International Journal of Neural Systems, Vol. 13, No. 3 (2003) 205–213

World Scientific
www.worldscientific.com

A HYBRID NEURAL NETWORK OF ADDRESSABLE AND CONTENT-ADDRESSABLE MEMORY

JINWEN MA
Department of Information Science, School of Mathematical Sciences, and LMAM, Peking University, Beijing, 100871, China
jwma@math.pku.edu.cn

Received 26 July 2001
Revised 26 March 2003
Accepted 10 May 2003

We investigate the memory structure and retrieval of the brain and propose a hybrid neural network of addressable and content-addressable memory which is a special database model and can memorize and retrieve any piece of information (a binary pattern) both addressably and content-addressably. The architecture of this hybrid neural network is hierarchical and takes the form of a tree of slabs which consist of binary neurons with the same array. Simplex memory neural networks are considered as the slabs of basic memory units, being distributed on the terminal vertexes of the tree. It is shown by theoretical analysis that the hybrid neural network is able to be constructed with Hebbian and competitive learning rules, and some other important characteristics of its learning and memory behavior are also consistent with those of the brain. Moreover, we demonstrate the hybrid neural network on a set of ten binary numeral patters.

Keywords: Forward neural network; addressable memory; content-addressable memory; associative memory; Hebbian learning; competitive learning.

1. Introduction

The key to understanding and modelling human intelligence is how information is represented and stored in memory. While there has been a great deal of work on the structure of memory,[1] it is clear that definitive answers have not been found yet. However, it has been shown by many psychological experiments that information items in memory are both addressable and content-addressable. In fact, as one recognizes his friend from a photo, or even a part of it, it is certain that the figure of his friend has been memorized as well as retrieved content-addressably in his brain. Content-addressable memory is a typical kind of associative memory in the brain and there have been several models to describe it (e.g., Refs. 2 and 3). On the other hand, temporally modulated information-processing and sequential behavior are also essential to our everyday functioning, such as in interpreting auditory stimuli and in using language. So the information items in memory should be also in serial order.[4,5] Since there is a huge amount of information in memory and the speed of information transmission is limited in the brain, a sequential information item can be only retrieved from some cue which functions as the address of the information item in memory. Therefore, information items should be addressable in memory. Although a lot of evidence has been found to support the argument that the memory in the brain is both addressable and content-addressable, there has not yet any reasonable theory or model which can unify the characteristics of addressable and content-addressable memory, to our best knowledge.

With the development of neurobiology and anatomy, researchers have been trying to investigate the learning and memory mechanisms structurally on the basis of the neural cells (neurons). In this way, the first breakthrough was the M-P model which mathematically describes the function or behavior of a neuron.[6] Then, Hebb proposed a learning supposal that the strength of efficiency of the synapse between two neurons will increase when they both activate at the same time, which is now known as Hebbian learning rule.[7] Anderson further proposed a simple neural network which can generate an interactive memory.[8] Moreover, several neural network models of content-addressable or associative memory were established(e.g., Refs. 9–11). Especially, Hopfield proposed a recurrent neural network which was late referred to as Hopfield network.[12] Under the learning scheme of sum-of-outer product, a group of sample patterns can be stored in the Hopfield network with function of associative memory. It has been recently shown that the asymmetric or generalized Hopfield network has a similar behavior of associative memory as Hopfield network.[13–15]

Based on the psychological fact that the memory is stored in the brain by groups individually, the author has proposed a biological neural network, called simplex memory neural network(SMNN),[16] which can learn and memorize (store) a single binary pattern with content-addressable memory function. Moreover, it can be constructed in the brain with Hebbian learning rule. In certain sense, this network can be considered as a basic unit of the memory in the brain. In the light of this idea, we now assume that each item of information (as a binary pattern) is stored in its SMNN in the brain. Then, the memory of the brain will be considered as a SMNN bank. One SMNN may have connection with other related SMNNs in the SMNN bank, that is, the retrieval of one pattern may cause the retrieval of the other related patterns as a hetero-associative memory. In such a way, a number of SMNNs may connect together in a certain order to learn and memorize a spatio-temporal pattern sequence. Certainly, the information association among these SMNNs is very important to thinking, inference, decision, etc., however, for simplicity, we will neglect these hetero-associative memory connections among these SMNNs in the current paper.

