A Cerebral Cortex Model that Self-Organizes Conditional Probability Tables and Executes Belief Propagation

Yuuji ICHISUGI

Abstract—**This paper describes a neural network model of cerebral cortex, *BESOM model*, that acquires conditional probability tables for a Bayesian network using self-organizing maps and estimates states of random variables with an approximate belief propagation algorithm. The approximate algorithm is derived from some assumptions. A neural network that executes the derived algorithm is in good agreement with six-layer and column structures that represent the anatomical characteristics of a cerebral cortex in many respects. This model has scalable time and space complexities and is therefore qualified to be a model of the brain, a large-scale information processor.**

I. INTRODUCTION

Through intensive neuroscience studies in recent years, the roles of individual areas in the cerebral cortex and enormous amounts of information on the anatomical structure of area connections have been accumulated and organized. However, understanding of the cortex is not enough to reproduce the main functions of the cortex on a computer.

Self-organizing map (SOM) [1] is a model that reproduces some cortex functions on a computer. SOM is a machine learning algorithm that is characterized by competitive learning and neighborhood learning. An SOM can achieve clustering high-dimensional input under unsupervised conditions and online. This agrees with one of the features of cerebral information processing. The column structure seen in the primary visual cortex and other areas is evidence that suggests that the cortex is a sort of SOM.

A Bayesian network[2] has also been used as a model for the cortex[6]. It is a graphical model that represents a directed acyclic graph of causal relations between random variables. Observations of some random variables within the network allow for estimation of the values of the remaining random variables based on conditional probability tables. For this estimation, an efficient calculation method, called a belief propagation algorithm, is used. This network has features such as noise-resistant pattern recognition, ambiguous information-based plausible inference and real-time operation that are also in agreement with features of cerebral information processing. The cortex areas form a bidirectionally linked network. This structure is similar to that of the Bayesian network.

One visual cortex model that has a mechanism similar to both SOM and a Bayesian network is the Selective Attention Model (SAM)[3]. The SAM, a hierarchical combination of competitive learning nodes, has bidirectional signal routes, a bottom-up signal that sends the results of individual nodes' recognition, and a top-down one that sends predictions based on past experiences and contexts. The SAM reproduces some of the features of cerebral visual information processing such as noise-resistant recognition and target segmentation, and is considered to be a plausible model for a cerebral cortex.

In addition to the SAM, there are models for the cortex in which a top-down signal represents predictions [4][5][6]. These models, however, have no self-organizing mechanism.

Traditional models also have the issue of scalability. The cortex is a large-scale information processor that consists of 14 billion neurons and should use a computational algorithm that runs at a realistic speed through parallel processing. However, the order of time and space complexity is not discussed in traditional models.

In this paper, we propose a BESOM (BidirEctional SOM) as a model for the cortex that has both the features of the SOM and the Bayesian network. The BESOM is a hierarchical SOM with bidirectional connections. Each SOM is used to self-organize and obtain a conditional probability table for the network. The BESOM replaces the SAM's learning and recognition calculating formula with one based on a belief propagation algorithm, thereby providing a footing for further expansion of the model and for efficient calculation.

This paper consists of five subsequent sections. Section II provides an overview of the BESOM model's architecture. Sections III and IV detail the learning and recognition steps, respectively. In Section IV, scalability is also discussed. Section V describes the agreement between this model and the anatomical features of a cerebral cortex. Section VI provides conclusions.

II. ARCHITECTURE

The BESOM consists of *nodes* that are connected in the form of a directed acyclic graph. Each node consists of several *units*. If two nodes are linked by an edge, the units included in each node are connected in a complete bipartite graph (Fig. 1). Each unit connection has a weight that varies with learning. (Section IV describes the structure of units more accurately.) It was assumed that a network structure does not vary with learning and is given as a prior knowledge.

