Learning Efficiency and Temperature Coefficient of Forward Type Three Layer Neural Networks

Placed on: Osaka City University Repository

Learning Efficiency and Temperature Coefficient of Forward Type Three Layer Neural Networks

Jiro OKAMOTO [*], Shigeyoshi NAKAJIMA [†]and Shoichi HOSOKAWA [‡]

(Received September 30,1997)

Synopsis: In this paper, (1) in case of the forward type three layer neural network (NN), we make clear that there exists the most suitable number of units in the hidden layer. (2) In case of the multi-layer forward NN, when the total number of the units in the hidden layers is limited, the success rate of the learning is increased to arrange their units numbers as descending order from the input layer to the output one. (3) The learning of NN, whose temperature coefficient T is not one, is transformed into the equivalent learning scheme whose T is one, by changing the learning coefficient α to α/T^2, so that we need not treat the temperature coefficient as an independent coefficient concerning the learning of NN.

Keywords: *neural network, parity discrimination, the number of hidden units, temperature coefficient, learning.*

1 Introduction

In this paper we only treat the forward type neural networks (NN's), especially three layer NN's. Concerning the learning of these NN's, those are usually problems that; (1) what is the suitable number of units in the hidden layer?; (2) which temperature coefficient is desirable?

In spite of the importance of these problems, they are determined by one's intuition and experience at present.

Regarding the problem (2), it is assertened that while learning the convergence rate is increased by decreasing the temperature coefficient [1].

In this paper we deal with these problems taking the parity discrimination problem as an example and make clear the efficiency of NN learning concerning the number of units in the hidden layer and concerning the temperature coefficient.

The reason why we take the parity discrimination problem as an example is that the minimum number of units in the hidden layer is proved to be $[N/2]+1$ for N bits parity discrimination [2].

One of the main results in this paper lies in the clarification that there is a reasonable number of units in the hidden layer by which an efficient learning is done and the number is experimentally obtained as four times as large as this minimum number of units in the hidden layer regarding to the parity discrimination.

The other lies in that the learning constant α and the temperature coefficient T are mutually connected and by changing α to α/T^2 and weight coefficient ω_{ij} or ω_i to ω_{ij}/T or ω_i/T, we can theoretically and experimentally get the same learning result or process of NN in which $T=1.0$ as the process in which $T \neq 1.0$.

Using this rule, we are able to execute the NN learning depending only on α, but not concerning the temperature coefficient.

If we know the times of units in the hidden layer compared with the minimum number, in parity problem, we may conjecture the preferable number of units for the other problems.

[*] Associate Professor, Department of Information and Computer Engineering

[†] Research Associate, Department of Information and Computer Engineering

[‡] Professor, Department of Information and Computer Engineering

2 Parity Discrimination

There is 2^N different signals for N bits signals. To discriminate these signals by parity is to classify them into two categories, in which adjacent vertex to each other are classified as the different categories respectively, considering the signals as the vertexes of a N dimensional hyper-cube. This classification is regarded as one of the most difficult classification problem for binary signals.

The minimum number of units in the hidden layer for a forward type three layer NN to N bits parity discrimination is already proved to be $[N/2]+1$ [2] [3] [4].

But as this number is a theoretically possible minimum number of hidden units, actually this number is an impossible number to discriminate the parity of the signals if the input number N is seven or more. Accordingly to know what times of the theoretical minimum number of hidden units to be converged is efficient, is thought to be very useful, even for other discrimination problems to conjecture the necessary number of hidden units.

In the sequel we abbreviate this number as the minimum number of hidden units or more simply the min number.

3 Experiments and their results

Fig.1 shows the relation between the convergent rates on vertical axis and the number of hidden units on horizontal axis. The values of the parameters in these experiments are (1) 0.2 for learning rate, (2) 0.9 for momentum (3) 2,000 trials in learning cycle, and 1.0 for temperature. All other experiments in this section are done by the same conditions as this.

According to this graph in Fig.1, we can conclude that the convergent rate becomes higher as the number of the hidden units is larger, but saturates at points near about four times of the min number.

Table 1 shows the convergent rates when the numbers of units in two hidden layers are changed on the condition that the total hidden number of units is constant. Judging from this table the arrangement of the number of the hidden units in each layer may be recommended to arrange as the descending order from the input layer to the output one.

