ICROS-SICE International Joint Conference 2009
August 18-21, 2009, Fukuoka International Congress Center, Japan

Effect of number of hidden neurons on learning in large-scale layered neural networks

Katsunari Shibata (Oita Univ.; e-mail:shibata@cc.oita-u.ac.jp) and Yusuke Ikeda (Oita Univ.)

Abstract: In order to provide a guideline about the number of hidden neurons $N^{(h)}$ and learning rate η for large-scale neural networks from the viewpoint of stable learning, the authors try to formulate the boundary of stable learning roughly, and to adjust it to the actual learning results of random number mapping problems. It is confirmed in the simulation that the hidden-output connection weights become small as the number of hidden neurons becomes large, and also that the trade-off in the learning stability between input-hidden and hidden-output connections exists. Finally, two equations $N^{(h)} = \sqrt{N^{(i)}N^{(o)}}$ and $\eta = 32/\sqrt{N^{(i)}N^{(o)}}$ are roughly introduced where $N^{(i)}$ and $N^{(o)}$ are the number of input and output neurons respectively even though further adjustment is necessary for other problems or conditions.

Keywords: large-scale layered neural network, error back propagation, supervised learning, learning stability

1. Introduction

A neural network with error back propagation (BP) [1] has a strong supervised learning ability, and is very powerful when desired functions cannot be written easily by human hands. In the robotic field, activities in the real world should be considered increasingly hereafter, but the programs developed for such a complicated environment have not been working so effectively until now. For example, human's flexible image recognition is far superior to those developed by human hands. Furthermore, the development of higher-order functions has been far slower than expected. That might be because the human brain is massively parallel and cohesively flexible with huge sensor signals, and our sequential consciousness cannot perceive precisely what the brain is doing[2]. Therefore, the authors have advocated that a large-scale neural network as a parallel and flexible learning system is essential to develop human-like intelligence.

In general, many of the researches concerning about large-scale neural networks seems to focus on the hardware implementation such as [3]. In order to realize complicated functions using a neural network, the main stream seems to direct to "modularization" such as [4]. The authors think that non-uniform connections should be introduced, but closer connections between modules are required than in the present modularized neural network. The possibility of a non-modularized large-scale neural network as a parallel and flexible system in flexible image recognition can be seen in [2].

When considering the use of a large-scale neural network, the concern is that learning becomes more unstable as the number of inputs becomes larger. However, the authors noticed in reinforcement learning of a task with a layered neural network that learning was unexpectedly stable even with a large number of inputs[5]. In this paper, the authors try to formulize appropriate number of hidden neurons and learning rate roughly and to examine them in random number mapping problems. Of course, since learning depends deeply on a given task, that cannot be applied generally, but the authors expect it to be useful as a guideline for constructing a large-scale neural network.

2. Formulation

Since neural networks have a large variety, several assumptions are given as follows before starting the discussion to prevent divergence and to give a trigger.

(1) The neural network has three layers and no recurrent connections. Each neuron has connections from all the neurons in the previous layer below. There are no direct connections from input to output layer. Bias input is introduced in each neuron. In the followings, the upper suffixes (*i*)(*h*)(*o*) indicate input, hidden and output layer respectively.
(2) Regular Error Back Propagation (BP)[1] learning is used. No inertia term is introduced.
(3) The output function of each neuron except for input neurons is sigmoid function ranged from 0.0 to 1.0.
(4) Input and output patterns are generated by uniform random numbers that are independent of each other. The value range of input is from 0.0 to 1.0 and that of output (training signal) is from 0.1 to 0.9.
(5) The initial hidden-output connection weights $w^{(o)}$ are all 0.0. That means that all the outputs are initially 0.5. The connection weights $w^{(o)}$ and also the error signal $\delta^{(o)}$ in the output layer are assumed to have a symmetrical distribution with respect to the origin.
(6) Learning rate η is the same for all the connections.
(7) The increase of the computation time due to the increase of number of hidden neurons is ignored by expecting parallel computation in some future.

Here, as a rough standard for the stability of learning, the change of internal state (net value) Δu through learning for the same input patterns is focused on. The modification of the weight from the *i*-th input to the *j*-th hidden neuron can be written as

$$\Delta w_{ji}^{(h)} = \eta \delta_j^{(h)} x_i^{(i)} \qquad (1)$$

where η is a learning rate, $x_i^{(i)}$ is the output of the *i*-th input neuron that means *i*-th input from outside. $\delta_j^{(h)}$ is the propagated error signal for the *j*-th hidden neuron defined as $\delta_j^{(h)} = \frac{\partial E}{\partial u_j^{(h)}}$ where E is the squared error

PR0002/09/0000-5008 ¥400

defined as $E = \frac{1}{2}\sum_{k=1}^{N^{(o)}}\left(t_k - x_k^{(o)}\right)^2$. t_k and $x_k^{(o)}$ are the training signal and output for the k-th output neuron, and $N^{(o)}$ is the number of output neurons.

