand of customers for products and services. Since the future is uncertain in nature, it is too difficult for us to predict exactly what will happen. Therefore, when the forecasting is performed upon the uncertain future, it is realistic to estimate the value of demand as an interval or a fuzzy number instead of a crisp unmber. In this paper, we propose a demand forecasting method using the standard back-propagation algorithm and then we extend the method to the case of interval inputs. Next, we demonstrate that the proposed method using the interval neural networks can represent the fuzziness of forecasting values as intervals. Last, we propose a demand forecasting method using the transformed input variables that can be obtained by taking account of the degree of influence between an input and an output.

1. Introduction

Forecasting methods can be classified by the following three categories: *qualitative techniques, time series analysis and causal models* [2]. The qualitative technique is a method that performs an analysis by using the expert opinion and judgement, while time series analysis is performed on the basis of historical data. The causal method is the most sophisticated kind of forecasting method. It expresses mathematically the causal relationship between the variable to be forecasted and other variables. A good forecasting can be achieved by choosing an appropriate forecasting method from various methods. For example, Parker et al.[8] showed that a good result can be obtained by the forecasting method using

* Department of Industrial Engineering, College of Engineering, University of Osaka Prefecture, Osaka 593, JAPAN(大阪府立大學 工學部 經營工學科)

regression analysis. Kimura et al.[6] performed time series analysis by layered neural networks.

Since there is an uncertainty in many variables, it is realistic to estimate the value of the future demand as an interval or a fuzzy number insted of a crisp number. Tanaka et al.[10] proposed fuzzy linear regression analysis in order to interpret a fuzzy phenomenon and formulated fuzzy regression analysis with the degree of similarity[11] which can be viewed as the degree of importance. Thus, input-output data with large importance are weighed in fuzzy regression analysis. Also, fuzzy GMDH[12] and fuzzy regression analysis using neural networks[3-5] were proposed to derive a nonlinear fuzzy regression model.

In this paper, we propose a demand forecasting method using interval neural networks. First, we propose a demand forecasting method using the standard back-propagation algorithm[9] and then we extend the method to the case of interval inputs. Next, we demonstrate that the proposed forecasting method using the interval neural networks[4] can represent the fuzziness of forecasting values as intervals. Last, we demonstrate that the forecasting ability of the neural networks can be improved by introducing the transformation of input variables which takes account of the degree of influence between an input variable and an output variable.

2. Forecasting method using neural networsks

2.1 Learning algorithm

Let us assume that m pairs of input vectors and target outputs are given as $(\boldsymbol{x}_p, y_p)$, $p=1,2,\cdots,m$, where $\boldsymbol{x}_p=(x_{p1},x_{p2},\cdots,x_{pn})$ is an n-dimensional input vector and y_p is an output variable (i.e., a target output of neural networks). The input-output relation of a three-layer feedforward neural network with n input units, n_2 hidden units and a single output unit is defined for the n-dimensional real number input vector $\boldsymbol{x}_p=(x_{p1},x_{p2},\cdots,x_{pn})$ as follows[9].

Input layer:

$$o_{pi}=x_{pi},\quad i=1,2,\cdots,n, \tag{1}$$

Hidden layer:

$$o_{pj}=f(net_{pj}),\quad j=1,2,\cdots,n_2, \tag{2}$$

$$net_{pj}=\sum_{i=1}^{n} w_{ji}o_{pi}+\theta_j,\quad j=1,2,\cdots,n_2, \tag{3}$$

Output layer:

$$o_p=f(net_p), \tag{4}$$

$$net_p=\sum_{j=1}^{n_2} w_j o_{pj}+\theta, \tag{5}$$

where the activation function $f(\cdot)$ is the sigmoid function:

$$f(x) = 1/(1 + exp(-x)), \tag{6}$$

and the weights w_{ji}, w_j and the biases θ_j, θ are real numbers.

The learning of the neural network for the training pattern $(\boldsymbol{x}_p, y_p)$ is performed in order to minimize the cost function:

$$E_p = (y_p - o_p)^2/2. \tag{7}$$

The learning algorithm can be derived from the cost function E_p in (7). The weights w_j and w_{ji} are changed by the following rules:

$$\Delta w_j(t+1) = -\eta(\partial E_p/\partial w_j) + \alpha\Delta w_j(t), \tag{8}$$
$$\Delta w_{ji}(t+1) = -\eta(\partial E_p/\partial w_{ji}) + \alpha\Delta w_{ji}(t), \tag{9}$$

where η is the learning rate, α is the momentum constant and t indexes the number of adjustments of the weights. The derivatives of $\partial E_p/\partial w_j$ and $\partial E_p/\partial w_{ji}$are explicitly caculated as follows.

$$\partial E_p/\partial w_j = \delta_p o_{pj}, \tag{10}$$
$$\partial E_p/\partial w_{ji} = \delta_{pj} o_{pi}, \tag{11}$$

where

$$\delta_p = -(y_p - o_p)o_p(1-o_p),$$
$$\delta_{pj} = o_{pj}(1-o_{pj})\delta_p w_j.$$

The biases θ and θ_j are changed in the same manner as the weights w_j and w_{ji}, respectively.

