A Hybrid Learning Algorithm for Multilayer Perceptrons to Improve Generalization under Sparse Training Data Conditions

Masanobu TONOMURA Kenji NAKAYAMA

Graduate School of Natural Science and Technology, Kanazawa Univ.
2–40–20 Kodatsuno, Kanazawa, Ishikawa, 920–8667, Japan
e-mail: tonomura@leo.ec.t.kanazawa-u.ac.jp

Abstract

The back-propagation algorithm is mainly used for multilayer perceptrons. This algorithm is, however, difficult to achieve high generalization when the number of training data is limited, that is sparse training data. In this paper, a new learning algorithm is proposed. It combines the BP algorithm and modifies hyperplanes taking internal information into account. In other words, the hyperplanes are controlled by the distance between the hyperplanes and the critical training data, which locate close to the boundary. This algorithm works well for the sparse training data to achieve high generalization. In order to evaluate generalization, it is supposed that all data are normally distributed around the training data. Several simulations of pattern classification demonstrate efficiency of the proposed.

1 Introduction

The back-propagation (BP) algorithm [1] is mainly used for multilayer perceptrons (MLP). This algorithm can approximate Bayes boundary using a sufficient number of training data in the statistical sense [2],[3]. This is the theoretical foundation which MLP are used as a classifier. This condition is, however, not always satisfied in the actual applications. Therefore, how to improve generalization ability using a limited number of the training data is very important [4]-[7].

Regarding conventional methods, generalization of the BP algorithm highly depends on the training data. The regularization learning method [4] requires more computation and results in slow convergence rate as the input dimension becomes high. Generalization by the weight elimination [5] is not sufficient for sparse training data. Performance of the SVM [7] depends on selection of the kernel function and other parameters.

In this paper, first, an internal information optimum (IIO) algorithm for single-layer perceptron is proposed, which is a basic algorithm for the optimization. The distance between the training data and the hyperplanes formed by connection weights from the input layer to the first hidden layer is called "internal infomation" in this paper. The IIO algorithm moves the hyperplanes taking the internal information into account and improves generalization. Convergence of this algorithm is theoretically proved. Furthermore, a hybrid learning algorithm for MLP is proposed, which combines the BP algorithm and the IIO algorithm. It does not require many training data and computational load, and is useful for not only sparse data but also mixed sparse and dense data distribution.

2 Assumption of Distribution

2.1 Optimum Boundary for Sparse Training Data

Under the sparse training data condition, if the categories are formed by using only the training data, the region of each category shrink compared with the optimum region, which can cover the other data than the training data. In such cases, the position of the hyperplane formed by the BP algorithm is not uniquely decided, rather it depends on the relative size of the connection weights, a learning-rate and a slope of sigmoidal functions. Then, the input potential, that is a linear combination of the inputs reaches a saturation point, where generalization is not sufficient, and the learning stops. Therefore, the BP algorithm cannot approximate the Bayes boundary, which is generally a quadratic hypersurface [8].

Since, it is difficult to conjecture the distribution of each category precisely by using the sparse training data, in this paper, it is supposed that all data are nor-

0-7803-7044-9/01/$10.00 ©2001 IEEE

mally distributed around the training data, occurrence rate of data of each category are equiprobable and all eigenvalues of the covariance matrix are the same.

Under the above assumptions, the optimum boundary orthogonally crosses the line at the center point, which connects the nearest-neighbor training data of two classes.

2.2 Evaluation of Generalization

There are deterministic and statistical evaluation of generalization. The former supposes that the training data are sampled from the data set with high density. The latter uses the samples which occur in accordance with the training data distribution density, and evaluates generalization on the average of many trials. First, it is shown that the IIO algorithm improves generalization in deterministic from Sec.3.1 to Sec.3.4 to make it easy to understand. Next, it is shown statistically by using the experiment in Sec.3.5.

