Notions of Intuition and Attention Modeled by a Hierarchically Arranged Generalized Regression Neural Network

Tetsuya Hoya, *Member, IEEE*

***Abstract*—In this paper, two psychological functions, intuition and attention, are modeled by a newly proposed hierarchically arranged generalized regression neural network (HA-GRNN). The main contribution of the paper is two-fold:**

1) **to provide an engineering basis for a macroscopic representation of psychology-oriented functions by means of artificial neural networks;**
2) **to propose a concrete model for the two functions, intuition and attention, in terms of the associated interactive and evolutionary processes within an HA-GRNN.**

In the simulation study, the effectiveness of an HA-GRNN is justified within the context of pattern classification tasks.

***Index Terms*—Attention, generalized regression neural networks, intuition, psychological functions.**

I. Introduction

INTERPRETING the notions of actual brain by means of artificial neural networks is really a challenging problem. Historically, the interpretation of psychological functions has raised a number of issues and in due course lead to controversies among many disciplines from biology to philosophy. Now, with the advancements in both biological and psychological studies as well as information technology, one of which we wish to achieve in near future is to develop so called the "brain-style" computers.

It is said that one of the key issues toward the development of brain-style computers is how to elucidate the notion of "intuition" in terms of artificial neural networks (e.g., See [1]).

On the other hand, modeling the notion of "consciousness" has recently been a topic of great interest in robotics [2]–[5]. For instance, such a concrete modeling can be found in [5], in which the authors developed an intelligent robot which can to some extent mimic an actual animal's behavior by means of the consciousness model. In contrast, in [2], a virtual machine (referred to as Magnus) in which the author modeled the notion of artificial consciousness is proposed.

In this paper, it is addressed that such psychological functions, "intuition" and "attention," can be interpreted in terms of the associated interactive processes and evolution of the novel hierarchically arranged generalized regression neural network (HA-GRNN) in which each sub-network is based upon a memory-based architecture. The effectiveness of the HA-GRNN is then justified within the context of pattern classification tasks.

The generalized regression neural networks (GRNNs) [6]/probabilistic neural networks (PNNs) [7] fall in the category of radial basis function neural networks (RBF-NNs) [8], while, unlike the standard RBF-NNs, they share a special property that the weight vectors between the RBFs and output neurons are fixed to the target vectors. By virtue of this attractive property, a dynamic neural system can be modeled without any complex mathematical operations. As an example, in [9], a two-stage memory system, the model of which was inspired from the psychological study of the short-term (STM) and long-term memory (LTM) concept [10], is proposed using GRNN/PNNs. Then, in [9] the model is applied to on-line batch pattern correction of both digit voice/character recognition tasks.

In addition, an RBF-NN has often been referred to as a Gaussian mixture model (GMM) [11], or it even can be subsumed into the concept of support vector machine (SVM) [12] by regarding the RBFs as Gaussian kernel machines. So far, these concepts have been successfully applied to a wide variety of signal processing-oriented problems, such as pattern classification, signal modeling and prediction (e.g., see [13]), or adaptive control (e.g., see [6]).

The organization of the paper is given as follows: in the next section, the configuration of a GRNN is summarized. In Section III, a hierarchically arranged generalized regression neural network (HA-GRNN) is newly proposed, which provides a basis for modeling psychological functions. The interpretation of the two notions, intuition and attention, is then described in Section IV in terms of the associated interactive processes and the evolution of an HA-GRNN. In Section IV, the linkage between the hierarchical memory system and the notions of both intuition and attention is elucidated. Section V is devoted to the simulation study of modeling these psychological functions by constructing an HA-GRNN using the three domain pattern sets of digit voice/character classification tasks. Finally, the paper is concluded in Section VI.

II. Configuration of a GRNN

A multilayered generalized regression neural networks (ML-GRNN) [14] with N_i input neurons, N_h radial basis functions (RBFs), and N_o output neurons is illustrated in the upper part in Fig. 1. In the figure, each input neuron $x_i (i = 1, 2, \cdots, N_i)$ corresponds to the element in the input vector $\boldsymbol{x} = [x_1, x_2, \cdots, x_{N_i}]^T$ (T: vector transpose),

Manuscript received July 14, 2001; revised August 13, 2002. This paper recommended by Associate Editor Y. Pan.

The author is with the Laboratory for Advanced Brain Signal Processing, Saitama 351-0198, Japan (e-mail: hoya@bsp.brain.riken.go.jp).

Digital Object Identifier 10.1109/TSMCB.2003.811758

$h_j (j = 1, 2, \cdots, N_h)$ is the j-th RBF (note that N_h is varied), $\|\cdots\|_2^2$ denotes the squared L_2 norm, and the output of each neuron $o_k (k = 1, 2, \cdots, N_o)$ is calculated as

$$o_k = \frac{1}{\delta} \sum_{j=1}^{N_h} w_{j,k} h_j \tag{1}$$

where:

$$\begin{aligned} \delta &= \sum_{k=1}^{N_o} \sum_{j=1}^{N_h} w_{j,k} h_j, \\ \boldsymbol{w}_j &= [w_{j,1}, w_{j,2}, \ldots, w_{j,N_o}]^T, \\ h_j &= f(\boldsymbol{x}, \boldsymbol{c}_j, \sigma_j) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_j\|_2^2}{2\sigma_j^2}\right) \end{aligned} \tag{2}$$

where $\boldsymbol{c}_j$ is called the centroid vector, σ_j is the radius, and $\boldsymbol{w}_j$ denotes the weight vector between the j-th RBF and the output neurons.

As in Fig. 1 in the upper, the structure of an ML-GRNN is similar to the well-known multilayered perceptron neural network (MLP-NN) [8] except that RBFs are used in the hidden layer and linear functions in the output layer.

In comparison with the conventional RBF-NNs, the GRNNs have a special property, namely that no iterative training of the weight vectors is required. That is, like other RBF-NNs, any input-output mapping is possible, by simply assigning the input vectors to the centroid vectors and fixing the weight vectors between the RBFs and outputs identical to the corresponding target vectors. This is quite attractive, since conventional MLP-NNs with backpropagation type weight adaptation involve long and iterative training, and there even may be a danger of their being stuck in local minima (this is serious as the size of the training set becomes large).

Moreover, the special property of GRNNs enables us to flexibly configure the network depending on the tasks given, which is considered to be beneficial to real hardware implementation, with only two parameters, $\boldsymbol{c}_j$ and σ_j, to be adjusted. The only disadvantage of GRNNs in comparison with MLP-NNs seems to be, due to the memory-based architecture, the need for storing all the centroid vectors into memory space, which can sometimes be exhaustive for on-line data processing, and hence, the utility is slow in the reference mode (i.e., the testing phase). Nevertheless, with the flexible configuration property, the GRNNs can be exploited for interpretation of the notions relevant to actual brain, such as "intuition," or other psychological functions.