2.2 Application to demand forecasting

The standard back-propagation algorithm[9] shown in the previous section is applied to the demand forecasting of Cherryoak Company. The demand data for 24 years (1947~1970) of the Cherryoak Company are given as shown in Table 1[8]. In this paper, the learning of the neutal network is performed with the data from the p-th to the $(p+t-1)$-th year in Table, i.e., $((H_p,I_p,M_p),S_p)\sim((H_{p+t-1}, I_{p+t-1}, M_{p+t-1}), S_{p+t-1})$, and then the demand of the next year (i.e.,$(p+t)$-th year) is forecasted, where $p=1947$ and $t=12,13,\cdots,23$. In the learning of the neural network, the output variable y_p is the actual sales S_p that is normalized into a real number in the closed interval [0.1,0.9] and the input variables (x_{p1},x_{p2},x_{p3}) are given by the input variables $[H_p,I_p,M_p]$ which are also normalized in the same manner as the output variable. Using the normalized training data, we trained the neural network with three input units, six hidden units and a single output unit by the learning algorithm wiht $\eta=0.5$ and $\alpha=0.9$. The initial values of the weights and the biases were randomly specified as real numbers in the closed interval [-1, 1].

The explicit caculation was done through the following steps:

[Step 1] Let us assume that training patterns are given as the data from the p-th to the $(p+t-1)$-th year in Table 1, where $p=1947$ and $t=12$.

[Step 2] Specify the initial values of the

Table 1. Data for 24 years of Cherryoak Co.(quoted from [8])

Year (p)	Housing starts(H_p) [thousands]	Disposable personal income(I_p) [$ billions]	New marriages(M_p) [thousands]	Company sales(S_p) [$ millions]
1947	744	158.9	2,291	92.920
1948	942	169.5	1,991	122.440
1949	1,033	188.3	1,811	125.570
1950	1,138	187.2	1,580	110.460
1951	1,549	205.8	1,667	139.400
1952	1,211	224.9	1,595	154.020
1953	1,251	235.0	1,539	157.590
1954	1,225	247.9	1,546	152.230
1955	1,354	254.4	1,490	139.130
1956	1,475	274.4	1,531	156.330
1957	1,240	292.9	1,585	140.470
1958	1,157	308.5	1,518	128.240
1959	1,341	318.8	1,451	117.450
1960	1,531	337.7	1,494	132.640
1961	1,274	350.0	1,527	126.160
1962	1,327	364.4	1,547	116.990
1963	1,469	385.3	1,580	123.900
1964	1,615	404.6	1,654	141.320
1965	1,538	436.6	1,719	156.710
1966	1,488	469.1	1,789	171.930
1967	1,173	505.3	1,844	184.790
1968	1,299	546.3	1,913	202.700
1969	1,524	590.0	2,059	237.340
1970	1,479	629.6	2,132	254.930

weights and the biases.

[Step 3] Calculate the output o_p corresponding to the input vector.

[Step 4] Compare the actual output o_p with the target output y_p, and caculate the value of the cost function.

[Step 5] Adjust the weights and the biases using the cost function.

[Step 6] Repeat from [Step 3] to [Step 5] for all the input-output data.

[Step 7] If a prespecified stopping condition is not satisfied then return to [Step 3] else go to [Step 8]. In this paper, we used the total number of iterations of these steps (from [Step 3] to [Step 6]) as a stopping condition

[Step 8] If a prespecified stopping condition is satisfied then estimate the output of the $(p+t)$-th year using the input values of the

$(p+t)$-th year.

[Step 9] If a prespecified experiment condition is not satisfied then return to [Step 2] else go to [Step 10]. In this paper, we used the total number of trials of these steps (from [Step 2] to [Step 8]) as a experiment condition.

[Step 10] If a prespecified experiment condition is satisfied then forecast the demand of the $(p+t)$-th year using the average of the estimated outputs.

[Step 11] Let $t := t+1$

[Step 12] If $t=24$ the stop else return to [Step 2].

We here assume that the stopping condition is 10000 iterations and experiment condition is 10 trials. Since the result of the learning of neural networks depends on the initial values of the weights and the biases, we use the average of the outputs over 10 experiments as the demand estimate. The average of the outputs obtained from the trained neural networks are shown in Fig.1, where the solid line stands for the average outputs for the training data and the dotted line stands for the average outputs for the test data. From Fig.1, we can observe both the good result to the training data and the good forecasting to the test data.

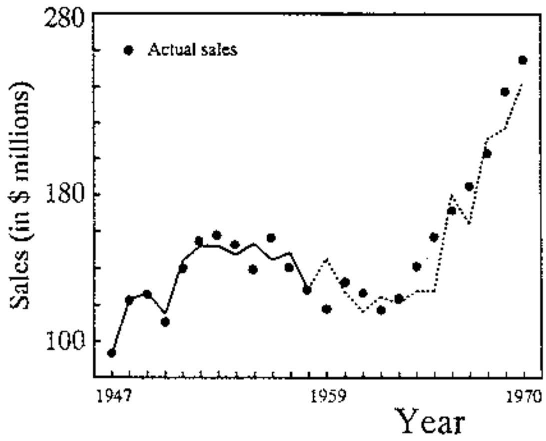

Fig. 1 Demand forecasting by the neural network.

