2011 International Symposium on Nonlinear Theory and its Applications
NOLTA2011, Kobe, Japan, September 4-7, 2011

A theory of distributed code integration for cortical circuits

Taichi Kiwaki†, Tetsuya J. Kobayashi‡, and Kazuyuki Aihara‡

†Graduate School of Engineering, University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8685, Japan
‡Institute of Industrial Science, University of Tokyo,
4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan
Email: kiwaki@sat.t.u-tokyo.ac.jp, tetsuya@mail.crmind.net, aihara@sat.t.u-tokyo.ac.jp

Abstract— The brain is a parallel distributed system, where its information representations are thought to be dynamically formed by experience. The problem here is how those representations, or neural codes, are coherently integrated throughout the brain. In this article, we address this problem based on the hypothesis that Bayesian computation is taken place in the brain. We formulate biologically plausible forms of information representations, computation, and learning for the integration of neural codes over distant cortical areas. We then provide a simulation result to demonstrate the effectiveness of our theory.

1. Introduction

1.1. A problem of code integration in the brain

The brain seems to carry out computation in a parallel fashion[1]. For coherent computation, information encoded in a local cortical circuit should be accessible for the other circuits to proceed their task; this can be done through interareal communication within the brain. We address the problem of how these cortical circuits can know the foreign encoding format of information to decode the received signals.

This problem seems relative easy if the neural representation of information, or neural code, is fixed. However, it is considered that the neural code is dynamically formed, influenced by experience as well as static innate effects that are genetically programmed[2]. Therefore it is implausible that the way to decode the foreign codes is hard-coded. Alternative possibility is that the neural codes are integrated dynamically throughout the cortical circuits; the circuits learn how to decode signals only with locally available information.

In this article we aim at deriving a reasonable learning algorithm to realize this code integration. Historically, this problem has been regarded as a variant of the famous binding problem and attracted much attention in the field of neuroscience[3]. We deal with this problem by providing a computational model, based on the idea that Bayesian computation is employed in the brain.

1.2. Bayesian computation in the brain

It is recently demonstrated that Bayesian computation might be performed in the brain, by some psychological experiments[4, 5]. Although this possibility is currently only suggested in primitive cognition, this idea is considered to provide an unique computational framework in the brain in the long run. In this study we hypothesize that Bayesian computation is carried out in the brain, so that we can maintain the universality of the theory.

2. A computation model and algorithms

2.1. Bayesian computation

Here we formulate the computation that we consider in this article. Although many types of computation can be considered, we focus on a problem called multimodal integration in the field of neuroscience. In multimodal integration, the neural system estimates the states of the environment by integrating more than two cues which are gained through the different sensory systems. This problem is often used in the theoretical studies on the Bayesian computation in the brain and captures the essence of the idea[6].

It is straightforward to formulate the problem of multimodal integration in the line of the Bayesian computation; we suppose that probabilistic density functions on the environment are somehow represented in the brain and estimation is made by manipulating them [5, 6]. For simplicity, we here only deal with the case where the hidden environmental state, denoted by h, is a scalar. Moreover, we also assume that the number of the sensory systems is two, and the information on the real value of h, h^* is obtained through observations made by these sensory systems. Those observations are denoted by o_0 and o_1, and the information on h^* that they contain is represented by posterior distributions as $p(h = h^*|o_0)$ and $p(h = h^*|o_1)$. Here we make another assumption that those observations are statistically independent with given h^*.

In the Bayesian framework, the information in those two observations, o_0 and o_1 is integrated coherently by constructing a new posterior distribution based both on o_0 and o_1. Under above conditions, this integrated posterior can be obtained by simply multiplying the two posteriors asso-

ciated with o_0 and o_1,

$$p(h|o_0,\ o_1) \propto p(h|o_0)\, p(h|o_1), \tag{1}$$

where a flat prior is assumed and $h = h^*$ is abbreviated to h[5, 6]. The outline of this computation is illustrated in Fig. 1. The final decision on h is made with the integrated posterior through a proper estimation scheme, for example, maximum posterior (MAP) estimation [7].