In Fig. 1, when the target vector $\boldsymbol{t}(\boldsymbol{x})$ corresponding to the input pattern vector $\boldsymbol{x}$ is given as a vector of indicator functions

$$\begin{aligned} \boldsymbol{t}(\boldsymbol{x}) &= (\delta_1, \delta_2, \ldots, \delta_{N_o}), \\ \delta_j &= \begin{cases} 1 & \text{if } \boldsymbol{x} \text{ belongs to the class corresponding to } o_k \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{3}$$

and when the RBF h_j is assigned for $\boldsymbol{x}$, with utilizing the special property of GRNNs, $\boldsymbol{w}_j = \boldsymbol{t}(\boldsymbol{x})$, the entire network becomes topologically equivalent to the network with a decision unit and N_o sub-networks as in the lower part in Fig. 1.[1]

[1]Hereafter, a "GRNN" is referred to as the network with the structure in the lower part of Fig. 1, unless explicitly denoted otherwise.

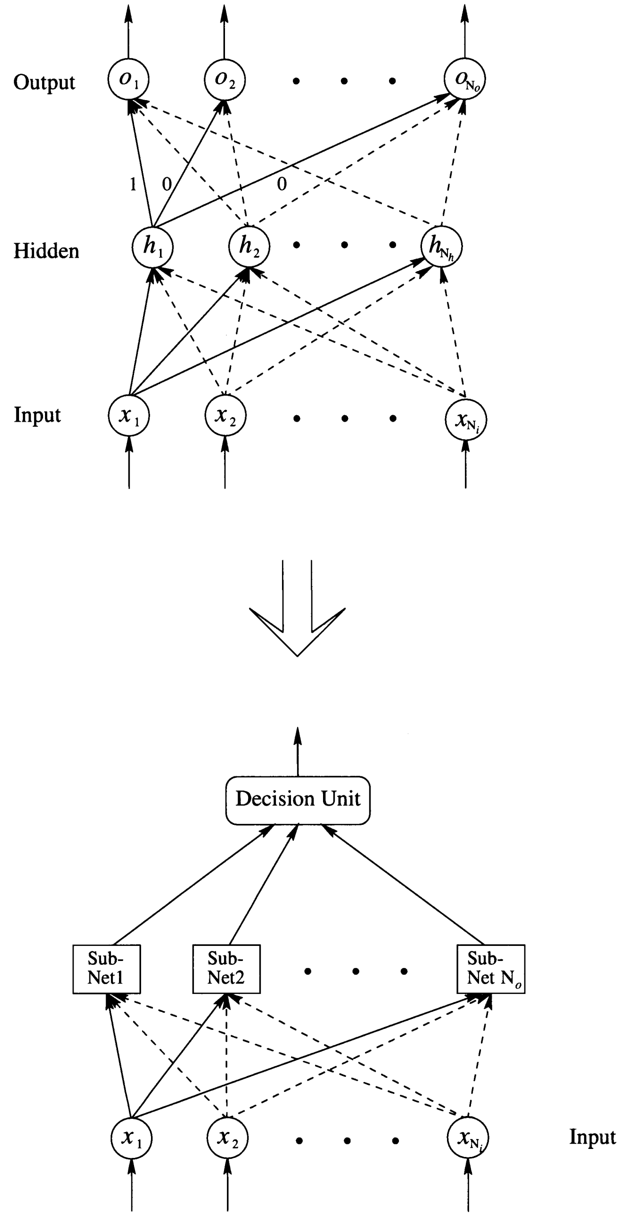

Fig. 1. Illustration of topological equivalence between the ML-GRNN with M hidden and N output units and the assembly of the N distinct sub-networks.

In summary, the network configuration[2] by means of an ML-GRNN is simply achieved as in the following.

Network Growing: Set $\boldsymbol{c}_j = \boldsymbol{x}$ and fix σ_j, then add the term $w_{jk} h_j$ in (2). The target vector $\boldsymbol{t}(\boldsymbol{x})$ is thus used as a class "label" indicating the sub-network number to which the RBF belongs. (Namely, this operation is equivalent to add the j-th RBF in the corresponding (the k-th) Sub-Net in the lower part in Fig. 1.)

Network Shrinking: Delete the term $w_{jk} h_j$ from (2).

[2]In neural networks community, this configuration is often referred to as "learning." Strictly speaking, the usage of the terminology is, however, rather limited, since the network is grown/shrunk by fixing the network parameters for a particular set of patterns other than "tuning" them, e.g., by repetitive adjustment of the weight vectors as in the backpropagation algorithms.

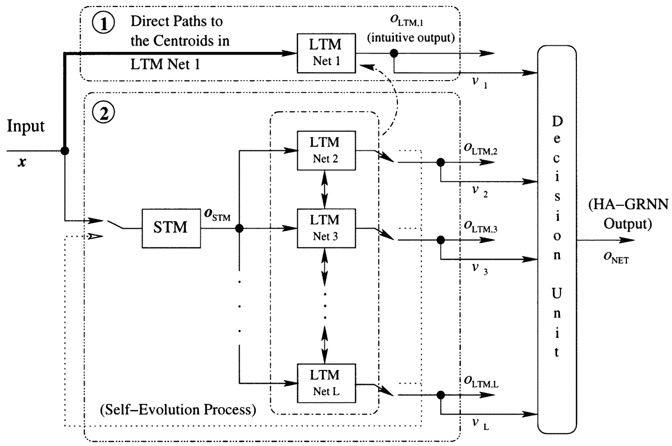

Fig. 2. Schematic representation of a hierarchically arranged GRNN.

Then, the above network configuration gives a basis for forming a dynamic pattern classifier and hence a hierarchically arranged GRNN (HA-GRNN) to be described in Section III.

A. The Radii Setting

For the design of RBF-NNs, the setting of radii values is also a significant factor and still remains an open issue (e.g., see [8]); this often requires a trade-off between the generalization performance and the computation time for the tuning. In this paper, by taking into account the flexibility in reconfiguration of HA-GRNN, a unique radius setting is considered and used for all the RBFs (of the whole network) according to

$$\sigma = \gamma d_{max} \tag{4}$$

where γ is a scalar constant and d_{max} is the maximum Euclidean distance searched among all the pairs of the centroid vectors. The constant γ can be chosen a priori so that the entire hyperspace formed during the training phase is moderately (or reasonably) covered by the RBFs. In this paper, the radii values are updated according to the above when the number of the RBFs is varied or when the parameters of the RBFs (e.g., the centroid vector) are changed.

III. Hierarchically Arranged Generalized Regression Neural Networks

The structure of a hierarchically arranged GRNN (HA-GRNN) is illustrated in Fig. 2. As in the figure, the HA-GRNN consists of a multiple of neural networks and its associated data processing mechanisms:

1) a modified RBF network representing short-term memory (STM);
2) a multiple of GRNNs representing long-term memory (LTM) networks (denoted "LTM Net 1-L" in the figure);
3) a decision unit (following the "winner-take-all" strategy).

