Vol. 42 No. 5 IPSJ Journal May 2001

Regular Paper

Learning from Imperfect Supervisor Using Neural Network Ensemble

Pitoyo Hartono† and Shuji Hashimoto†

In training a conventional multilayer perceptron (MLP), the existence of perfect learning data or a perfect supervisor usually has to be assumed. But in on-line training processes in which a human expert has no direct access to the training data, it is probable that erroneous data due to imperfection of the learning object (supervisor) or observation noise will be included in the training data. This kind of learning data will cause a non-optimal learning process that produces an unreliable neural network. In this paper we propose a model of a neural network ensemble consisting of a number of MLPs in which each MLP can be trained to obtain a unique type of expertise. On the basis of each MLP's expertise, the ensemble will be able to distinguish correct learning data from erroneous data and automatically assign one of its members to be trained only on correct data, allowing an optimal learning process with an imperfect learning supervisor. The proposed ensemble model, eliminates the requirement to design a perfect learning supervisor to train neural networks, and can thus help to make neural networks more widely applicable in real-world problems.

1. Introduction

One of the important factors in the successful training of a neural network is the selection of a good training set. The creation of a good training set is less troublesome if a human expert is available. In the case of on-line training in which the neural network has to obtain the data by observing the behavior of a learning object (supervisor) without the help of a human expert, it is always possible that contradictory or incorrect learning data may be generated as a result of observation error or the imperfection of the supervisor. Such cases may occur when a neural network has to undergo on-line training to approximate the transfer function of a plant in which reliability is not 100%. Even if a human expert exists, he or she may make errors when dealing with a complicated problem. Erroneous data in the training sets may prevent the neural network from learning optimally, producing an unreliable neural network.

Some work shows that injecting small amount of noise into the teacher signal may enable the neural network to achieve a better performance [1]. A number of papers also show that introducing noise into the input may help the neural network to achieve better generalization ability [2)~6)]. But to our knowledge, there has been no research on situations involving imperfect supervisor that sometimes produces a completely wrong answer to a given problem, even though such situations are likely to occur in real-world problems.

There has also been some research on how to extract a conditional distribution of learning signals produced by a supervisor that behaves in a probabilistic way [7], but this has not resulted in a method of training a neural network with such a supervisor without excessively compromising the performance.

We previously have proposed a model of a neural network ensemble composed of a number of multilayer perceptrons (MLP) [8]. In the proposed ensemble model, each member is permitted to learn its own type expertise through competition, and the model has a unique characteristic not observed in previous ensemble models [9)~14)]: we introduced a temperature control mechanism that automatically assigns the member with the most relevant expertise regarding the given problem. Our model has a similar structure to some existing ensemble models [15)~17)], but it differs from them in two respects. The first is that the proposed model does not require a gating mechanism to select the most relevant member with respect to the problem to be solved, and the second is that it is able to train each of its members to deal with a particular problem, whereas the previous models allocate individual members to deal only with subproblems.

The multi-neural network model proposed in Müller, et al. [18] is also similar to ours in some respects. The objective of this model is to realize unsupervised switching to deal with non-

† Department of Pure and Applied Physics, Graduate School of Science and Engineering, Waseda University

stationary signals originating from different dynamical systems which alternate in time. Our ensemble model is differentiated from the latter by its switching mechanism.

As one promising application of our model, in this paper we investigate the behavior of the ensemble when it is trained with an imperfect supervisor. We consider that the imperfect supervisor generates two different environments (we define an environment as a set of input patterns with desired outputs) with a particular probability: one is a clean environment containing correct learning data, and the other contains only false data. It can be expected that the ensemble will be able to assign a particular member to learn in the correct environment while allocating the other members to absorb the erroneous one, implying that one of the members is trained only with correct learning data, producing a neural network that is effectively trained even if the supervisor is not completely reliable.

It should be noted that in this research we are trying to train a neural network not with an environment that requires a probabilistic answer for a given problem, but with a deterministic environment. Only the imperfection of the supervisor causes a probabilistic relation between the input and the desired output. Consequently, the objective of the training is not to reflect the probabilistic behavior of the training supervisor, but to suppress the neural network's performance degradation due to the imperfection of the training supervisor.

In Section 2 the structure and dynamics of the proposed neural network ensemble will be explained. In Section 3 the details of the temperature control will be given. Some experimental results will be given in Section 4, and our conclusions will be presented in the final section.

