IEEE International Conference on Intelligent System Applications to Power Systems (ISAP'2001), Budapest, Hungary, June 18th-21st, 2001

WATER FLOW FORECASTING USING ANALYZABLE STRUCTURED NEURAL NETWORK

Tatsuya Iizaka **Tetsuro Matsui** **Yoshikazu Fukuyama, member, IEEE**
(iizaka-tatuya, matsui-tetsuro, fukuyama-yoshikazu)@fujielectric.co.jp
Fuji Electric Corporate R & D, Ltd.
1, Fuji-machi, Hino-city, Tokyo, 191-8502, Japan

Abstract - This paper presents water flow forecasting using an analyzable neural network. The proposed analyzable neural network is a kind of three layers feed foreword neural networks. The proposed neural network consists of two types of hidden units. One type of hidden units has weights connect to only one group of input units. Another one has weights connected to all input units. First type of hidden units allows to analyze each relation between a certain input data and a corresponding output data. Namely, dam operators can notice output results and extract knowledge from the neural network. Another type of hidden units ensure the performance of the neural network as same as the conventional neural networks. The effectiveness of the proposed method is shown using actual water flow data.

Keywords: water flow forecasting, neural network, knowledge extraction, structural learning

I. INTRODUCTION

Forecasting water flow volumes flowing into a dam is a very important task for economic and safe operation of the hydraulic plant. The energy stored in water reservoirs should be converted into electric energy as efficiently as possible. The gates of the dam must be operated safely in order to protect downstream areas from flood. Usually, the operators of the dam forecast the water flow. However, in order to improve forecasting performance and realize automatically forecasting to help operators, an accurate computer-based forecasting system is eagerly required.

The water flow in a river is a complex natural phenomenon. Therefore, the system performance is influenced by the features of the river basin. There are many conventional mathematical methods for forecasting water flow. However, it is difficult to make a reliable forecast using these methods and it is difficult for the conventional methods to tune their parameters.

Recently, many studies of neural networks for the water flow have been conducted [1-3]. The neural network is regarded as a powerful method for handling nonlinear complex phenomenon. It is able to develop a neural network automatically only by training past data about these target problems [4]. However, neural networks are said to be a black box. Namely, it is difficult to explain the output results. Many researchers investigated the ways to explain the output results [5-7]. However, these methods are effective only for the discrete value outputs, and cannot be applied to the continuous value outputs problems such as the water flow forecasting.

This paper proposes the water flow forecasting using an analyzable structured neural network (ASNN). ASNN allows the dam operators to notice reasons of forecasting results and extract knowledge about the forecasting from ASNN. The rough relations between input and output data can be presented by the skeletal structure of ASNN. The detail relations can be presented by extracted relations from ASNN.

The concept of the proposed method is demonstrated by approximation of a simple quadric function. The effectiveness of the proposed method is also shown by the water flow forecasting with actual data. Forecasting performance using the proposed method is verified by a comparison with forecasting results using a conventional neural network trained by a back propagation algorithm (BP).

II. ANALYZABLE STRUCTURED NEURAL NETWORK

Fig. 1 shows the proposed ASNN, which has some network modules. The ASNN consists of two types of hidden units. One type of hidden units has weights connected to only one group of input units. The network module with this type of hidden units is called a *sparse connecting module*. Another one has weights connected to all input units. The network module with this type of hidden units is called an *all-connecting module*. The former type of hidden units allows to analyze each relation between a certain input data and a corresponding output data. The latter type of hidden units ensure the performance of the neural network as same as the conventional neural networks.

The concept of a proposed analysis method is explained using function approximation as follows:

$$y = x_1 + x_2 + x_1 x_2 \qquad x1, x2 = \{0.0 \sim 1.0\} \tag{1}$$

Fig. 2 shows the proposed neural network structure after training. It also shows the rough relations between input and output units. First and second hidden units indicate the independent relations of x1 and x2. Third hidden unit indicates the interaction between x1 and x2.