The change of the internal state can be written as

$$\Delta u_j^{(h)} = \sum_{i=1}^{N^{(i)}} \Delta w_{ji}^{(h)} x_i^{(i)} = \eta \delta_j^{(h)} \sum_{i=1}^{N^{(i)}} \left(x_i^{(i)}\right)^2. \qquad (2)$$

Because $\left(x_i^{(i)}\right)^2 \geq 0$ and $x_i^{(i)}$ is assumed to have the same probability distribution between neurons, the expectation of $\Delta u_j^{(h)}$ is proportional to the number of inputs $N^{(i)}$. If $\Delta u_j^{(h)}$ becomes large, the output of the neuron is completely different from the previous output value and that causes the instability of learning. This is the origin of the concern that learning becomes unstable when the neural network becomes large-scale. However, actually, since the next inputs are usually different from the previous inputs, the boundary of stable learning with respect to $N^{(i)}$ might be between $\sqrt{N^{(i)}}$ and $N^{(i)}$. $\sqrt{N^{(i)}}$ is derived from the standard deviation of the sum of $N^{(i)}$ independent 0-mean variables. In the followings, $N^{(i)}$ is used as a representative for the moment.

Here, the authors set up a hypothesis from their experiences that when the number of hidden neurons becomes large and redundant, the number of hidden neurons with similar response increases. Then the hypothesis is that the level of absolute value of hidden-output connection weights $\left|w^{(o)}\right|$ is inversely proportional to the number of hidden neurons $N^{(h)}$. The propagated error signal $\delta_j^{(h)}$ in the hidden layer is computed from the sum of the output error signal $\delta_k^{(o)}$ weighted by $w_{kj}^{(o)}$ as

$$\delta_j^{(h)} = f'(u_j^{(h)}) \sum_{k=1}^{N^{(o)}} w_{kj}^{(o)} \delta_k^{(o)} \qquad (3)$$

where $f()$ is the sigmoid function and $f'()$ is its derivative. Accordingly, $\delta_j^{(h)}$ is also inversely proportional to the number of hidden neurons $N^{(h)}$.

Finally, the effect of the number of output neurons is considered. The propagated error on a hidden neuron is computed as the above Eq. (3). Under the assumption that $f'(u_j^{(h)})$ and $\sum_{k=1}^{N^{(o)}} w_{kj}^{(o)} \delta_k^{(o)}$ are independent, the standard deviation of $\delta_j^{(h)}$is indicated as

$$\begin{aligned}\sigma_{\delta_j^{(h)}} &= \sqrt{\overline{\left(f'(u_j^{(h)}) \sum_{k=1}^{N^{(o)}} w_{kj}^{(o)} \delta_k^{(o)} - \overline{f'(u_j^{(h)}) \sum_{k=1}^{N^{(o)}} w_{kj}^{(o)} \delta_k^{(o)}}\right)^2}} \\ &= \sqrt{\left\{\overline{f'(u_j^{(h)})}^2 + var(f'(u_j^{(h)}))\right\} \overline{\left(\sum_{k=1}^{N^{(o)}} w_{kj}^{(o)} \delta_k^{(o)}\right)^2}} \qquad (4)\end{aligned}$$

where *var*() indicates the variance of a random variable, and the over-bar indicates the expectation. If it is assumed that the propagated error signals $w_{kj}^{(o)}\delta_k^{(o)}$ from different output neurons are completely independent of each other, the deviation becomes as

$$\begin{aligned}\sigma_{\delta_j^{(h)}} &= \sqrt{\left\{\overline{f'(u_j^{(h)})}^2 + var(f'(u_j^{(h)}))\right\} \sum_{k=1}^{N^{(o)}} \overline{\left(w_{kj}^{(o)} \delta_k^{(o)}\right)^2}} \\ &= \sqrt{\left\{\overline{f'(u_j^{(h)})}^2 + var(f'(u_j^{(h)}))\right\} N^{(o)} \overline{\left(w_{1j}^{(o)} \delta_1^{(o)}\right)^2}} \\ &= \sqrt{N^{(o)}} \sqrt{\left\{\overline{f'(u_j^{(h)})}^2 + var(f'(u_j^{(h)}))\right\} \overline{\left(w_{1j}^{(o)} \delta_1^{(o)}\right)^2}}, \qquad (5)\end{aligned}$$