3 Basic Algorithm

3.1 Basic Network

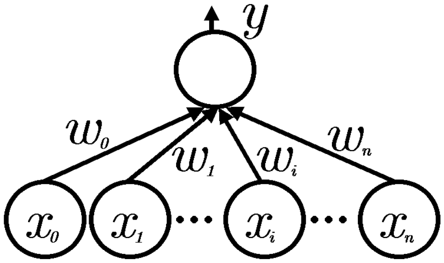

Figure 1: Single-layer perceptron

We consider the single-layer perceptron model with the n-dimensional inputs and one output as shown in Fig.1. These are defined as follows:

$$\begin{aligned} \mathbf{x} &= [x_0, x_1, \cdots, x_n]^t, x_0 = 1, x_i \in J & (1) \\ \mathbf{w} &= [w_0, w_1, \cdots, w_n]^t & (2) \\ y &= f(\mathbf{w}^t\mathbf{x}) & (3) \\ f(x) &= 1/(1 + e^{-x/u}) & (4) \end{aligned}$$

where x_0 is a bias ($x_0 = 1$), w_0. $J \equiv [0, 1]$. $\mathbf{w}^t$ is the transpose of the weight vector $\mathbf{w}$, and $f(\cdot)$ is a nonlinear activation function.

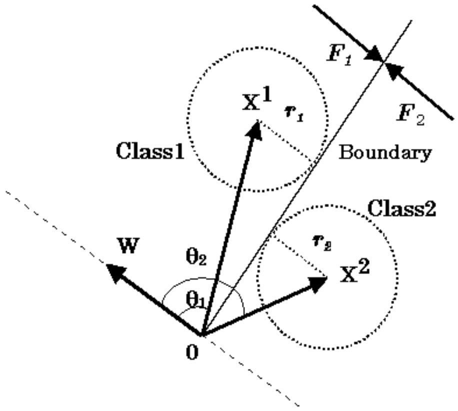

Figure 2: Vector space

3.2 Estimate Function

The multi-dimensional vector space which consists of a weight vector $\mathbf{w}$ and the training vectors $\mathbf{x}^1 \in Class1, \mathbf{x}^2 \in Class2$ is shown in Fig.2. If we assume that the test vectors are normally distributed around the training vectors and all the eigenvalue σ^2 of the n-by-n covariance matrix are the same, it is necessary that the radius (example:3σ) of the hypersphere does not exceed the hyperplane so that the test vectors may be classified properly. The radius r_p is a projection on the weight vector $\mathbf{w}$ of the training vector $\mathbf{x}^p (p=1, 2)$ and is given by

$$r_p = \frac{|\,\mathbf{w}^t\mathbf{x}^p\,|}{\|\,\mathbf{w}\,\|} = \|\,\mathbf{x}^p\,\|\,|\cos\theta_p\,| \qquad (5)$$

where $\|\cdot\|$ denotes the Euclidean norm. When $r_1 = r_2$, the noise permissible level of the training vector $\mathbf{x}^1$, $\mathbf{x}^2$ becomes equal without leaning. The inverse of Eq.(5) can be considered the force F_p which $\mathbf{x}^p$ pushes a hyperplane. The criterion, which measures a balance of these forces, is given by

$$E_{iio} \equiv \frac{1}{2\,|X\,|} \sum_{\mathbf{x} \in X} \Big(\beta(\mathbf{x})\frac{\|\,\mathbf{w}\,\|}{\mathbf{w}^t\mathbf{x}}\Big)^2 \qquad (6)$$

where $|X|$ is the number of the patterns which belong to a finite subset X of J^n. $\beta(\mathbf{x})$ is the weight function which makes the statistic nature of the training vectors reflect on the evaluation function. In this paper, $\beta = 1$.

3.3 IIO Algorithm

The IIO algorithm is based on the gradient descent algorithm, which minimizes the mean squared error E_{iio}. $\mathbf{w}(n)$ is updated as follows:

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) + \Delta\mathbf{w}\,|_{\mathbf{w}=\mathbf{w}(n)} & (7) \\ \Delta\mathbf{w} &= \mu \tanh\Big(-\frac{\partial E_{iio}}{\partial \mathbf{w}}\Big/T\Big) & (8) \end{aligned}$$

where n is an iteration number. μ is a leaning-rate parameter, $0 < \mu \ll 1$. A slope of the tanh function is controlled by T. The purpose of using the tanh function is to assure convergence even when a hyperplane is in the neighborhood of the training vectors.

