人工知能学会誌 Vol. 5 No. 5 (Sept. 1990)〔別刷〕

論文 Technical Paper

Computational Semantics of a Neural Network System for Thought Process Simulation and its Applications

Takeshi Chusho*

* Systems Development Laboratory, Hitachi Ltd., Kawasaki 215, Japan.

Received March 6, 1989

Keywords: neural network, thought process model, learning, computational semantics.

Summary

One major defect in neural network models in earlier research is nonclarity of computational semantics.

This paper describes a neural network system for thouth process simulation called "Repetition of Association and Concentration Model". Repetition of concept association is simulated by a neural network model with a feedback loop. However such a loop results in a stable state which is contradictory to the dynamics of the thought process. In this model, dynamics are maintained by nonlinearity to simulate consciousness, that is, thought energy distribution is enhanced to facilitate concept recollection.

Computational semantics of this model are defined as a parallel production system by correspondence of the synaptic connections between neurons to production rules. Three types of learning facilities are described as automatic knowledge acquisition. The relevance of this model is suggested by simulating the learning procedures for recognition of relations between concepts.

1. Introduction

For the past few years, considerable efforts have been spent on the improvement of neural network models and demonstrations of their applications [16]. A neural network system will be a breakthrough in computing technology because it will eliminate the defects of conventional computers, which are not suitable for pattern recognition and parallel computation.

The basic neural network idea was developed a few decades ago. In 1943, W. S. McCulloch and W. Pitts modeled a neural logic element with a linear threshold function based on neurophysiology [11]. In about 1960, F. Rosenblatt demonstrated that a neural network could recognize patterns by adjusting synaptic connections [13]. In 1961, E. R. Caianiello designed a neural network model with a self-organization facility [2]. In 1969, M. Minsky and S. Papert determined the ability and limitations of perceptrons with one layer [12]. In 1970, S. Amari provided an approach for a mathematical foundation of concept formation for self-organizing neural networks [1].

Around 1980, models based on psychological findings were proposed. For example, S. Grossberg provided an approach to pattern recognition with the focus on psychological stability and plasticity. He also developed an adaptive resonance theory, ART, based on both short-term memory and long-term memory [6]. In this model, the recognition ability of a particular pattern was reinforced by matching the input pattern with an expected pattern in the long-term memory. J. L. McClelland and D. E. Rumelhart also provided a pattern recognition model referred to as an interactive activation model based on findings that word knowledge helped in recognition of letters within words [10].

By the early 80's, some of the practical use problems had been solved from the engineering viewpoint. J. Hopfield showed that a simple model based on nonlinear neurons organized into networks with symmetric connections had the capacity to solve optimization problems [8]. D. Rumelhart developed an effective learning algorithm on a multilayer network, which was called an error back propagation algorithm [14]. T. Kohonen demonstrated the highly adaptive properties of neural networks, which allow a very accurate, nonlinear statistical analysis of real signals, such as speech recognition [9].

Around 1970, this author also researched neural networks. First, an electronic circuit with a learning facility was developed to imitate neuron activity [3]. This was composed of pnpn switch elements, registers and a constant current source. This circuit was described as a simple two-layer perceptron, and had the ability to learn boolean logic by using insulation break comparable to recent programable logic array circuits. Such a micro model of a neural network, however, had two defects : the ability was simple and only small circuits could be implemented. This latter problem has been solved by recent LSI technology. Next, a macro model of a neural network was developed for simulating the thought process based on neuron activity [4].

In earlier research, however, one major defect in neural network models encountered was unclear computational semantics. This problem still exists today. Although computational semantics of neural network elements have been clarified, the theoretical foundations of the dynamic features for a whole neural network system are still weak. For example, it is still not clear which learning algorithm is suitable for a particular application or in which order sample data should be given to the system for efficient and effective learning results. At present an experimental approach, rather than a theoretical approach, is being adopted. Clarification of computational semantics to construct neural network models should be the first step toward solving these problems.

Initially this paper defines the computational semantics of a neural network model, called a "Repetition of Association and Concentration Model", as a parallel production system with three correspondences : ① synaptic connection between neurons to a production rule, ② connectivity to a certainty factor of that rule, and ③ neuron activity to a certainty factor of a fact. The feedback loop for the RAC model is analogous to the recognize-act-cycle of a production system. A concentration function with nonlinearity is used instead of a conflict resolution strategy.