while, if $w_{kj}^{(o)}\delta_k^{(o)}$ is completely dependent on each other, the deviation becomes as

$$\begin{aligned}\sigma_{\delta_j^{(h)}} &= \sqrt{\left\{\overline{f'(u_j^{(h)})}^2 + var(f'(u_j^{(h)}))\right\} N^{(o)2} \overline{\left(w_{1j}^{(o)} \delta_1^{(o)}\right)^2}} \\ &= N^{(o)} \sqrt{\left\{\overline{f'(u_j^{(h)})}^2 + var(f'(u_j^{(h)}))\right\} \overline{\left(w_{1j}^{(o)} \delta_1^{(o)}\right)^2}}. \qquad (6)\end{aligned}$$

Actually, $\Delta u_j^{(h)}$ would be proportional to the value between $\sqrt{N^{(o)}}$ and $N^{(o)}$. In the followings, $N^{(o)}$ is used as a representative. The boundary of stable learning of a hidden neuron is summarized as $\eta \frac{N^{(i)}}{N^{(h)}} N^{(o)}$ by adding the effect of learning rate η to the above discussions.

When the change of internal state in an output neuron $\Delta u_k^{(o)}$ is focused on, the error signal $\delta_k^{(o)}$ is not much influenced by the number of neurons because the propagated error in the output layer is directly calculated from the difference between the training signal and output in each neuron. Therefore, $\Delta u_k^{(o)}$ is thought to be proportional only to $N^{(h)}$ or $\sqrt{N^{(h)}}$.

Here, a trade-off is formed that if the number of hidden neurons becomes too large, the output neurons becomes unstable, and if the number of hidden neurons becomes too small, the hidden neurons becomes unstable again. Then an equation can be introduced to balance the trade-off as

$$\alpha\eta \frac{N^{(i)}}{N^{(h)}} N^{(o)} = \eta N^{(h)} \qquad (7)$$

where the right and left hand sides are derived from the viewpoint of the boundary of stable learning in hidden neurons and output neurons respectively, and α is a constant for consistency. The equation is rough and tentative, and so tuning is done after the following simulations. Now the tentative guideline to decide the number of hidden neurons is written as

$$N^{(h)} = \sqrt{\alpha N^{(i)} N^{(o)}}. \qquad (8)$$

The guideline for an appropriate learning rate η can be considered as follows. The left and right hand sides of Eq. (7) also influence the learning speed. Accordingly,

appropriate learning rate keeps each side of Eq. (7) constant that is the maximum value in the range of stable learning. The appropriate values of α and η must be deeply depending on the given task. Therefore, it is a good idea that the appropriate values for one condition are found through a simulation, and then appropriate values are guessed for the other conditions.

3. Simulation

In this section, learning of random input-output patterns is done and the results are used to examine the validity of and adjust the formulation introduced in the previous section. The followings are the setups in the simulation.

(1) The task is supervised learning of random input-output patterns. Each input is randomly chosen from 0.0 to 1.0 and the training signal for each output neuron is also chosen randomly from 0.1 to 0.9 to prevent the output from being in the saturation range of sigmoid function. The number of patterns is 20. The patterns are presented in a fixed order.

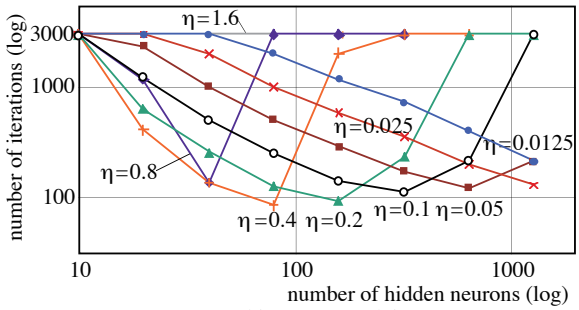
(2) The condition of successful learning is that the difference between output and training signal for all patterns and all outputs is less than 0.01.