The BESOM repeats the *learning step* and the *recognition step* alternately. A learning step updates weights of connections based on the result of the previous recognition step. A recognition step calculates outputs of nodes, which represent the state of the world, based on the observed values

Yuuji ICHISUGI is with National Institute of Advanced Industrial Science and Technology(AIST), Tsukuba Central 2, Tsukuba, Ibaraki 305-8568, Japan. (e-mail: y-ichisugi@aist.go.jp)

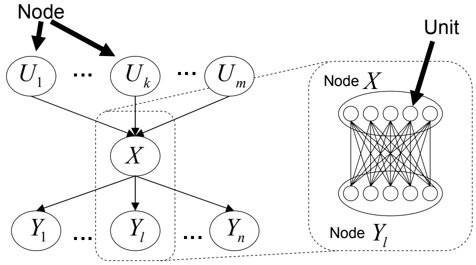

Fig. 1. BESOM Architecture. See Fig. 6 for unit structure and detailed connections among units.

and the current conditional probability tables.

In the learning step, each node works as a SOM's competitive layer. Each node clusters input vectors that are sent from child nodes. The result of the SOM's learning can be regarded as a conditional probability, as described in the next section. The updated conditional probabilities are used at the next recognition step. The network of SOMs form a hierarchical structure such that SOMs at the higher layers of a network express more abstract information that compresses more input information than those at the lower layers.

In the recognition step, a network of nodes works as a Bayesian network. Each node represents a random variable. Each unit included in one node corresponds to a possible value of the random variable. External inputs from sensors (observed values) are given as outputs of the lowest nodes (nodes that have no child nodes). In the recognition step, in accordance with external inputs and each node's conditional probability table, an approximate belief propagation algorithm (described in Section IV) is used to compute posterior probabilities of individual variables. The results are used in the next learning step.

In terms of a model for a cerebral cortex, the BESOM is interpreted as follows. The BESOM's nodes and units are equivalent to the hyper-columns and columns of the cortex. In the primary visual cortex, individual units are equivalent to the orientation columns.

III. Learning Step

In the learning step, each node works as an SOM competitive layer and clusters inputs from its child nodes.

Assume that Node X has n child nodes $Y_l (l = 1,\cdots,n)$. In this step, the SOM receives estimated values of the individual child nodes as inputs. Given that a estimated posterior probability of Node Y_l's unit y_i^l is $BEL(y_i^l)$, an element of an input vector v^l from Y_l is expressed as follows.

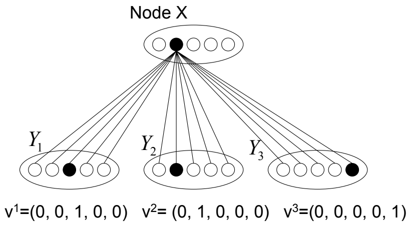

Fig. 2. Input vectors to the SOM in the learning step. Each black unit corresponds to the value with maximum posterior probability.

$$v_j^l = \begin{cases} 1 \ (\text{if } BEL(y_j^l) = \max_i BEL(y_i^l)) \\ 0 \ (\text{otherwise}) \end{cases} \quad (1)$$

Namely, an element corresponding to a unit with a maximum posterior probability is 1, while other elements are 0 (Fig. 2).

The estimated value of Node X becomes a winner for competitive learning. In the winner unit, a reference vector (a vector of weights) is brought close to an input vector. Assuming that the weight of the connection between Node X 's winner unit x_i and Node Y_l 's unit y_j^l is w_{ij}^l, this weight is then updated with the following rule:

$$w_{ij}^l \leftarrow (1-\alpha) w_{ij}^l + \alpha v_j^l \quad (2)$$

where α is a learning rate. In this step, neighborhood learning should also be done using a proper neighborhood function.

Here, given that the neighborhood radius is sufficiently small and negligible, and a learning rate α is equal to $1/n$ for n-th learning of each unit, a weight w_{ij}^l is then equal to a conditional probability $P(Y_l = y_j^l \mid X = x_i)$. (For details, see the appendix.)

IV. Recognition Step

In the recognition step, an approximate belief propagation algorithm is executed to estimate values of the individual random variables. This section indicates the problems in introducing a belief propagation algorithm into a neural network model and provides solutions to these problems. The characteristics of the derived approximate algorithm are then discussed.