As each pattern is stored in its SMNN in a content-addressable way and it can be retrieved only from its own SMNN but the other SMNNs, one simple mechanism of the retrieval of a pattern on the SMNN bank is to transmit the input pattern to all the SMNNs in a full parallel way. Certainly, the corresponding pattern can be retrieved from some SMNN. If this mechanism is true, all neurons in the brain will work together when we see or hear something. But this is inconsistent with the biological finding that the neurons are active only in some local region of the brain when we see or hear something. Moreover, it is also inconsistent with the result of psychological experiments that the memory is retrieved both addressably and content-addressably. How does the brain retrieve a piece of information (pattern) in the SMNN bank addressably? The classification system of books in a library gives us a clue. We can imagine that the information is memorized into classes, subclasses, etc., in the brain. In fact, when we get some new information, we always think how it relates to the other memorized information. In this way, the new information will be memorized by its class, subclass, etc., as well as its content. When we retrieve an item of information from the input, we may search its class, subclass, etc., and finally retrieve it by a certain SMNN in the SMNN bank. Therefore, the membership of the class, subclass, etc. of the information may serve as an address for the information. With such an address, we may retrieve the information quickly and correctly. In the light of this idea, we will propose a hybrid neural network of addressable and content-addressable memory on the SMNN bank by which an item of information (as a binary pattern) is memorized and retrieved both addressably and content-addressably. Consequently, this hybrid model makes it possible that the content-addressable memory is compatible with the addressable memory.

In sequel, we propose the hybrid neural network of addressable and content-addressable memory in Sec. 2. We further discuss the learning rules of the hybrid neural network in Sec. 3. In Sec. 4, a simple simulation experiment is conducted to demonstrate our proposed network. Finally, we give a brief conclusion in Sec. 5.

2. The Model of the Hybrid Neural Network

We begin with a brief description of simplex memory neural network(SMNN).[16] Generally, an SMNN consists of n connected binary or M-P neurons defined by a weight matrix $\mathbf{W} = (w_{ij})_{n\times n}$ and a threshold vector θ with certain content-addressable memory function. The SMNN for a binary pattern $U = [u_1, u_2, \ldots, u_n]^T \in \{0,1\}^n$ with an error-correcting capacity t can be constructed by

$$w_{ij} = \begin{cases} 1 & \text{if } u_i + u_j = 2, i \neq j \\ -1 & \text{if } u_i + u_j = 1, i \neq j \\ 0 & \text{if } u_i + u_j = 0, i \neq j \\ 0 & \text{if } i = j \end{cases}, \quad \theta_i = d_H(U) - (t+1) \tag{1}$$

for all $i, j = 1, 2, \ldots, n$, where $d_H(U)$ $(= \sum_{i=1}^{n} u_i)$ is the Hamming weight of U. Clearly, it is a Hopfield network of n binary neurons. For clarity, we let $d_H(X, U) = \sum_{i=1}^{n} |x_i - u_i|$ be the Hamming distance between X and U. Indeed, such constructed SMNN has the retrieval characteristic that when the input pattern $X = [x_1, x_2, \ldots, x_n]^T \in \{0,1\}^n$, satisfies $d_H(X, U) \leq t$, i.e., the number of errors appearing on the bits of the input pattern in contrast to U is no more than t, the network will be stable at U (U is retrieved); otherwise, the network will be stable at 0 (U is not retrieved and the network is in a quiescent state). Clearly, the pattern U is stored in the SMNN in a distributable and content-addressable way. It is also shown by theoretical analysis in Ref. 16 that an SMNN for a pattern can be constructed with Hebbian learning rule.

We now propose our hybrid neural network of addressable and content-addressable memory on the SMNN bank. Structurally, it can be described as the graph of a tree of slabs. That is, each vertex of the tree is a slab which consists of n binary neurons with the same array. The root vertex is just the unique input slab of our network. The terminal vertexes are slabs of SMNNs in the SMNN bank, i.e., the SMNNs are considered as the slabs of basic memory units, being distributed on the terminal vertexes of the tree. There are a number of intermediate or hidden slabs between the input slab and these SMNN slabs, which will be used for information transmission and processing. The neurons of any slab but an SMNN, have no interconnections, but they are connected to the corresponding neurons of each of connected slabs to transmit the information forward, and also dominate the communication through it in a certain way.