Moreover, the LTM networks can be subdivided into two types of networks; one for generating "intuitive outputs" ("LTM Net 1") and the others ("LTM Net 2" to "LTM Net L") for the regular outputs. For the regular LTM, each of LTM Nets 2 to L has the same structure as in the lower part in Fig. 1, whereas both the STM and LTM Net 1 are the modified RBF-NNs as illustrated

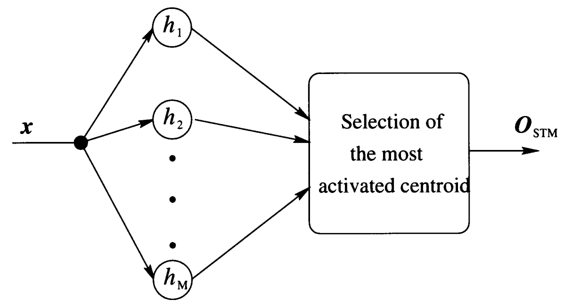

Fig. 3. Illustration of a modified RBF-NN for STM.

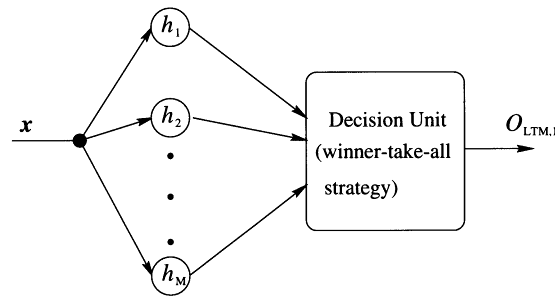

Fig. 4. Illustration of a network representing LTM Net 1.

in Figs. 3 and 4, respectively. (However, as described in Section III-B, the associated data processing within the STM is different from that in LTM Net 1.) In Fig. 2, $\boldsymbol{x}$ denotes an incoming input pattern vector to the HA-GRNN, $\boldsymbol{o}_{STM}$ is the STM output vector, $o_{LTM,i}$ $(i = 1, 2, \cdots, L)$ are the LTM network outputs, v_i are the weight values to the LTM network outputs, and o_{NET} is the final HA-GRNN output.

A. Evolution of the HA-GRNN

The concept of the HA-GRNN is inspired from both biology and psychology-oriented studies of the memory system in actual brain [10], [15]–[20]. In the HA-GRNN, the role of STM is to "buffer" the incoming input pattern vectors, before storing them to the LTM. It is then hypothesized that the LTM intrinsically has a layered structure representing a hierarchical pattern classification system and that the formation of such classification system depends upon such criterion as "significance" or "attractiveness" of information received which is of crucial/beneficial to the whole system. In this paper, without loss of generality, the criterion is simply based upon the activation of RBFs within the LTM networks.

In summary, the evolution of the HA-GRNN is divided into five phases.

Phase 1: The STM and LTM Net 2 formation.
Phase 2: Formation of LTM Nets 2 to L.
Phase 3: Reconfiguration of LTM Nets 2 to L (self-evolution).
Phase 4: Formation of LTM Net 1 (for generating intuitive outputs).
Phase 5: Formation of the attentive states.

In the following subsections and Sections IV-A and IV-B, each of the five phases above will be described in detail.

1) Phase 1: Formation of the STM Network and LTM Net 2: In Phase 1, the STM network is firstly formed in the manner to be described in detail in Section III-B, then LTM Net 2 is formed by directly assigning the output vectors of the STM network to the RBFs in LTM Net 2. In other words, at the initial stage of the evolutionary process (i.e., from the very first presentation of the incoming input pattern vector until LTM Net 2 is filled), since each LTM network (except LTM Net 1) is represented by a GRNN, the RBFs within LTM Net 2 are distributed into the respective sub-networks, according to the class "label" [i.e., the label is set by the target vector consisting of a series of indicator functions as defined in (3)] associated with each centroid vector.

2) Phase 2: Formation of LTM Nets 2 to L: The addition of the RBFs in Sub-Net i ($i = 1, 2, \ldots, N_{cl}$, where N_{cl} is the number of classes which is given identical to the number of the sub-nets in each LTM network[3]) of LTM Net 2 is repeated until the total number of RBFs in Sub-Net i reaches a maximum $M_{LTM_2,i}$. Otherwise, the least activated RBF in Sub-Net i is moved to LTM Net 3. This process corresponds to Phase 2 and is summarized as follows:

[Phase 2: Formation of LTM Nets 2 to L]

Step 1: Provided that the STM output vector belongs to Class i,
for $j = 1$ to $L-1$, do the following:
If the number of RBFs in Sub-Net i of LTM Net j
reaches $M_{LTM_{j,i}}$, move the least activated RBF
within Sub-Net i of LTM Net j to that of LTM Net
$j+1$.

Step 2: If the number of RBFs in Sub-Net i of LTM Net L reaches
$M_{LTM_{L,i}}$ (i.e., all the i-th sub-networks within LTM
Nets 2 to L are filled), there is no entry to store the new STM
output vector. Then, do the following:
Step 2.1: Discard the least activated RBF in Sub-Net i of LTM
Net L.
Step 2.2: Shift one by one all the least activated RBFs in Sub-Net
i of LTM Nets (L-1) to 2 into that of LTM Nets L to 3.
Step 2.3: Then, store the new STM output vector in Sub-Net i
of LTM Net 2. (Thus, this procedure is similar to a
last-in-first-out (LIFO) stack.)

The previous is based on the hypothesis that long-term memory has a layered structure in itself, where in the HA-GRNN context the long-term memory is represented as a group of LTM Nets 2 to L. In Fig. 2, the final output of the HA-GRNN o_{NET} is given as the largest value among the weighted LTM network outputs $o_{LTM,i}$ $(i = 1, 2, \cdots, L)$

$$o_{NET} = \max(v_1 \cdot o_{LTM,1},\ v_2 \cdot o_{LTM,2},\ \cdots,\ v_L \cdot o_{LTM,L}) \quad (5)$$

where

$$v_1 \gg v_2 > v_3 > \cdots > v_L. \quad (6)$$

Note that the weight value v_1 for $o_{LTM,1}$ is given relatively larger than the others. This discrimination indicates the formation of the "intuitive output" from the HA-GRNN, to be described later.

3) Phase 3: Reconfiguration of LTM Nets 2 to L (Self-Evolution): After the formation of LTM Nets 2 to L, the reconfiguration of the LTM networks is considered to be occurred in Phase 3. This process may be invoked either at a particular (period of) time or due to the strong excitation of some RBFs in the LTM networks driven by a particular input pattern vector(s).[4] During the reconfiguration phase, the presentation of the incoming input pattern vector is not allowed to process at all (hence the term "self-evolution") by the HA-GRNN. Then, the reconfiguration procedure is given as follows.

[Phase 3: Reconfiguration of LTM Nets 2 to L (Self-Evolution)]

Step 1) Collect all the centroid vectors within LTM Nets 2 to l ($l <= L$).
Step 2) Set the centroid vectors so collected as the incoming input pattern vectors.
Step 3) Present them to the HA-GRNN again, one by one. This process is repeated for p times. (In Fig. 2, this flow is depicted in a dotted line.)

It is then considered that the above reconfiguration process invoked at a particular time period is effective for "shaping up" the pattern space spanned by the RBFs within LTM Nets 2 to L.