2.3 Forecasting with interval input variables

In this section, we perform a demand forecasting with interval input variables. To consider th fuzziness of the input variables, let us use $H_{p+t} \pm \Delta H_{p+t}$(thousand), $I_{p+t} \pm \Delta I_{p+t}$(billions) and $M_{p+t} \pm \Delta M_{p+t}$(thousand) as input variables where ΔH_{p+t}, ΔI_{p+t} and ΔM_{p+t} stand for the fuzziness of all input variables. Therefore, the input vector of $(p+t)$-th year becomes a three-dimensional interval input vector (i.e., $([H_{p+t}-\Delta H_{p+t}, H_{p+t}+\Delta H_{p+t}], [I_{p+t}-\Delta I_{p+t}, I_{p+t}+\Delta I_{p+t}], [M_{p+t}-\Delta M_{p+t}, M_{p+t}+\Delta M_{p+t}])$) The input-output relation of neural network in (1)~(5) can be extended to the case of the input vector $\boldsymbol{X}_p$ as follow[7], where $\boldsymbol{X}_p=(X_{p1}, X_{p2}, \cdots, X_{pn})$ is an n-dimensional interval vector.

Input layer:

$$O_{pi} = X_{pi}, \quad i=1,2,\cdots,n, \tag{14}$$

Hidden layer:

$$O_{pj} = f(net_{pj}), \quad j=1,2,\cdots,n_2, \tag{15}$$

$$Net_{pj} = \sum_{i=1}^{n} w_{ji} O_{pi} + \theta_j, \ j = 1,2,\cdots,n_2, \tag{16}$$

Output layer:

$$O_p = f(Net_p), \tag{17}$$

$$Net_p = \sum_{j=1}^{n_2} w_j O_{pj} + \theta, \tag{18}$$

Where O_{pi}, O_{pj}, O_p, Net_{pj} and Net_p are intervals. The input-output relation in (14)~(18) can be explicitly caculated by interval arithmetic[1] (see Kwon et al. [7]).

In this paper, let us assume that the input variables have the fuzziness of $\Delta H_{p+l} = 30$, $\Delta I_{p+l} = 5$, and $\Delta M_{p+l} = 30$. Demand forecasting results for the interval inputs are shown in Fig.2. In Fig.2, the dotted two lines are the upper limit (F^u) and the lower limit (F^L) of the forecasted interval demand, respectively.

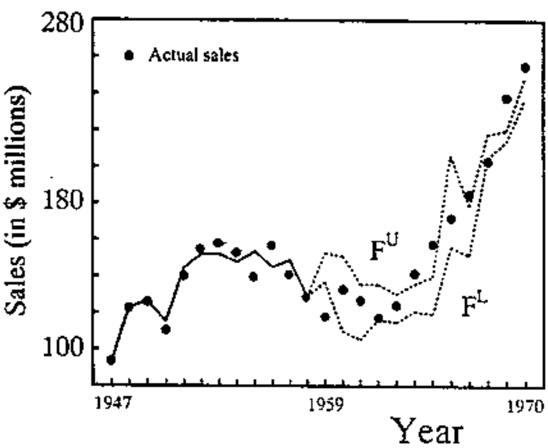

Fig. 2 Demand forecasting with the interval inputs.

3. Forecasting method using interval neural networks

3.1 Learning algorithm

Let us define a three-layer feedforward neural network that has interval weights and interval biases. The input-output relation of the neural network with n inputs, $n_2$hidden units and a single output is defined for the n-dimensional real vector $\boldsymbol{x}_p = (x_{p1}, x_{p2}, \cdots, x_{pn})$ as follows[4].

Input layer:

$$o_{pi} = x_{pi}, \quad i = 1,2,\cdots,n, \tag{19}$$

Hidden layer:

$$O_{pj} = f(Net_{pj}), \quad j = 1,2,\cdots,n_2, \tag{20}$$

$$Net_{pj} = \sum_{i=1}^{n} W_{ji} o_{pi} + \Theta_j, \quad j = 1,2,\cdots,n_2, \tag{21}$$

Output layer:

$$O_p = f(Net_p), \tag{22}$$

$$Net_p = \sum_{j=1}^{n_2} W_j O_{pj} + \Theta, \tag{23}$$

where the weights W_{ji}, W_j and the biase Θ_j, Θ are intervals. For simplicity, We assume that the input vectors are non-negative (i.e., $x_{pi} \geq 0$).

The input-output relation in (19)~(23) can be explicitly calculated by interval arithmetic[1] as follows.

Input layer:

$$o_{pi} = x_{pi}, \quad i = 1,2,\cdots,n, \tag{24}$$

Hidden layer:

$$O_{pj} = [o_{pj}^L,\ o_{pj}^U] = f(Net_{pj}) = [f(net_{pj}^L),\ f(net_{pj}^U)],$$

$$j = 1,2,\cdots,n_2, \tag{25}$$

$$net_{pj}^L = \sum_{i=1}^{n} w_{ji}^L o_{pi} + \theta_j^L, \tag{26}$$

$$net_{pj}^U = \sum_{i=1}^{n} w_{ji}^U o_{pi} + \theta_j^U, \tag{27}$$

Output layer:

$$O_p = [o_p^L,\ o_p^U] = f(Net_p) = [f(net_p^L),\ f(net_p^U)], \tag{28}$$

$$net_p^L = \sum_{\substack{j=1 \\ w_j^L \geq 0}}^{n_2} w_j^L o_{pj}^L + \sum_{\substack{j=1 \\ w_j^L < 0}}^{n_2} w_j^L o_{pj}^U + \theta^L, \tag{29}$$

$$net_p^U = \sum_{\substack{j=1 \\ w_j^U \geq 0}}^{n_2} w_j^U o_{pj}^U + \sum_{\substack{j=1 \\ w_j^U < 0}}^{n_2} w_j^U o_{pj}^L + \theta^U, \tag{30}$$