More precisely, the network can be further described as a multi-layer forward neural network. For the sake of clarity, we assume that the network consists of l layers and the ith layer has m_i slabs. The first layer is just a single input slab (or the root vertex in the tree), i.e., $m_1 = 1$. It receives the information, i.e., a binary pattern, from the outside environment and transmits this information to the slabs on the second layer. The m_2 slabs on the second layer divide all the memory units into m_2 classes. Here, a slab actually represents a class of the memory. The m_3 slabs on the third layer further divide all the memory units into m_3 subclasses belonging to m_2 classes. A slab on the second layer transmits the information only to the slabs on the third layer which represent the subclasses belonging to the class of this slab. In this way, each slab except an SMNN or the input slab has one original slab on the preceding (left) layer and a number of generative slabs on the next (right) layer. So it transmits the information only to its generative slabs. However, such a slab should have an additional dominant neuron which contains the information of the class it represents. When the information is transmitted from the original slab to this slab, the dominant neuron also receives this information and matches it with the class information. If the input information belongs to the class, the dominant neuron spreads a positive signal to let the information pass through it to the generative slabs. Otherwise, the dominant neuron spreads an inhibitory signal (or nothing) to prevent the information from passing through it to the generative slabs. That is, the information stops and disappears at this slab. Since the memory units may have different number of the classifications, SMNN slabs may appear on the hidden layers as the terminal vertexes of the tree.

As shown in Fig. 1, the architecture sketch of a simple hybrid neural network is given. It is a three-layer forward neural network with eight slabs. Obviously, it can be considered as the graph of a tree

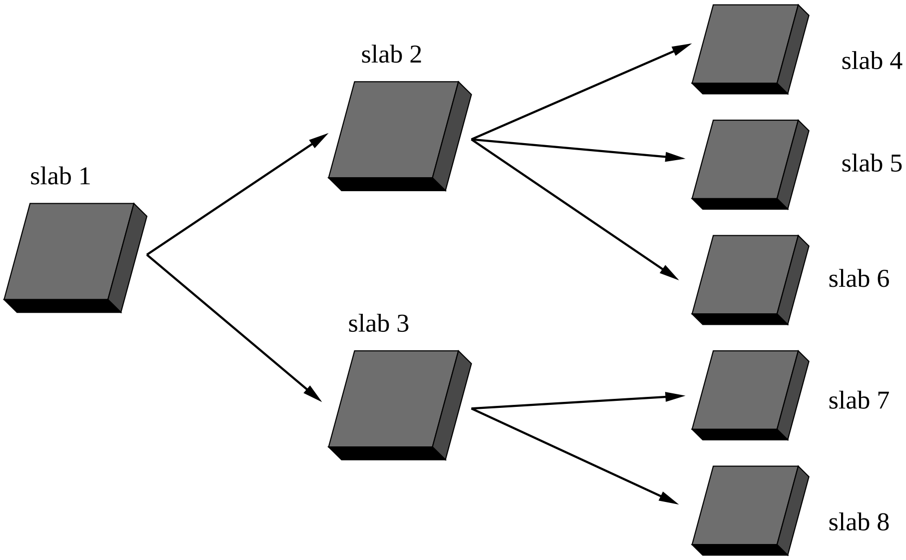

Fig. 1. The architecture sketch of a simple hybrid neural network.