Alternative to the above, a clustering like method in [9] could be considered for the reconfiguration of the LTM networks. The approach in [9] is, however, rather different from the above instance-based operation in the sense that a new RBF set for LTM is obtained by *compressing* the existing LTM using the clustering techniques, which, as reported, may sometimes collapse the pattern space as the number of representative vectors becomes small.

4) Phase 4: Formation of LTM Net 1: In Phase 4, a certain number of the RBFs in LTM Nets 2 to L which keep relatively strong activation in a certain period of the pattern presentation are transferred to LTM Net 1. Each RBF newly assigned in LTM Net 1 eventually forms an RBF-NN and will have a direct connection with the incoming input vector, instead of the output vector from the STM. The formation of LTM Net 1 is then summarized as follows.

[3]Here, without loss of generality, it is assumed that the number of the sub-nets is unique in each of LTM Nets 2 to L.

[4]In the simulation study, due to the analytical difficulty, the latter case was not considered.

[Phase 4: Formation of LTM Net 1]

Step 1) In Phases 2 and 3 (i.e., during the formation/reconfiguration of the LTM networks), given an output vector from the STM, the most activated RBFs in LTM Nets 2 to L are monitored; each RBF has an auxiliary variable which is initially set to 0 and is incremented, whenever the corresponding RBF is most activated *and* the class ID of the given incoming pattern vector matches the Sub-Net number to which the RBF belongs.

Step 2) Then, at a certain time or period (q, say), list up all the auxiliary variables (or, activation counter) of the RBFs in LTM Nets 2 to L, and obtain the N RBFs with the N largest numbers, where the number N can be set as

$$N \ll \sum_i \sum_j M_{LTM_{j,i}} (j = 2, 3, \ldots, L).$$

Step 3) If the total number of RBFs in LTM Net 1 is currently less than or equal to $M_{LTM_1} - N$ (i.e., M_{LTM_1} denotes the maximum number of the RBFs in LTM Net 1, assuming $N \leq M_{LTM_1}$), move all the N RBFs to LTM Net 1. Otherwise, remain intact the original $M_{LTM_1} - N$ RBFs within LTM Net 1, while fill/replace the remaining RBFs in LTM Net 1 with the N newly obtained RBFs.

Step 4) Create a direct path to the incoming input pattern vector for each RBF added in the previous step. (This data flow is illustrated in bold-line in Fig. 2.)

Note that, unlike other LTM networks, the radii values of the RBFs in LTM Net 1 must not be varied during the evolution, since the strong activation from each RBF is expected to continue after the transfer with the current radii values.

Up to here, the technical detail of the first four phases within the evolutionary process of HA-GRNN has been described. In Section IV-A, it will be elucidated that how the process in Phase 4 above can be interpreted as the notion of intuition. In the sequel, the remaining Phase 5, which is relevant to the other notion, i.e., attention, will be discussed in detail in Section IV-B. Then, the rest of this section is devoted to the description of the STM network.

B. The STM Network

In Fig. 3, the output of the RBF-NN is given in a *vector* form rather than a scalar value (e.g., the value calculated as the sum of the RBF outputs in the conventional RBF-NN). Note that, unlike the regular LTM networks, the STM network does not have any sub-networks, i.e., it is based upon a single layered structure, with a maximum number of RBFs M_{STM}. Thus, as in the formation of LTM Nets 2 to L in Section III-A-4, the STM is also equipped with a mechanism similar to a last-in-first-out (LIFO) stack system, by the introduction of the factor M_{STM}. The learning of the STM network is then summarized in the following.

Step 1) If the number of the RBFs M is currently less than M_{STM}, add an RBF with activation h_i (calculated by (2)) and its centroid vector $\boldsymbol{c}_i = \boldsymbol{x}$ in the STM network. The STM network output vector $\boldsymbol{o}_{STM}$ is then equal to $\boldsymbol{x}$.

Step 2) Otherwise,

- If the activation of the least activated RBF (h_j, say) $h_j < th_{STM}$, replace it with a new one with the centroid vector $\boldsymbol{c}_j = \boldsymbol{x}$. In this case, the network output $\boldsymbol{o}_{STM}$ is the same as $\boldsymbol{x}$.
- Otherwise, the network output vector $\boldsymbol{o}_{STM}$ is adjusted according to:

$$\boldsymbol{o}_{STM} = \lambda \boldsymbol{c}_k + (1 - \lambda)\boldsymbol{x} \quad (7)$$

where $\boldsymbol{c}_k$ is the centroid vector of the most activated RBF (k-th, say) h_k and λ is a *smoothing* factor ($0 \leq \lambda \leq 1$).

In Step 2 above, a smoothing factor λ is introduced in order to regulate how fast the STM network is evolved by a new incoming pattern vector given to the network. In other words, the role of this factor is to determine how quick the STM network *switches* the focus to other patterns. This may be regarded as "selective attention" [18] to a particular object/event. For example, if the factor is set small, the output $\boldsymbol{o}_{STM}$ becomes more likely to $\boldsymbol{x}$, then this mimics "carelessness." In contrast, if the factor is set large, the STM network becomes "sticky" to a particular set of patterns. This also relates to the notion of attention to be described in Section IV-B.

IV. Interpretation of "Intuition" and "Attention"

A. A Model of "Intuition" by an HA-GRNN

In our daily life, we sometimes have a feeling that the thing/matter is true but can neither explain the reason why nor find the the evidence of such feeling. This is referred to as the notion of, what is called, "intuition."

Conjecture 1: In the context of HA-GRNN, *intuition* can be interpreted in such a way that, for a particular incoming input pattern vector there exists a certain set of RBFs with *abnormally* strong activation within the LTM networks.

The above conjecture also agrees with the standpoint that the notion of intuition can be explained in terms of the information processing pertaining to a particular activity of neurons within brain (e.g., see [21]).

In Fig. 2, there are two paths for the incoming input pattern vectors to the HA-GRNN. The point of having these paths within the HA-GRNN is that for the *regular* incoming input pattern vectors the final output will be generated after the associated processing within the two-stage memory, namely the STM and LTM, while a certain set of input patterns may excite the neurons within LTM Net 1, which is enough to yield "intuitive" outputs from the HA-GRNN. Then, the evidence for referring to the output of LTM Net 1 as intuitive output is that, as in the description of the evolution of HA-GRNN in Section III-A, LTM Net 1 will be formed *after a relatively long and iterative exposition* of incoming input pattern vectors, which results in the strong excitation of (a certain number of) the RBFs in LTM Nets 2 to L. In other words, the transition of the RBFs from the STM to LTM Nets (2 to L) is referred to as *normal* learning process, whereas, in counter-wise, that from LTM Nets (2 to L) to LTM Net 1 gives the chances of generating "intuitive" outputs from an HA-GRNN.

In practice, this feature is particularly useful, since it is highly expected that the HA-GRNN can generate better and faster pattern classification results from LTM Net 1, while keeping the network size smaller than the conventional networks trained by an iterative algorithm with a large amount of training data such as MLP-NNs, than the ordinary reasoning process, i.e., the reasoning process through STM + regular LTM Nets (2 to L).