2. Neural Network Ensemble

2.1 Structure of the Ensemble

An outline of the proposed ensemble is shown in **Fig. 1**. The ensemble is composed of a number of independent MLPs. There are no restrictions on the number of layers in an MLP, but in this research we use a three-layered MLP. The ensemble is provided with an input layer to receive input from the learning object (supervisor). The input is then propagated to each of the MLP's input layers, to be processed by each MLP independently. The temperature control mechanism controls the learning intensity of each MLP on the basis of the members' outputs and supervisor's desired output.

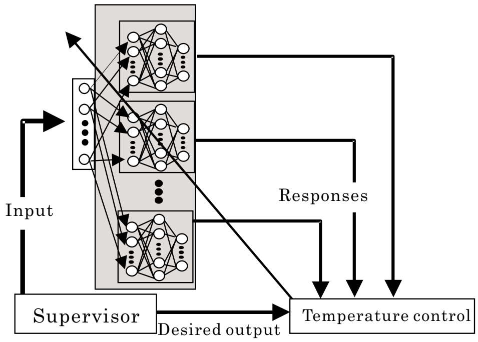

Fig. 1 Neural network ensemble.

The connection weights of the neurons in each MLP are initialized randomly. Because the objective of the learning process is to train each MLP to obtain a unique type of expertise, it is preferable that all MLPs should also have different structures. The structures are diversified by setting different number of neurons in the middle layer of each MLP.

2.2 Ensemble's Dynamic

In the proposed ensemble, the activation function of neurons in the middle layer is defined as follows:

$$O_m^{i,mid} = \frac{1}{1+\exp(-u_m^{i,mid})}, \tag{1}$$

$$u_m^{i,mid} = \sum_{n=1}^{N_{in}} w_{nm}^{i,in} O_n^{i,in},$$

where $O_m^{i,mid}$ and $u_m^{i,mid}$ are the output and the potential of the m-th neuron in the middle layer of the i-th MLP, respectively. $O_n^{i,in}$ is the output of the n-th neuron in the input layer and $w_{nm}^{i,in}$ is the connection weight between the n-th neuron in the input layer and the m-th neuron in the middle layer of the i-th MLP. N_{in} defines the number of input neurons which are common to all the ensemble's members.

The activation function of neurons in the output layer of the i-th MLP is defined as follows:

$$O_k^{i,out} = \frac{1}{1+\exp(-\frac{u_k^{i,out}}{T_i})}, \tag{2}$$

$$u_k^{i,out} = \sum_{m=1}^{N_{mid}^i} w_{mk}^{i,mid} O_m^{i,mid},$$

where $O_k^{i,out}$ is the output of the k-th neuron in the output layer of i-th MLP and $u_k^{i,out}$ is its potential. $w_{mk}^{i,mid}$ is a connection weight between the m-th neuron in the middle layer and the k-th neuron in the output layer of the i-th MLP, while N_{mid}^{i} is the number of middle neurons in the i-th MLP. T_i indicates the temperature of the i-th MLP that is shared by all the output neurons in the MLP.

From Eq. (2) it is clear that if the temperature T_i is high enough, the neurons will always produce an output value in the vicinity of 0.5, regardless of their potentials. In problems requiring binary responses, such an output can be regarded as insignificant. A neuron that gives an insignificant output can be thought of as an inactive neuron, and consequently an ensemble's member containing inactive neurons is regarded as an inactive member. On the other hand, a member with a low temperature can be considered to be active, because its output is sensitive to its potential.

In the training process for the proposed ensemble, the members with relatively low abilities to learn the expertise offered by the learning supervisor can be inactivated by increasing their temperatures.

Each MLP is trained according to the following backpropagation training rule [19]:

$$\begin{aligned} W^i(t+1) &= W^i(t) + \eta\Delta W^i(t) \\ &\quad + \mu\Delta W^i(t-1), \qquad (3) \\ \Delta W^i(t) &= -\frac{\partial E^i(t)}{\partial W^i(t)}, \\ E^i(t) &= \frac{1}{2}\sum_{k=1}^{N_{out}}(d_k(t) - O_k^{i,out}(t))^2, \end{aligned}$$

where W^i is the weight vector of the i-th MLP. $E^i(t)$ is the error of the i-th MLP at time t, regarding the desired output $D(d_1, d_2, \cdots, d_{N_out})$ and the MLP output $O^{i,out}$ at time t, and N_{out} is the number of neurons in the output layer. η and μ indicate the learning rate and momentum rate, respectively.