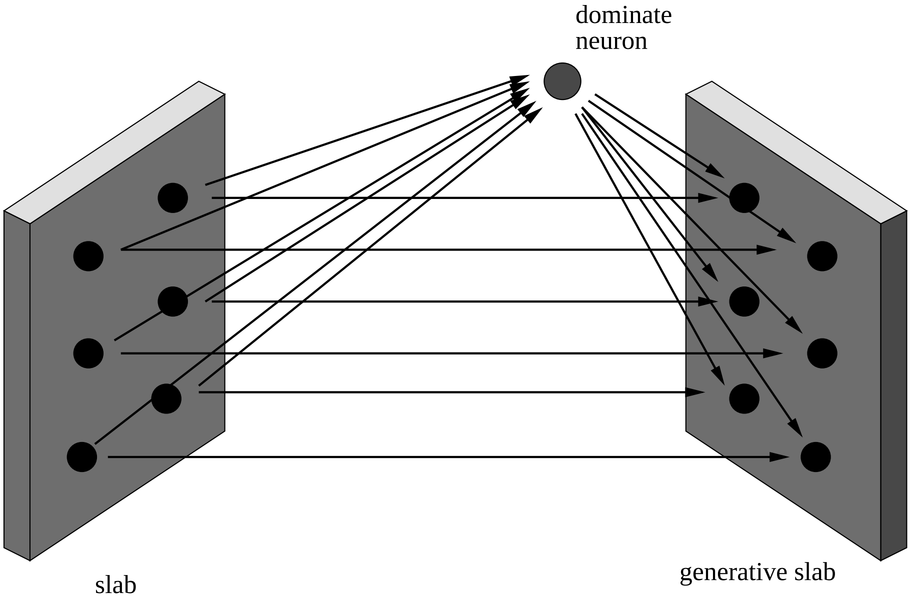

Fig. 2. The sketch of the connections from a slab to its generative slab.

with eight vertexes. Here, the final layer consists of the five slabs which are the SMNNs for five binary patterns. Clearly, slab 2 has three generative slabs and slab 3 has two generative slabs. In this simple case, the five binary patterns belonging to two classes are stored addressably and content-addressably.

We further explain how a slab connects to a generative slab which is not an SMNN (slab). As shown in Fig. 2, every neuron in the array of the left slab connects only to the corresponding neuron in the array of the generative (right) slab. Furthermore, all the neurons in the left slab connect to the dominant neuron of the generative slab which then connects to all the neurons of the generative slab. Certainly, we should assume that when the activating signals of the neurons on the left slab are transmitted to the corresponding neurons on the right slab, the dominant neuron has processed the input signal from the left slab and sent the output signal to these neurons on its slab in the same time. Obviously, it is easy to design the slabs and their connections. However, it may be difficult to design the dominant neurons for the slabs. We will discuss the construction of the dominant neurons in the next section. Since an SMNN slab does not need the dominant neuron, the information (a binary pattern) on its original slab will be transmitted only to it and then it will retrieve the memorized pattern or evolve to the zero state.

With such architecture, this neural network can retrieve or restore a piece of memorized information (pattern) from one of the SMNNs in the memory, i.e., the SMNN bank, as follows. When a pattern (a piece of information) is inputted on the input layer or slab, it will be transmitted only to its class slab, subclass slab, etc., through the dominant neurons' positive signals. That is, when the dominant neuron of a slab is activated and sends a positive signal to each neuron of the slab, the threshold value of the neurons of the slab decreases enough to let the input pattern pass through the slab. Otherwise, the threshold value remains high enough to prevent the input pattern from passing through the slab and the input pattern disappears at the slab. In this way, it will be finally transmitted to a number of SMNNs (SMNN slabs) in the parallel way. By the content-addressable memory function of the SMNNs, the pattern will be certainly retrieved or restored from its SMNN even if the input pattern contains some errors in contrast to the memorized pattern. In the case that there are some errors on the input pattern, we assume that the pattern with a small number of errors still has the same class as the pattern itself by the decision of the dominant neurons. Obviously, this neural network is a hybrid model of addressable and content-addressable memory. The information or pattern is always transmitted in a small passage and directed by the dominant neurons. Therefore, the membership of the class, subclass, etc., of a pattern directs the input signal to retrieve it, that is, the membership serves as its address in the SMNN bank. The pattern is finally retrieved content-addressably in its SMNN from a number of SMNNs.

3. The Learning Rules and Biological Discussions

We further discuss the learning rules and biological characteristics of our proposed hybrid neural network of addressable and content-addressable memory. In this situation, the hybrid neural network is considered biologically. The weight on the connection between two neurons is the efficiency of the synapse between them. As being expressed by the SMNN model, the neurons in a slab evolve synchronously and a pattern is retrieved or restored when its SMNN is stable at this pattern, that is, the SMNN can maintain the state as this pattern for a period of time.

We suppose that the learning process is just a process of synapse modification, while the threshold values of neurons make no contribution to the learning result. So we further suppose that all the neurons in a local field or slab have the same threshold value, however, it can change with the environment. With above facts and supposals, it is already shown in Ref. 16 that an SMNN can be constructed with Hebbian learning rule. Clearly, the connections from one slab to its generative slab are simple and can be also constructed by Hebbian learning rule. Then, we only need to consider the learning rule to construct a dominant neuron for a slab.