In contrast, we quite often hear such episodes as, "I have got a flash to a brilliant idea!" or "While I was asleep, I was suddenly awaken by a horrible nightmare." It can also be postulated that these all are the phenomena occurred in the brain, similar to the intuition, during the self-evolution process of memory. In the context of HA-GRNN, this is relevant to Phase 3 in which, during the reconfiguration (or, reconstruction, in other words) phase of the long-term memory, some of the RBFs in LTM are so excited enough to exceed a certain level of activation. Then, these RBFs remain in LTM relatively for a long period or (almost) permanently because of such memorable events. This interpretation also agrees with the biological findings [22], [23] in which the authors state that, once one has acquired the behavioral skill, the person would not forget it for a long time.

B. *Interpreting the Notion of "Attention" by an HA-GRNN*

In psychology, it is generally acknowledged that "attention" is one of the constituents describing "consciousness" (e.g., [24]–[26]). The word "consciousness" is, however, quite intangible and the explicit definition of consciousness is almost impossible, due to its inherently too broad and complicated issues involved.

Nevertheless, the notion of "consciousness" (in a narrow sense) has motivated and been utilized in the development of intelligent robotics [2]–[5]. In [5] and [27], such concrete models can be found in which the maze-path finding pursuit is achieved by an artificial mouse. In [27], the movement of the (artificial) mouse is controlled by a hierarchically conceptual model, so-called the "consciousness architecture" (strictly, the utility of the term "awareness" seems more appropriate in the context). In the model, the mouse robot can continue the maze-path finding by the introduction of a higher layer of memory representing the state of "being aware" of the path-finding pursuit, while the lower part is used for the actual movement.

The notion of attention here is to focus the HA-GRNN to a particular set of incoming patterns, e.g., paying attention to someone's voice or the facial image in order to acquire further information of interest, in parallel to process other incoming patterns received by the HA-GRNN, and, as described in Section III-B, the STM has the role.

1) *Phase 5: Formation of Attentive States:*

In the context of HA-GRNN, as described above, the model in [27] coincides with the evidence of having a "hierarchical" structure representing the notion of attention. This hierarchy can be represented simply by the number of RBFs within the STM network:

Conjecture 2: The state of being "attentive" of something is represented in terms of the RBFs within the STM.

As discussed earlier, in the HA-GRNN, since the transition of the RBFs from the STM to the LTM networks can be regarded as the "feedforward" transition, it is natural to consider, in counter-wise, the feedback transition, namely that from the LTM networks to the STM. Therefore, the attentive states (given in Phase 5) can be formulated as the feedforward transition operation in contrast to Phases 2 and 3 defined in Section III-A:

[Phase 5: Formation of Attentive States]

Step 1) Collect m RBFs of which the auxiliary variables are the first m largest within the LTM networks for particular classes (this collection then meets the attentive states of the HA-GRNN).

Step 2) Add the *copies* of the m RBFs back into the STM, while the $M_{STM} - m$ most activated RBFs in the STM network remain intact. The m RBFs so selected remain within the STM for a certain long period, without updating their centroid vectors (whereas the radii may be updated).

In Phase 5, the m RBFs so collected make the HA-GRNN to focus upon a particular set of incoming input vectors, and, by increasing m, it is considered that the filtering process in transferring incoming pattern vectors to the LTM networks becomes more accurate. For instance, if the HA-GRNN is applied to pattern classification tasks, it is expected that the system can compensate for the misclassified patterns that fall in a certain class(es). In addition, the radii values of the m RBFs so copied may be updated, since the parameters of other remaining RBFs within the STM can be varied during the learning. In other words, it is postulated that the ratio between the m RBFs and the rest of the $M_{STM} - m$ RBFs in the STM determines the "level of attention." Therefore, the following conjecture can be drawn;

Conjecture 3: The level of attention can be determined by the ratio between the number of m most activated RBFs selected from the LTM networks and that of the remaining $M_{STM} - m$ RBFs within the STM. This is referred to as a part of the "learning" process.

Conjecture 3 is also correlated with the neurophysiological evidence of "rehearsing" activity [17] in which the information acquired during learning would be gradually stored as long-term memory after rehearsing. In the context of an HA-GRNN, an incoming input pattern vector (or a set of the input pattern vectors) can be compared to the input information to the brain and is temporally stored within the STM (hence the function of filtering or "buffering"). Then, during the evolution, the information represented by the RBFs within the STM network is selectively transferred to the LTM, as in the Phases 1–3. In contrast, the RBFs within the LTM networks may be transferred back to the STM, because "attention" of certain classes (of those RBFs) is occurred at particular moments. (This interaction is therefore regarded as the "learning" process in [17]).

In the HA-GRNN context, since the evolution process given earlier is, strictly speaking, not autonomous, we may want to design the state of the "attention" in advance, according to the problems given in practical situations. (But, it is still possible to evolve HA-GRNN autonomously by appropriately setting the transition operations suited for the applications, though such a case is not considered in this paper.) For instance, in the context of pattern classification tasks, one may limit the number of the classes to $N < N_{max}$ in such a way that "For a certain period of the pattern presentations, the HA-GRNN must be attentive to only N classes among a total of N_{max}, in order to reinforce the HA-GRNN for the N classes."

V. Simulation Study

In the simulation study, three different domain data sets were used and extracted from

1) SFS [28];
2) OptDigit;
3) PenDigit databases.

The SFS database is a public domain database containing raw speech utterances of single digits, which has generally been used for spoken language-oriented tasks but also for a benchmark of digit voice pattern classification, while the latter two contain the feature vectors ready for character recognition tasks, both of which can be obtained from "UCI Machine Learning Repository" of the University of California.

For the SFS, the data set consisted of a total of 900 utterances of the digits from /ZERO/ to /NINE/ recorded in English by nine different speakers (including both the female and male speakers). The data set was then arbitrarily partitioned into two sets; one for constructing an HA-GRNN (i.e., the incoming pattern set) and the other for testing. The incoming pattern set contains a total of 540 speech samples, where 54 samples were chosen for each digit, while the testing consists of a total of 360 samples (36 samples per each digit). In both the sets, each utterance is sampled at 20 kHz and was converted into a feature vector with a normalized set of 256 data points obtained by the well-known LPC-mel-cepstral analysis (e.g., see [29]). The feature vector was therefore used as the input pattern vector of the HA-GRNN.

In contrast, both the OptDigit and PenDigit data sets were composed of 1200 and 400 feature vectors for the training and testing sets, respectively. Each of the feature vectors has 64 data points for the OptDigit, while 16 data points for the PenDigit, respectively.