From Eq. (3), the correction of weights between neurons in the middle and output layers of the i-th MLP can be written as

$$\begin{aligned} \Delta w_{mk}^{i,mid} &= -\frac{\partial E^i}{\partial w_{mk}^{i,mid}} \qquad (4) \\ &= \frac{1}{T_i} O_m^{i,mid}\delta_k^{out}, \\ \delta_k^{out} &= (d_k - O_k^{i,out})O_k^{i,out}(1 - O_k^{i,out}). \end{aligned}$$

From Eq. (4), it is clear that if the temperature T_i is high enough then the weight correction can be ignored. This implies that we can prevent the erroneous training data from destroying the expertise of an MLP that is trained with the correct data by increasing the temperature of the MLP. The decision to increase or decrease a particular MLP's temperature is made in accordance with the temperature control, which will be explained in the next section.

High temperature will also prevent a particular item of learning data from changing the connection weights between neurons in the input layer and middle layer, because the correction value can be written as

$$\begin{aligned} \Delta w_{nm}^{i,in} &= -\frac{\partial E^i}{\partial w_{nm}^{i,in}} \qquad (5) \\ &= \frac{1}{T_i} O_n^{i,in} O_m^{i,mid}(1 - O_m^{i,mid})\delta_m^{mid}, \\ \delta_m^{mid} &= \sum_{k=1}^{N_{out}} w_{mk}^{i,mid}\delta_k^{out}. \end{aligned}$$

Equations (4) and (5) show that, by increasing their temperatures, the ensemble can prevent members with poor learning performances with respect to particular training data from forwarding the learning process, while continuing to train members with good learning performances by keeping their temperature low.

3. Temperature Control

The temperature control is introduced to automatically activate an MLP that has a relatively high ability to learn the expertise offered by the correct learning data, and to inactivate the other MLPs with lower abilities.

The temperature control is implemented by decreasing the temperature of the MLP whose output has the smallest error with respect to the desired output, and increasing the temperature of the MLP with relatively large errors.

It has to be noticed that the MLP that performs well with regard to the correct learning data will perform badly when erroneous data are given, so the erroneous data will have no effect on the MLP in the training process.

Because it is preferable that there should be only one active MLP in the ensemble at a given time, we create competition for domination among members of the ensemble, so a particular MLP that performs well not only rewards itself by decreasing its own temperature but also punishes the others by increasing their temperatures. On the other hand, an MLP that performs badly has to punish itself by increasing its own temperature and giving the other MLPs chances to learn by decreasing their temperatures.

The temperature control is written as follows:

$$T_i(t+1) = T_i(t) + \Delta T_i(t) - C, \quad (6)$$
$$\Delta T_i(t) = -p^{self}(1 - N\tau^i(t)) + p^{cross}\sum_{\substack{j=1\\j\neq i}}^{N}(1 - N\tau^j(t)),$$

$$\tau^i(t) = \frac{\sum_{k=1}^{N_{out}}(d_k - O_k^{i,out})^2}{\sum_{j=1}^{N}\sum_{k=1}^{N_{out}}(d_k - O_k^{j,out})^2}, \quad (7)$$

where N is the number of MLPs in the ensemble and p^{self}, p^{cross}, and C are the self-penalty, cross-penalty, and cool-down constant, respectively. $\tau^i(t)$ is the error reference of the i-th MLP, which measures how well it performs in comparison with other MLPs at time t.

The first term of the temperature correction, ΔT_i, is the self-penalty term which will decrease the temperature of a particular MLP if the MLP performs relatively well, and increase it if it performs badly. The second term is the cross-penalty term, which will increase other MLPs' temperatures if an MLP performs well and decrease them if it performs badly. The cool-down term C is intended to speed up the temperature competition.

The temperature is limited between 1 and T_{max} by the following limiting function:

$$T_i(t+1) = T_{\max} \quad if \quad T_i(t) + \Delta T_i > T_{\max}$$
$$T_i(t+1) = 1 \quad if \quad T_i(t) + \Delta T_i < 1 \quad (8)$$

The T_{max} in Eq. (8) is empirically determined to ensure that the connection weights' renewals in Eqs. (4) and (5) are insignificant when the performance of an MLP is below average.

From Eq. (7),

$$1 - \tau^i(t) = \sum_{\substack{j=1\\j\neq i}}^{N}\tau^j(t), \quad (9)$$

so the temperature renewal in Eq. (6) can be written as

$$\triangle T_i(t) = (p^{self} + p^{cross})(N\tau^i(t) - 1) \quad (10)$$

Equation (10) implies that an MLP with good performance (low $\tau(t)$) is rewarded by decreasing its temperature proportional to $p^{self} + p^{cross}$, and penalized heavily when the performance is bad, so that the training process has no effect on its expertise.