Second, learning facilities of the RAC model are presented. The learning ability of a neural network is the most important factor for applying it. The basic learning algorithm is based on Hebb's law [7]. Three types of variation are introduced, which correspond to dialogues with both a teacher and a dialogist as well as a monologue. These have different selection rules as to which synaptic connections between neurons should be reinforced. This learning procedure is described as automatic knowledge acquisition, and it is expected to reduce Feigenbaum's bottleneck, which states that it is difficult to extract knowledge from a domain expert when constructing an expert system.

Finally, the relevance of this model is suggested by simulating the learning procedure for recognition of relations between concepts.

2. A Neural Network Model for Thought Process Simulation

2・1 Thought Process Modeling

Research on the human thought process has been undertaken from various viewpoints, such as neurophysiology, neurobiology, psychology, cognitive science, artificial intelligence and knowledge engineering. In this paper, a thought process model is discussed for use in developing an artificial intelligence system. Since human thought ability has a close relation to natural language, it is desirable that the thought process model be made on the language level. However, the natural language processing technique is not far enough advanced. To overcome this difficulty, it is necessary to develop a model on a basic language concept level.

Memory, learning, association and concentration are considered essential factors to thought ability and are defined as follows :

Memory : The ability to recognize concepts.
Learning : The ability to recognize a relation between concepts.
Association : Recollection of a concept by using another related concept.
Concentration : Vivid recollection of one concept.

Based on these definitions, the thought process is assumed to be as follows :

(1) A thought process is described as a sequence of concepts successively recalled.
(2) Concept sequence comes from repetition of association and concentration.

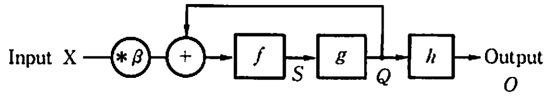
The basic model is constructed using an abstract idea referred to as thought energy.

① Concentration is the operation concentrating thought energy, which is distributed in a human brain. Sufficient concentration of thought energy causes concept recollection.
② Association is the operation diffusing concentrated thought energy. Sufficient thought energy diffusion causes recollection of another concept.
③ The thought process is a variation on thought energy distribution caused by the constant operations of ① and ②.
④ Learning is the operation facilitating the diffusion of thought energy in a particular direction. This operation corresponds to reinforcing connections between neurons in a neural network.

This model is called the Repetition of Association and Concentration Model (RAC).

2・2 *Mathematical Formulation of the RAC Model*

The RAC model is shown in Fig. 1. Assuming that the number of concepts is n, the symbols in



f: Diffusion function g: Concentration function
h: Linguistic function

Fig. 1 A repetition of association and concentration model for thought process simulation.

Fig. 1 represent the following :

X : an n component row vector of input. Each component has a value of 0 or 1. The j-th component $x[j]=1$ implies that words corresponding to the j-th concept are input.

O : an n component row vector of the output. Each component has a value of 0 or 1. The j-th component $o[j]=1$ implies that words corresponding to the j-th concept are output.

Q : an n component row vector of the state. Each component has a value range between 0 and 1. The j-th component $q[j]$ represents the degree to which the j-th concept is recalled in the RAC model. It is assumed that $\sum_j q[j]=1$, that is, thought energy is considered constant.

S : an n component row vector of the transient state introduced to facilitate sequential computation of f and g.

f : the diffusion function of thought energy, corresponding to association.

g : the concentration function of thought enegy.

h : the linguistic function corresponding to spoken language.

β : the degree of attention paid to an input with a non-negative value.

In this model, assuming that time is discrete for convenience, functions f, g, h are defined as follows :

(1) Diffusion function (f) This function is based on a classic neural network and is as

$$S=f(Q,X)=M(Q+\beta X),$$

where M is an n by n connection matrix. Each component $m[j,k]$ represents connectivity between the j-th and k-th concepts, that is, the k-th concept is associated with the j-th concept. The value of $m[j,k]$ is determined by a previous learning experiment, described later, and is between 0 and 1. It is assumed that $\sum_j m[j,k]=1$

to manage relative associativities from the k-th concept to n concepts. The initial state of M is a unit matrix which implies that each concept has already been committed to memory and that no relation between any concepts has yet been learned.