Fig. 3 shows relations of trained neural network for the simple quadric function. The relations by the first and the second hidden units are almost liner and have same values. The relation by the third hidden unit shows a quadric relation. These characteristics are the same as that of the function shown in (1).

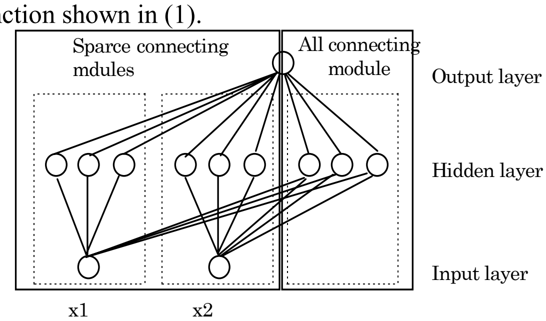


Fig. 1 Structure of the proposed neural network

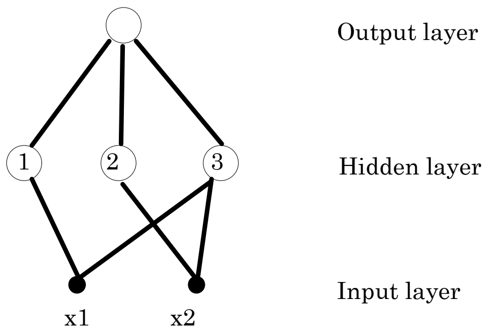


Fig. 2 A neural network structure after training a quadric function.

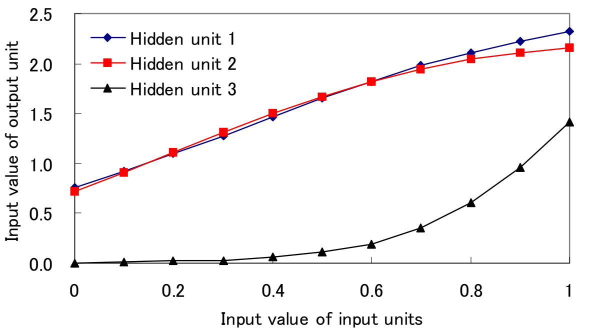


Fig. 3 An example of relations of the proposed neural network.

III. TRAINING METHOD FOR ASNN

A. Formulation of Training Algorithm

If a trained neural network includes useless hidden units and useless connections, analysis and extraction of knowledge are difficult, and the neural network cannot have high generalization ability. The proposed neural network is trained by a superposed energy function [8] and a structural learning algorithm with forgetting (SLF) [5, 6] for easy analysis and high generalization ability.

The superposed energy function lets important units grow up, and the variance of their outputs becomes big. SLF lets useless weights degenerate, and the variance of their outputs becomes small. Therefore, useless units can be distinguished easily by the variance of output of hidden units, and it also can be combined easily to bias units as follows [9,10]:

$$\omega_{bj} = \omega_{bj} + \overline{O}_i \omega_{ij} \tag{2}$$

where,

- w_{bj} : weight of bias,
- w_{ij} : weight of the useless hidden unit,
- O_i : average output of the useless units.

In order to use the superposed energy function, some sub-perceptrons have to be defined. Connections of small number of sub-perceptrons are grown up early, and connections of large number of sub-perceptrons are grown up slowly. The SLF removes useless connections using the model complexity penalty term in the energy function. The proposed energy function, which utilizes the superposed energy function and SLF concurrently is as shown by:

$$F = \sum_{i}^{H} \beta_i E_i + \varepsilon' \sum |\omega_{ij}| \tag{3}$$

$$E_i = \frac{1}{2}(y_t - y_i)^2 \tag{4}$$

$$\beta_i = \frac{6i^2}{H(H+1)(2H+1)} \tag{5}$$

where,

- F :energy function,
- H :a number of sub-perceptrons,
- β :weight factor of sub-perceptron,
- ε' :forgetting factor,
- E_i :energy function of sub-perceptron i,
- y_t :training data,
- y_i :output of sub-perceptron i.