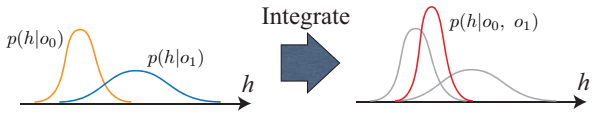

Figure 1: Formulation of multimodal integration in the Bayesian framework: two posterior distributions are first computed based on the observations (the orange and blue curves in the left), and then they are multiplied to yield an integrated posterior (the red curve in the right).

2.2. On the neural codes

We next consider how this Bayesian computation can be achieved in the neural systems. Particularly in this section, it is discussed what limitations are present in the neural systems and how they can affect possible realization of the neural codes.

If we hypothesize that Bayesian computation is taken place in the brain, the information in the circuits is represented in the form of posterior density functions. In that case, what limitations are there in expressing posterior distributions in the neural system? It is at least asserted that the neural systems are incapable of directly expressing continuous functions like posterior distributions, since this requires infinite computational capacity. Therefore, instead of functions, it seems reasonable to consider that the system manages vectors which approximate the original distribution. This approximation would naturally be done by quantizing the corresponding distribution function to yield vector components,

$$p(h^* \in H_i) = \int_{H_i} p(h)\, \mathrm{d}h, \tag{2}$$

where H_i is a quantization interval and i is its index.

As for the neural representation of this vector, we here hypothesize that the population code is used; each neuron in the circuit corresponds to a quantization interval and the value of the component associated with this interval is represented by the firing rate of the neuron as

$$p(h^* \in H_i) \propto q_i, \tag{3}$$

where the firing rate of ith neuron is denoted by q_i. This is illustrated in Fig. 2.

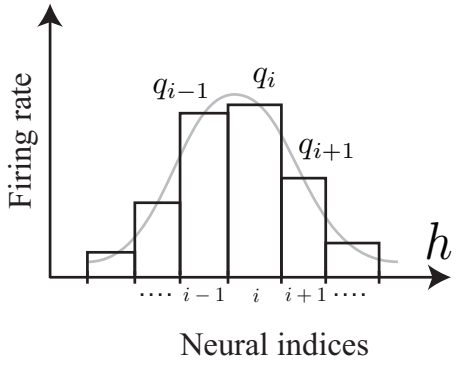

Figure 2: Neural representation of a density function: the neurons are numbered in the same manner as the corresponding interval and their firing rate represents the components of the quantized density function.

2.3. Discrete forms of computation

When considering Bayesian computation in the brain, we assume that the computation of Eq. (1) is somehow implemented in the neural system. Particularly, because we model the neural system as a distributed system, the computation of Eq. (1) is independently realized over the distant cortical circuits. Since we are concerning the case where only two sensory systems are involved, we here consider two cortical circuits, denoted by $\mathcal{S}^0$ and $\mathcal{S}^1$, each of them is responsible for each sensory system. Accordingly, one of the two posteriors in the r.h.s. of Eq. (1) is directly available in a local circuit, and the other is retrieved via communication; we call them local and distant posteriors, respectively.

As we discussed in the previous section, posterior distributions must be represented in a discrete form, so must be the expression of the computation, Eq. (1). If the quantization intervals and the neural mapping in the cortical circuits are the same, it is straightforward to derive a discrete form. With an assumption that quantization does not violate the independence condition, Eq. (1) can be expressed as

$$p(H_i|o_0,\ o_1) \propto p(H_i|o_0)\, p(H_i|o_1), \tag{4}$$

where $h^* \in H_i$ is abbreviated to H_i. However, this is mere a special case. Generally speaking, the intervals and the mapping do not coincide between distant cortical circuits; the foreign code of the distant circuit must be interpreted before integration. Fortunately, this interpretation seems to be realized with a relative simple form as we derive

$$p\left(H_k^0|o_0,\ o_1\right) \propto p\left(H_k^0|o_0\right) \left\{ \sum_{l'} p\left(H_k^0|H_{l'}^1\right) p\left(H_{l'}^1|o_1\right) \right\}, \tag{5}$$