This definition of $f(Q, X)$ is derived from the assumption that thought energy distribution at time t is determined by the previous thought energy distribution and input at time $t-1$. Since the expression $(Q+\beta X)$ implies that each component of an input vector X is multiplied by attention degree β and added to the corresponding component of state vector Q, each component value of transient state S is obtained using the following equation :

$$s[j]=\sum_k m[j,k](q[k]+\beta x[k]).$$

(2) **Concentration function (g)** This function is a nonlinear function to enhance thought energy distribution by increasing the larger component values of state vector Q and decreasing the smaller component values. This operation is described as conscious concentration in the human brain, and avoids a state of no concept recollection due to excessive thought energy distribution.

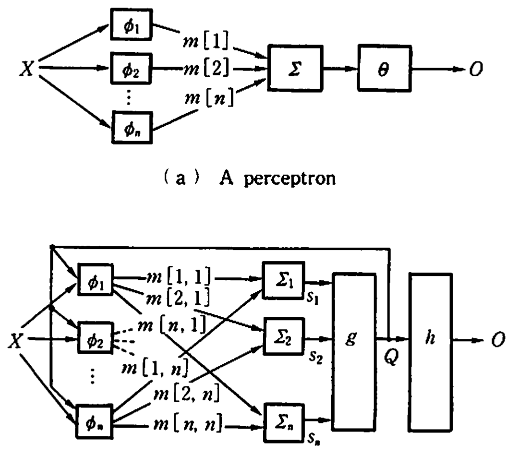
Although there are many functions with such characteristics, the power function is used for simplicity. Concentration function is described as

$$Q=g(S) \quad \text{where } q[j]=s[j]^p/\sum_k s[k]^p \text{ and } p>1,$$

where p is the degree of concentration bacause the larger the value of p, the stronger conscious concentration. The RAC model is considered to be a generalization of Caianiello's model [2] because it actually becomes Caianiello's model by using a threshold function as the concentration function. In other words, concentration operation is the main feature of the RAC model.

(3) **Linguistic function (h)** This function is the following threshold function which outputs only concepts with sufficient thought energy as words :

$$O=h(Q) \quad \text{where } o[j]=1 \text{ if } q[j]>\gamma \text{ and } o[j]=0 \text{ otherwise.}$$



(a) A perceptron

(b) A RAC model

Fig. 2 Comparison between a perceptron and a RAC model.

γ is a threshold value between 0 and 1.

2・3 Comparison with Perceptron

To clarify the intuitional semantics for the RAC model, it is compared with the classic neural network system perceptron shown in Fig. 2 (a) [12]. ϕ_j is a function that represents a sensory neuron transforming input X to an internal signal. Σ and Θ are functions representing internal neurons that discriminate a particular input from among the others. $m[j]$ is connectivity between the j-th sensory neuron and the internal neuron and it corresponds to a synaptic connection. The perceptron function is

$$O=1 \text{ if } \sum_j m[j]\,\phi_j(X)>\theta \text{ and}$$
$$O=0 \text{ otherwise.}$$

$\sum_j m[j]\,\phi_j(X)$ is linear with respect to $\{\phi_1,\cdots,\phi_n\}$.

The RAC model in Fig. 1 is represented as the perceptron-like figure in Fig. 2 (b). New symbols are defined as follows :

$$\phi_k: \phi_k(Q,X)=q[k]+\beta x[k].$$
$$\Sigma_j: s[j]=\Sigma_j(M,\phi_1,\cdots,\phi_n)=\sum_k m[j,k]\,\phi_k(Q,X).$$

There are four main differences between the RAC model and the perceptron :

(1) **Modeling levels** Perceptrons make decisions—determine whether or not an event fits a certain "pattern"—by adding up evidence obtained from many small experiments. They are considered to be a form of micro-level modeling, since the recognition process of a human brain is modeled on neurophysiology. The RAC model is considered to be a form of macro-level modeling, since the thought process is modeled on concept recollection.

(2) **Feedback loop** Although original perceptrons have no loops or feedback paths, a feedback loop is indispensable to the RAC model to simulate the thought process. There are many other neural network models with loops. In particular, the feature that the RAC model's internal activation state influences perceptual processing is similar to Grossberg's ART model and McClelland's interactive activation model mentioned in the introduction. However, the RAC model is based on associations between concepts whereas their models are based on pattern recognition.