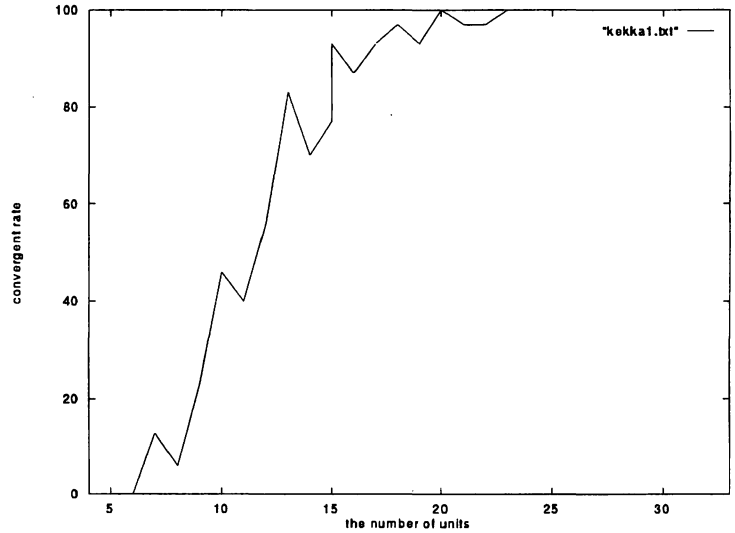

Fig.1 Relation between convergent rates and numbers of hidden units

Fig.2 is a graph showing the relation between the computation efficiency, that is, [convergent rate (%)/ computing time until converge] and the number of hidden units in three layer NN as horizontal axis. The definition of the computational efficiency is based on the idea that the convergent rate is gradually saturated when the number of hidden units is increased, but the computation time is much increased than the increment of this convergent rate. So that the efficienct number of units is at the point that this ration is maximum.

According to this graph we can clearly see that there exists the most suitable number of hidden units concerning this computational efficiency.

Table 1 Convergent rate when the numbers of hidden units are changed

construction of NN	convergent rate
6-4-12-1	13%
6-5-11-1	30%
6-6-10-1	37%
6-7-9-1	60%
6-8-8-1	53%
6-9-7-1	60%
6-10-6-1	80%
6-11-5-1	77%
6-12-4-1	77%

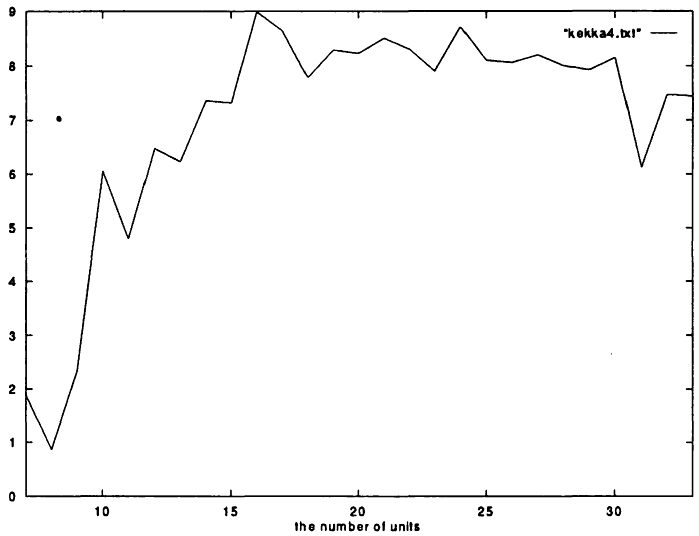

Fig.2 Relation between the computation efficiency and the number of hidden units

4 Application of This Efficiency to a Numeric Recognition

We applied the above mentioned proposition to the proposition that there exists the most suitable number of hidden units to a numeric recognition in order to ascertain it.

Fig.3 shows the relation between the number of hidden units (as horizontal axis) and the computational efficiency concerning a numeric recognition in which the mesh of a numeric character is divided into 7 × 5. The construction of the NN is 35 - [the number of hidden units] - 10. In this case we can also see that the most suitable number of hidden units exists.

As $2^4 > 10 > 2^3$ where 10 is the number of numeric and 4 is the number of inner expression needed to express its number, so that the minimum number of hidden units may be thought to be 4.

Accordingly the most efficient number is also four times of this minimum number 4, that is 16, as same as the case of the parity discrimination.