To prevent the learning from getting stuck when all of the MLPs have average performances, a positive cool down-constant C is needed, and to ensure that the temperature increases when the performance is bad, C is set as

$$0 < C \ll (p^{self} + p^{cross})(N-1). \quad (11)$$

Furthermore, to ensure that an MLP is sufficiently penalized when the performance is bad and at the same time to prevent the cross-penalty term from nullifying the self-penalty term, the penalty constants are set as follows:

$$(p^{self} + p^{cross}) \approx \frac{T_{\max}}{N-1},$$
$$p^{cross} < \frac{p^{self}}{N-1}. \quad (12)$$

From Eq. (10) it is clear that, when at time t a correct learning pattern is given and that the winner is the i-th MLP, then the expected temperature at time $t+1$ is:

$$\langle T_i(t+1)\rangle = (1-\epsilon) + \epsilon\{1 + p^{tot}(N-1) - C\} = 1 + \epsilon\{(N-1)p^{tot} - C\}, \quad (13)$$
$$p^{tot} = (p^{self} + p^{cross}).$$

Equation (13) shows that, when the supervisor is perfect, the temperature of the winner converges to 1, and when the ϵ is non-zero then on the average the temperature of the winner fluctuates around a small value while the temperatures of the losers fluctuate around a larger value. The center of the winner's temperature fluctuation is determined by the error rate ϵ.

The temperature control is applied only to the neurons in the output layer, because to inactivate an MLP it is sufficient to inhibit the output neurons. Furthermore, the penalty given to an MLP is considered to be a penalty for the output neurons and not for the middle neurons.

The proposed temperature control will allow each member of the ensemble to learn a unique type of expertise. It can be expected that a particular ensemble member will be able to master the input-output relation presented in correct learning data by isolating itself from the erroneous data in the learning process, while letting the other members absorb the erroneous learning data.

4. Experiment

We conducted an experiment in which the ensemble was trained with a supervisor that generated a particular percentage of classification errors, where the error rate ϵ is defined as follows:

$$P(D_i = D_i^{false}|\mathbf{x}_i) = \epsilon, \quad (14)$$
$$P(D_i = D_i^{true}|\mathbf{x}_i) = 1 - \epsilon,$$
$$0 \leq \epsilon \leq 1.$$

Here, $P(D_i|\mathbf{x}_i)$ indicates the probability that the supervisor will give D_i as desired output for $\mathbf{x}_i$ input.

The reason why a single MLP will fail to deal with the problem in Eq. (14) is shown in Eqs. (15) and (16). For simplicity but without loss of generality, an MLP with one output neuron is considered. With an imperfect supervisor as in Eq. (14), the expected error of the MLP whenever input $\mathbf{x}_i$ is given, $\langle E(\mathbf{x}_i)\rangle$, can be written as

$$\langle E(\mathbf{x}_i)\rangle = \epsilon(O(\mathbf{x}_i) - D_i^{false})^2 + (1-\epsilon)(O(\mathbf{x}_i) - D_i^{true})^2. \quad (15)$$

For binary problems, Eq. (15) can be developed as

$$\langle E(\mathbf{x}_i)\rangle = [O(\mathbf{x}_i) - \langle D_i\rangle]^2 + \epsilon(1-\epsilon), \quad (16)$$
$$\langle D_i\rangle = \epsilon D_i^{false} + (1-\epsilon) D_i^{true},$$

where $O(\mathbf{x}_i)$ is the MLP's response for input $\mathbf{x}_i$ and $\langle D_i\rangle$ is the expected teacher signal for input $\mathbf{x}_i$.

The learning process will minimize the first term of Eq. (16), so that after the learning process, the output of the MLP will be $\langle D_i\rangle$ instead of the ideal response, D_i^{true} for input $\mathbf{x}_i$, implying that a single MLP is not appropriate for dealing with the given problem.

It can be expected that through the temperature control, the ensemble will be able to distinguish correct learning data from erroneous data and automatically allocate one of its members to learn only correct data, while letting the other members absorb the erroneous learning data. This implies that the effect of the supervisor's imperfection can be reduced, because the ensemble will produce a member that is trained only with the correct learning data. It can be predicted that learning performance will deteriorate along with an increase in the error rate, because the competition inside the ensemble will become frequent and consequently effective learning will rarely occur. Still, in contrast to the case for a conventional MLP, "graceful degradation" can be expected.

Table 1 Parameter settings for Experiment 1.