The weights change, Δw_{ij} is as shown by

$$\Delta w_{ij} = -\eta \frac{\partial F}{\partial w_{ij}} = \Delta w'_{ij} - \varepsilon \operatorname{sgn}(w_{ij}) \tag{6}$$

$$\operatorname{sgn}(x) = \begin{cases} -1 & (x < 0) \\ 0 & (x = 0) \\ 1 & (x > 0) \end{cases}$$

where,

- ε :amount of forgetting factor,
- $\Delta w_{ij}'$:weight change by the superposed energy function.

It is important to adjust the amount of forgetting factor ε. If ε is too big, even useful connections are removed. If ε is too small, useless connections cannot be removed. Adjustment of ε is very important and difficult on continuous value problems. The big value of ε (ex. 10^{-4}) for a discrete value problem [5] may cause removing the useful connections. Moreover, suitable ε is changing while training. This paper also proposes an adjusting method of the amount of forgetting factor ε. One random input unit is added to the neural network, and the amount of forgetting factor ε is calculated by the following equation:

$$\varepsilon = \text{average (weights)/left training times} \tag{7}$$

where,

weights :weights connected to the random input.

B. Training Method for ASNN

The proposed ASNN is trained by the following three steps. Fig. 4 shows each step trained by the simple quadric function shown in (1).

Fig. 4 (a) shows the initial structure before training. Arbitrary connections are set to zero, and the other connections are set to random values. The analyzing purpose of this simple example is to extract two relations of x1 and x2. Therefore, there are two sparse connecting modules.

Step 1 is shown by Fig. 4 (b). Each type of modules is allocated alternately. A purpose of this step is decision of rough structure for training data. Most useless hidden units and connections are pruned through step 1.

Step 2 is shown by Fig. 4 (c). A purpose of this training step is increase of connecting weight values of sparse-connecting modules. The sub-perceptron 1 consists of the all sparse-connecting modules, and sub-perceptron 2 consists of the all-connecting module. Training of the sub-perceptron 1 is accelerated compared with that of the sub-perceptron 2. Namely, the independent relations between an input and output unit are constructed especially for sparse connecting module.

Finally, the trained neural network is obtained by step 3 in Fig. 4 (d) and Fig. 4 (e). This step is carried out for optimization of the neural network. Hidden units are re-allocated according to the following goodness factor, which presents an importance of hidden units [10]. Namely, a hidden unit with the largest goodness factor value is allocated to sub-perceptron1.

$$G_i = \sum^{P} (w_i O_i)^2 \tag{8}$$

where,

G_i :goodness factor of hidden unit i,
O_i :output of hidden unit i,
P :total number of training patterns.

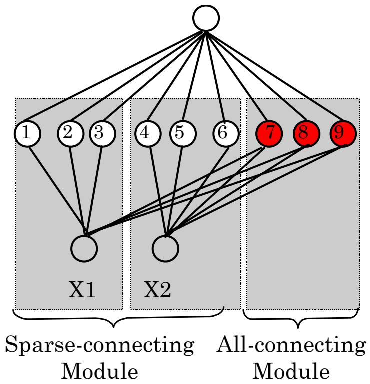
IV. NUMERICAL EXAMPLES

A. Approximations of a Simple Quadratic Function

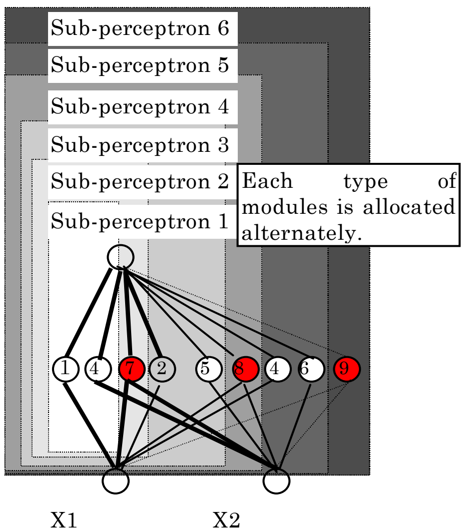
The effectiveness of the proposed ASNN is demonstrated by the function shown in (1).