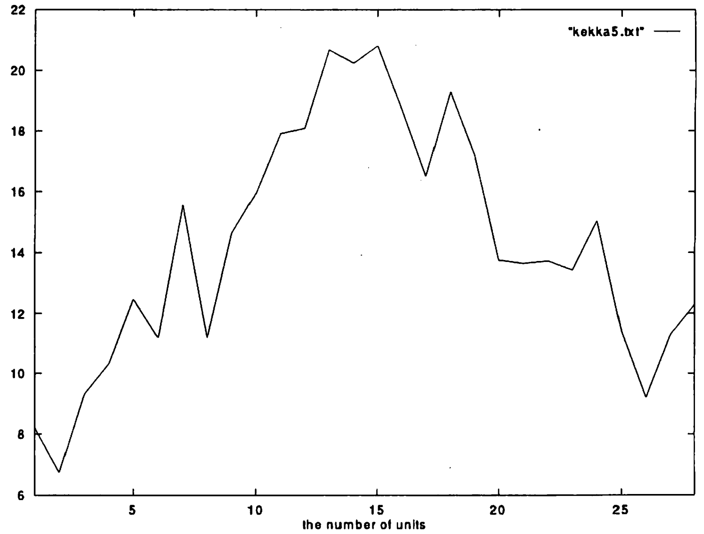

Fig.3 The computation efficiency in case of the recognition of numeric

5 The Relation between Learning Constant and Temperature Coefficient

5.1 Learning and local minimum

We define here the construction of a three layer parity discriminating NN precisely in Fig.4 in order to obtain the relation between learning constant and temperature coefficient. Fig.5 is a simplified construction of Fig.4 using matrix and vector notations.

Where $\mathbf{X}$ is input, $\mathbf{W}^{(i)}$ is weight coefficient to i-th layer, $\theta_j^{(i)}$ is threshold of j-th unit in the i-th layer, $I_j^{(i)}$ is a net input to the j-th unit in the i-th layer Y is the output, those are expressed as follows;

$$\mathbf{X} = (x_1, x_2, \cdots, x_N)^t,\ x_i = 0 \text{ or } 1, \tag{1}$$

where t is transpose.

$$\mathbf{W}^{(1)} = \begin{pmatrix} \theta_1^{(1)} & \omega_{11} & \omega_{21} & \cdots & \omega_{N1} \\ \theta_2^{(1)} & \omega_{12} & \omega_{22} & \cdots & \omega_{N2} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \theta_M^{(1)} & \omega_{1N} & \omega_{2N} & \cdots & \omega_{NM} \end{pmatrix} \tag{2}$$

$$\begin{aligned} \mathbf{I}^{(1)} &= (1, I_1^{(1)}, I_2^{(1)}, \cdots, I_M^{(1)})^t & (3) \\ \mathbf{I}^{(1)} &= W^{(1)}\mathbf{X} & (4) \\ \mathbf{W}^{(2)} &= (\theta^{(2)}, \omega_1, \omega_2, \cdots, \omega_M). & (5) \end{aligned}$$

The local minimum is the point where

$$\nabla E = \mathbf{0}, \tag{6}$$

where

$$E = \sum_P (Y_P - Z_P)^2/2. \tag{7}$$

Where P means the P-th pattern of binary signals. Suffix P is used only when the discrimination for patterns is necessary.

And ∇ is defined as;

$$\nabla = (\partial/\partial\omega_{11}, \partial/\partial\omega_{12}, \cdots, \partial/\omega_{NM}, \partial/\partial\omega_1, \cdots, \partial/\partial\omega_M, \partial/\partial\theta_1^{(1)}, \cdots, \partial/\partial\theta_M^{(1)}, \partial/\partial\theta^{(2)}). \tag{8}$$

If we assume that each unit has the same time constant, the output Y is expressed as;

$$Y = f_T(\mathbf{W}^{(2)} f_T(\mathbf{W}^{(1)}\mathbf{X})). \tag{9}$$

If we transform $\mathbf{W}^{(i)'} = T\mathbf{W}^{(i)}$ in the above equation, we can get the equivalent NN where its weight parameters are $\mathbf{W}^{(i)}$ and temperature is $T = 1$. Both outputs are equal and furthermore both constructions are topologically the same and they have the same local minimum. That is, in the equivalent NN's, all relations are not changed between the NN of $T \neq 1.0$ and the NN of T=1.0 by the transformation from $\mathbf{W}$ to $\mathbf{W}'$. Accordingly the expectation to get higher convergent rate by changing T from large to small as for the case of the Hopfield type NN's[1] is denied at least for this forward type NN's.