The learning of the neural network is performed so that the target output y_p may be included in the actual interval output $O_p = [o_p^L, o_p^U]$. The cost function to be minimized in the learning of the interval neural network is defined as follows[4].

$$e_p = v \cdot e_p^L + \mu \cdot e_p^U, \tag{31}$$

where

$$v = \begin{cases} \omega, & \text{if } o_p^L \leq y_p, \\ 1, & \text{if } y_p < o_p^L, \end{cases} \tag{32}$$

$$\mu = \begin{cases} 1, & \text{if } o_p^U < y_p, \\ \omega, & \text{if } y_p \leq o_p^U, \end{cases} \tag{33}$$

$$e_p^L = (y_p - o_p^L)^2/2, \tag{34}$$

$$e_p^U = (y_p - o_p^U)^2/2, \tag{35}$$

where ω is a small positive constant such the $\omega \ll 1$.

We use the following decreasing function instead of the constant ω.

$$\omega(u) = \frac{1}{1 + (u/2000)^3}. \tag{36}$$

In a similar manner as the back-propagation algorithm[9], the learning algorithm for the neural network defined by (19)~(23) or (24)~(30) can be derived from the cost function e_p in (31). The interval weight $W_j = [w_j^L, w_j^U]$ is changed by the following rules:

$$\Delta w_j^L(t+1) = -\eta(\partial e_p / \partial w_j^L) + \alpha \Delta w_j^L(t), \tag{37}$$

$$\Delta w_j^U(t+1) = -\eta(\partial e_p / \partial w_j^U) + \alpha \Delta w_j^U(t), \tag{38}$$

The derivatives in (37), (38) can be calculated from the cost function in (31) (see Ishibuchi et al.[4]). The lower limit w_j^L and the upper limit w_j^U of the interval weight W_j are changed as:

$$w_j^L(t+1) = w_j^L(t) + \Delta w_j^L(t+1), \tag{39}$$

$$w_j^U(t+1) = w_j^U(t) + \Delta w_j^U(t+1). \tag{40}$$

In order to cope with the situation that the lower limit of the interval weight exceeds its upper limit after the adjustment by (37), (38), the interval weight is defined as:

$$W_j(t+1) = [min\{w_j^L(t+1),\ w_j^U(t+1)\},\ max\{w_j^L(t+1),\ w_j^U(t+1)\}], \quad (41)$$

The interval weight W_{ji} and the interval biases Θ, Θ_j are changed in the same manner as the interval weight W_j.

3.2 Application to demand forecasting

Let us apply the interval neural network shown in the previous section to a demand forecasting problem. A demand forecasting is performed in the same manner as described in the section 2.2. An average of interval outputs obtained from the trained neural networks for ten trials is shown in Fig.3. Fig.3 shows the upper estimate F^U and the lower estimate F^L of the demand. From Fig.3, we can see that the actual outputs include all the training data (the solid lines). For the test data which are not used in the learning phase, we can observe the difference between the interval forecasts (the dotted lines) and the actual sales. In the section 4, this method will be improved to obtain better results.

3.3 Forecasting with interval input variables

In this section, we propose a demand forecasting method for the case of interval input variables. The input-output relation of each unit in (19)~(23) is defined by interval arithmetic[1] for the interval input vector $\boldsymbol{X}_p = (X_{p1}, X_{p2}, \cdots, X_{pn})$ as follows[7].

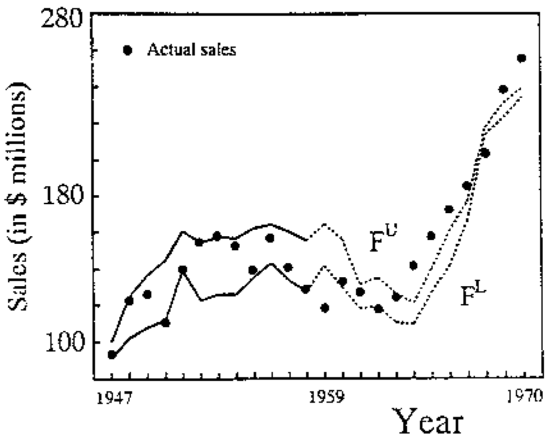

Fig. 3 Demand forecasting by the interval neural network.

Input layer:

$$O_{pi} = X_{pi}, \quad i = 1,2,\cdots,n, \quad (42)$$

Hidden layer:

$$O_{pj} = f(Net_{pj}), \quad j = 1,2,\cdots,n_2, \quad (43)$$

$$Net_{pj} = \sum_{i=1}^{n} W_{ji} O_{pi} + \Theta_j \quad j = 1,2,\cdots,n_2, \quad (44)$$

Output layer:

$$O_p = f(Net_p), \quad (45)$$

$$Net_p = \sum_{j=1}^{n_2} W_j O_{pj} + \Theta. \quad (46)$$

The above input-output relation in (42)~(46) can be explicitly caculated by interval arithmetic[1] (see Kwon et al.[7]).

Demand forecasting results for the interval inputs are shown in Fig. 4.