where H_k^0 and H_l^1 are the quantization interval of $\mathcal{S}^0$ and $\mathcal{S}^1$, k and l are their neural index, and it is assumed that

events $h^* \in H^0_k$ and o_1 is statistically independent under a condition of $h^* \in H^0_k$. Although Eq. (5) is only for $\mathcal{S}^0$, the expression for $\mathcal{S}^1$ is the same except for the indices. We only show the results on $\mathcal{S}^0$ in the rest of this article, without loosing generality.

Next, let us consider what neural architecture can implement the computation of Eq. (5). First of all, it is necessary for the local circuits to have a set of neurons which encodes the local posterior, $p(h|o_0)$ or $p(h|o_1)$. We here denote its population activity by $\hat{q}^0_k(o_0)$ and $\hat{q}^1_l(o_1)$. In addition to that, another set of neurons is needed in order for the integrated posterior, $p(h|o_0,\ o_1)$ to be expressed in the neural system; we denote its activity by $q^0_k(o_0, o_1)$ and $q^1_l(o_0, o_1)$. We assume that the same neural code is used in those two neural populations. Equation (5) suggests that $q^0_k(o_0, o_1)$ successfully approximates the integrated posterior if it is computed as

$$q^0_k(o_0, o_1) \propto \hat{q}^0_k(o_0) \sum_{l'} A^{01}_{k,l'} \hat{q}^1_{l'}(o_1) \tag{6}$$

in the neural circuit, where we assume $A^{01}_{k,l}$ is given as a close approximation of $p\left(H^0_k|H^1_l\right)$. The main problem here is how $A^{01}_{k,l}$ and $A^{10}_{l,k}$ can be generated in the local circuit. If they can be learned independently in the cortical circuits based only on locally available information, we can provide a possible answer to the problem of code integration in the brain.

2.4. A learning algorithm for code integration

We here formulate a learning algorithm for $A^{01}_{k,l}$ and $A^{10}_{l,k}$. As shown in the previous section, the ideal values of $A^{01}_{k,l}$ and $A^{10}_{l,k}$ are the conditional distributions, $p\left(H^0_k|H^1_l\right)$ and $p\left(H^1_l|H^0_k\right)$. Since both of them can be obtained from the joint distribution, $p\left(H^0_k,\ H^1_l\right)$, we first examine the possibility of learning this joint distribution.

By expressing the joint distribution as a marginal distribution of $p\left(H^0_k,\ H^1_l,\ o_0,\ o_1\right)$, we obtain

$$p\left(H^0_k,\ H^1_l\right) = \iint p\left(H^0_k|\boldsymbol{o}\right)\, p\left(H^1_l|\boldsymbol{o}\right)\, p(\boldsymbol{o})\, \mathrm{d}o_0\, \mathrm{d}o_1, \tag{7}$$

where we assume that events, $h^* \in H^0_k$ and $h^* \in H^1_l$ become statistically independent with a given pair of o_0 and o_1, and use a vector notation of o_0 and o_1, $\boldsymbol{o}$. Notice that the r.h.s. of Eq. (7) can be seen as an ensemble mean of the product between $p\left(H^0_k|\boldsymbol{o}\right)$ and $p\left(H^1_l|\boldsymbol{o}\right)$, over any possible $\boldsymbol{o}$. When the time course of $\boldsymbol{o}$ is given to obey $p(\boldsymbol{o})$, this mean can be equivalently expressed as a temporal mean,

$$p\left(H^0_k,\ H^1_l\right) = \lim_{T\to\infty} \frac{1}{T} \int_0^T p\left(H^0_k|\boldsymbol{o}(\tau)\right)\, p\left(H^1_l|\boldsymbol{o}(\tau)\right)\, \mathrm{d}\tau, \tag{8}$$

where the time course is denoted by $\boldsymbol{o}(t)$.