⟨The condition of the convergence⟩

Convergence can be assured by deciding a learning rate μ in the range that it does not exceed the shortest distance between the regions of the different classes. The sufficient condition for the convergence is given by

$$\mid \max[\Delta w] \mid = \mu < \min\{d[x_i^p, x_i^q]\} \tag{9}$$

where $d[x_i^p, x_i^q]$ is a distance between the elements x_i^p, x_i^q, $(i=1,\cdots,n)$, $p \in X^p, q \in X^q, p \neq q$. Actually, small value which satisfies Eq.(9) is used to restrain vibration.

3.4 The Nature of IIO Algorithm

By minimizing r_p, the hyperplane can cross the straight line, connecting the training vectors in two classes, at the center point. The inverse of r_p also moves the hyperplane like this. Furthermore, they are orthogonal to each other. This is the important nature realized by using the inverse of r_p.

In the case of two-pattern two-class classification, from Eq.(8), the correction is

$$\begin{aligned} \Delta\mathbf{w} &= \Big(\frac{\|\mathbf{w}\|^2\mathbf{x}^1}{(\mathbf{w}^t\mathbf{x}^1)^3} - \frac{\mathbf{w}}{(\mathbf{w}^t\mathbf{x}^1)^2}\Big) \\ &\quad + \Big(\frac{\|\mathbf{w}\|^2\mathbf{x}^2}{(\mathbf{w}^t\mathbf{x}^2)^3} - \frac{\mathbf{w}}{(\mathbf{w}^t\mathbf{x}^2)^2}\Big) \end{aligned} \tag{10}$$

where $\mu, \mid X \mid, the$ tanh function are omitted for simplisity. Letting $\Delta\mathbf{w} = 0$ in Eq.(10), the optimum $\mathbf{w}$ is given by

$$\mathbf{w} = \frac{\|\mathbf{w}\|^2 \left((\mathbf{w}^t\mathbf{x}^2)^3\mathbf{x}^1 + (\mathbf{w}^t\mathbf{x}^1)^3\mathbf{x}^2\right)}{(\mathbf{w}^t\mathbf{x}^1)^3(\mathbf{w}^t\mathbf{x}^2) + (\mathbf{w}^t\mathbf{x}^2)^3(\mathbf{w}^t\mathbf{x}^1)} \tag{11}$$

Furthermore, supposing both radiuses are equal ($r_1 = r_2$)

$$\mathbf{w}^t\mathbf{x}^1 = -\mathbf{w}^t\mathbf{x}^2 \tag{12}$$

By substituting Eq.(12) into Eq.(11)

$$\mathbf{w} = \frac{\|\mathbf{w}\|^2}{2(\mathbf{w}^t\mathbf{x}^2)}(\mathbf{x}^2 - \mathbf{x}^1) \tag{13}$$

From this analysis, it is confirmed that the IIO algorithm can results $\mathbf{w}$ being parallel to the straight line, connecting $\mathbf{x}^1$ and $\mathbf{x}^2$, that is the adjusted hyperplane and the straight line are orthogonal to each other.

3.5 Statistical Generalization of IIO Algorithm

In this section, statistical generalization of the BP algorithm and the IIO learning algorithm, applied after the BP learning, is compared under the sparse training data condition. Two-dimensional inputs, two patterns and two classes are taken into account. The center of each category are $\mathbf{x}_c^1(\in Class1) = (0.3{,}0.7), \mathbf{x}_c^2(\in Class2) = (0.7{,}0.3)$. Standard deviation for each class are equal ($\sigma = 0.1$). 1000 test vectors are used in each class. The average recognition rate and the dispersion (variance) of 100 trials are evaluated. Simulation results are shown in Fig.3. The horizontal axis indicates the number of the training vectors which occurs in accordance with the distribution density. As the number of the training data is increased, the recognition rate improves and the dispersion becomes small. They are saturated at 7 training data. The recognition rate of the IIO learning algorithm is beyond that of the BP algorithm in all cases. Thus, the IIO learning algorithm can improve generalization based on the statistical evaluation.