According to its function, the dominant neuron for a slab detects whether the input pattern belongs to the class it (or the slab) represents. When the input pattern belongs to the class, the dominant neuron sends a positive signal to each of the neurons in the slab to let the pattern pass through. Otherwise, when the input pattern does not belong to the class, the dominant neuron sends an inhibitory signal (or nothing) to each of the neurons in the slab to make the pattern disappear at the slab. Essentially, these dominate neurons of the slabs in one layer compete each input pattern for their classes or subclasses. So, we can apply the winner-take-all rule[18] of competitive learning to constructing these dominate neurons of the slabs in the hidden layers of the hybrid neural network as follows.

For clarity, we let $\mathcal{V} = \{V^0, V^1, \ldots, V^{N-1}\}$ be the set of binary sample (or standard) patterns to be processed. We first consider the dominate neurons of the slabs in the second layer which classify all the N binary sample patterns into m_2 classes. Actually, these m_2 dominate neurons form a competitive layer for the input patterns. Here, we simply denote the weight vector of the dominate neuron of slab i by $W_i = [w_{i1}, \ldots, w_{in}]^T, i = 1, 2, \ldots, m_2$. We also assume these weight vectors to be normalized, i.e.,

$\|W_i\| = 1$. The winner-take-all (or classical competitive learning) rule for these dominate neurons or weight vectors consists of the following two steps.

Step 1: Randomly take a sample V from $\mathcal{V}$, and for $i = 1, \ldots, m_2$, let

$$u_i = \begin{cases} 1\,, & \text{if } i = c \text{ such that } W_c^T V = \max_j W_j^T V\,, \\ 0\,, & \text{otherwise}\,. \end{cases} \tag{2}$$

where u_i denotes the output of the dominate neuron of slab i.

Step 2: Update the weight vectors W_i by

$$\Delta W_i = \eta u_i (V - W_i)\,, \tag{3}$$

with normalizing.

Clearly, for each pattern, only the winning unit W_c is updated. The parameter η with $0 \leq \eta \leq 1$ is the learning rate that either is a small positive number or starts from a reasonable initial value and then reduces to zero according to the so called Robbin–Monro stochastic approximation procedure.[19]

As the learning process is completed, we have obtained the weight vectors of these dominate neurons. Each dominate neuron represents a class of the sample patterns it wins. That is, if the dominate neuron i wins the sample pattern V^j in the competition, its output u_i is 1 and the outputs of the other dominate neurons are zero. According to the function of the competitive learning, the classification via these dominate neurons is based on the similarity of the sample patterns, which is consistent with the function of our hybrid neural network. Especially, if the dominate neuron i only wins one sample pattern after the competitive learning is completed, we will use the SMNN of this pattern instead of this slab.

For the dominate neuron of a slab in the sequential hidden layers, we can use the same method to train its weight vector. In this situation, the training set of sample patterns becomes the class of sample patterns the dominate neuron of its original slab represents and the dominate neurons in the competition are those of the generative slabs with the same original slab. Therefore, we can construct all the dominant neurons with the winner-take-all rule of competitive learning.

By above discussions, the dominant neuron can be constructed with certain competitive learning rule. Therefore, the hybrid neural network can be constructed with Hebbian and competitive learning rules. We leave the discussion of its performance in the next section. Here, we give more discussion on its biological significance. Since Hebbian learning and competitive learning are strongly believed to exist in the brain, it is possible that there exists such a biological hybrid network in the brain for the information storage (memory) and retrieval. We further have two arguments for the existence of the biological hybrid network in the brain as follows.

- First, according to the anatomy of the brain, it is known that the memory or information is distributed on the cortex. However, the input signals from the eyes, ears, etc., will be transmitted through a number of layers of intermediate neurons to reach at the cortex and retrieve the information there. Since the cortex covers on all the information channels from the sensory organs, the neural network architecture from the input layer of one sensory organ to the cortex takes the form of a tree if we do not consider the hetero-associative memory in the cortex. This is consistent with our proposed hybrid network.
- Second, the dominant neurons can serve as the address of information in a simple and natural way. By above discussions, the dominant neuron is a detector of certain similarity in the sample patterns and we can design it by the competitive learning. If the similarity is detected, the dominant neuron will be excitatory and send some (chemical) substances to decrease the threshold value of the neurons on the slab. (Note that all the neuron on a slab have the same threshold value by our supposal.) Then, the pattern will pass through the slab for further transmission. Otherwise, if the similarity is not detected, the dominant neuron will be inhibitory and send some other substances (or nothing) to increase (or remain) the threshold value so that all the neurons will keep quiescent even if the pattern is transmitted at the slab. Therefore, the pattern will certainly disappear at this slab.