A. Approximation of conditional probability tables

Bayesian networks have a problem in that the size of a conditional probability table $P(X \mid U_1,\cdots,U_m)$ increases exponentially against the number of parent nodes, m . At the same time, the execution cost of the belief propagation algorithm [2] (Fig. 3) also increases in the same order. (In Fig.3, the variable $BEL(x)$ is an abbreviation of

$BEL(X = x_i)$, that is an estimated posterior probability of a value $x_i \, (i = 1, \cdots, s)$ of a random variable X, where s is a number of possible values of X. The variable $\pi_X(u_k)$ is an abbreviation of $\pi_X(U_k = u_i^k)$, that is a message to a node X from a parent node U_k concerning a value u_i^k. Other variables are abbreviated in the same manner.)

In the case of BESOM as a model of a cerebral cortex, it may be assumed that the individual nodes included in the same layer learn different sets of features, and that their individual recognition results are almost independent of each other. In addition, the number of input features required by the status of a certain node may be assumed to be small because of the sparseness of neuron firings. In this situation, if at least one of Node X's parent nodes requires a feature x, it can be estimated that this feature will exist. Based on these assumptions, a conditional probability table for Node X can be approximated with the sum of conditional probabilities learned by parent's SOMs.

$$P(X \mid U_1, \cdots, U_m) \approx \sum_{i=1}^{m} P(X \mid U_i) \qquad (3)$$

This assumption is similar to that of the noisy-OR model [2], so it can be expected that this situation will result in a simple belief propagation algorithm and reduce the time and space complexity substantially.

It is necessary to verify the establishment of (3) with a physiological experiment or a computer simulation. In this paper, (3) is assumed to be given and an approximate belief propagation algorithm is derived from this assumption.

B. *Information from message receiver*

Each formula for $\pi_{Y_l}(x)$ and $\lambda_X(u_k)$ in the algorithm in Figure 3 excludes information came from the node that will receive a messages calculated by the formula. This causes complexity of the algorithm. If a network has a tree structure, the algorithm becomes simpler, as used in [6]. In cerebral cortex, however, generally one area has connections to several higher areas, so we cannot assume that the network is structured in a tree form.

In this paper, conversely, we assume that one node has considerably many parent and child nodes. By assuming this, information that supports values of a certain node's random variable are normally obtained from several nodes, so inclusion of information from a message receiver may not greatly affect the estimation results.

Based on this assumption, we use an approximation of "inclusion of information from a message receiver."

C. *Approximate belief propagation algorithm*

In addition to the two kinds of approximations described above, it is assumed that messages from parent nodes are normalized.

$$\sum_{u_k} \pi_X(u_k) = 1 \qquad (4)$$

$$BEL(x) = \alpha \lambda(x) \pi(x)$$
$$\pi(x) = \sum_{u_1, \cdots, u_m} P(x \mid u_1, \cdots, u_m) \prod_k \pi_X(u_k)$$
$$\lambda(x) = \prod_l \lambda_{Y_l}(x)$$
$$\pi_{Y_l}(x) = \pi(x) \prod_{j \neq l} \lambda_{Y_j}(x)$$
$$\lambda_X(u_k) = \sum_x \lambda(x) \sum_{u_1, \cdots, u_m / u_k} P(x \mid u_1, \cdots, u_m) \prod_{i \neq k} \pi_X(u_i)$$

Fig. 3. Original belief propagation algorithm.

Then, a belief propagation algorithm can be approximated, as shown in Figure 4. (See the appendix for a detailed derivation.)

In the approximated belief propagation algorithm, each node receives the values of $\lambda(y_l)$ and Z_{Y_l} from its child nodes and the values of $BEL(u_k)$ from its parent nodes and determines values of its own $\lambda(x)$, $BEL(x)$ and Z_X. The algorithm repeats these calculations until the values of the individual variables converge, like an original loopy belief propagation algorithm.

$\lambda(y_l)$ is the results of the child nodes' recognition with the use of mainly bottom-up information. $\kappa_{u_k}(x)$ is a prediction based on information from parent nodes. $BEL(x)$ is a posterior probability. Z_X is used for the normalization of $BEL(x)$ and also represents the degree of agreement between the predictions and observations in Node X.