(3) The initial hidden-output connection weights $w^{(o)}$ are all 0.0 as mentioned. In the preliminary simulations for the case of 1,000 input neurons, learning is the fastest when the initial weights are chosen randomly from -0.5 to 0.5. In order to keep the value level of hidden neurons constant, the value range of the initial input-hidden connection weights is inversely proportional to $\sqrt{N^{(i)}}$ with reference to the range from -0.5 to 0.5 at $N^{(i)}$=1,000. The expectation value of internal state of hidden neurons is 0.0 because the distribution of initial $w^{(h)}$ is symmetrical. The standard deviation is proportional to $\sqrt{N^{(i)}}$ according to the equation as

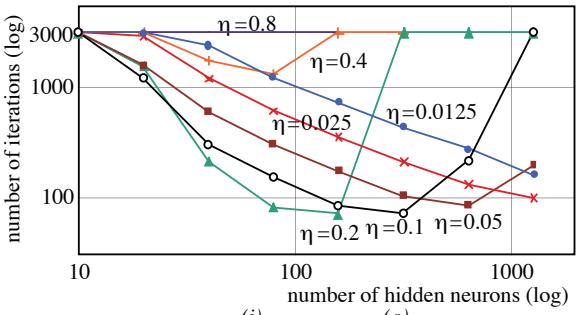
$$\sigma_{u^{(h)}} = \sqrt{\overline{\left(\sum_{i=1}^{N^{(i)}} w_{ji}^{(h)} x_i^{(i)}\right)^2}} = \sqrt{\overline{\sum_{i=1}^{N^{(i)}} \left(w_{ji}^{(h)} x_i^{(i)}\right)^2}} = \sqrt{N^{(i)}} \sqrt{\overline{\left(w_{11}^{(h)} x_1^{(i)}\right)^2}} \quad (9)$$

The number of input $N^{(i)}$, hidden $N^{(h)}$ and output neurons $N^{(o)}$ and also learning rate η are varied, and the average number of iterations over 20 simulation runs to reach the successful learning are observed, and it did not reach the success in 3,000 iterations, the number is set to 3,000.

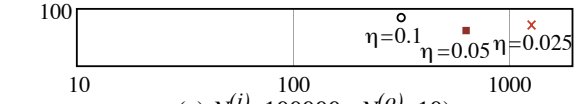
Figure 1 shows the results for 5 combinations of the number of inputs and outputs. By comparing the result between (a)(b)(c), the effect of number of inputs $N^{(i)}$ can be seen. While, by comparing the result between (a)(d)(e), the effect of number of outputs $N^{(o)}$ can be seen.



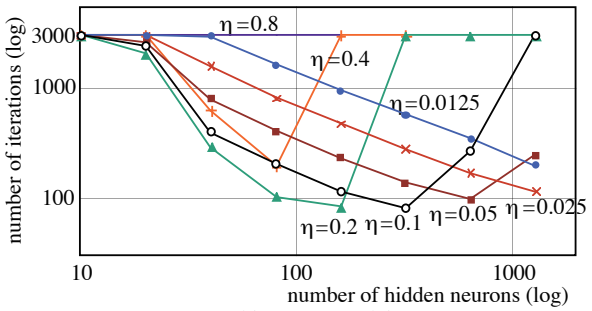
(a) $N^{(i)}$: 1,000, $N^{(o)}$: 10



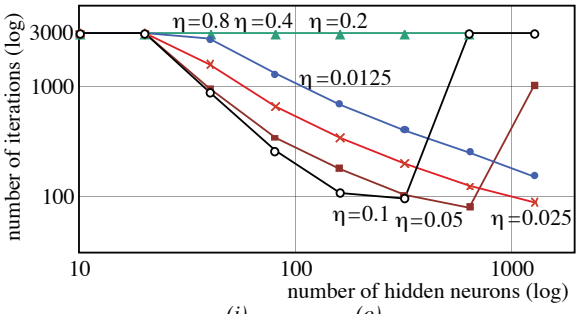
(b) ($N^{(i)}$: 10,000, $N^{(o)}$: 10



(c) $N^{(i)}$: 100000, $N^{(o)}$: 10)



(d) $N^{(i)}$: 1,000, $N^{(o)}$: 100



(e) $N^{(i)}$: 1,000, $N^{(o)}$: 1,000

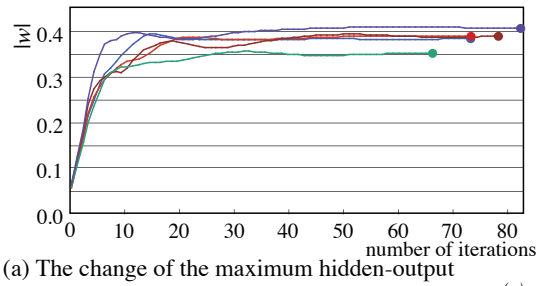
Fig. 1 The number of iterations to meet the condition of successful learning as a function of the number of hidden neurons $N^{(h)}$ for the 5 combinations of the number of inputs and outputs, $N^{(i)}$ and $N^{(o)}$. Each plot shows the average of 20 simulation runs with a different random sequence. When $N^{(i)}$=100,000 (Fig. (c)), only 3 points are plotted to save the time.