In Eq. (8), $p\left(H^0_k|\boldsymbol{o}\right)$ and $p\left(H^1_l|\boldsymbol{o}\right)$ are the expressions of integrated posterior in the neural code of $\mathcal{S}^0$ and $\mathcal{S}^1$. Therefore, by substituting them by their neural representation, $q^0_k(\boldsymbol{o})$ and $q^1_l(\boldsymbol{o})$, we may be able to derive an expression for learning on the joint distribution. Based on this idea, we define a neural expression for the joint distribution, $\tilde{A}_{k,l}(t)$ as

$$\tilde{A}_{k,l}(t) = \int_0^t q^0_k(\boldsymbol{o}(\tau))\, q^1_l(\boldsymbol{o}(\tau))\, \exp\left(-\frac{t-\tau}{\tau_{\mathrm{L}}}\right) \mathrm{d}\tau, \tag{9}$$

where we made the substitution and introduced a learning time constant, τ_{L}, for the system to be able to accommodate a change in $p(\boldsymbol{o})$. Since $\tilde{A}_{k,l}(t)$ may approximate the joint distribution, we can obtain $A^{01}_{k,l}$ and $A^{10}_{l,k}$ from it just as we get conditional distributions from a joint one.

To run this algorithm, all needed for the local circuits is the activities of the distant neurons that encode the integrated posterior, $p(h|\boldsymbol{o})$; this is thought to be available through interareal communication. Figure 3 illustrates our neural model for the learning algorithm.

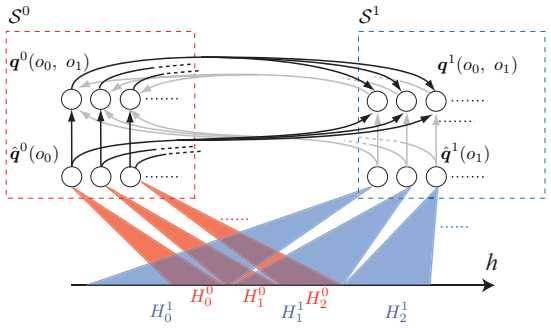

Figure 3: The model architecture. Two models of cortical circuit, $\mathcal{S}^0$ and $\mathcal{S}^1$ have two layers of neurons, the upper one for the integrated posterior and the lower one for the local posterior. These two layers are mapped onto a set of intervals in h, $\{H^0_k\}$ for $\mathcal{S}^0$ and $\{H^1_l\}$ for $\mathcal{S}^1$. The neural connections from $\mathcal{S}^0$ is displayed in black and those from $\mathcal{S}^1$ is in gray. For computation, the upper layer receives parallel connections from the local lower layer and cross connections from all of the neurons in the distant lower layer. For leaning, neurons in one of the upper layers receive connections from all of the neurons in the other upper layer.

When we see this learning algorithm as a dynamical system driven by a stochastic variable, $\boldsymbol{o}(t)$, it seems that the line, $\tilde{A}_{k,l} \propto p\left(H^0_k,\ H^1_l\right)$ becomes a set of fixed points of this dynamics. The reason for this is as follows. First, if $\tilde{A}_{k,l} \propto p\left(H^0_k,\ H^1_l\right)$ holds, then the cortical circuits can successfully compute $q^0_k(\boldsymbol{o})$ and $q^1_l(\boldsymbol{o})$ through Eq. (6). Second, it is asserted that $q^0_k(\boldsymbol{o})$ and $q^1_l(\boldsymbol{o})$ now closely approximate $p\left(H^0_k|\boldsymbol{o}\right)$ and $p\left(H^1_l|\boldsymbol{o}\right)$, Eq. (8) therefore suggests that $\tilde{A}_{k,l}$ will not deviate from the original line as far as its development conforms to Eq. (9). The problem left here is its stability and basin structure. If this line is shown to be a

stable attractor, then it is possible to learn proper values of $\tilde{A}_{k,l}$, from a certain initial values.