It is interesting that the term Z_{Y_l} appears in a formula for $\lambda_{Y_l}(x)$. Inputs from a node with a good agreement between predictions and observations have a smaller effect on the recognition of parent nodes. In other words, the features not agreed with predictions are emphasized on the recognition. (This effect results from Assumption (3).)

The values of individual variables, which only arise from a very simple calculation, can be sufficiently implemented with a neural network. In particular, an inner product calculation conducted using $\lambda_{Y_l}(x)$ or $\kappa_{u_k}(x)$ is suitable for neuron execution. In addition, a conditional probability table can be obtained from Hebbian learning by synapses linking to these neurons. (Note that one conditional probability $P(x \mid u_k)$ is learned by two neurons, $\kappa_{u_k}(x)$ and $\lambda_X(u_k)$, simultaneously.)

D. *Scalability*

Because input vectors in the learning steps are sparse, the obtained conditional probability table may also be sparse.

$$\lambda_{Y_l}^{t+1}(x) = Z_{Y_l}^{t} + \sum_{y_l} \lambda^{t}(y_l) P(y_l \mid x)$$

$$\lambda^{t+1}(x) = \prod_{l=1}^{n} \lambda_{Y_l}^{t}(x)$$

$$\kappa_{U_k}^{t+1}(x) = \sum_{u_k} P(x \mid u_k) BEL^{t}(u_k)$$

$$\pi^{t+1}(x) = \sum_{k=1}^{m} \kappa_{U_k}^{t+1}(x)$$

$$\rho^{t+1}(x) = \lambda^{t+1}(x) \pi^{t+1}(x)$$

$$Z_X^{t+1} = \sum_{x} \rho^{t+1}(x)$$

$$BEL^{t+1}(x) = \rho^{t+1}(x) / Z_X^{t+1}$$

Fig. 4. Derived approximate belief propagation algorithm.

The number of non-zero elements of an input vector is a constant number (number of child nodes) and independent of the number of units s within each child node. The resulting conditional probability table may inevitably have the same features. Taking advantage of this characteristic, we can lower the order of time and space complexities. In the case of cortex, synapses with a weight of 0 can be eliminated. This can limit the explosive increase of synapses required.

A space complexity that is required to express one unit's conditional probability table (reference vector) is $O(s)$, which can be reduced to $O(1)$ by taking advantage of table's sparseness.

A time factor that requires an inner product calculation made by $\lambda_{Y_l}(x)$ and $\kappa_{u_k}(x)$ will reach $O(1)$ by the same reasoning. Therefore, a time for all inner product calculations in parallel computation within a node is $O(1)$. The time complexities in parallel computation for other variables are $O(\log s)$ or less.

As shown above, the approximate algorithm derived in this paper is scalable. Hence, it can be said to be qualified as a model for the cerebral cortex's information processing algorithm in terms of time and space complexities.

V. Corresponding to Neuroscience Findings

A. Anatomical characteristics of a cerebral cortex

The cerebral cortex has a six-layer structure. Areas of the cortex are bidirectionally connected. These connections are known to have the following regularities[7]. Bottom-up connections are directed from layer III to IV. Some areas also have connections from layer V to IV. Top-down connections are directed mainly from layers V and VI to layer I. There are also a few connections from layer III to layer I.

Furthermore, based on a column's anatomical structure, information input to layer IV is considered to be output from a-layer V via layers II and III within the column[8].

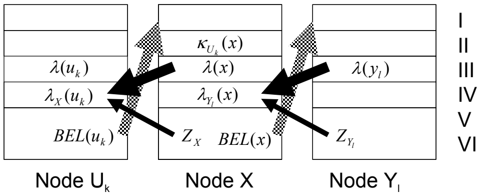

Fig. 5. Correspondence of approximate belief propagation algorithm and the six-layer structure of cerebral cortex.

Considering these two findings, the six-layer cortex structure forms a very strange construction in which the information in layer III, a provisional result of information processing within the column, is sent to a higher area, and the information in layer V, a final result, is sent back to a lower area. The functional meaning of this structure is unknown.