In all the graphs, we can see that when the number of hidden neurons is fixed, learning becomes faster in proportion to the learning rate at first, but when it becomes too large, the learning performance becomes worse suddenly. As is well known, that suggests the existence of appropriate step size in learning. When the learning rate is fixed, the downward slope of the lines is observed. It might be explained by the following two reasons. The increase of the number of hidden neurons $N^{(h)}$ causes the appearance of similar hidden neurons, and works as well as the increase of the learning rate. Actually, when the input-hidden weights for the first 320 hidden neurons are copied 3 times to the other 960 hidden neurons, the learning became faster than the case of 320 hidden neurons, and the speed is less than, but close to the case of 1280 hidden neurons. Furthermore, a large number of hidden neurons also cause a large possibility of existence of useful hidden neurons before learning and also a large repertoire of solutions. The optimal learning rate seems to depend only on the number of hidden neurons. It looks that when the number of hidden neurons becomes twice, the optimal learning rate becomes 1/2. That roughly matches the discussion in the previous section.

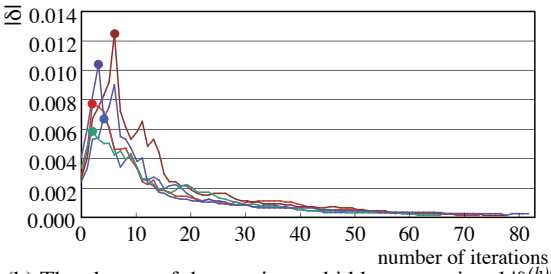
When the minimum number of iterations for each number of hidden neurons is observed, the difference is not so large, but the optimal number of hidden neurons looks to exist for each combination of $N^{(i)}$ and $N^{(o)}$. For example, in Fig. (a), the minimum number of iterations for the case of 80 hidden neurons is less than 100 and it is the minimum value when the number of hidden neurons is varied. It can be seen that according to the increase of the number of input neurons, the optimal number of hidden neurons becomes larger from around 80 in Fig. (a) to around 640 in Fig. (c). As for the increase of the number of output neurons, it also becomes larger from around 80 in Fig. (a) to around 640 in Fig. (e). Then, it is noticed that when the number of inputs is 10,000 (Fig. (b)), learning is so slow in the case of $N^{(h)}$=20 comparing with the case of $N^{(h)}$=160 even though learning converged successfully even in the case of $N^{(h)}$=20 for all the 20 simulation runs with η=0.1. This result matches that a large number of hidden neurons made learning stable in our previous simulation.

Figure 2 shows the change of the maximum absolute value of hidden-output connection weights $\left|w^{(o)}\right|$ and propagated error $\left|\delta^{(h)}\right|$ in the neural network for 5 simulation runs for the case of $N^{(i)}$=10,000, $N^{(h)}$=160, $N^{(o)}$=10 and η=0.2. In order to examine whether $\left|w^{(o)}\right|$ and $\left|\delta^{(h)}\right|$ decreased as the increase of number of hidden neurons, $\left|w^{(o)}\right|$ at the end of learning and the maximum $\left|\delta^{(h)}\right|$ during learning are observed in the followings.

Figure 3 shows the maximum absolute value of hidden-output connection weights $\left|w^{(o)}\right|$ after learning for 3 combinations of $N^{(i)}$ and $N^{(o)}$. It can be seen that the maximum connection weight decreases when the number of hidden neurons becomes large along a straight