The learning is done to minimize the error E. In the sequel we put T as a suffix when the parameter $T \neq 1$ in NN as follows. For sigmoid functions;

$$f_T(x) = 1/(1 + e^{-x/T}) \tag{10}$$
$$f(x) = 1/(1 + e^{-x}). \tag{11}$$

According to Eq.10 and Eq.11, we get the next relation as;

$$f(x/T) = f_T(x) \tag{12}$$

Using this relation, we get a next relation

$$\begin{aligned} Y_T &= f_T(\mathbf{W}_T^{(2)} f_T(\mathbf{W}_T^{(1)} \cdot \mathbf{X})) && (13)\\ &= f(\frac{\mathbf{W}_T^{(2)}}{T} f(\frac{\mathbf{W}_T(1)}{T} \cdot \mathbf{X})) && (14)\\ &= f(\mathbf{W}^{(2)} f(\mathbf{W}^{(1)})) && (15)\\ &= Y, && (16)\end{aligned}$$

where we used the relation as next transformation as;

$$\mathbf{W}^{(i)} = \mathbf{W}_T^{(i)}/T \tag{17}$$

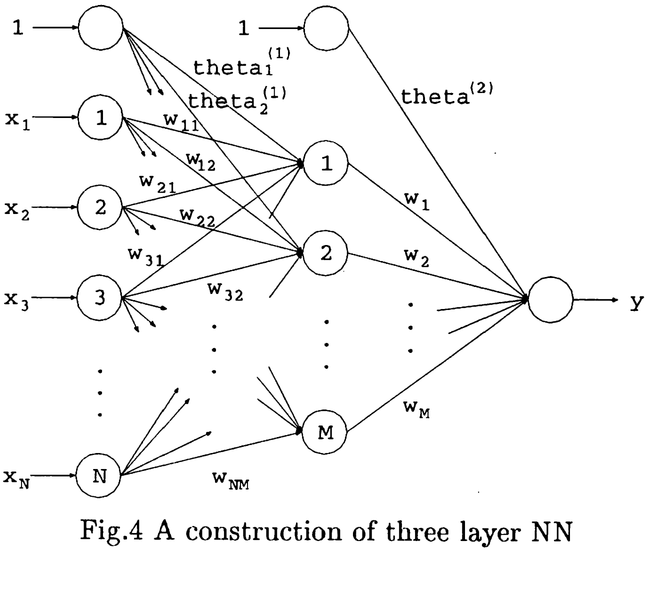

Fig.4 A construction of three layer NN

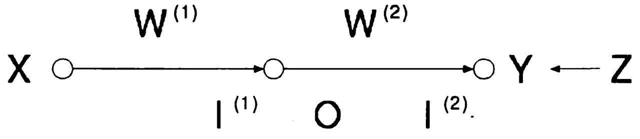

Fig.5 A simplified construction of three layer NN

Next we will get the learning rule for the NN of $T = 1$ which is equivalent to the NN of $T \neq 1$ dividing into two cases.

(1) From the hidden layer to the output layer

The learning rule is

$$\begin{aligned} \Delta\omega_T i^{(2)}(t) &= -\alpha_T d_T^{(2)} O_{Ti} + \beta_T \Delta\omega_{Ti}^{(2)}(t-1) & (18) \\ d_T^{(2)} &= 2(Y_P - Z_P) df_T(I_T^{(2)})/dI_T^{(2)} & (19) \end{aligned}$$

Using Eq.17 we get

$$\begin{aligned} T\Delta\omega_i^{(2)}(t) &= -\alpha_T^2(y_P - t_P)\frac{df(TI^{(2)}/T)}{TdI^{(2)}}O_{Ti} + \beta_T T\Delta\omega_i^{(2)}(t-1) & (20) \\ &= -\alpha_T^2(Y_P - Z_P)\frac{df(I^{(2)})}{dI^{(2)}}O_i + \beta_T T\Delta\omega_i^{(2)}(t-1), & (21) \\ & & (22) \end{aligned}$$

therefore,

$$\Delta\omega_i^{(2)}(t) = -\alpha_T\frac{2}{T^2}(Y_P - Z_P)\frac{df(I^{(2)})}{dI^{(2)}}O_i + \beta_T\Delta\omega_i^{(2)}(t-1). \quad (23)$$