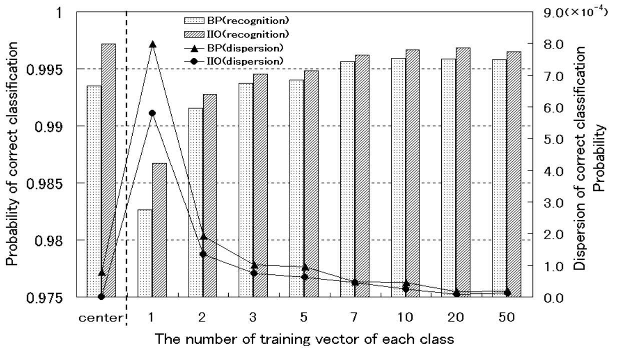

Figure 3: Compare of statistical generalization ability of the BP learning and the IIO learning.

The "center" on the horizontal axis means the case that the training vectors are fixed at the center of each class. In this case, the recognition rate is higher than the other cases. This means that the recognition rate can be improved by collecting the training vectors around the central of the distribution.

4 A Hybrid Learning Algorithm for MLP

If the input vectors are mapped onto around the apex of the hypercube through the first hidden layer with a sigmoidal nonlinear function, the generalization ability is not affected by the hyperplanes formed with the connection weights in the upper layers. Because the outputs of the first hidden layer are not affected by disturbance in the input data. The same situation is observed in the upper layers. On the other hand, effects of the hyperplanes formed with the connection weights from the input layer to the first hidden layer on the generalization ability is high. Therefore, these hyperplanes should be moved to the suitable position to improve generalization under the sparse training data condition. So, we propose a hybrid learning algorithm for MLP. The correction of the IIO algorithm is combined to that of the BP algorithm in learning the connection weights from the input layer to the first hidden layer.

For $N+1$ multilayer perceptron with n_0-input, n_N-output, the connection weight w_{pij} is updated as follows:

$$w_{pij}(n+1) = w_{pij}(n) + \Delta w_{pij}|_{w=w(n)} \tag{14}$$

$$\Delta w_{pij} = \begin{cases} \Delta w^{bp}_{pij} + \Delta w^{iio}_{pij} & (p=1) \\ \Delta w^{bp}_{pij} & (2 \le p < N) \end{cases} \tag{15}$$

$$\Delta w^{bp}_{pij} = \eta \frac{1}{|X|} \sum_{\mathbf{x}\in X} \delta_{pj}(\mathbf{x}) g_{p-1,i}(\mathbf{x}) \tag{16}$$

$$\Delta w^{iio}_{1ij} = \mu \tanh\Big(\frac{1}{|X|}\sum_{\mathbf{x}\in X}\frac{\beta^2(\mathbf{x})}{(\mathbf{w}^t_{1j}\mathbf{x})^2}\Big(\frac{\|\mathbf{w}_{1j}\|^2}{\mathbf{w}^t_{1j}\mathbf{x}}x_i - w_{1ij}\Big)\Big/T\Big) \tag{17}$$

where $w_{pij}(p=1,\cdots,N, i=0,\cdots,n_{p-1}, j=1,\cdots,n_p)$ are the connection weights of jth unit in $p+1$th layer from ith unit in pth layer. The correction Δw_{pij} shown in Eq.(15) is a combination of Δw^{bp}_{pij} and Δw^{iio}_{pij}. The correction Δw^{bp}_{pij} shown in Eq.(16) is defined by a delta rule. η is a learning rate, $0 < \eta < 1$. Equation(17) is indicated with the element of Eq.(8). $\mathbf{w}_{1j}$ is a weight vector of jth unit in the first hidden layer. μ must satisfies both Eq.(9) and $(0 < \mu/\eta \ll 1)$ which does not affect pattern classification by the BP algorithm. The correction Δw_{1ij} propagates as shown in Fig.4.