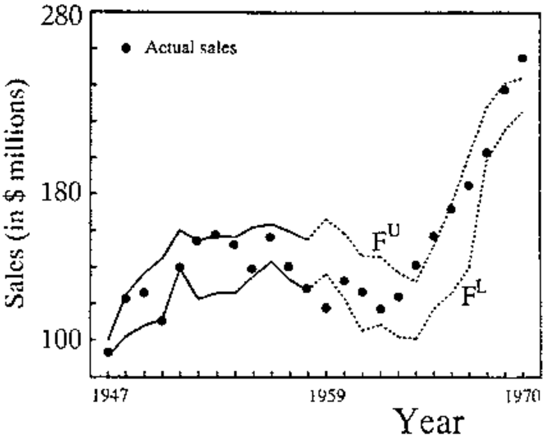

Fig. 4. Demand forecasting by the interval neural network with the interval inputs.

4. Transformation of input variables

In this section, we improve the performance of the demand forecasting by introducing a transformation of input variables which takes account of the degree of influence between an input variable and an output variable.

4.1 The degree of influence of input variables

In order to introduce the transformation of input variables considering the degree of influence, we define the degree of influence of an input variable on an output variable. Let us assume that m pairs of input vectors and the corresponding outputs are given as$(\boldsymbol{x}_p, y_p)$, $p=1,2,\cdots,m$, where $\boldsymbol{x}_p=(x_{p1},x_{p2},\cdots,x_{pn})$ is an n-dimensional input vector and y_p is an output variable. In this case, we can consider a polynomial of degree k as a regression model for $y=f(x_i)$, $i=1,2,\cdots,n$. i.e.,

$$y=\beta_0+\beta_1 x_i+\beta_2 x_i^2+\beta_3 x_i^3+\cdots+\beta_k x_i^k+\varepsilon_i,$$
$$i=1,2,\cdots,n, \tag{47}$$

whereβ_j is an unkown parameter andε_i is the error term.

Using the equation (47), let us assume that the degree of influence of the input variable $x_i,i=1,2,\cdots,n$, is proportional to the well known coefficient of determination, r_i^2, $i=1,2,\cdots,n$, as:

$$r_i^2=1-\frac{SS(E)}{SS(t)}=\frac{SS(T)-SS(E)}{SS(T)}=\frac{SS(R)}{SS(T)},$$
$$i=1,2,\cdots,n, \tag{48}$$

where $SS(E)=\Sigma_{p=1}^m(y_p-\hat{y}_p)^2$, $SS(R)=\Sigma_{p=1}^m(\hat{y}_p-\bar{y})^2$ and $SS(T)=\Sigma_{p=1}^m(y_p-\bar{y})^2$. Here, $\bar{y}=\Sigma_{p=1}^m y_p/m$, $\hat{y}_p$ is the estimate derived from a regression model. We assume that the regression model can be determined from the scatter diagram between an input variable and the corresponding output variable. The range of r_i^2 is $0\le r_i^2\le 1$, and we can consider that the closer value of r_i^2 to 1 means the more influence of the input variable x_i on the output variable y.

Using the coefficient of determination r_i^2 in (48), we can derive the degree of influence $\rho(x_i)$, $i=1,2,\cdots,n$, between each input variable x_i, $i=1,2,\cdots,n$, and the output variable y. The degree of influence is heuristically defined as:

$$\rho(x_i) = 7r_i^2 + 1, \quad i = 1,2,\cdots,n, \tag{49}$$

that is, a real number in the closed interval [1, 8] is given as the degree of influence. As $\rho(x_i)$ goes closer to 8, the degree of influence becomes higher.

4.2 Transformation of input variables by the degree of influence

Taking account of the degree of influence, we transform the input variables $x_{pi}, i = 1,2,\cdots,n$, into real numbers in the interval $[0.1 \times \rho(x_{pi}), 0.9 \times \rho(x_{pi})]$. i.e.,

$$x_{pi} \in [0.1, 0.9] \Rightarrow x_{pi} \in [0.1, \times \rho(x_{pi}), 0.9 \times \rho(x_{pi})], \ i = 1,2,\cdots,n, \tag{50}$$

where $\rho(x_{pi}), i = 1,2,\cdots,n$, is given by equation (49). In other words, the transformed input variable x_{pi} is a real number in the interval [0.1, 7.2].

In the case of the neural network using an input variable with the degree of influence, the initial values of the weights and the biases are given as follows:

- The weights between the input layer and the hidden layer: 1
- The weights between the hidden layer and the output layer: Random real number in the closed interval [-1, 1]
- The biases of the hidden layer: -3
- The biases of the output layer: Random real number in the closed interval [-1, 1]

By setting the initial values of the weights and the biases as the above, the learning of the neural network is performed with priority given to the input variable which has a high degree of influence. It is because the weight change dictated by the delta rule corresponds to performing steepest descent on a surface in weight space whose heights at any point in weight space is equal to the error measure[9]. Thus, the outputs of the neural network are strongly influenced by the input variable that has a high degree of influnce. Let us demonstrate this with a simple numerical example. In this numerical example, we assume that the training data for a two-input and single-output system are given in Table 2. The interrelation between each input variable and the output variable in Table 2 is shown in Fig.5.