Accordingly we get the relation for the equivalent learning constant α, concerning the NN when $T = 1$ as;

$$\begin{aligned} \alpha &= \alpha_T/T^2, & (24) \\ \beta &= \beta_T. & (25) \end{aligned}$$

(2) From the input layer to the hidden layer

The learning rule is;

$$\begin{aligned} \Delta\omega_{Tij}^{(1)}(t) &= -\alpha_T^{(1)} d_{Tj}^{(1)} x_i + \beta_T\Delta\omega_{Tij}^{(1)}(t-1) & (26) \\ d_{Tj}^{(1)} &= \sum_i d_t^{(2)}\omega_{Ti}^{(2)}\frac{df_T(I_{Ti}^{(1)})}{dI_{Tj}^{(1)}}, & (27) \end{aligned}$$

therefore,

$$\begin{aligned} T\Delta\omega_{ij}^{(1)}(t) &= -\alpha_T^{(1)} d_{Tj}^{(1)} x_i + \beta_T T\Delta\omega_{ij}^{(1)}(t-1) & (28) \\ d_{Tj}^{(1)} &= \sum d^{(2)}\frac{1}{T}\cdot T\omega_i^{(2)}\cdot\frac{1}{T}\cdot\frac{df(I^{(1)})}{dI^{(1)}} & (29) \\ &= \frac{1}{T}\sum d^{(2)}\omega_i^{(2)}\frac{df(I^{(1)})}{dI^{(1)}} & (30) \\ &= d^{(1)}/T. & (31) \end{aligned}$$

Therefore concerning the learning constant and momentum, we get the same relation as the case (1).

We do not describe the case of multi-layer NN at present, but the results is the same as this three layer case. We can describe the result as follows to summarize above two cases.

The rule that the equivalent NN of $T = 1$ works as same as the NN of $T \neq 1$ is ;

(1) to change the initial weights as
$\omega_{ij} = \omega_{Tij}/T$ or $\omega_i = \omega_{Ti}/T$,

(2) to change learning constant
$\alpha = \alpha_T/T^2$,

(3) to maintain the momentum to be the original value.

5.2 Simulation Results

To ascertain this rule to be true, we used seven bits parity discrimination NN taking the proper size of NN into consideration, in which we selected the number of the hidden units as fifteen for input signals of seven bits.

The solid line in Fig.6 shows the convergent rate of NN as vertical axis against temperature as horizontal axis. The dotted line shows the convergent rate when the learning coefficient α_T is changed into $T^2\alpha_T$ for each NN with temperature T's. As we can easily see that in every case the convergent rates agree well with a case of $T = 1$. In these cases, we set the initial weight region as $[-1 \sim 1]/T$.

In Fig.7, the solid line is the same as the solid line in Fig.6 and the dotted line shows the convergent rate when the learning constant is changed to α_T/T^2 in an equivalent circuit of $T = 1$. Both lines agree so well that it shows the rule we proposed is right.