Summing up all the discussions above, we have found that this hybrid neural network is a reasonable model to describe the memory structure or information storage and retrieval mechanism in the brain.

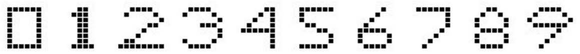

Fig. 3. The ten binary numeral patterns for the simulation experiment.

4. The Simulation Results

In this section, a simulation experiment on the ten binary numeral patters is carried out to demonstrate our proposed hybrid neural network. Then, we need to only memorize the ten binary patterns of Arabic numerals in our hybrid neural network. As shown in Fig. 3, these ten Arabic numerals $\{0,1,2,3,4,5,6,7,8,9\}$ for the simulation are expressed by 8×8 pixels. Specifically, each number i is expressed by a binary matrix $S^k = (s^k_{ij})_{8\times 8}$, where $s^k_{ij} = 1$ represents the black pixel. Actually, we consider it as a vector binary pattern, i.e., $V^k = vec[S_k]$ in the experiment. For analysis, we introduce the minimum Hamming distance of each sample pattern V^k to the other ones by

$$d^*_k = \min_{j\neq k} d_H(V^k, V^j)$$
$$= \min\{d_H(V^k, V^j) : j = 0, \ldots, k-1, k+1, \ldots, 9\}.$$

As is well-known in coding theory, $d^*_0, d^*_1, \ldots, d^*_9$ reflect the bounds of radiuses of attraction or error-correcting hyperspheres of the binary numeral patterns(codes) in 64-dim binary space. In fact, the reasonable and largest radius of attraction hypersphere of each V^k should be $t^*_k = [\frac{d^*_k-1}{2}]$. (Here $[x]$ denotes the integer part of the real number x). For an associative memory system, only if the radius of the attracting hypersphere of each V^k is just t^*_k, the error probability of retrieval of the numeral patterns in an equally distributed noisy environment reaches the minimum.

Based on the Hamming distances between these ten binary numeral patterns, we have

$$(t^*_0, t^*_1, t^*_2, t^*_3, t^*_4, t^*_5, t^*_6, t^*_7, t^*_8, t^*_9)$$
$$= (6, 8, 5, 5, 5, 6, 5, 5, 6, 5)\,.$$

We then use these t^*_k to design the threshold values of the neurons of the SMNN slabs in the output layer of the hybrid neural network for the ten numeral patterns. That is, we just let the threshold value of the neurons in the SMNN slab for V^k be

$$\theta^k = d_H(V^k) - (t^*_k + 1)\,, \quad k = 0, 1, \ldots, 9\,. \tag{4}$$

In the simulation experiment, we take the four-layer hybrid neural network in which there are one input slab, two hidden slabs in the second layer and four hidden slabs in the third layer for classification, and ten SMNN slabs in the output layer for these ten binary numeral patterns. There are two generative slabs in the third layer for each slab in the second layer. Clearly, the ten SMNN slabs can be constructed directly from these ten binary numeral patterns via Eqs. (1) and (4). For the two hidden slabs in the second layer, we use the winner-take-all rule to train the weight vectors of their dominate neurons respectively. After the competitive learning, we have found that the first hidden slab (or the dominate neuron of it) wins the numeral patterns 0, 2, 3, 5, 7, 9, while the second hidden slab wins the other numeral patterns 1, 4, 6, 8. Furthermore, we train the weight vectors of the dominate neurons of the slabs in the third layer by the winner-take-all rule. We begin to train the first two slabs(the two generative slabs of the first slab in the second layer) with the binary numeral patterns 0, 2, 3, 5, 7, 9 and find out that the first slab wins the binary numeral patterns 7, 9, while the second slab wins the other four numeral patterns 0, 2, 3, 5. We then train the last two slabs(the two generative slabs of the second slab in the second layer) with the binary numeral patterns 1, 4, 6, 8 and find out that the third slab just wins the single numeral pattern 1, while the fourth slab wins the other three numeral patterns 4, 6, 8. After all, we get the structure of the hybrid neural network for the ten binary numeral patterns shown in Fig. 4.