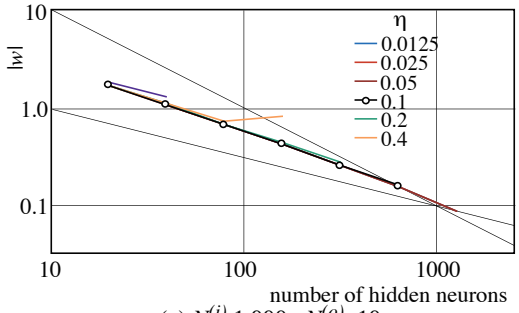


(a) The change of the maximum hidden-output connection weight $|w^{(o)}|$

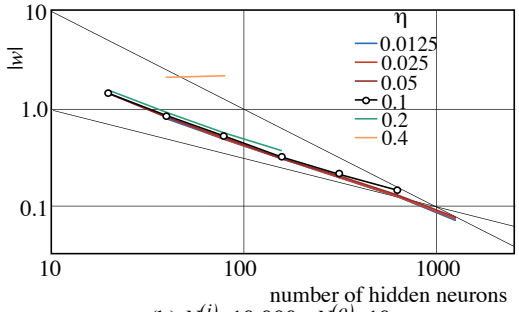


(b) The change of the maximum hidden error signal $|\delta^{(h)}|$

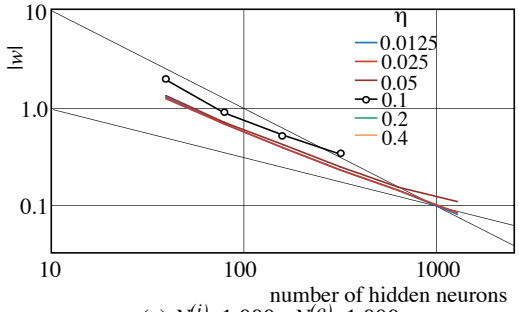
Fig. 2 The change of hidden-output connection weights and propagated error during learning.



(a) $N^{(i)}$ 1,000, $N^{(o)}$: 10



(b) $N^{(i)}$: 10,000, $N^{(o)}$: 10



(c) $N^{(i)}$: 1,000, $N^{(o)}$: 1,000

Fig. 3 The maximum hidden-output connections weights $\left|w^{(o)}\right|$ after learning as a function of the number of hidden neurons for 3 cases.

line. The value does not change so much depending on the number of input neurons $N^{(i)}$, output neurons $N^{(o)}$, or the learning rate η. From the slope of the line in Fig. 3, the relation of $|w^{(o)}|$ and $N^{(h)}$ is $|w^{(o)}| = 10/N^{(h)\frac{2}{3}}$.

Figure 4 shows the maximum absolute value of the propagated error $|\delta^{(h)}|$ during learning. It can be seen that $|\delta^{(h)}|$ also decreases as the increase of $N^{(h)}$ even though it is varied by the learning rate η. The slope is not so clear as the case of $|w^{(o)}|$ (Fig. 3), and looks depending on $N^{(i)}$ or $N^{(o)}$. When investigating the reason of the gentle slope in the case of $N^{(o)}$=1,000 (Fig. (d)) and η=0.025, the absolute value of correlation coefficient between $|w^{(o)}|$ and $|\delta^{(o)}|$ was large when $N^{(h)}$=1,280, but it was small when $N^{(h)}$=80. When $N^{(i)}$ changes, the value does not change so much by comparing (a) and (b), but when $N^{(o)}$ becomes 100 times, the value becomes more than 10 times by comparing (a) (c) and (d).

Finally, the trade-off between the stable learning of input-hidden and hidden-output is examined. Figure 5 shows the change of outputs of hidden and output neurons for one of the 20 input patterns for the three cases of number of hidden neurons when $N^{(i)}$=10,000, $N^{(o)}$=10 and η=0.2. In the case of $N^{(h)}$=20, the output of output neurons changes slowly, while that of hidden neurons changes fast and finally stay around 0.0 or 1.0 that is the saturation ranges of sigmoid function. In the case of $N^{(h)}$=160, the output of both kinds of neurons looks to change smoothly. In the case of $N^{(h)}$=320, the output of output neurons looks to oscillate hard. This represents that small number of hidden neurons causes the instability of input-hidden connections, while large number of hidden neurons causes the instability of hidden-output connections. In all the graphs, many of the hidden neurons are biased to 0.0 or 1.0. Some of them change its value depending on the presented patterns, but some of them do not change. The value range of initial input-hidden weights was decided with respect to the learning speed in the preliminary simulations, but the number of biased hidden outputs is larger than expected.

4. Tuning of Formulation

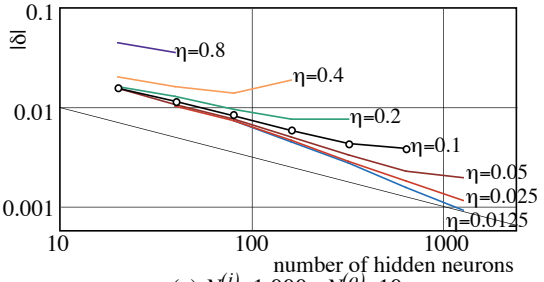
From Fig. 4 (b) or (c), the effect of number of hidden neurons on the size of hidden propagated error is set to $O(N^{(h)\frac{1}{2}})$ that is less than $O(N^{(h)})$ in the formulation. The reason can be thought that twice of the same hidden neurons does not lead to twice of learning speed and also that twice of hidden neurons does not generate twice of the similar hidden neurons.