TABLE I
Network Configuration Parameters for the HA-GRNN

Parameter	SFS	OptDigit	PenDigit
Max. num. of centroids in STM, M_{STM}	30	30	30
Total num. of LTM networks, $(L+1)$	3	2	4
Max. num. of centroids in LTM Net 1, M_{LTM_1}	5	25	15
Num. of sub-networks in LTM Nets 2-L, N_{cl}	10	10	10
Max. num. of centroids in each subnet, $M_{LTM_{j,i}}$ $(j = 2, 3, ..., L, i = 1, 2, \cdots, 10)$	4	2	4

A. Parameter Setting of the HA-GRNN

In Table I, the network configuration parameters of the HA-GRNN used in the simulation study are summarized. In the table, M_{LTM_1}, $M_{LTM_{2,i}}$, and $M_{LTM_{3,i}}$ $(i = 1, 2, \cdots, 10$, corresponding to the respective class ID $1, 2, \cdots, 10)$ were arbitrarily chosen, while N_{cl} was fixed to the number of the classes (i.e., the ten digits). With this setting, the total number of RBFs in LTM Nets (1 to 3), $M_{LTM,Total}$, is thus calculated as

$$M_{LTM,Total} = M_{LTM,1} + N_{cl}(M_{LTM,2} + M_{LTM,3}) \quad (8)$$

which yields

1) 85 for the SFS;
2) 65 for the OptDigit;
3) 175 for the PenDigit data set, respectively.

1) The STM Network Setting: For the STM network, the choice of both M_{STM} (as shown in Table I) and $\gamma = 2$ [in (4)] was, respectively, made *a priori* so that the STM network functions as a "buffer" to the LTM networks with sparsely but reasonably covering all the ten classes during the evolution. Then, the setting of $th_{STM} = 0.1$ and the smoothing factor $\lambda = 0.6$ [in (7)] was used for all the three data sets. In the preliminary simulation study, it was empirically found that the choice of $\lambda = 0.6$ yields a reasonable generalization performance of the HA-GRNN.

2) Parameter Setting of the Regular LTM Networks: For the radii setting of LTM Nets 2 to L, the setting of $\gamma = 0.25$ for the SFS and OptDigit or $\gamma = 0.05$ for the PenDigit [in (4)] was empirically found to be a choice for maintaining a reasonably good generalization performance during the evolution. Then, to give the "intuitive outputs" from LTM Net 1, v_1 was fixed to 2.0, while v_i $(i = 2, 3, \cdots, L)$ were given by the linear decay

$$v_i = 0.8\,(1 - 0.05(i-2)). \quad (9)$$

B. Evolution Schedule

Fig. 5 shows the evolution schedule used for the simulation study. In the figure, t corresponds to the presentation of the t-th incoming pattern vector to the HA-GRNN. In the simulation, the setting $t_2 = t_1 + 1$ was used. Note that the formation of LTM Net 1 was scheduled to occur after a relatively long exposition of incoming input pattern vectors, as described in Section IV-A. Note also that, with this setting, it requires that the RBFs in LTM Net 1 should be effectively selected from the previously (i.e., the

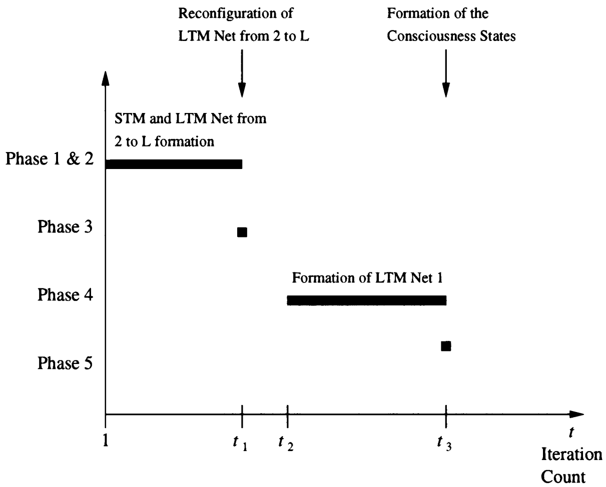

Fig. 5. Evolution schedule setup for the simulation study.

TABLE II
PARAMETERS FOR THE EVOLUTION OF THE HA-GRNN

Parameter	SFS	OptDigit	PenDigit
t_1	200	400	400
t_3	400	800	800

time before t_1) spanned pattern space. Thus, the self-evolution (in Phase 3) was schedule to occur at t_1 with $p = 2$ (i.e., the self-evolution was performed twice, and it was empirically found that this setting does not give any impact on the generalization performance) in the simulation. In Table II, the setting of t_1 and t_3 (which covers all the five phases) is summarized. The evolution was eventually stopped when all the incoming vectors in the training set were presented to the HA-GRNN.

C. Simulation Results

To test the classification accuracy of the HA-GRNN, the STM network was bypassed and the generalization performance over the testing was evaluated using only LTM Nets 1 to L, after the evolution. Moreover, the generalization performance of intuitive outputs generated during testing was also considered as another criterion.

Due to the limit of the space, only the confusion matrices of using the SFS data set are presented in this paper. Table III shows the confusion matrix obtained by the HA-GRNN after the evolution. In this case, no attentive states were formed at t_3. For comparison of the generalization capability, Table IV shows the confusion matrix obtained using a conventional GRNN with the same number of RBFs in each subnet as the HA-GRNN (i.e., a total of 85 RBFs were used.), where the respective RBFs are found by the well-known MacQueen's k-means clustering method [30]. To give a fair comparison, the RBFs in each subnet were obtained by applying the k-means clustering method to the respective (incoming pattern vector) subsets containing 54 samples per each digit (i.e., from Digit /ZERO/ to /NINE/).

In comparison with the conventional GRNN as in Table IV, it is clearly observed in Table III that, besides the superiority in

TABLE III
CONFUSION MATRIX OBTAINED BY THE HA-GRNN AFTER THE EVOLUTION

Digit	0	1	2	3	4	5	6	7	8	9	Total	Generalization Performance
0	29				3		2	1		1	29/36	80.6%
1		31			1	2				2	31/36	86.1%
2	1	0	28	2	2		1	2			28/36	77.8%
3				32	2	1		1			32/36	88.9%
4					36						36/36	100.0%
5		3			1	27		2		3	27/36	75.0%
6							32	2	2		32/36	88.9%
7								36			36/36	100.0%
8							1	1	34		34/36	94.4%
9		4				10		1		21	21/36	58.3%
Total											306/360	85.0%

TABLE IV
CONFUSION MATRIX OBTAINED BY THE CONVENTIONAL GRNN USING k-MEANS CLUSTERING METHOD

Digit	0	1	2	3	4	5	6	7	8	9	Total	Generalization Performance
0	34			1	1						34/36	94.4%
1		17				19					17/36	47.2%
2			28		8						28/36	77.8%
3			3	22	10	1					22/36	61.1%
4					36						36/36	100.0%
5						36					36/36	100.0%
6							34				36/36	100.0%
7	1		3		2	5		19			19/36	52.8%
8					2	1	7		26		26/36	72.2%
9			1			27				8	8/36	22.2%
Total											262/360	72.8%

the overall generalization performance of the HA-GRNN, the generalization performance in each digit, except Digit /NINE/, is relatively consistent, while the performance with the conventional GRNN varies from digit to digit as in Table IV. This indicates that the pattern space spanned by the RBFs obtained by the k-means clustering method is rather biased.