Figure 5 shows the relative relation of Δw^{iio}_{1ij}(solid line) and Δw^{bp}_{1ij}(dotted line). When the leaning does not converge, Δw^{bp}_{1ij} is dominant because μ is very small in comparison with η($Case1$). As the learning converges to a certain extent, Δw^{iio}_{1ij} becomes dominant. When the sign of Δw^{iio}_{1ij} and Δw^{bp}_{1ij} is the same, Δw^{iio}_{1ij} accelerates the learning, and slows down it with the oposite sign. When the sign of Δw^{bp}_{1ij} and Δw^{iio}_{1ij} are different as shown in $Case2, 3$, it becomes $\Delta w_{1ij} = 0$ in the position of the thin dotted line and the learning stops. Although the learning should converge at $\Delta w^{iio}_{1ij} = 0$ ideally, there is a gap. In order to restrain this gap to be the minimum, T in Eq.(17) is made as small as possible to make the width of Δw^{iio}_{1ij} short. Actually, the hyperplane is adjusted in the optimum position, because Δw^{bp}_{1ij} decreases monotonously with a certain sign after the classification ability is constructed, and the generalization is further improved.

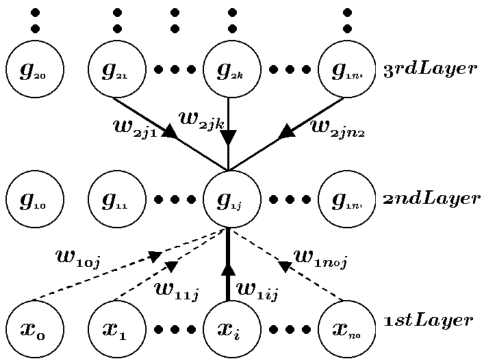

Figure 4: The propagation of the modification of the connection weight w_{1ij}

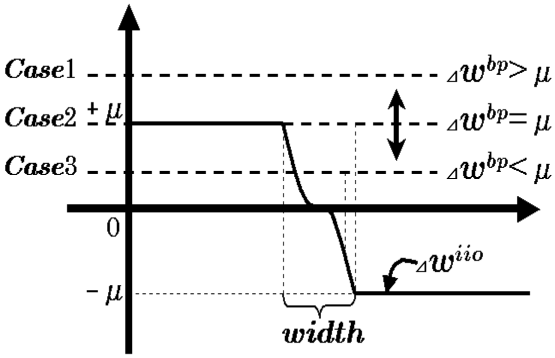

Figure 5: Relations between the correction Δw^{bp} and Δw^{iio}

Under the sparse training data condition, if categories are linearly separable, then the generalization can be improved by using the two-step learning method, in which the IIO learning algorithm is applied after the BP learning converges. However, in actual applications, we must presume mixed sparse and dense data distribution, in which the training data are overlapped between the categories. In this case, it is may be expected that the quadratic hypersurface, adjusted by using the two-step learning algorithm which emploies the distance between the training data and the hyperplanes, is not good for the generalization. However, the hybrid learning algorithm has the following feature, it moves the hyperplanes to improve generalization in the sparse distribution and does not influence for the dense distribution, in which Δw^{bp}_{1ij} is dominant. Even if Δw^{bp}_{1ij} is obstructed by Δw^{iio}_{1ij}, Δw^{bp}_{1ij} increases due to the increase of the BP error, and $|\Delta w^{bp}_{1ij}+\Delta w^{iio}_{1ij}| > 0$.

Table 1: Compare of recognition rate of the BP learning and a hybrid learning in each example.

	BP learn	Hybrid learn
Example 1	0.953	0.993
Example 2	0.950	0.950
Example 3	0.898	0.901

5 Simulation Results

5.1 Two-Class Problem

In order to observe the hyperplanes and recognition regions visually, two-dimensional two-class classification is employed for computer simulations. Multilayer perceptrons with two hidden layers are used. The number of the input units, the 1st hidden units, the 2nd hidden units, and the output units are 2, 10, 5 and 2, respectively. An offset unit is included in each layer. The recognition rate of the BP algorithm and the hybrid learning algorithm are compared by using the same initial connection weights. The targets 0.99, 0.01 are used to destinguish two classes. The classification was done using the threshold 0.5. Regions where the output ≥ 0.5 and < 0.5, are drown with black and white colors, respectively. Gray straight lines indicate the hyperplanes from the input layer to the first hidden layer. Recognition rates are shown in Table 1.