Parameter	Value
Learning rate	0.4
Momentum	0.1
Self-penalty	40
Cross-penalty	5
Cool-down	20
$T_{\max}$	100

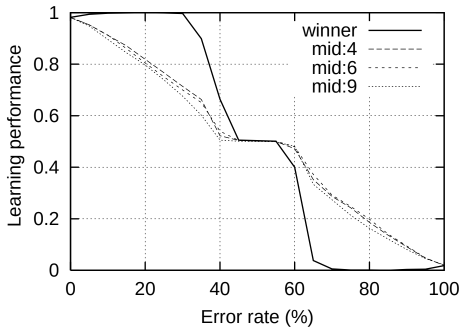

Fig. 2 Performances of the winner and single MLPs in XOR problem.

4.1 Experiment on Logic Function

In this experiment the ensemble was tested on XOR problem in which the supervisor generated a particular percentage of classification errors. The classification errors were generated by flipping the correct desired answer according to ϵ in Eq. (14). The ensemble consisted of 3 MLPs, each with 2 input neurons, 1 output neuron and 4, 6, and 9 middle neurons, respectively. Learning was iterated 40,000 times, and each time a pattern was randomly selected. The parameter settings are shown in **Table 1**.

At the end of the training process a winner, which is the member that benefits most from the training process, is chosen. The winner is selected by choosing the member with the lowest average temperature during the learning process. We iterated the experiment by changing the error rate and evaluate the learning performance of the winner with respect to errorless learning data. The learning performance A^i of the i-th MLP is calculated as follows:

$$A^i = 1 - \frac{1}{N_{pat}} \sum_{j}^{N_{pat}} |O^i(\mathbf{x}_j) - D^{true}(\mathbf{x}_j)|, \quad (17)$$

where N_{pat} is the number of testing patterns, $O^i(\mathbf{x}_j)$ is the output of the i-th MLP when pattern $\mathbf{x}_j$ is given, and $D^{true}(\mathbf{x}_j)$ is the correct desired output for the given input. **Figure 2** shows a comparison of the learning per-

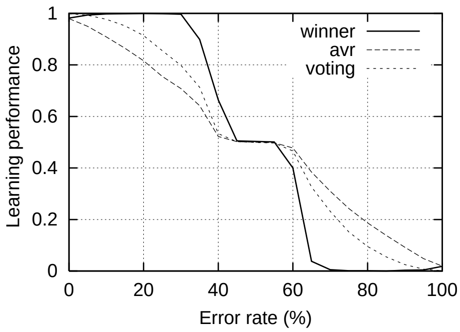

Fig. 3 Performances of the winner and multi-systems in XOR problem.

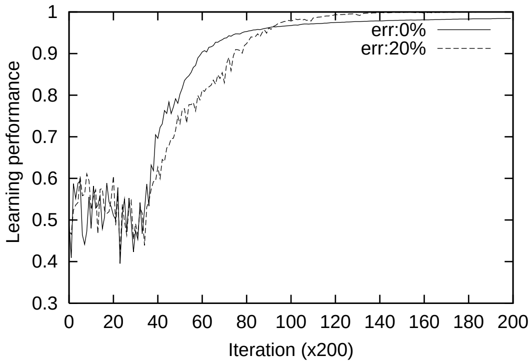

Fig. 4 Performance during training.

formance of the winner and each of the ensemble's members provided that they are independently trained using identical training data.

In Fig. 2, "winner" indicates the learning performance degradation of the winner of the ensemble with respect to the error rate, while "mid:4", "mid:6", and "mid:9" represent the degradation of independently trained MLP with 4, 6, and 9 middle neurons, respectively.

Figure 3 shows a comparison of the winner with two multi-neural-network systems using three independently trained neural networks. The first multinetwork system averages the members' outputs according to Eq. (18) as system output.

$$O^{avr} = \frac{1}{N_{mlp}} \sum_{i=1}^{N_{mlp}} O^i, \tag{18}$$

where O^{avr} is the system's output, O^i is the output of the i-th MLP, and N_{mlp} is the number of MLPs in the ensemble.

The second multinetwork system is based on statistical calculation with the assumption that each output represents the probability that the response of the MLP is 1. The system output is the probability O^{vot} that a majority of the members output 1. It can be considered that the system is a kind of majority voting system. Since in this experiment the system contains three members the output can be calculated as

$$O^{vot} = \prod_{i=1}^{3} O^i + \sum \sum_{i \neq j \neq k}^{3} \sum O^i O^j (1 - O^k). \tag{19}$$

From Figs. 2 and 3 it can be seen that the winner of the ensemble can retain more than 90% of its optimum performance until the error rate reaches 30%. The performance of the winner degrades much more gracefully than that of the conventional MLP or other multi-neural-network systems. This experiment is reiterated several times by changing the initial connections inside the ensemble's members, and the result shows that the learning ability of the ensemble with respect to an imperfect supervisor is independent of its initial weights, although the training process may produce a different winner for different initial conditions.