(3) **Linearity vs. nonlinearity** Perceptron linearity, that is $\sum_j m[j]\ \phi_j(X)$, is convenient for mathematical analysis. For perceptrons with loops, iteration of linear calculation results in a stable state, which is contradictory to thought process dynamics. In this model, nonlinearity is introduced for simulation of conscious concentration in order to make the system dynamics. Nonlinearity is used to recall vividly a particular concept by enhancing thought energy distribution.

(4) **Initial learning state** The final results of learning in a perceptron are dependent on the initial connectivity state among the neurons, namely initial values of $\{m[1], \cdots, m[n]\}$. This is because its learning procedures are based on Hebb's Law that synaptic connections between neurons are reinforced when both source and destination neurons are active.

In the RAC model, this problem does not arise because the initial state of connection matrix M is a unit matrix which implies no connectivity between concepts. Human beings develop thought abilities through various dialogs. Their initial learning is performed through dialogs with parents in their infancy. It correspond to initial variation of connection matrx M.

3. Computational Semantics for the RAC Model

One main defect of a neural network model is unclear computational semantics. Computational semantics for the RAC model can be defined based on the computational semantics for a production system (PS) which is common knowledge engineering methodology. Regarding the RAC model as a parallel production system, the semantics are defined using comparison with a PS model as follows :

(1) **Cognition model** The repetition of association and concentration corresponds to the recognize-act-cycle of a PS model :

recognize $\Rightarrow$ act $\Rightarrow$ recognize $\Rightarrow$ act $\Rightarrow \cdots$

During operation of a PS model, an action implies often writing on an internal working memory rather than output and recognition implies often reading a working memory rather than input. Consequently, since recognition depends on previous recognitions via their actions, such a sequence of recognitions in a PS model can be described as a kind of thought process. This is the same as a sequence of concept recollections in the RAC model.

(2) **Current state** The current state of the PS model during execution is stored in the working memory. That of the RAC model is stored as a state vector which represents current thought energy distribution. The state vector components indicate the degree of recollection for each concept.

(3) **Knowledge base** In a PS model, knowledge is stored in a production memory. In the RAC model, knowledge is stored as a connection matrix.

(4) **Knowledge representation** In a PS model, knowledge is represented as a production rule in the "if~then~" form :

if condition (a) then action (b)

In the RAC model, knowledge is represented as connectivity between concepts in the form of $m[j, k]$, that is, a connection matrix component. Its logical meaning is interpreted as the following production rule :

if condition (concept k is recalled) then action (recall concept j)

In a general neural network model, connectivity between a source neuron k and a destination neuron j is interpreted as follows :

if condition (neuron k is activated) then action (activate neuron j)

(5) **Knowledge ambiguity** Knowledge ambiguity includes uncertainty, fuzziness, incompleteness, polymeaning and non-determinism. Uncertainty is a common ambiguity of knowledge in the form of production rules for a PS model. Therefore a certainty factor (CF) was introduced for the first time in MYCIN [15] and has since been used in many PS models.

For the rule "if a then b", CF (a, b) and CF (a) represent certainty factors of the rule and its condition, respectively. The value range of CF is between -1.0 and 1.0 where 1.0 implies that the rule is always true and -1.0 implies that the rule is always false.

In the RAC model, connectivity between concepts, $m\,[j, k]$, corresponds to CF (a, b) and is represented as CF (k, j). The degree of concept recollection, $q\,[k]$, coressponds to CF (a) and is represented as CF (k). However, CF (k, j) and CF (k) of the RAC model differ from CF (a, b) and CF (a) of the PS model as follows :

① The value ranges of CF (k,j) and CF (k) are between 0.0 and 1.0.

② $\sum_j \mathrm{CF}\,(k,j) = \sum_j m\,[j,k] = 1.$

③ $\sum_k \mathrm{CF}\,(k) = \sum_k q\,[k] = 1.$

It is possible to extend the lower limit of the value range ① to -1.0 by assuming that a negative value implies negation. In a general neural network model, negative connections between neurons imply inhibitory synapses based on neurophysiology.

The normalizations of ② and ③ are assumptions mentioned previously. CF (k) should be called a consciousness factor rather than a certainty factor because CF (k) implies a relative degree of concept recollection.