5.1.1 Example 1: Sparse Distribution. As shown in Fig.6, five training data are used, where 1 datum (O) belongs to Class 1 and 4 data (X) belong to Class 2. The dotted circles are the assistants of the same distance from each pattern. Recognition rates are calculated by using 1000 test data, which occur around the training data. Figure 6(a) shows the boundary obtained through the BP algorithm. From this result, the generalization is not good. Figure 6(b) shows the result by the proposed hybrid learning algorithm. The hyperplanes shown in Fig.6(a) are further modified to improve the generalization.

5.1.2 Example 2: Overlap Distribution. 200 training data in each category are sampled in accordance with the specified distribution. The data distribution is given by the center coordinate, $\mathbf{x}_c^1(\in Class1) = (0.4,0.6)$, the standard deviation $\sigma_1 = 0.05$, and $\mathbf{x}_c^2(\in Class2) = (0.6,0.4)$, $\sigma_2 = 0.15$. The recognition rate is calculated with 1000 test data, sampled in the same way as the training data. Simulation results are shown in Fig.7, in which dots indicate the training data. The boundary approximates the Bayes discriminant function. The hyperplane positions of the hybrid learning algorithm are different from that of the BP algorithm. However, their regions are almost the same. The recognition rate is also equal from Table 1. From this result, it can confirm that the proposed does not influence the quadratic hypersurface made by the BP algorithm.

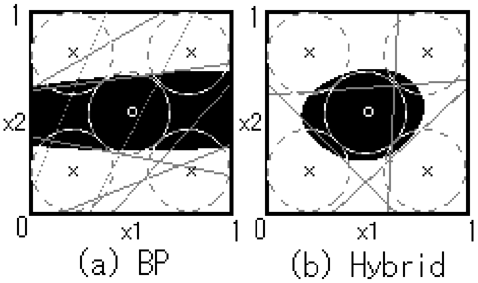

Figure 6: Classification regions and hyperplane in the case that a space of the categories is sparse.

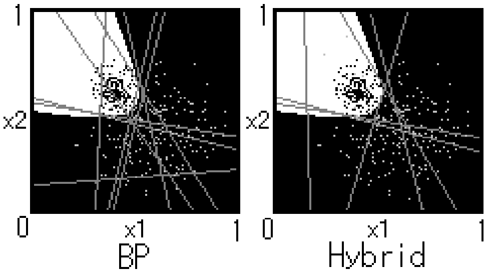

Figure 7: Classification regions and hyperplane in the case that a space of the categories is dense.

5.1.3 Example 3 : Mixed Sparse and Dense Distribution. 100 training data in each category are sampled from the distribution, specified with the center coordinate, $\mathbf{x}_c^1(\in Class1) = (0.4,0.6)$, the standard deviation $\sigma_1 = 0.1$, $\mathbf{x}_c^2(\in Class2) = (0.9,0.1)$, $\sigma_2 = 0.02$, and $\mathbf{x}_c^3(\in Class2) = (0.3,0.7)$, $\sigma_3 = 0.05$. The recognition rate is calculated with 1000 test data. Simulation results are shown in Fig.8. The proposed algorithm does not influence the dense distribution, and moves the hyperplanes to improve the generalization in the sparse distribution.