Table 2. Training data

Input variables		Output variable
x_1	x_2	y
0.50	0.10	0.1
0.36	0.87	0.2
0.70	0.15	0.3
0.25	0.81	0.4
0.78	0.22	0.5
0.18	0.75	0.6
0.15	0.30	0.7
0.87	0.64	0.8
0.10	0.50	0.9

Using the training data in Table 2, we trained the standard neural network of the section 2.1 with the input variables transformed by the degree of influence. Moreover, we

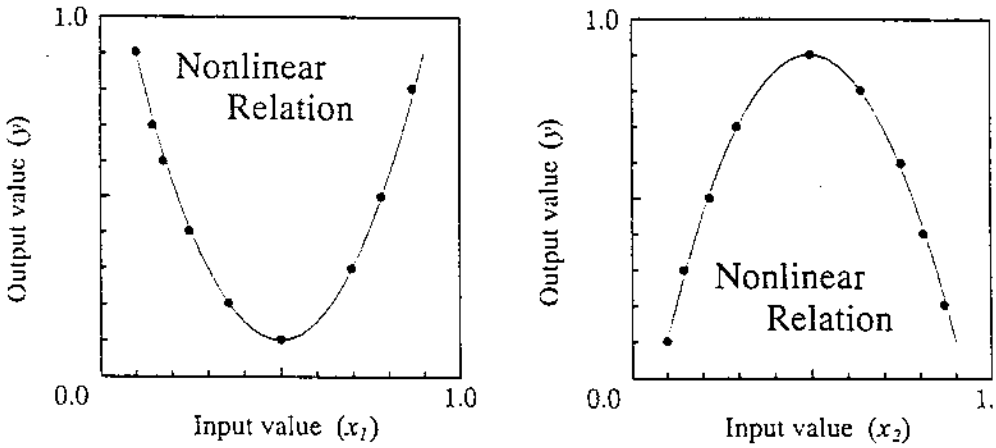

Fig. 5 Relation between an input variable and an output variable

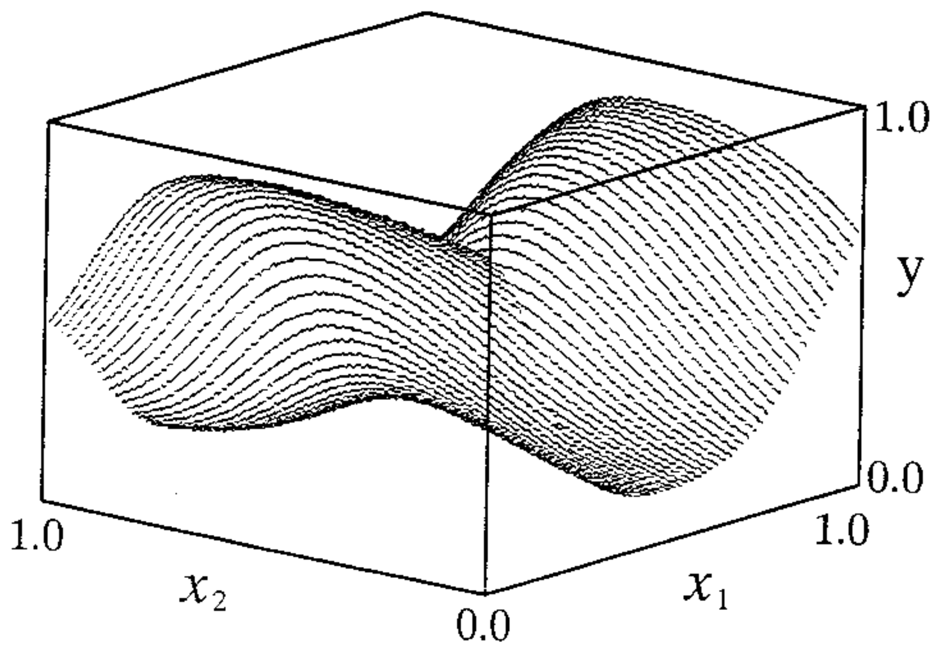

Fig. 6 In The case of the non-transformed inputs.

trained it for the non-transformed input variables for comparison. In the case where the transformed input variables are used, the initial values of the weights and the biases are given as the above-mentioned case. The number of hidden units is 6, the momentum constant is 0.9 and the learning rate is 0.3. On the other hand, in the case where the non-transformed

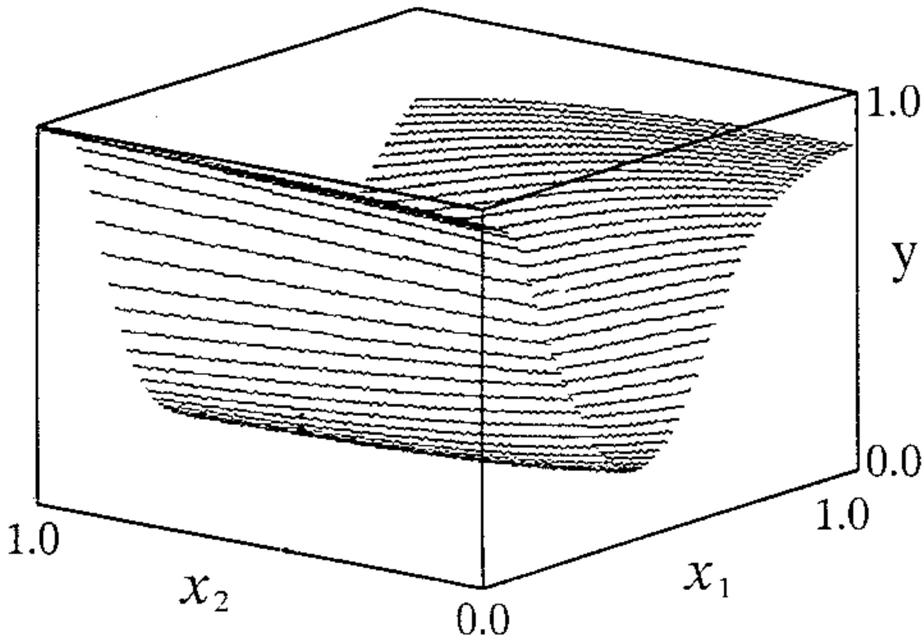

Fig. 7 In The case of the transformed inputs [$\rho(x_1)=8$, $\rho(x_2)=1$].