1) Generation of the Intuitive Outputs: For the SFS data set, the intuitive outputs were generated three times during the evolution, and all the three patterns were correctly classified for Digits /FOUR/ and /EIGHT/. In contrast, during testing, 13 pattern vectors among 360 yielded the generation of the intuitive outputs from LTM Net 1 in which 12 out of the 13 patterns were correctly classified. It was then observed that the Euclidean distances between the twelve pattern vectors and the respective centroid vectors corresponding to their class IDs (i.e., digit number) were relatively small and, for some patterns, close to the minimum (i.e., the distance between that of Pattern Nos. 77, 88, 104, and 113, and the RBFs for Digits /SEVEN/, /EIGHT/, /FOUR/ and /THREE/, respectively, in LTM Net 1 were minimal). From this observation, it can therefore be confirmed that intuitive outputs are likely to be generated when the incoming pattern vectors are very close to the respective centroid vectors in LTM Net 1.

For the OptDigit, despite the slightly worse generalization performance by HA-GRNN (87.0%) compared with that of the GRNN with k-means (88.8%), the generalization performance for the 174 out of the 360 testing patterns which yielded the intuitive outputs was better, i.e., 95.1%. This indicates that the LTM Net 1 was successfully formed and contributed to the better performance. Moreover, as discussed in Section IV-A, this lead to a faster decision-making, since the intuitive outputs were generated, e.g., without the processing within the STM and the regular LTM Nets.

In contrast, for the PenDigit, while overall a better generalization performance was obtained by the HA-GRNN (89.3%) in comparison with that of the conventional GRNN (88.0%), only a single testing pattern yielded the intuitive output (in which the pattern was correctly classified). Then, by increasing the maximum number of allowable RBFs in LTM Net 1 (as in Table I, which was initially fixed to 15) to 100, the simulation was performed again. As expected, the number of generating intuitive outputs was increased to 14, in which all the 14 testing patterns were correctly classified.

2) Simulations on Modeling the Attentive States: In Table III, it is observed that the generalization performance for Digits /FIVE/ and /NINE/ is relatively poor. To study the effectiveness of having the attentive states within the HA-GRNN, the attentive states were considered for both Digits /FIVE/ and /NINE/.

Then, by following both the conjectures 2 and 3 in Section IV-B, 10 (20 for the PenDigit) among 30 RBFs within the STM network were fixed for the respective digits after evolution time at t_3. In addition, since the poor generalization performance for Digits /FIVE/ and /NINE/ was (perhaps) due to the insufficient number of the RBFs for those classes, the maximum numbers of the RBFs within LTM Net from 2 to 3, $M_{LTM_{2,i}}$ and $M_{LTM_{3,i}}$ ($i = 5$ and 10), respectively, were also increased.

In Table V, the confusion matrix obtained by the HA-GRNN with an attentive state of only Digit /NINE/ is shown. For this case, a total of 8 more RBFs in LTM Nets 2 and 3 (i.e., 4 more each in LTM Nets 2 and 3) which correspond to the first 8 (instead of 4) strongest activations were selected (following Phase 4 in Section III-A) and added into Sub-Net 10 within both the LTM Nets 2 and 3 (i.e., the total number of RBFs in LTM Nets 1 to 3 was increased to 93). As in the table, the generalization performance for Digit /NINE/ was improved at 63.9%, in comparison with that in Table III, while preserving the same generalization performance for other digits. It is interesting to note that the generalization performance for Digits /FIVE/ is also improved.

In contrast, Table VI shows a confusion matrix obtained with having the attentive states of both the digits /FIVE/ and /NINE/. Similar to the case with an attentive state of Digit /NINE/, a total of 16 such RBFs for the two digits were respectively added into Sub-Nets 6 and 10 within both the LTM Nets 2 and 3. (Therefore, the total number of RBFs in LTM Nets 1 to 3 was increased to 101). In comparison with Table III, the generalization performance for Digit /FIVE/ was remarkably improved, as well as Digit /NINE/.

TABLE V
CONFUSION MATRIX OBTAINED BY THE HA-GRNN AFTER THE EVOLUTION (WITH AN ATTENTIVE STATE OF DIGIT 9)

Digit	0	1	2	3	4	5	6	7	8	9	Total	Generalization Performance
0	29			1	3		2	1			29/36	80.6%
1		31			2	2				1	31/36	86.1%
2	1		28	2	2		1	2			28/36	77.8%
3				32	2	1		1			32/36	88.9%
4					36						36/36	100.0%
5		2			1	29		2		2	29/36	80.6%
6							32	2	2		32/36	88.9%
7								36			36/36	100.0%
8							1	1	34		34/36	94.4%
9		2				11				23	23/36	63.9%
Total											310/360	86.1%

TABLE VI
CONFUSION MATRIX OBTAINED BY THE HA-GRNN AFTER THE EVOLUTION (WITH ATTENTIVE STATES OF DIGITS 5 AND 9)

Digit	0	1	2	3	4	5	6	7	8	9	Total	Generalization Performance
0	29			1	3		2				29/36	80.6%
1		31			2	2				1	31/36	86.1%
2	1		28	2	2		1	2			28/36	77.8%
3				33	2			1			33/36	91.7%
4					36						36/36	100.0%
5		1			1	33				1	33/36	91.7%
6							32	2	2		32/36	88.9%
7			4					36			36/36	100.0%
8							1	1	34		34/36	94.4%
9		3		1		8				24	24/36	66.7%
Total											316/360	87.8%

From these observations, it is considered that, since the performance improvement for Digit /NINE/ in both the cases was not more than expected, the pattern space for Digit /NINE/ is much harder to fully cover than other digits.

For both the OptDigit and PenDigit data sets, a similar performance improvement to the SFS case was obtained; for the OptDigit, the performance of Digit /NINE/ was relatively poor (57.5%), then the number of the RBFs within each of LTM Nets 2 to 3 for Digit /NINE/ was increased from 2 to 8 (which yields the total number of RBFs in LTM Nets 1 to 3, 77) and the performance for Digit /NINE/ was remarkably increased at 67.5%, which resulted in the overall generalization performance of 87.5% (initially 87.0%).

Similarly, for the PenDigit, a performance improvement of 5.0% (i.e., from 80.0% to 85.0%) or Digit /NINE/ was obtained by increasing the number of RBFs from 4 to 6 in each of LTM Nets (2 to 5) for Digit /NINE/ only (then, the total number of RBFs in LTM Nets 1 to 5 is 183), which yielded the overall generalization performance of 89.8% (initially 89.3%).

VI. Conclusion

In this paper, the two psychological functions, intuition and attention, have been modeled using a newly proposed HA-GRNN. The concept of the HA-GRNN and its evolution have been motivated from both psychological and biological studies, in which the architecture is based upon an hierarchical memory system implemented with both the short-term and long-term memory models. It has been justified that the notions of intuition and attention can be interpreted within the framework of evolution of the HA-GRNN. In the simulation study, the models of both the psychological functions have been introduced to construct an HA-GRNN using the three domain data sets for pattern classification tasks. The effectiveness of the HA-GRNN has then been investigated and its superiority in comparison with a conventional GRNN using the k-means clustering method has been confirmed. Future work is directed toward the development of intelligent robots utilizing the concept of the HA-GRNN.