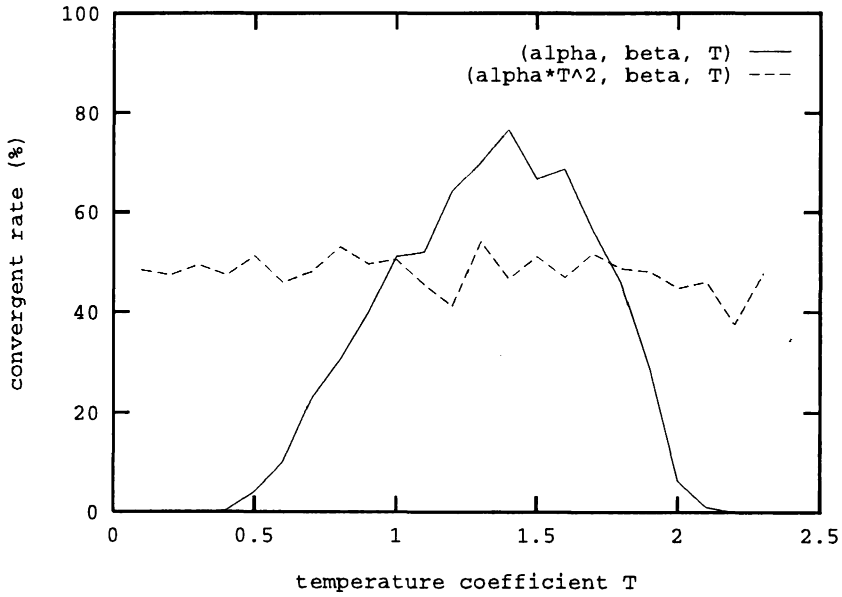

Fig.6 Relation between convergent rates and temperature and their converted relation

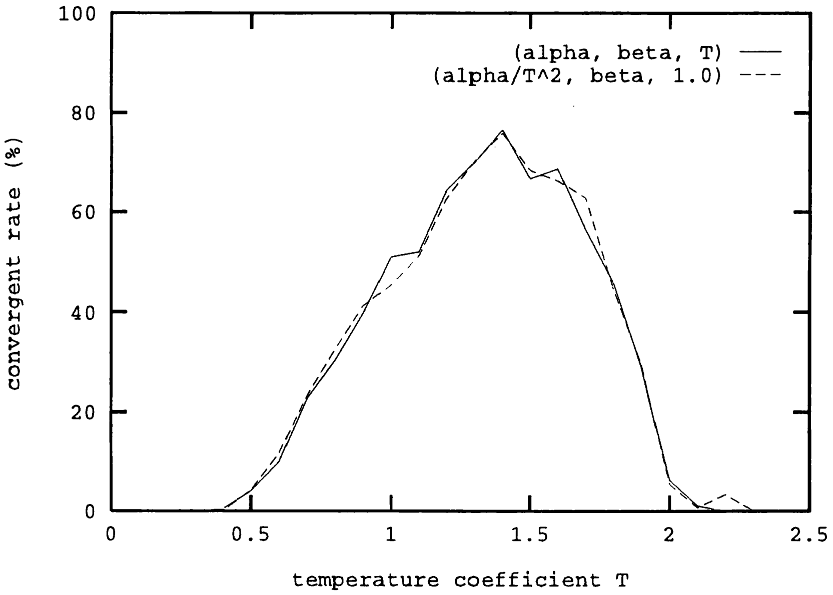

Fig.7 Convergent relations between original NN and equivalent NN

6 Conclusion

For the forward type three layer NN, we showed that there exists the most efficient number of the hidden units which is four times the minimum number of them, through parity discrimination. In addition to this assertion, for four layer NN under the condition of the total number of hidden units being fixed, we asserted that the convergent rate becomes higher if the numbers of hidden units are arranged in descending order. Next, we showed theoretically and experimentally that the back propagation learning of NN, when the temperature coefficients of its sigmoid functions for each unit have the same value, could be able to work equivalently as the equivalent NN of which temperature coefficient $T = 1$, by changing the weight coefficient ω_T to ω_T/T and the learning coefficient α_T to α_T/T^2, through seven bit parity discrimination problem.

A future subject is to obtain the most efficient number of hidden units for general problems, that is, to obtain the number of which is what times of the minimum number. We think it may be obtainable by using the principal value analysis.

Another subject is to obtain the changing rule of the learning constant on the way of the learning. Throughout these experiment we think that in NN the parity discrimination problem may play a rôle like benchmark test for von Neumann type computer.

Acknowledgment

We should like to express our thanks to a master course student Mr M.Hashimoto and an undergraduate student Mr T. Kanamaru for their great aids to execute the experiments of NN's.

References

[1] S.Kirkpatrick,C.D.Gelatt, and M.P.Becchi; Optimization by Simulated Annealing, Science 220, 671/680(1983)

[2] J.Okamoto and S.Hosokawa; On the Minimum Hidden Unit Numbers Necessary to Discriminate Input Parity, Trans. IEICE, **J77-DII**,9,1956/1959(1994)

[3] Ooyaen A.V. and Nienhais B.; Improving the convergence of the back-propagation algorithm, Neural Networks,**5**,3,463/471(1992)

[4] Okamoto J. and Kawamura Y.; On the Hidden Unit Numbers Necessary for 3 Bits Parity Discrimination, Trans. IEICE, **J76-DII**,8,2146/2147(1993)