(6) **Ambiguity computation method** MYCIN gives the following certainty factor computation method :

① After the rule "if a then b" is executed, the certainty factor of action (b) is the product of CF (a, b) and CF (a) :

$$\mathrm{CF}\,(b) = \mathrm{CF}\,(a, b) \cdot \mathrm{CF}\,(a).$$

② After different rules with the same action (b) are executed, the final certainty factor for b is caluculated by using the products of the uncertainty factors $(1-\mathrm{CF})$:

$$\mathrm{CF}\,(b) = 1 - \prod_i (1-\mathrm{CF}_i\,(b))$$

where $\mathrm{CF}_i(b)$ is the non-negative certainty factor of b obtained by execution of the i-th rule.

If $i=2$ in ②, the equation is the familiar

$$\mathrm{CF}\,(b) = \mathrm{CF}_1(b) + \mathrm{CF}_2(b) - \mathrm{CF}_1(b) \cdot \mathrm{CF}_2(b).$$

In the RAC model, the corresponding computation method is defined in accordance with the previous mathematical formulation as follows :

ⓐ After the k-th concept is associated with the j-th concept via connectivity $m\,[j, k]$, which implies "if k then j", a consciousness factor for the j-th concept, $\mathrm{CF}_k(j)$, is the product of CF (k, j) and CF (k) as follows :

$$\mathrm{CF}_k(j) = \mathrm{CF}\,(k,j) \cdot \mathrm{CF}\,(k) = m\,[j,k] \cdot q\,[k].$$

ⓑ After ⓐ is calculated for all concepts associated with the j-th concept by using $\{m\,[j, 1], \cdots, m\,[j, n]\}$, the final consciousness factor for the j-th concept is the summation of results :

$$\mathrm{CF}\,(j) = \sum_k \mathrm{CF}_k(j) = \sum_k m\,[j,k] \cdot q\,[k].$$

Although ⓐ is the same as for MYCIN, ⓑ is different. This is because CF (j) is not a certainty factor but a consciousness factor and the association mechanism of the RAC model is similar to a synaptic connection based on neurophysiology.

(7) **Sequenciality vs. parallelism** In the PS model, if there is more than one executable rule in a recognize-act-cycle, only one rule is selected based on a conflict resolution strategy.

In the RAC model, parallelism is adopted instead of sequenciality because it is essential for a neural network system to compute in parallel. The product of a connection matrix and a state vector in the diffusion function, MQ, implies that all n^2 rules corresponing to n^2 connection matrix components are executed in parallel.

Concentration function g is a kind of a conflict resolution strategy for concept recollection because the function operates so as to find a concept with a concentration degree larger than threshold value γ of linguistic function h as fast as possible. This operation is similar to best-first search for search space reduction in problem solving technology, assuming that component values of transient state vector S correspond to evaluation criterion for best-first search.

(8) **Knowledge acquisition** In the PS model, such knowledge acquisition as rule addition, deletion and modification, certainty factor adjustment, and conflict resolution strategy selection are independent of inference engine implementation and knowledge representation. Knowledge acquisition through the building of an expert system is called Feigenbaum's bottleneck, which states that it is difficult to extract expertise from a domain expert. A lot of research projects have already contributed toward automatic knowledge acquisition, but results are insufficient for practical application.

In the RAC model, however, automatic knowledge acquisition is realized by automatic modification of connection matrix component values based on learning facilities, as in other neural network models. The self-organization mechanism of the RAC model using learning procedures is described in the next chapter.

4. Learning Facilities

4・1 Learning Procedures

The most attractive feature of neural networks is their self-organization ability based on learning. For the RAC model, learning means to be capable of associating one concept with another by adjusting connection matrix component values. The RAC model provides the following three types of learning procedures:

(1) learning-by-teacher : $m[j, k] \Rightarrow m[j, k] + \delta_1$ if $x[k] = 1$ at t and $x[j] = 1$ at $t+1$.

(2) learning-by-dialog : $m[j, k] \Rightarrow m[j, k] + \delta_2$ if $o[k] = 1$ at t and $x[j] = 1$ at $t+1$.

(3) learning-by-monolog : $m[j, k] \Rightarrow m[j, k] + \delta_3$ if $o[k] = 1$ at t and $o[j] = 1$ at $t+1$.

Here, t is time and δ_i is variation in connectivity. After $m[j, k]$ increases by δ_i, the k-th column of M is normalized as $\sum_j m[j, k] = 1$.