input variables are used, the initial values of the weights and the biases are given as random real numbers in the closed interval [-1, 1]. The number of the hidden units is 6, the momentum constant is 0.9 and the learning rate is 0.5. For the data given in Table 2, the degrees of influence of input variables (e.g., in the case of the quadratic regression model) are $\rho(x_1)=\rho(x_2)=8$. However, we consider extra two cases [$(x_1)=8$, $\rho(x_2)=1$] and [$\rho(x_1)=1$, $\rho(x_2)=8$]to observe the outputs obtained from the trained neural networks.

Simulation results after 5000 iterations are shown in Fig.6~Fig.8. Fig.6 is the result for the non-transformed input variables. Fig.7 and Fig.8 are the results for the transformed input variables. From the comparison between Fig.7 and Fig.8, we can see that the outputs are different because the inputs were transformed by the degree of influence.

4.3 Forecasting with transformed input variables

By the transformation described in the previous section, the input variables x_{pi}, $i=1,2,3$, in Table 1 are transformed to real numbers in the interval [0.1,7.2]. We assume that the relation between an input variable and an output variable is linear in the case of the input-output pairs (x_{p1}, y_p) and (x_{p3}, y_p) and curvilinear in the case of the input-output pair (x_{p2}, y_p) (quadratic up to $p=1963$, cubic from $p=1964$). For example, the estimated regression equation (quadratic) between an input

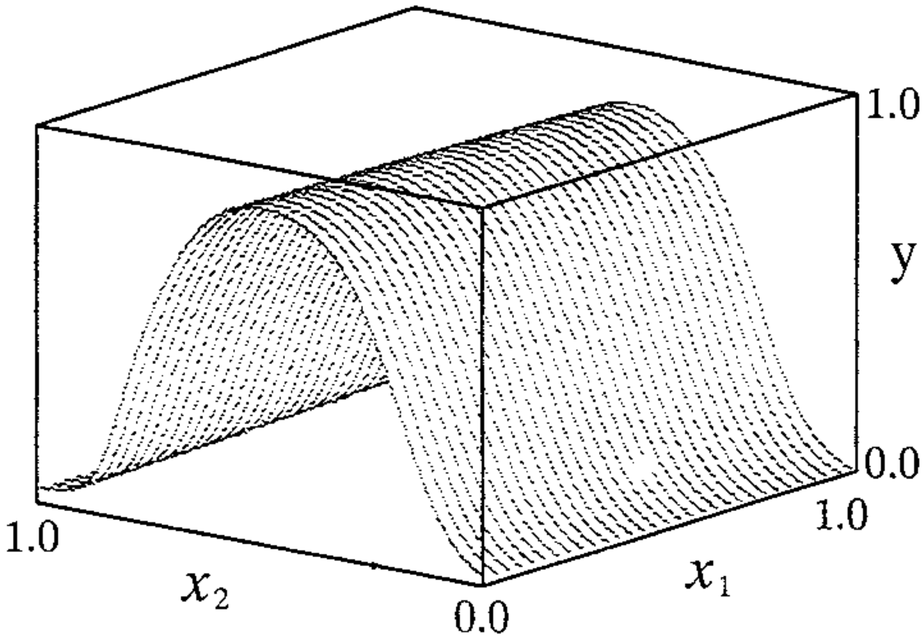

Fig. 8 In The case of the transformed inputs [$\rho(x_1)=1$, $\rho(x_2)=8$].

variable x_{p2} and an output variable y_p (up to $p=1958$) was derived as: $\hat{y_p} = -0.39+5.89x_{p2} -11.52x_{p2}^2$. From this regression equation, $r_2^2=0.827$ and $\rho(x_{p2})=6.789$. The results of demand forecasting using transformed input variables are shown in Fig.9 and Fig.10. Fig. 9 is the result for the standard neural network [9], while Fig.10 is the result for the interval neural network[4].

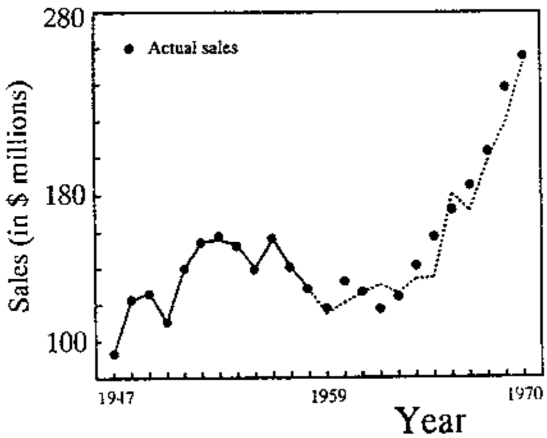

Fig. 9 Demand forecasting by the neural network with the transformed inputs.