We finally turn to the performance of the trained hybrid neural network. Clearly, if any numeral pattern V^k is inputted to the hybrid neural network, according to the construction of the hybrid neural network, the SMNN slab of V^k will be finally activated and stable at V^k, while all the other SMNN slabs will be always quiescent. We further verify the hybrid neural network by retrieving each numeral pattern in a noisy environment. In this situation, the performance of the hybrid neural network is dominated by the radiuses of attraction of these V^k under the hybrid neural network. For clarity, we let $\hat{t}^*_i$ be the radius of attraction of V^k. That is, $\hat{t}^*_k$ is the largest integer satisfying that if an input binary pattern V belongs to $N_{\hat{t}^*_k}(V^k) = \{V \in \{0,1\}^{64} : d_H(V, V^k) \le \hat{t}^*_k\}$ — the hypersphere of V^k with the

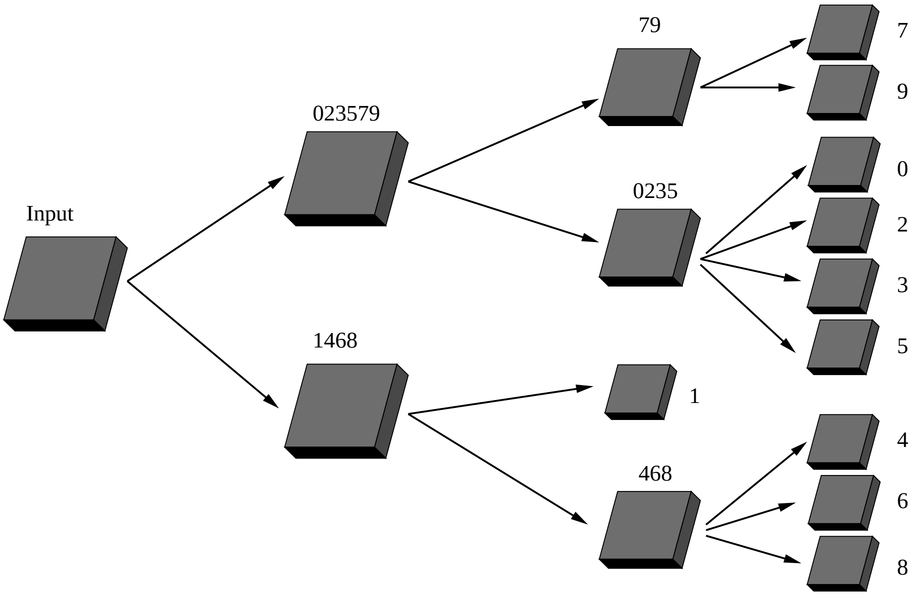

Fig. 4. The structure of the trained hybrid neural network for the ten binary numeral patterns. The numerals over each hidden slab represent the binary numeral patterns the slab wins. The slab with a single numeral is just the SMNN slab of this binary numeral pattern.

radius $\hat{t}_k^*$, the SMNN slab of V^k will be finally activated and stable at V^k, while all the other SMNN slabs will be always quiescent. According to the function of the constructed SMNN slabs, $\hat{t}_k^* \leq t_k^*$. However, $\hat{t}_k^*$ cannot be computed directly from the parameters of the hybrid neural network. We now estimate them from the simulation results in the following way.

For each V^k and an integer number $j (j \geq 0)$, we randomly select 1000 input binary patterns with a Hamming distance being j from V^k for the hybrid neural network.[a] These input binary patterns can be considered as V^k polluted by j errors in some j components of it. We then operate the hybrid neural network with each input binary pattern and check whether it finally activate the SMNN slab of V^k and make the other SMNN slabs be quiescent. If the hybrid neural network does so for all 1000 input binary patterns, we are sure that j is a possible radius of attraction of V^k. In this way for j from $0, 1, 2, \ldots$, we can get $\hat{t}_k^*$ — the largest possible radius of attraction of V^k.