The proposed ASNN and a conventional neural network trained by BP are tested one hundred times using different initial weights. Sixty patterns of training and evaluation data are selected randomly at each trial. The proposed ASNNs consist of nine hidden units initially. Neural networks trained by BP consist of one to nine hidden units.

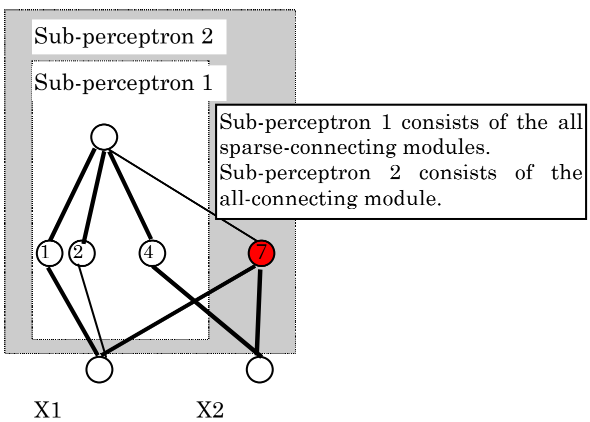
Table 1 shows training conditions. Fig. 5 (a) shows the hidden units variances of ASNN while training through step 1. Fig. 5 (b) shows the hidden units variance of the conventional neural network trained by BP. In Fig. 5 (a),



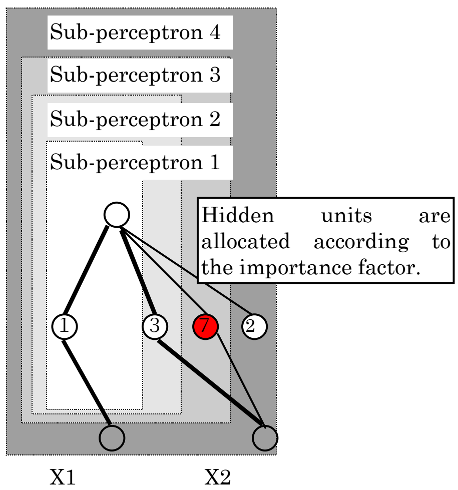
(a) Initial structure



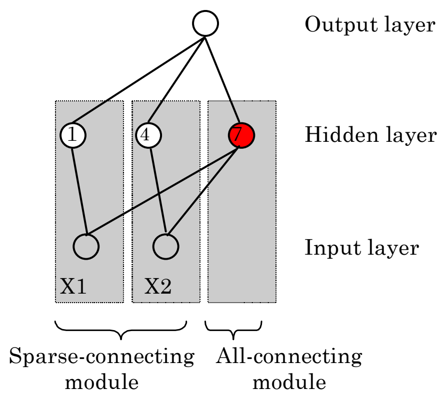
(b) Step 1 (decision of rough structure)



(c) Step 2 (increase of weight values of sparse-connecting module)



(d) Step 3(optimization)



(e) Final structure after training

Fig. 4 Each training step by ASNN

hidden units 1, 4, 7 that allocated in sub-perceptron 1 or 2 are grow up bigger than the other hidden units. In Fig. 5 (b), all hidden units have almost same variance. Fig. 5 indicates that the proposed method can distinguish and prune useless hidden units easily, because deference of variances is very big. Fig. 6 shows amount of forgetting factor ε. Fig. 6 indicates that the proposed adjusting method for ε is suitable because ε does not prevent growing of useful hidden units and can degenerate useless hidden units.

Fig. 7 shows the comparison of mean absolute errors between ASNNs and conventional neural networks trained by BP. Mean absolute error by ASNN is better than that by BP. The results indicate high generalization ability of the ASNN. After training, ASNNs consist of three or four hidden units. The best neural network trained by BP consists of three hidden units as shown in Fig. 7. The results show that the proposed training method can prune useless hidden units. Fig. 2 and Fig. 3 also show that the ASNN can extract knowledge of the quadric function.