The comparison between Fig.9 and Fig.1 is shown in Tables 3 and 4. Here, we trained the neural networks with six hidden units by the learning algorithms with $\eta=0.5$ and $\alpha=0.9$. Table 3 shows the averages of the errors over ten separate trials corresponding to different iterations of the learning algorithms using the data during the years 1947 to 1958. The mean absolute deviation (MAD) of forecasted sales during the years 1959 to 1970 are summarized in Table 4. The MAD is defined as:

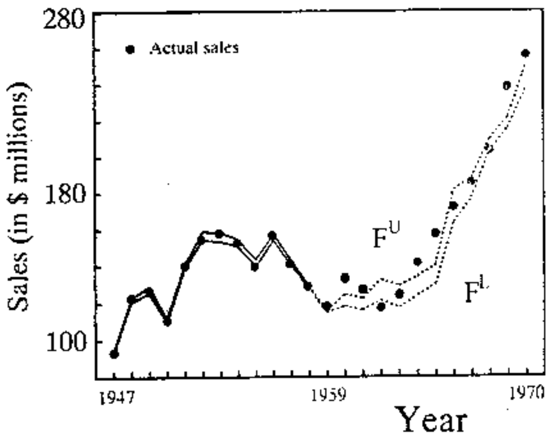

Fig. 10 Demand forecasting by the interval neural network with the transformed inputs.

Table 3: Average of the error function *E*

Iterations	Forecasting method without the transformation	Forecasting method with the transformation
0	0.11847	0.15890
1,000	0.01476	0.00633
5,000	0.01022	0.00113
10,000	0.00747	0.00065

Table 4. Mean absolute deviation (MAD)

	Forecasting method without the transformation	Forecasting method with the transformation
MAD	13.83	9.25

$$MAD = \sum_{p=1}^{n} |S_p - F_p| / n, \qquad (51)$$

where S_p is the actual sales of the p-th year, F_p is a forecasted sales of the p-th year and n is the number of forecasted years.

From Table 3 and Table 4, we can see that the results used the transformed input variables by the degree of influence outperforme those used the non-transformed input variable in the aspect of learning speed and forecasting error. By comparing Fig.3 with Fig.10, we can also see that the outputs of the trained interval neural network corresponding to the transformed input varibles are closer to the actual sales. By introuducing the degree of influence, we can get the better forecasting estimate because the variation of the input variable which has the highest influence to the output variable can be reflected more strongly (in this example, x_{p2}: disposable personal income). The degrees of influence of input variables for the data during the years 1947 to 1958 are as follows: $\rho(x_{p1}) = 4.9$, $\rho(x_{p2}) = 6.8$ and $\rho(x_{p3}) = 4.9$.

5. Conclusion

In this paper, we proposed a demand forecasting method using interval neural networks. First, we proposed a demand forecasting method using the standard back-propagation algorithm. Also, the interval forecasting method was presented by considering the fuzziness of input variables as intervals. Next, this paper proposed a demand forecasting method in which the actual sales should be included in the interval estimate. Last, we demonstrated that the forecasting ability of the neural networks was improved by introducing the degree of influence of input variables.

References

[1] Alefeld G. and Herzberger J., *Introduction to Interval Computations,* Academic Press, New York, 1983.

[2] Chambers J.C., Mulick S.K. and Smith D. D., "How to Choose the Right Forecasting Technique", *Harvard Business Review,* pp. 45-74, July-Aug. 1971.

[3] Ishibuchi H. and Tanaka H., "Fuzzy Regression Analysis Using Neural Networks", *Fuzzy sets and Systems,* Vol. 50, pp. 257-265, 1992.

[4] Ishibuchi H., Okada H. and Tanaka H., "An Architecture of Neural Networks with Interval Weights and Its Application to Fuzzy Regression Analysis", *Fuzzy Sets and Systems.* Vol. 57, pp. 27-39, 1992.

[5] Ishibuchi H., Okada H. and Tanaka H., "Fuzzy Neural Networks with Fuzzy Weights", *Trans. ISCIE Japan,* Vol. 6, No. 3, pp. 137-148, 1993.

[6] Kimura A., Arizono I. and Ohta H., "An Application of Layered Neural Networks to Demand Forecasting", *Journal of Japan Industrial Management Association,* Vol. 44, No. 5, pp. 401-407, 1993.

[7] Kwon K., Ishibuchi H. and Tanaka H., "Nonlinear Mapping of Interval Vectors by Neural Networks", *Proc. of IJCNN' 93-Nagoya,* Vol. 1, pp. 758-761, 1993.

[8] Parker G.G.C. and Segura E.L., "How to Get a Better Forecast", *Harvard Business Review,* pp. 99-109, Mar.-Apr. 1971.

[9] Rumelhart D.E., McClelland J.L. and the PDP Research Group, *Parallel Distributed Processing,* Vol. 1, MIT Press, Cambridge, 1986.

[10] Tanaka H., Hayashi I. and Watada J., "Possibilistic Linear Regression Analysis for Fuzzy Data", *European Journal of Operational Research,* Vol. 40, pp. 389-396, 1989.

[11] Tanaka H., Ishibuchi H. and Hwang S. G., "Fuzzy Model of the Number of Staff in Local Government by Fuzzy Regression Analysis with Similarity Relations", *Journal of Japan Industrial Management Association,* Vol. 41, No.2, pp. 99-104, 1990.

[12] Tanaka H. and Hayashi I., "The Fuzzy GMDH Algorithm by Possibility Models and Its Application", *Fuzzy Sets and Systems,* Vol. 36, pp. 245-258, 1990.