Figure 4 shows the winner's performance during the training process with respect to the supervisor with an error rate of 0% indicated by "err:0%" and 20% indicated by "err:20%" respectively. The performance of the winner is evaluated every 200 learning iterations. The oscillation seen in the early phase of performance evaluation line when the error rate 0% is caused not by the supervisor's imperfection but by the competition between the ensemble's members. Once the winner has been established the performance increases smoothly. When the error rate is 20%, the oscillation exists throughout the whole training phase; this not only results from the competition, but also reflects the imperfection of the supervisor.

Figures 5 and **6** show the fluctuations of each member's average temperature during learning when the error rates are 0% and 20%, respectively. In both graphs the temperatures are averaged after every 100 learning iterations. From Fig. 5, it is obvious that the temperature of the winner (MLP with 6 middle neurons) converges toward 1 after the winner is established, while the temperatures of the other members fluctuate in the higher region. Figure 6 shows that the temperature of the winner fluctuates in the low temperature region because of the imperfection of the supervisor, while the temperatures of the losers fluctuate

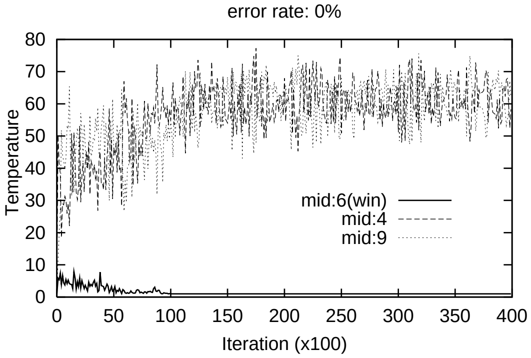

Fig. 5 Temperature during training (error rate: 0%).

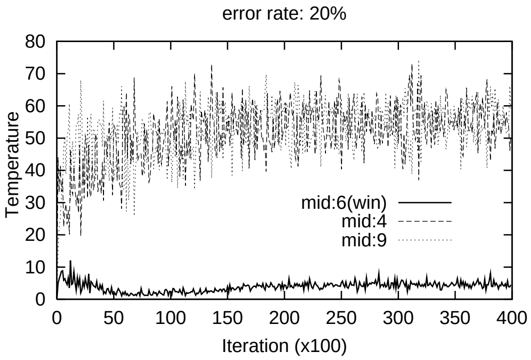

Fig. 6 Temperature during training (error rate: 20%).

Table 2 Learning performance for various structures.

	Structure	Performance
ensemble1	(1,1,1)	0.730
ensemble2	(1,6,1)	0.680
ensemble3	(2,6,1)	0.648
ensemble4	(4,5,6)	0.986
ensemble5	(4,6,9)	0.987

in a much higher temperature region. In both cases, the proposed temperature control algorithm produces a robust learning process.

We also conducted experiments on the ensemble's structure. In these experiments, we trained a number of ensembles with different structures to investigate the relation between the structure and learning performance. Ensembles with three members were used. Their structures were changed by altering the number of their middle neurons. The results of a performance test with an errorless supervisor are shown in **Table 2**.

The structure is represented by (), where the numbers inside the parentheses indicate the numbers of middle neurons in the respective members. The numbers of input and output neurons are fixed at 2, and 1, respectively.

Table 3 Parameter settings for Experiment 2.

Parameter	Value
Learning rate	0.4
Momentum	0.1
Self-penalty	25
Cross-penalty	6
Cool-down	20
$T_{\max}$	75

From Table 2, it is clear that an ensemble which has a member with only one middle neuron for which it is potentially difficult to learn XOR problem, cannot achieve satisfactory learning performance. We draw the conclusion that an ensemble can perform well if each member potentially has the ability to learn the given problem independently. Experiments with various problems and ensemble sizes also yielded similar results.

4.2 Experiment on Iris Data

We tested the performance of our neural network ensemble on Iris classification data, which is often used as a benchmark test problem in neural network research. There are three classes of Iris (iris-setosa, iris-versicolor, and iris-virginica). Because setosa is linearly separable, we considered the other two classes, making it a two-class problem. The input to the neural network is composed of four parameters, namely, the width and length of the sepal and the petal, respectively. All of the parameters are expressed in continuous values in centimeters.