B. Corresponding to the approximate algorithm

Of the seven variables appearing in the approximate belief propagation algorithm, five variables related to node communication are applied to an area connection rule. The result is shown in Figure 5. We selected a layer II, not I, for $\kappa_{u_k}(x)$ because the layer I contains small number of cells. Layers V and VI were selected for Z_X based on a depth related to $BEL(x)$, although a layer III can be considered.

As shown in the Figure 5, an approximate algorithm can be directly corresponded to the connection rule, and cannot be considered to be coincidentally correspondent to it. (Top-down connections from layer III to I cannot be explained by this model.)

Figure 6 shows that the seven kinds of variables are placed in a column so as to be consistent with anatomical findings. A flow of information in the order of layers IV, II/III and V, as described above, corresponds with that in an order of variables of $\lambda_{Y_l}(x)$, $\lambda(x)$, $\rho(x)$ and $BEL(x)$. (Based on the findings, information further flows in the order of layers V, VI and IV[8]. This information route cannot be explained by this model.)

In addition, the following agreements can be observed with the findings: 1) almost all information processing is conducted vertically within the column, 2) many horizontal fibers are seen in layers I and IV and 3) there are many small cells in layers II and IV.

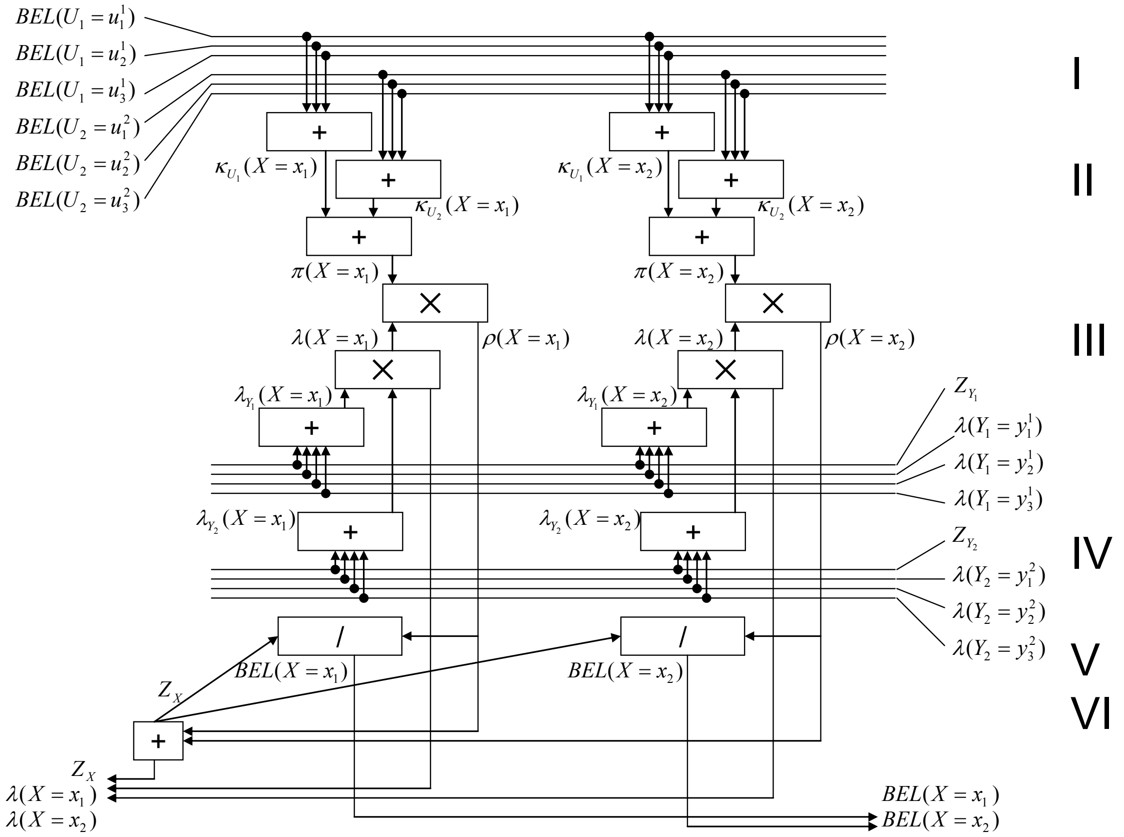

Fig. 6. Neural network which executes the approximate belief propagation algorithm. This figure shows the flow of information between variables within two units, x_1 and x_2, of Node X. Node X receives information from two parent nodes, U_1 and U_2, and two child nodes, Y_1 and Y_2. The network structure is similar to the column structure of a cerebral cortex in many respects (see main text).