B. Water Flow Forecasting into Dam

The water flow forecasting is to forecast the amount of river flow into the dam until one hour later [1, 3]. Fig. 8 shows a model of the target river basin. Three rivers join by one place.

Table 2 shows input data for the water flow forecasting into the dam. A number of input data is thirty-five. In order to understand analysis results easily, these inputs are divided into five groups. An initial ASNN consists of twelve hidden units. Namely, each five sparse modules and all-connecting module have two hidden units before training.

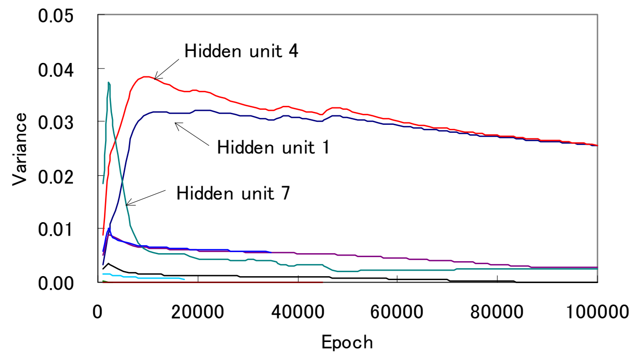
Fig. 10 shows correlation coefficients between output unit and each input unit. Correlation coefficients of upper river flows and water flow into the target dam are very small. That means these two groups are not important for forecasting. Correlation coefficients of deference of upper river flows and deference of water flow into the target dam are big. That means these two groups are important for forecasting. However, these data have same characteristics. Only observed points are different between these data. Therefore, only one group data is required for forecasting. Many correlation coefficients of rainfall are small. This means most rainfall units are not important for forecasting.

Fig. 9 shows a structure example of ASNN after training. It consists of only two sparse connecting modules of deference of upper river flows and rainfall. The other sparse connecting modules and many weights connected to rainfall are pruned. This means that the deference of upper river flow and the rainfall are very important groups, and the other groups and many pruned rainfall units are not important for forecasting. Fig. 10 shows the same characteristic of the structure of the trained ASNN.

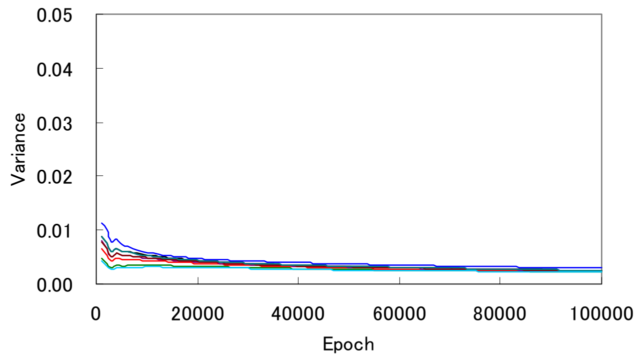
In Fig. 9, a number of large weights between rainfall and sparse connecting module are only two, which connect to rainfall (-1h) and rainfall (-3h). These sign of weight value are opposite. This means deference of these rainfall data is important. Fig. 11(b) shows actual correlation of deference of these rainfall data. Correlation coefficient of deference of these rainfall data is bigger (R^2=0.382) than

Table 1Training conditions.

Epoch	100000 in each training step
Combining of hidden units to bias unit	Distribution is less than 0.0001.
Calculation timing of amount of forgetting	Each 1000 epoch until 90000 epochs



(a) Variances of hidden units by ASNN through step 1.



(b) Variances of hidden units by BP.

Fig. 5 A sample of hidden units variance.

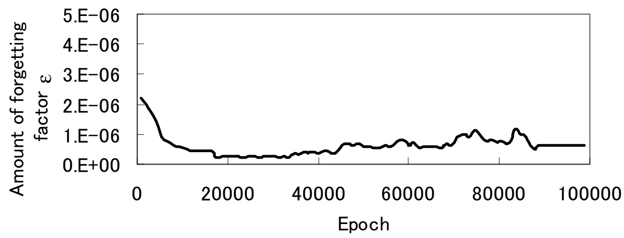


Fig. 6 A sample of amount of forgetting factor, ε through step 1.