Learning-by-teacher is used for directly memorizing a sequence of inputs. Learning-by-dialog is used for obtaining knowledge from a dialogist. Learning-by-monolog is used for strengthening one's own knowledge and is similar to the common learning procedure for a neural network.

4・2 Measurement of the Learning Results

This section introduces the following three measurements of the learning results. The first is measured by outside observation. The others are based on analysis of the internal state of a model represented by M.

(1) **Same-answer rate** This measurement indicates the similarity between two models, which is defined as follows:

Same-answer rate:

A = the number of same answers in the test cases/the number of test cases

(2) **Degree of learning** This measurement indicates the ability to associate one concept with another and is defined as

$$\text{Learning degree}: L = \sqrt{\sum_{j \neq k} m[j, k]^2}.$$

L is 0 in the initial state of the connection matrix, since all $m[j, k]$ where $j \neq k$ are 0. L becomes $\sqrt{n}$ of the maximum when every connection matrix column has a component with 1, except for the diagonal components.

(3) **Degree of difference** This measurement indicates the difference between two models as defined by the following distance function on M:

Difference degree:

$$D = \sqrt{\sum_{j, k} (m_1[j, k] - m_2[j, k])^2}$$

where m_1 and m_2 are the connection matrices for two models. D is 0 if M_1 is equal to M_2. D becomes $\sqrt{2n}$ in the worst possible case.

5. RAC Model Applications

RAC model learning facility features are confirmed by the following simulation programs :

5・1 Learning-by-teacher

This learning procedure is used for teaching known knowledge to a system from the outside. Since changing connectivity depends only on inputs and not on the state or outputs of the system, system behavior is easily adjusted.

5・2 Learning-by-dialog

This learning procedure is used by a system to obtain knowledge through dialog. Since changing connectivity depends on both the input and output of the system, learning results are influenced by the state of M at the beginning of learning and by such parameters as attention degree β of the diffusion function, concentration degree p of the concentration function, threshold value γ of the linguistic function, and variation δ in connectivity. This paper supposes standard parameters to be

$p=2.0$

$\beta=1.0$

$\gamma=0.5$

$\delta=0.02$

The number of concepts is supposed to be ten, therefore,

$n=10$

Two types of learning by dialog are simulated using two RAC models coupled to each other as shown in Fig. 3. One is learning by question and answer. The other is learning through debate.

(1) **Learning by question and answer** Of the two models in Fig. 3, it is assumed that one is a student, with a learning-by-dialog facility, and the other is a teacher, with no learning facility. That is, the student acquires teacher's knowledge by question and answer with the teacher. A part of the simulation results are shown in Fig. 4. At the beginning of these experiments, component values of M for the teacher are set as $L=\sqrt{10}$ since it is assumed that the teacher has already finished learning. Component values of M for the student are set as $L=1.66$ since it is assumed that the student has already learned a little. If it is assumed that the student has not learned anything like a baby and then a unit matrix is used as M, learning results become similar to results obtained by using the procedure of learning-by-teacher. In Fig. 4, (a) is the standard parameter results and (b) to (f) are results obtained by changing one different parameter from among the standard parameters.

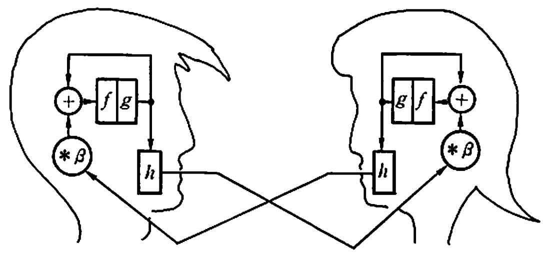


Fig. 3 Simulation of learning by dialog using two RAC models.

The x-axis represents the number of discussion rounds, r. One round implies that ten fg-cycles are executed for each one of ten questions, that is, a total of 100 fg-cycles. The y-axis represents the results of learning by A, L and D, as defined in the previous chapter.

The following features are confirmed by simulation :

① The larger the concentration degree p, the better the learning results.

② The greater or lesser the attention degree is from 1, the better the learning results.

③ The smaller the threshold value γ, the better the learning results.

④ The greater the change in varient δ for connectivity, the better the learning results.