For training, 20 data (10 for each class) were provided. Errors were generated by corrupting the teacher signal. The number of learning iterations was 60000. The ensemble consists of three members each with 4 input neurons, 1 output neuron and 6, 7, and 8 middle neurons, respectively. The parameter settings are shown in **Table 3**.

Figures 7 and **8** show that the ensemble can tolerate a supervisor error rate of up to 30% without degrading the learning performance in the Iris classification problem.

In this experiment we also conducted a test on the winner's generalization ability. The generalization performance test was conducted by classifying 20 test patterns (10 for each class) that were not used in the training process. The generalization performance comparisons with single MLPs and multi-system-neural-networks are shown in **Figs. 9** and **10**, respectively.

From this experiment it is clear that the proposed neural network ensemble is able to

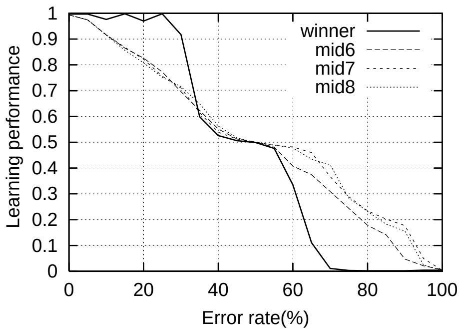

Fig. 7 Performances of the winner and single MLPs in the Iris problem.

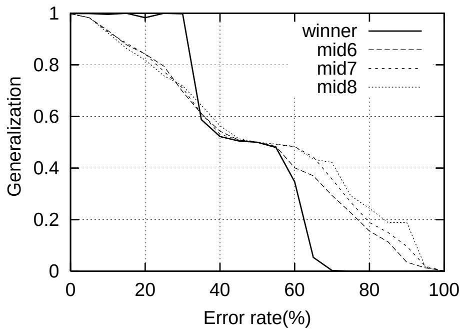

Fig. 9 Generalization of the winner and single MLPs on Iris problem.

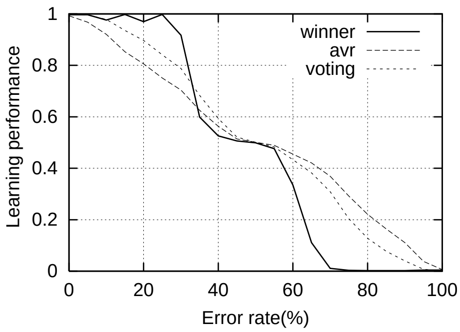

Fig. 8 Performances of the winner and multi-systems on Iris problem.

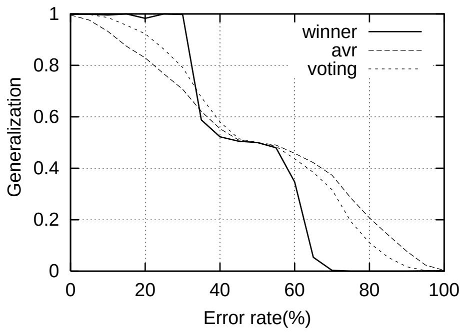

Fig. 10 Generalization of the winner and multi-systems in the Iris problem.

suppress the effects of a supervisor's imperfection to produce a member which has better learning and generalization abilities than conventional single-MLP and other multi-neural-network systems.

5. Conclusion

We have proposed a neural network ensemble model, in which each member is allowed to learn a unique type of expertise. Temperature control is introduced to the ensemble to realize automatic selection of the most appropriate member to deal with a given problem, allowing the ensemble to run effectively in non-static environments.

The experimental results show that the ensemble can be trained effectively even if the supervisor is imperfect, without excessively compromising the performance. The basic idea of the proposed method is that we can treat the correct input-output pairs as a particular environment and the incorrect ones as another environment. The ensemble can achieve good learning performance, because one of the ensemble's members is able to learn from correct learning patterns although the patterns are statistically mixed with erroneous learning patterns. From the experiment we conducted, it can be seen that the proposed neural network is able not only to learn better but also to generalize better than conventional MLP and other multi-neural-network systems in the presence of an imperfect supervisor. We can also argue that the proposed ensemble is superior to a single MLP with cooling-down temperature control or an MLP that rejects training data which generate large error after a maturation point, because the rejection of training data is based only on the performance of the single MLP, which is unreliable on account of the shifting of the learning target, which also produces large errors. In the proposed ensemble, no shifting of the learning target occurs, so a decision by the winner to reject training data is more reliable.