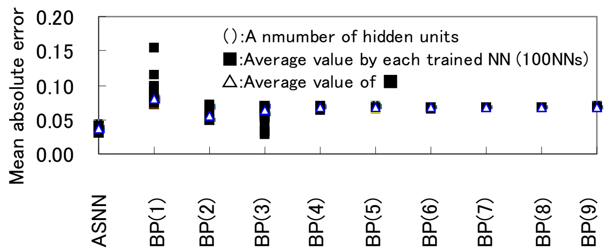


Fig. 7 Comparison of mean absolute errors between ASNN and conventional neural networks trained by BP.

that of original rainfall data (R^2=0.203). Fig. 12 shows extracted relations from trained ASNN. The extracted relations indicate the same characteristics of actual correlation as shown in Fig. 11. This indicates ASNN can analyze relation between input and output units and extract knowledge automatically.

Twenty-three actual flood cases are utilized for verification of the forecasting performance. From eleventh to twenty- third cases were utilized for evaluations. From first to previous target cases were utilized for training. Table 3 shows the evaluations of forecasting results. Fig. 10 and fig. 11 show samples of forecasting results. Forecasting performance is very good whenever the amount of water flow is heavy or light. ASNN can forecast better than a conventional neural network trained by BP.

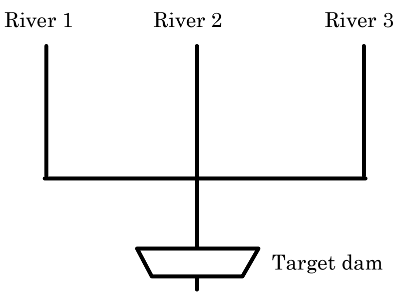


Fig. 8 Target river basin.

Table 2 Input data of water flow forecasting.

Input groups	Details
Upper river flows	River1 (0h), River2 (0h), River3 (0h)
Deference of upper river flows	River1 (-1h, -2h, -3h) River2 (-1h, -2h, -3h) River3 (-1h, -2h, -3h)
Water flow into the target dam	Water flow into the target dam (0h)
Deference of water flows into the target dam	Deference of water flow (-1h, -2h)
Rainfall	Rainfall (-1h,-2h,・・・-20h)

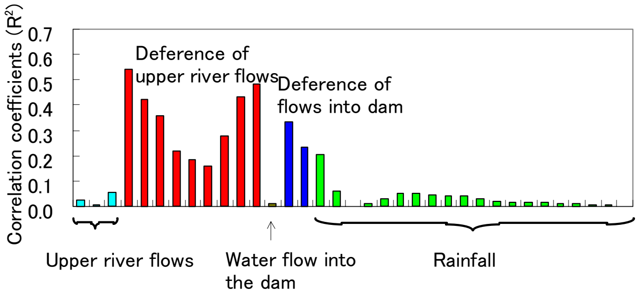
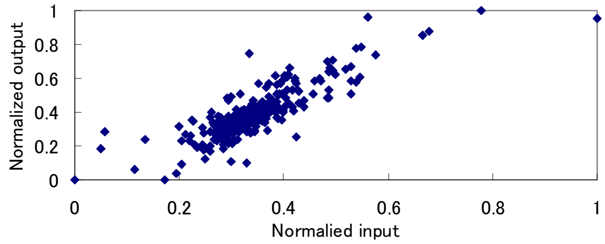
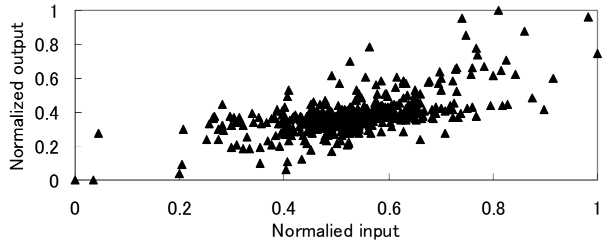


Fig. 10 Correlation coefficients between output unit and each input units.