These reasonable results are obtained through varying values of standard parameters in the range shown in Fig. 4, although trivial results are obtained through replacing values of standard parameters by extreme parameter values. In ②, $\beta=1$ implies that the input of the diffusion function becomes $Q+X$ and then enhances diffusion, so that output decreases, since in most cases Q and X are different. In ③, a large thresh-

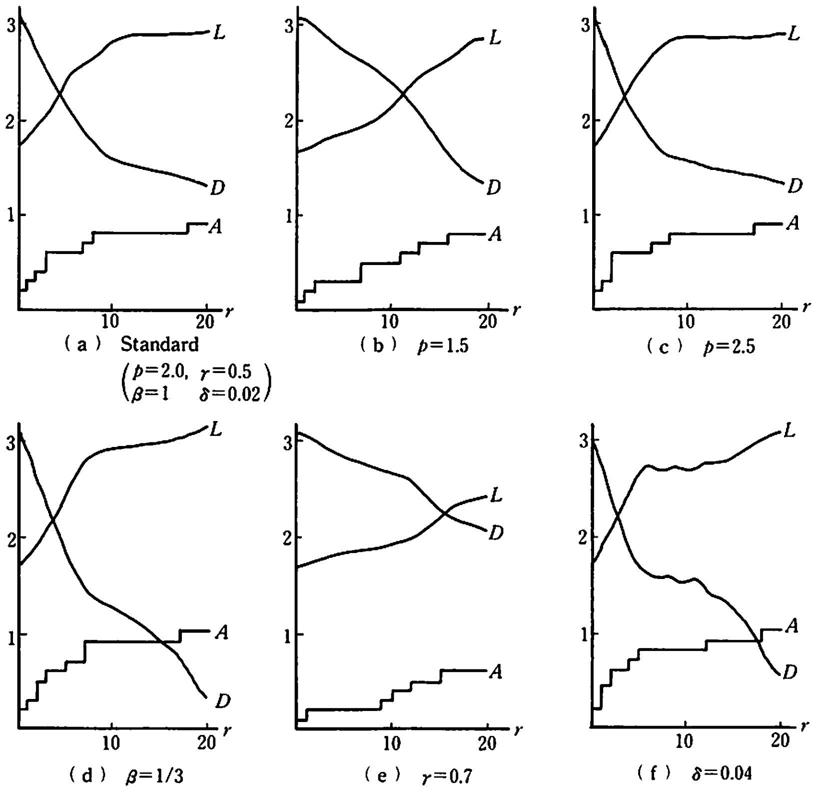


Fig. 4 Simulation results of learning by question and answer.

old value for γ inhibits learning-by-dialog, since output decreases.

(2) **Learning through debate** Both models in Fig. 3 have the same learning procedures. Learning-by-monolog is applied to one model if the other model does not answer the former's question. Learning-by-teacher is applied to the former if the former does not answer the latter's question. In this procedure, learning-by-monolog is used for reinforcing one's own knowledge. Learning-by-teacher is used for acquiring the debater's knowledge.

Two simulation results with variations in the initial degree of difference are shown in Fig. 5. The following features are confirmed by these results :

(1) If difference degree D between two models is large, the one with the lesser degree of learning learns from and comes equal to the one having the greater degree of learning.

(2) If difference degree D is small, the number of same answers increases rapidly during learning.

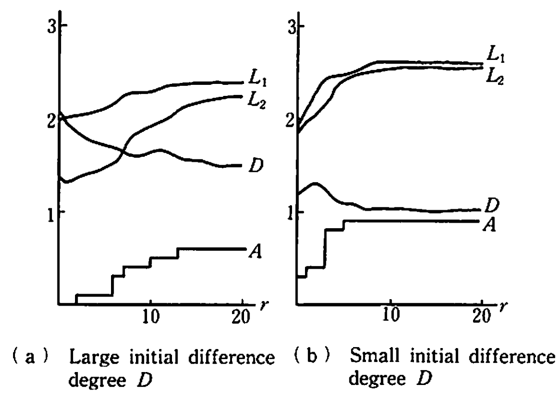


Fig. 5 Simulation results of learning through debate.

6. Conclusion

A neural network system for thought process simulation which is referred to as a repetition of association and concentration model is proposed. Associations between concepts were simulated by connectivity between the neurons. Furthermore, actual conscious concentration was simulated by adding nonlinearity to a neural network model. Computational semantics for this model are defined as a parallel production system based on three correspondences : ① synaptic connection between neurons to a production rule, ② connectivity to a certainty factor of that rule, and ③ activity of the neuron to a certainty factor of a fact.