Acknowledgment

The author would like to thank all the reviewers and associate editors for their encouraging/insightful comments to improve the first draft of the paper.

References

[1] G. Matsumoto, Y. Shigematsu, and M. Ichikawa, "The brain as a computer," in *Proceedings of the International Conference on Brain Processes, Theories and Models*. Cambridge, MA: MIT Press, 1995.

[2] I. Aleksander, *Impossible Minds: My Neurons, My Consciousness*. London, U.K.: Imperial College Press, 1996.

[3] R. A. Brooks, "From earwigs to humans," in *Proceedings of the International Symposium on "Concept Formation, Thinking, and Their Development"*. Kyoto, Japan: Int. Inst. Advanced Studies, 1996.

[4] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot Automat.*, pp. 14–23, 1986.

[5] T. Kitamura, Y. Otsuka, and Y. Nakao, "Imitation of animal behavior with use of a model of consciousness—behavior relation for a small robot," in *Proc. 4th IEEE Int. Workshop Robot Human Comm.*, Tokyo, Japan, 1995, pp. 313–316.

[6] D. F. Specht, "A general regression neural network," *IEEE Trans. Neural Networks*, vol. 2, pp. 568–576, Nov. 1991.

[7] D. F. Specht, "Probabilistic neural networks," *Neural Networks*, vol. 3, pp. 109–118, 1990.

[8] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan College, 1994.

[9] T. Hoya and J. A. Chambers, "Heuristic pattern correction scheme using adaptively trained generalized regression neural networks," *IEEE Trans. Neural Networks*, vol. 12, pp. 91–100, Jan. 2001.

[10] R. C. Atkinson and R. M. Shiffrin, "Human memory: a proposed system and its control processes," in *The Psychology of Learning and Motivation*, K. W. Spence and J. T. Spence, Eds. New York: Academic, 1968, vol. 2, pp. 89–195.

[11] D. A. Reynold and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. Speech Audio Processing*, vol. 3, pp. 72–83, Jan. 1995.

[12] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer Verlag, 1995.

[13] *Proc. IEEE Int. Workshop Neural Networks Signal Process*, 2002.

[14] P. D. Wasserman, "Advanced methods in neural computing," in *Radial-Basis Function Networks*. New York: Van Nostrand Reinhold, 1993, ch. 8, pp. 147–176.

[15] J. G. Taylor, B. Horwitz, N. J. Shah, W. A. Fellenz, H.-W. Mueller-Gaertner, and J. B. Krause, "Decomposing memory: functional assignments and brain traffic in paired word associate learning," *Neural Networks*, vol. 13, no. 8–9, pp. 923–940, Oct./Nov. 2000.

[16] M. S. Gazzaniga, R. B. Irvy, and G. R. Mangun, "Memory systems," in *Cognitive Neuroscience—The Biology of the Mind*. New York: W. W. Norton, 1998, ch. 7.

[17] O. Hikosaka, S. Miyachi, K. Miyashita, and M. K. Rand, "Procedural learning in monkeys—possible roles of the basal ganglia," in *Perception, Memory and Emotion: Frontiers in Neuroscience*, T. Ono, B. L. McNaughton, S. Molotchnikoff, E. T. Rolls, and H. Nishijo, Eds. New York: Elsevier, 1996, pp. 403–420.

[18] N. Osaka, *Brain and Consciousness*. Tokyo, Japan: Asakura-Shoten, 1997.

[19] Y. Shigematsu, M. Ichikawa, and G. Matsumoto, "Reconstitution studies on brain computing with the neural network engineering," in *Perception, Memory and Emotion: Frontiers in Neuroscience*, T. Ono, B. L. McNaughton, S. Molotchnikoff, E. T. Rolls, and H. Nishijo, Eds. New York: Elsevier, 1996, pp. 581–599.

[20] L. R. Squire, *Memory and Brain*. Oxford, U.K.: Oxford Univ. Press, 1987.

[21] M. Minsky, "Emotions and the society of mind," in *Emotions and Psychopathology*, M. Clynes and J. Panksepp, Eds. New York: Plenum, 1988.

[22] C. I. Hovland, "Human learning and retention," in *Handbook of Experimental Psychology*, S. S. Stevens, Ed. New York: Wiley, 1951, pp. 613–689.

[23] P. A. Kolers, "Reading a year later," *J. Exper. Psychol.: Human Learn. Memory*, vol. 2, pp. 554–565, 1976.

[24] J. Eccles, "Evolution of consciousness," in *Brain Mechanisms of Perception and Memory: From Neuron to Behavior*, T. Ono, L. R. Squire, M. E. Raichle, D. I. Perrett, and M. Fukuda, Eds. Oxford, U.K.: Oxford Univ. Press, 1993.

[25] J. A. Hobson, *Consciousness*. New York: W. H. Freeman, 1999.

[26] M. Jibu and K. Yasue, *Quantum Brain Dynamics and Consciousness: An Introduction*. Amsterdam, The Netherlands: John Benjamins, 1995.

[27] N. Matsumoto, *The Brain and Biophysics*. Tokyo, Japan: Kyo-ritsu Shuppan Press, 1997.

[28] M. Huckvale, *Speech Filing System Vs3.0—Computer Tools for Speech Research*. London, U.K.: University College London, Mar. 1996.

[29] J. R. Deller Jr, J. G. Proakis, and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals*. New York: Macmillan, 1993.

[30] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Symposium on Mathematics and Statistical Probability*, 5th ed. Berkeley, CA: Univ. of California Press, 1967, vol. 1, pp. 281–297.

Tetsuya Hoya (M'01) was born in Tokyo, Japan, on September 15, 1969. He received the B.Sc. and M.Sc. degrees in electrical engineering from Meiji University, Japan, in 1992 and 1994, respectively, and the Ph.D. degree from the Imperial College of Science, Technology and Medicine, University of London, U.K., in 1998.

From April 1994 to September 1994, he was a Research Assistant at the Department of Electronics and Communication, Graduate School of Meiji University, Japan. He was then a student at Department of Electrical and Electronics Engineering, Imperial College of Science, Technology and Medicine, from October 1994 to December 1997. He was a Postdoctoral Research Associate at the Department of Electrical and Electronics Engineering, Imperial College, London, from September 1997 to August 2000. Since October 2000, he has been a Research Scientist within the Brain Science Institute, RIKEN (The Institute of Physical and Chemical Research), Japan. His research interest focuses on a wide spectrum of brain science: artificial intelligence, cognitive neuroscience, combinatoric optimization, consciousness studies, electroencephalography, neural networks (connectionism), philosophy, psychology, robotics, and signal processing. He is a committee member of the Fourth International Symposium on Independent Component Analysis and Blind Signal Separation (ICA-2003).