(a) Total deference of upper river flows



(b) Deference between rainfall (-1h) and rainfall (-3h)

Fig. 11 Actual correlation.

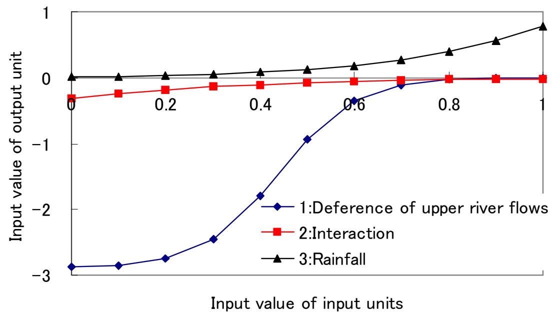


Fig. 12 Extracted relations from ASNN.

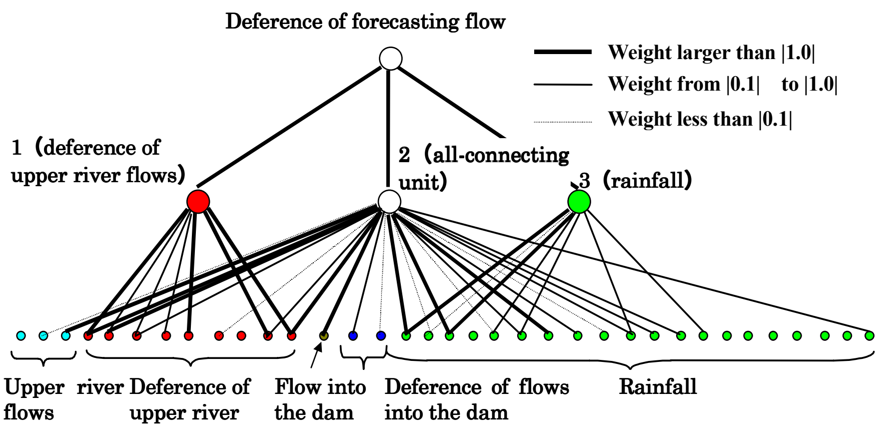


Fig. 9 A structure example of a trained neural network for water flow forecasting.

Table 3 Average evaluations of forecasting results in thirteen cases

	ASNN	BP
Mean absolute error	5.06 %	6.23 %
Standard deviation error	7.79 %	7.95 %
Correlation	0.989	0.988

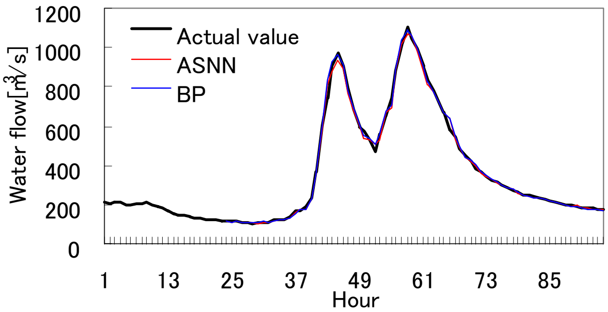


Fig. 13 An example of forecasting results when amount of water flow is heavy.

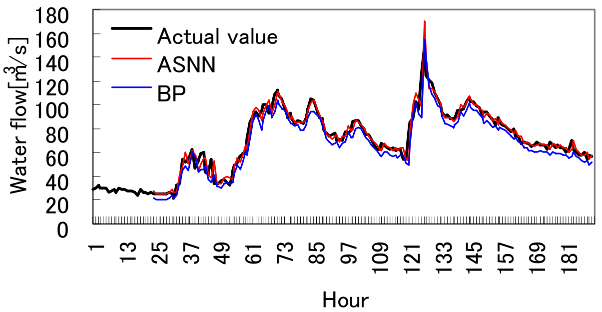


Fig. 14 An example of forecasting results when amount of water flow is light.