C. Same layer's area connection problems

The cerebral cortex sometimes has horizontal connections between two areas that cannot be explained in the current BESOM model. We think these connections are communication route with a different purpose, such as independent component analysis.

VI. Conclusion

An approximate belief propagation algorithm was derived for bidirectionally connected SOM network, and it was demonstrated that the algorithm both corresponds well to the six-layer and column structures of the cerebral cortex and is scalable. This model represents firing rate of neurons within the cortex that might be observed in physiological experiments.

In order to simulate this model and conduct large-scale tests, the mechanism to make nodes within the same layer independent each other must be elucidated. We are currently tackling this challenge, by introducing a mechanism of independent component analysis to this model. If this issue is solved, it will allow for simulations to be performed verifying the adequacy of an approximate expression (3) in a conditional probability table.

In addition to the basic functions of the BESOM model described in this paper, we think it is necessary to extend the model, for instance, to include selective attention and short-term memory mechanisms.

The BESOM network has a very high level of expressive power. We have started to model a mechanism for acquiring action sequences done by premotor and supplementary motor areas, and also model a mechanism for acquiring a state transition table of external world used for action planning by prefrontal area. Both models are expressed by BESOM network combined with a reinforcement learning mechanism that is a model of the basal ganglia.

In the future, we aim to reproduce main brain functions on computers.

Appendix

It can be shown that weights obtained in learning steps are regarded as conditional probabilities as follows. Let $v_j(n) \in \{0,1\}$ be an j-th element of an input vector from X's child node Y and let $w_{ij}(n)$ be learning result, where n means n-th learning of a unit x_i. (We do not consider neighborhood learning here.) Let $m(n)=\sum_{i=1}^{n} v_j(i)$ be the number of times that a unit y_j becomes the estimated value of Node Y. Let $\alpha = 1/n$ be a learning ratio. Given $w_{ij}(1)=v_j(1)=m(1)$, the value of $w_{ij}(n)$ $(n>1)$ equals the following.

$$
\begin{aligned}
& w_{ij}(n)=(1-\alpha)w_{ij}(n-1)+\alpha v_j(n) \\
& =((n-1)w_{ij}(n-1)+v_j(n))/n \\
& =((n-1)(m(n-1)/(n-1))+v_j(n))/n \\
& =(m(n-1)+v_j(n))/n \\
& =m(n)/n
\end{aligned}
$$

This value is a ratio between "the number of times that a unit y_j becomes the estimated value of Node Y" and "the number of times that a unit x_i becomes the estimated value of Node X." Assuming that the estimation results are right, this value is the conditional probability $P(y_j \mid x_i)$.

An approximate belief propagation algorithm used for learning steps is derived as follows.

First, $\pi(x)$ can be approximated as follows, using (3).

$$
\begin{aligned}
& \pi(x)=\sum_{u_1,\cdots,u_m} P(x \mid u_1,\cdots,u_m)\prod_{i}\pi_X(u_i) \\
& \approx \sum_{u_1,\cdots,u_m} (\sum_{k} P(x \mid u_k))\prod_{i}\pi_X(u_i) \\
& =\sum_{u_1,\cdots,u_m}\sum_{k} P(x \mid u_k)\prod_{i}\pi_X(u_i) \qquad (6) \\
& =\sum_{k}\sum_{u_k} P(x \mid u_k)\pi_X(u_k)\sum_{u_1,\cdots,u_m/u_k}\prod_{i\neq k}\pi_X(u_i) \qquad (7)
\end{aligned}
$$

Here, the following equation is established from (4), assumption for normalization.

$$
\sum_{u_1,\cdots,u_m/u_k}\prod_{i\neq k}\pi_X(u_i)=1 \qquad (8)
$$

Hence, (7) is converted to the following expression.