The effect of $N^{(o)}$ is not so clear in the simulation, but $O(N^{(o)\frac{2}{3}})$ is used here by comparing Fig. 4 (a) and (d).

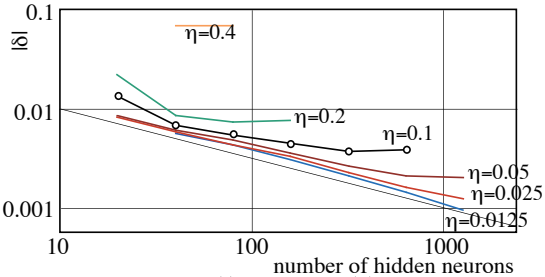
Since the effect of $N^{(i)}$ and $N^{(o)}$ is very similar from Fig. 1, the effect of $N^{(i)}$ is also set to $O(N^{(i)\frac{2}{3}})$. Furthermore, from Fig. 1, the appropriate learning rate is inversely proportional to the number of hidden neurons $N^{(h)}$, and the effect of $N^{(h)}$ on the learning stability of hidden-output connection is set to $O(N^{(h)})$. Then Eq. (7) and (8) can be rewritten as

$$\alpha N^{(i)\frac{2}{3}} N^{(o)\frac{2}{3}} / N^{(h)\frac{1}{2}} = N^{(h)} \quad (10)$$

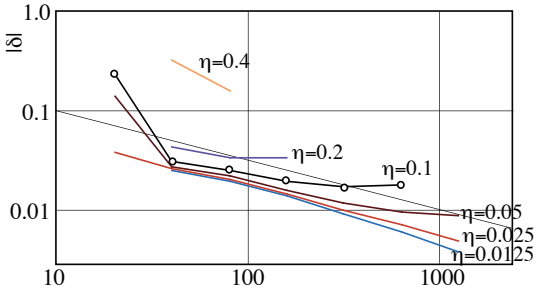
$$N^{(h)\frac{9}{8}} = \sqrt{\alpha N^{(i)} N^{(o)}}. \quad (11)$$



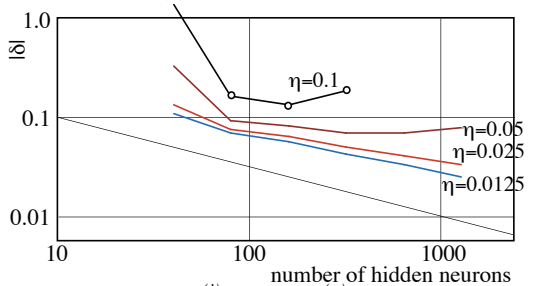
(a) $N^{(i)}$: 1,000, $N^{(o)}$: 10



(b) $N^{(i)}$: 10,000, $N^{(o)}$: 10



(c) $N^{(i)}$: 1.000, $N^{(o)}$: 100



(d) $N^{(i)}$: 1,000, $N^{(o)}$: 1,000

Fig. 4 The maximum propagated error $|\delta^{(o)}|$ during learning as a function of the number of hidden neurons. Please note that the scale of the vertical axis is different between upper and lower two graphs.

In the simulation, since the number of hidden neurons is doubled and doubled, and the number of the other neurons is decupled and decupled, the precision is not so high originally. Then, the exponent 9/8 in Eq. (11) is approximated roughly to 1. Since the optimal number of hidden neurons $N^{(h)}$ when $N^{(i)}$=1,000, $N^{(o)}$=1,000 is around 1,000, they are substituted in Eq. (11). Then, α=1 is derived and Eq. (8) becomes as simple as

$$N^{(h)} = \sqrt{N^{(i)}N^{(o)}}. \qquad (12)$$

As for the learning rate η, the optimal value is depending only on the number of hidden neurons, and the number of hidden neurons becomes twice, the optimal learning rate becomes 1/2. When $N^{(h)}$=320, the optimal η=0.1. The relation can be written as

$$\eta = \frac{32}{N^{(h)}} = \frac{32}{\sqrt{N^{(i)}N^{(o)}}} \qquad (13)$$

5. Conclusion

Learning stability in large-scale neural networks has been investigated. It was confirmed in the simulation of random pattern mapping problems that hidden-output connection weights become small when the number of hidden neurons becomes large, and also that the trade-off in the learning stability between input-hidden and hidden-output connections exists. With respect to the learning stability, a rough guideline of appropriate number of hidden neurons $N^{(h)}$ and learning rate η are introduced as $N^{(h)} = \sqrt{N^{(i)}N^{(o)}}$ and $\eta = 32/\sqrt{N^{(i)}N^{(o)}}$ where $N^{(i)}$ and $N^{(o)}$ are the number of input and output neurons respectively. Since the guideline was obtained based on the learning results of random pattern mapping problems on some conditions, it should be adjusted for other problems or other conditions.