Based on the simulation results, we have estimated these $\hat{t}_i^*$ as follows.

$$(\hat{t}_0^*, \hat{t}_1^*, \hat{t}_2^*, \hat{t}_3^*, \hat{t}_4^*, \hat{t}_5^*, \hat{t}_6^*, \hat{t}_7^*, \hat{t}_8^*, \hat{t}_9^*)$$
$$= (6, 7, 5, 4, 5, 6, 5, 5, 5, 5)\,.$$

By comparing $\hat{t}_k^*$ with t_k^*, we find that the ten binary numeral patterns can be reasonably retrieved from the hybrid neural network when they are in a noisy environment. Since $t_k^* - \hat{t}_k^* \leq 1$, i.e., $\hat{t}_k^*$ is close to t_k^*, it is further shown that the weight vector of each dominate neuron obtained from the competitive learning is reasonable and robust for the numeral patterns, although it has been trained only from the particular numeral patterns. In a very special situation that the input binary pattern V does not belong to any $N_{t_k^*}(V^k)$, the hybrid neural network may have a wrong retrieval result or be quiescent at all the SMNN slabs, i.e., no numeral pattern is retrieved.

5. Conclusion

We have investigated the memory structure and retrieving mechanisms in the brain. Regarding the SMNNs as the memory units in the brain, we have proposed a hybrid neural network of addressable and content-addressable memory. By this hybrid model,

[a] If the number of all the input binary patterns with a Hamming distance being j from V^k, is less than 1000, we will use all the possible input binary patterns.

a binary pattern is memorized and retrieved both addressably and content-addressably. By the analysis, we have found that the hybrid neural network can be constructed with Hebbian and competitive learning rules. Moreover, it has certain important functions in accord with the memory behavior of the brain. Finally, we have demonstrated the hybrid neural network on a set of ten binary numeral patterns.

Acknowledgment

This work was supported by the NSFC(60071004, 19701022) and the Natural Science Foundation of Guangdong Province(970377).

References

1. J. R. Anderson 1983, *The Architecture of Cognition* (Cambridge, Mass.: Harvard University Press).
2. B. B. Murdock 1982, "A theory for the storage and retrieval of item and associative information," *Psychological Review* **89**, pp. 609–626.
3. G. Gillund and R. M. Shiffrin 1984, "A retrieval model for both recognition and recall," *Psychological Review* **91**, pp. 1–67.
4. K. S. Lashley 1951, "The problem of serial order in behavior," In L. A. Jeffress (ed.), *Cerebral Mechanisms in Behavior* (New York: Wiley).
5. N. Burgess and G. J. Hitch 1999, "Memory for serial order: A network model of the phonological loop and its timing," *Psychological Review* **106**, pp. 551–581.
6. W. S. McCulloch and W. Pitts 1942, "A logical calculus of the ideas imminent in nervous activity," *Bulletin of Mathematical Biophysics* **5**, pp. 115–133.
7. D. Hebb 1949, *The organization of behavior* (New York: Wiley).
8. J. A. Anderson 1972, "A simple neural network generating an interactive memory," *Mathematical Biosciences* **14**, pp. 197–220.
9. W. A. Little 1974, "The existence of persistent states in the brain," *Mathematical Biosciences* **19**, pp. 101–120.
10. T. Kohonen 1978, *Associative memory: A system-theoretical approach* (Berlin, Federal Republic of Germany: Springer-Verlag).
11. J. A. Anderson and G. E. Hinton 1981, "Models of information processing in the brain," In G. E. Hinton and J. A. Anderson (eds.), *Parallel Models of Associative Memory* (Hillsdale, NJ: Erlbaum).
12. J. J. Hopfield 1982, "Neural networks and physical systems with emergent collective computational abilities," *Proceeding of the National Academy Sciences, USA* **79**, pp. 2245–2558.
13. J. Ma 1997, "The stability of the generalized Hopfield networks in randomly asynchronous mode," *Neural Networks* **10**, pp. 1109–1116.
14. J. Ma 1999, "The object perceptron learning algorithm on generalised Hopfield net works for associative memory," *Neural Computing & Applications* **8**, pp. 25–32.
15. J. Ma 1999, "The asymptotic memory capacity of the generalized Hopfield networks," *Neural Networks* **12**, pp. 1207–1212.
16. J. Ma 1997, "Simplex memory neural networks," *Neural Networks* **10**, pp. 25–29.
17. J. Ma 2001, "A neural network approach to real-time pattern recognition," *International Journal of Pattern Recognition and Artificial Intelligence* **15**, pp. 937–47.
18. R. Hecht-Nielsen 1990, *Neurocomputing* (Reading, MA: Addison-Wesley).
19. H. Robbins and S. Monro 1951, "A stochastic approximation method," *Ann. Math. Stat.* **22**, pp. 400–407.