5.2 Multi-Class Problem

A multi-class problem with 6-dimensional input and 4-classes is employed for computer simulations. A multilayer perceptron with two hidden layers is used. The number of the input units, the 1st hidden units, the 2nd hidden units and the output units are 7, 20, 7 and 5, respectively. The bias unit is included in each layer. Four training data generated, which have distance of 1 among them, and assigned to each class. The de-

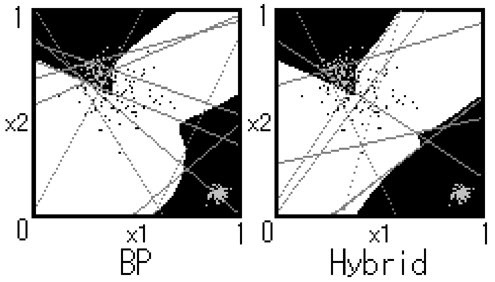

Figure 8: Classification regions and hyperplane in the mixed case that a space of the categories are dense and sparse.

sired outputs of the assigned class is 0.99, and other classes are 0.01. Recognition rate is calculated by using 1000 test data, which distribute around the training data with the same standard deviation, $\sigma = 0.2$. The hybrid learning algorithm is compared with the BP algorithm by using the same initial connection weights. The learning is stopped at 50,000 times, resulting in well reduced error. The learning is tried 10 times by changing the initial connection weights.

Simulation results are shown in Fig.9. In any cases, recognition rates of the proposed are better than those of the BP algorithm. Improvement of the recognition rate is large in the 3rd trial, while small in the 5th trial. This difference is caused by the relative size of connection weights, a learning-rate parameter and a slope of sigmoidal function as stated in 2.1. The recognition rates in the 3rd and the 4th trials are low in comparison with the other trials, where the proposed learning algorithm is used. This is caused by the difference in the combination of the hyperplanes constructed by the BP algorithm. It shows that the proposed hybrid algorithm improves the generalization ability within this combination.

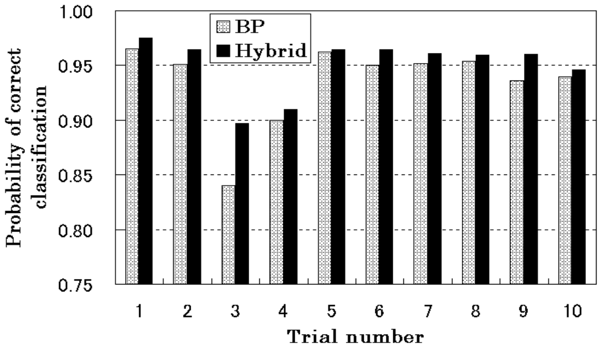

Figure 9: Compare of recognition rate of the BP learning and a hybrid learning.

6 Conclusions

We have proposed the learning algorithm for the sparse training data to achieve high generalization. The generalization can be drastically improved compared with the BP algorithm with sparse training data and without increasing computational load.

In this paper, we have used the ratio $\mu/\eta = 10^{-3} \sim 10^{-4}$, the learning-rate μ of the IIO algorithm and η of the BP algorithm in the computer simulations. We must clear the effective range of the value μ/η theoretically as a future subject.

References

[1] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning internal representations by error propagation," in Parallel Distributed Processing vol.1, eds. J.L.McCleland, D.E.Rumelhart, and The PDP Research group, MIT press, 1986.

[2] D.W.Ruck, S. Rogers, M. Kabrisky, H. Oxley, and B. Suter, "The multilayer Perceptron as an approximator to a Bayes optimal discriminant function," IEEE Trans. Neural Networks, vol.1, no.4, pp.296-298, 1990.

[3] K. Funahashi, "Multilayer neural networks and Bayes decision theory," Neural Networks, vol.11, pp.209-213, 1998.

[4] S.Akaho, "Regularization learning of neural networks for generalization," Proc. Workshop on Algorithmic Learning Theory, pp.99-110, 1992.

[5] A.S. Weigend, D.E. Rumelhart, and B.A. Huberman, "Generalization by weight elimination applied to currency exchange rate prediction," Proc. International Joint Conference on Neural Networks, vol.3, pp.2374-2379, Singapore, 1990.

[6] K.Nakayama and Y.Kimura, "Optimization of activation functions in multilayer neural network," Proc.IEEE ICNN'94, Orlando, pp.431-436, June 1994.

[7] V.Vapnic, Statistical Learning Theory, Wiley, 1998.

[8] R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, John Wiley & Sons, 1973.