$$
\pi(x)=\sum_{k}\sum_{u_k} P(x \mid u_k)\pi_X(u_k)
$$

$\pi_{Y_l}(x)$ can be approximated as follows by "including information from a message receiver".

$$
\begin{aligned}
& \pi_{Y_l}(x)=\pi(x)\prod_{j\neq l}\lambda_{Y_j}(x) \\
& \approx \pi(x)\prod_{j}\lambda_{Y_j}(x) \\
& =\lambda(x)\pi(x)
\end{aligned}
$$

The following equation included in a formula for $\lambda_X(u_k)$ can be approximated as follows, using (3) and (8).

$$
\begin{aligned}
& \sum_{u_1,\cdots,u_m/u_k} P(x \mid u_1,\cdots,u_m)\prod_{i\neq k}\pi_X(u_i) \\
& \approx \sum_{u_1,\cdots,u_m/u_k} (\sum_{j\neq k} P(x \mid u_j)+P(x \mid u_k))\prod_{i\neq k}\pi_X(u_i) \\
& =\sum_{u_1,\cdots,u_m/u_k}\sum_{j\neq k} P(x \mid u_j)\prod_{i\neq k}\pi_X(u_i) \\
& +P(x \mid u_k)\sum_{u_1,\cdots,u_m/u_k}\prod_{i\neq k}\pi_X(u_i) \\
& =\sum_{u_1,\cdots,u_m/u_k}\sum_{j\neq k} P(x \mid u_j)\prod_{i\neq k}\pi_X(u_i)+P(x \mid u_k)
\end{aligned}
$$

By "including information from a message receiver" and (6), the above formula is approximated as follows.

$$
\begin{aligned}
& \sum_{u_1,\cdots,u_m/u_k}\sum_{j\neq k} P(x \mid u_j)\prod_{i\neq k}\pi_X(u_i)+P(x \mid u_k) \\
& \approx \sum_{u_1,\cdots,u_m}\sum_{j} P(x \mid u_j)\prod_{i}\pi_X(u_i)+P(x \mid u_k) \\
& \approx \pi(x)+P(x \mid u_k)
\end{aligned}
$$

By substitution of the above equation, $\lambda_X(u_k)$ can be approximated as follows.

$$
\begin{aligned}
& \lambda_X(u_k)=\sum_{x}\lambda(x)\sum_{u_1,\cdots,u_m/u_k} P(x|u_1,\cdots,u_m)\prod_{i\neq k}\pi_X(u_i) \\
& \approx \sum_{x}\lambda(x)(\pi(x)+P(x \mid u_k)) \\
& =\sum_{x}\lambda(x)\pi(x)+\sum_{x}\lambda(x)P(x \mid u_k)
\end{aligned}
$$

Arranging the above results, we obtain the algorithm shown in Figure 4.

References

[1] T. Kohonen, Self-Organizing Maps. Springer-Verlag, 1995.

[2] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, 1988.

[3] K. Fukushima, Neural network model for selective attention in visual-pattern recognition and associative recall, APPLIED OPTICS 26 (23): 4985-4992 Dec 1 1987.

[4] M. Kawato, H. Hayakawa, T. Inui: A forward-inverse optics model of reciprocal connections between visual areas. Network: Computation in Neural Systems 4, 415-422, 1993.

[5] R.P.N. Rao and D.H. Ballard, Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects, Nature Neuroscience, Vol.2 No.1, pp.79-87, Jan 1999.

[6] George, D. Hawkins, J., A hierarchical Bayesian model of invariant pattern recognition in the visual cortex, in proc. of IJCNN 2005, vol. 3, pp.1812-1817, 2005.

[7] Pandya, D.N. and Yeterian, E.H., Architecture and connections of cortical association areas. In: Peters A, Jones EG, eds. Cerebral Cortex (Vol. 4): Association and Auditory Cortices. New York: Plenum Press, 3-61, 1985.

[8] Gilbert, C.D., Microcircuitry of the visual-cortex, Annual review of neuroscience, 6: 217-247, 1983.