Three types of learning facilities are described as automatic knowledge acquisition. The relevance of this model was confirmed by computer simulation of learning procedures for recognition of relations between concepts. Further studies are suggested on the stable state analysis of this model and the convergence problems of learning procedures.

Acknowledgements

The author wishes to express his gratitude to Prof. Masao Saito of Tokyo University for his valuable suggestions and advice. He is also indebted to Nobuyoshi Dohmen and Koichirou Ishihara for providing the opportunity to conduct this study.

◇ References ◇

(1) Amari, S. : *Mathematical Foundation of Neural Network* (Japanese), Sangyo-tosho, Tokyo (1978).

(2) Caianiello, E. R. : Outline of a Theory of Thought-process and Thinking Machine, *J. Theoretical Biology*, Vol. 2, pp. 204-235 (1961).

(3) Chusho, T. and Saito, M. : Circuits with Learning Facilities Equal to Conditioned Reflex (Japanese), *Annual Conf. of IECEJ*, p. 242 (1969).

(4) Chusho, T. and Saito, M. : Learning by Discussion Using a Simple Thought Process Model (Japanese), *Annual Conf. of IECEJ*, p. 231 (1970).

(5) Chusho, T., Masui, S., Haga, H. and Yoshiura, H. : Amalgamation of Conflicting Concepts for Knowledge Processing Language Supporting Multiparadigms (Japanese), *J. of Japanese Society for Artif. Intell.*, Vol. 4, No. 1, pp. 77-87 (1989).

(6) Grossberg, S. : How Does a Brain Build a Cognitive Code?, *Psychological Review*, Vol. 87, No. 1, pp. 1-51 (1980).

(7) Hebb, D. O. : *The Organization of Behavior*, Wiley, N. Y. (1949).

(8) Hopfield, J. J. and Tank, D. W. : Computing with Neural Circuits : a Model, *Science*, Vol. 233, pp. 625-633 (1986).

(9) Kohonen, T. : The "Neural" Phonetic Typewriter, *Computer*, Vol. 21, No. 3, pp. 11-22 (1988).

(10) McClelland, J. L. and Rumelhart, D. E. : An Interactive Activation Model of Context Effects in Letter Perception : Part 1. An Account of Basic Findings, *Psychological Review*, Vol. 88, No. 5, pp. 375-407 (1981).

(11) McCulloch, W. S. and Pitts, W. : A Logical Calculus of the Ideas Immanent in Neural Net, *Bullentin of Mathematical Biophysics*, Vol. 5, pp. 115-137 (1943).

(12) Minsky, M. and Papert, S. : *Perceptrons*, MIT Press, Cambridge, Mass. (1969).

(13) Rosenblatt, F. : *Principles of Neurodynamics*, Spartan, Washington, DC (1961).

(14) Rumelhart, D. E., Hinton, G. E. and Williams, R. J. : Learning Representations by Back-propagating Errors, *Nature*, Vol. 323, pp. 533-536 (1986).

(15) Shortliffe, E. H., *Computer-based Medical Consultations : MYCIN*, American Elsevier (1976).

(16) Wasserman, P. D. and Schwartz, T. : Neural Networks, Part 1 and Part 2, *IEEE EXPERT*, Vol. 2, No. 4, pp. 10-14 (1987), and Vol. 3, No. 1, pp. 10-15 (1988).

〔担当編集委員：中野　良平，査読者：河原　英紀〕

Author's Profile

Takeshi Chusho (Member)

Takeshi Chusho received the B. S. and M. S. degrees in Electronic Engineering from the University of Tokyo, in 1969 and 1971, respectively, and Ph. D. degree in Computer Science from the University of Tokyo in 1984. He is a Senior Researcher of Systems Development Laboratory, Hitachi Ltd. Kawasaki, Japan and a Lecturer on artificial intelligence at Tokyo Institute of Tokyo, Tokyo, Japan. Dr. Chusho received the 1982 Best Paper Award from Information Processing Society of Japan and the 1987 Okochi Memorial Technology Prize. His current research interests include knowledge Engineering and Software Engineering. He is the co-author of two books : "Artificial Intelligence" and "Programming Tools" (Shokodo). He is a member of editorial board of the Institute of Electronics Information and Communication Engineers of Japan and editorial board of Japan Society for Software Science and Technology. He is a member of the IEEE Computer Society.