IV. CONCLUSIONS

This paper proposes a water flow forecasting using an analyzable structured neural network. The analyzable structured neural network can extract independent relations between input and output units. Therefore, operators of dam can recognize reasons of forecasting results. Moreover, forecasting performance is very excellent whenever amount of water flow is heavy or light, and better than conventional neural networks trained by a back propagation algorithm. The effectiveness of proposed method is verified using twenty-three actual flood data in summer time with heavy rain.

V. REFERENCES

[1] K.Ichiyanagi, Y.Goto, et al. "A prediction of river flow into a dam for hydro-power plant by artificial neural network taking account of accumulated rain fall", *Proc. of IPEC*, Vol.2, pp.702-706, 1995.

[2] A.F.Atiya, S.M.El-Shoura, S.I.Shaheen, M.S.El-Sherif, "A Comparison Between Neural Network Forecasting Techniques -Case Study: River Flow Forecasting-", *IEEE, Trans. on Neural Networks*, Vol.10, No.2, March, (1999).

[3] T.iizaka, T. Matsui, Y. Ueki, "Forecasting of Water flow into a Dam using Neural networks and Fuzzy Theory" , *Proc. of ICEE*, B3-03, Hong Kong, 1999.

[4] D. E. Rumelhart, J. L. McClelland and the PDP Research Group, *Parallel distributed processing* (Vol.1-2), The MIT Press ,1986.

[5] M. Ishikawa, "Rule Extraction by Successive Regularization", *Proc. of IEEE ICNN*, pp.1139-1143, 1996.

[6] D.A.Miller, J.M.Zurada, "A Dynamical System Perspective of Structural Learning with Forgetting", *IEEE trans. on Neural Networks*, Vol.9, No3, pp.508-515, MAY, 1998.

[7] LiMin Fu, "Rule Generation from Neural Networks", *IEEE Trans. on SMC*, Vol.24-8, pp.1114-1124, 1994.

[8] T.Takahashi, R.Tokunaga, "Removing the redundancy of perceptorons in terms of a simple energy function", *Proc. of ICONIP'97*, Vol.1, pp.271-274, 1997.

[9] J. Sietsma, R. J. F. Dow, "Creating artificial neural networks that generalize", *Neural Networks*, Vol.4, pp.67-79, 1991.

[10] Y.Matsunaga, Y.Nakade, O.Yamakawa, K.Murase, "A back-propagation algorithm with reduction of association units in multi-layered neural network", *Trans. of the institute of electronics, information and communication engineers*, D-2, Vol. J74-D-2, No.8, pp.1118-1121, 1991, (in Japanese).

VI. BIOGRAPHIES

TATSUYA IIZAKA received B.S. and M.S. degrees in electrical engineering in 1992 and 1994, respectively, from Saitama University, Saitama, Japan. He has been working at Fuji Electric Co. R&D Japan from 1994. His research interests include application of intelligent systems such as neural network, and fuzzy inference techniques to power systems. He is a member of IEE of Japan.

TETSURO MATSUI received B.S. degree in information engineering in 1988, from Yokohama National University, Kanagawa, Japan. He has been working at Fuji Electric Co. R&D Japan from 1988. His research interests include application of intelligent systems such as expert system, neural network, and fuzzy inference techniques to power systems. He is a member of IEE of Japan.

YOSHIKAZU FUKUYAMA (M'90) received B.S., M.S., and PhD degrees in electrical engineering in 1985, 1987, and 1997, respectively, from Waseda University, Tokyo, Japan. He has been working at Fuji Electric Co. R&D Japan from 1987. He was a visiting scientist at Cornell University from 1993 to 1994. His research interests include application of intelligent systems such as expert system, neural network, and modern heuristic techniques to power systems and power system analysis including voltage stability and load flow. He is also interested in applications of modern heuristic techniques to practical and general optimization problems. He is a member of IEEE and IEE of Japan.