Employing symmetrical output function and also employing different learning rate for each layer were not considered in this paper, but they might be promising from the discussions in this paper.

Reference

[1] D. E. Rumelhart, et al., "Parallel Distributed Processing", MIT Press, Vol.1, pp. 318-364, 1986

[2] K. Shibata and T. Kawano, "Acquisition of Flexible Image Recognition by Coupling of Reinforcement Learning and a Neural Network", *SICE JCMSI*, Vol. 2, No. 2, pp. 122-129, 2009

[3] N. Miyakawa, M. Ichikawa, and G. Matsumoto, "Large-scale neural network method for brain computing", *Applied Mathematics and Computation*, Vol.111, Issues 2-3, pp.203-208, 2000

[4] D. M. Wolpert and M. Kawato, "Multiple paired forward and inverse models for motor control", *Neural Networks*, Vol.11, pp.1317-1329, 1998

[5] K. Yuki and K. Shibata, The 23th SICE Kyushu-branch Conference, pp. 369-372, 2004 (in Japanese)

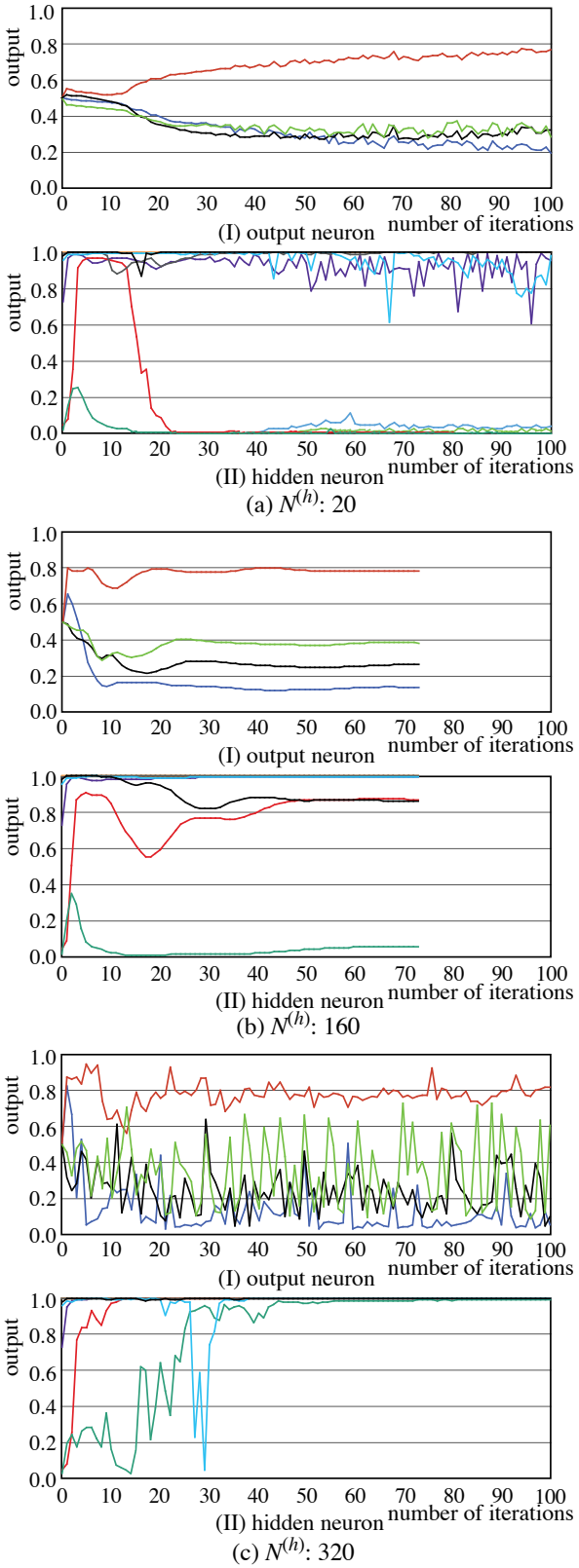


Fig. 5 The change of the outputs of 10 hidden and 4 output neurons during learning for the 3 cases of number of hidden neurons $N^{(h)}$, 20, 160 and 320, with $N^{(i)}$=10,000, $N^{(0)}$=10, η=0.2. Many of the 10 hidden neurons cannot be seen because they take the value around 0.0 or 1.0.