The strict requirement that errorless training data be provided to train a neural network can be loosened in the proposed neural network ensemble, and thus we have more freedom in designing a learning supervisor, because the ensemble can tolerate imperfect data to some extent.

Although in this paper we have explored

the ability of the proposed ensemble in only two class problems, it can be easily developed to perform multi-class classification problems without increasing the calculation complexity.

Future applications of the proposed ensemble will include on-line training, design of fault-tolerant systems, error detection and correction, probability learning, and non-linear equalization.

References

1) An, G.: The Effect of Adding Noise during Backpropagation Training on Generalization Performance, *Neural Computation* 8, pp.643–674 (1996).
2) Reed, R.: Regularization Using Jittered Training Data, *Proc. Int. Joint Conf. Neural Networks 92*, III-147 (1992).
3) Reed, R. and Marks, R.: *Neural Smithing—Supervised Learning in Feedforward Artificial Neural Networks*, pp.277–292, MIT Press (1999).
4) Matsuoka, K.: Noise Injection into Inputs in Back-Propagation Learning, *IEEE Trans. System, Man, and Cybernetics*, Vol.22, No.3, pp.436–440 (1992).
5) Holmstrom, L. and Koistinen, P.: Using Additive Noise in Back-propagation Training, *IEEE Trans. Neural Networks*, Vol.3, No.1, pp.24–38 (1992).
6) Minnix, J.: Fault Tolerance of the Back-propagation Neural Network Trained on Noisy Inputs, *Proc. IJCNN 92*, Vol.1, pp.847–852 (1992).
7) Thathachar, M. and Arvind, M.: Global Boltzman Perceptron Network for On-Line Learning of Conditional Distribution, *IEEE Trans. Neural Networks*, Vol.10, No.5, pp.1090–1098 (1999).
8) Hartono, P. and Hashimoto, S.: Temperature Switching in Neural Network Ensemble, *Journal of Signal Processing*, Vol.4, No.5, pp.395–402 (2000).
9) Hashem, S.: Optimal Linear Combinations of Neural Networks, *Neural Networks*, Vol.10, No.4, pp.559–614 (1997).
10) Sharkey, A.: On Combining Artificial Neural Nets, *Connection Science*, Vol.8, Nos.3 & 4, pp.299–313 (1996).
11) Rosen, B.: Ensemble Learning Using Decorrelated Neural Networks, *Connection Science*, Vol.8, Nos.3 & 4, pp.373–383 (1996).
12) Tumer, K. and Ghosh, J.: Error Correlation and Error Reduction in Ensemble Classifiers, *Connection Science*, Vol.8, Nos.3 & 4, pp.385–404 (1996).
13) Baxt, W.: Improving the Accuracy of an Artificial Neural Network Using Multiple Differently Trained Networks, *Neural Computation*, Vol.4, pp.772–780 (1992).
14) Hansen, L.K. and Salamon, P.: Neural Network Ensemble, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.12, No.10, pp.993–1001 (1990).
15) Jacobs, R., Jordan, M., Nowlan, S. and Hinton, G.: Adaptive Mixture of Local Experts, *Neural Computation*, Vol.3, pp.79–87 (1991).
16) Jordan, M. and Jacobs, R.: Hierarchical Mixture of Experts and the EM Algorithm, *Neural Computation*, Vol.6, pp.181–214 (1994).
17) Jordan, M. and Xu, L.: Convergence Results for the EM Approach to Mixture of Experts Architecture, *Neural Networks*, Vol.8, No.9, pp.1409–1431 (1995).
18) Müller, K.-R., Kohlmorgen, J. and Pawelzik, K.: Analysis of Switching Dynamics with Competing Neural Networks, *IEICE Trans. Fundamentals*, Vol.E78-A, No.10, pp.1306–1315 (1995).
19) Rumelhart, D.E.: Learning Internal Representation by Error Propagation, *Parallel Distributed Processing*, Vol.1, pp.318–362, MIT Press (1986).

(Received March 9, 2000)
(Accepted February 1, 2001)

Pitoyo Hartono received his B.S. and M.S. degrees from the Department of Applied Physics, Waseda University in 1993 and 1995, respectively. From 1998, he has been a Ph.D. candidate in Waseda University. His interests include Neural Networks and Evolutionary Computations.

Shuji Hashimoto received his B.S. and Ph.D. degrees from Waseda University in 1970 and 1977, respectively. He is currently a Professor at the Department of Applied Physics, Waseda University. His research interests include Neural Networks, Image Processing, Humanoid Robots and Kansei Information Processing.