3. Numerical verification

To verify whether the learning algorithm proposed in **2.4** functions properly or not, we here show some results of a numerical simulation. In this simulation, we randomly generate h^* from an uniform distribution on an interval, $[0, 1]$, and then generate local posteriors, $p(h|o_0)$ and $p(h|o_1)$ to run the learning algorithm. These distributions are given as a gaussian distribution; we use their variance as one of the principal parameters in this experiment. It is also important how the quantization intervals are defined; in this experiment, they are independently made in the cortical circuits, with a fixed interval width. These widths are another parameter of this experiment.

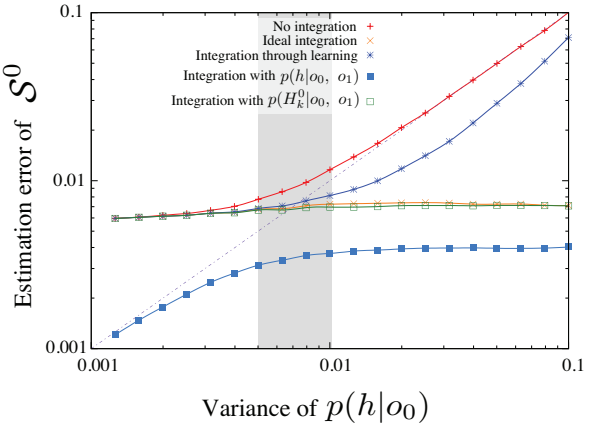

Figure 4: The performance of the system. The vertical axis shows the deviation of the estimation in $\mathcal{S}^0$ from the real value, h^*, and the horizontal axis shows the variance of the local posterior. Estimation is made by MAP estimation. The widths of quantization intervals in $\mathcal{S}^0$ and $\mathcal{S}^1$ are 0.02 and 0.005, respectively. The variance of the distant posterior, $p(h|o_1)$ is 0.003. The each curve in the graph represents the case without integration (shown in red +), the case where ideal values of $\tilde{A}_{k,l}$ is used (orange $\times$), the case where $\tilde{A}_{k,l}$ is learned (purple $*$), the case where integration is made by $p(h|o_0,\ o_1)$ (blue ■), and the case where integration is made by $p\left(H_k^0|o_0,\ o_1\right)$ (green ⊡).

Figure 4 shows the performance of $\mathcal{S}^0$ on the estimation of h^* with respect to the variance of $p(h|o_0)$. The shadowed region in the graph indicates the range of the variance in which the width of the peak in $p(h|o_0)$ almost matches the quantization width. This range seems to be a biologically plausible one. Out of this range, the activities of different neurons get strongly correlated or information provided by the sensory systems gets largely discarded; both of these effects are thought to be unfavorable in the sense of computational efficiency. Around this range, it is clearly seen that the performance achieved through learning closely approaches its ideal value. This shows that the learning algorithm is effective in that range.

4. Conclusion

In this article, we addressed a problem of code integration in the brain. By assuming that Bayesian computation is carried out in the brain, we formulated a possible form of computation and a learning algorithm to resolve the code difference between the cortical circuits. We then made a numerical experiment to show the effectiveness of the learning algorithm. The numerical result indicates that the learning algorithm can configure the interpreting rule for a foreign neural code under biologically relevant conditions. Those results show that it is possible for the local cortical circuits to learn how they should process signals from other cortical circuits in a dynamical and distributed fashion for coherent computation to proceed in the brain.

This research is partially supported by Grant-in-Aid for Scientific Research (A) (20246026) from MEXT of Japan.

References

[1] T.J. Sejnowski, C. Koch, and P.S. Churchland, "Computational neuroscience", **Science**, vol. 241, no. 4871, pp. 1299, 1988.

[2] R. Held, Y. Ostrovsky, B. Degelder, T. Gandhi, S. Ganesh, U. Mathur, and P. Sinha, "The newly sighted fail to match seen with felt.", **Nature neuroscience**, vol. advance online publication, Apr. 2011.

[3] A.K. Engel and W. Singer, "Temporal binding and the neural correlates of sensory awareness", **Trends in cognitive sciences**, vol. 5, no. 1, pp. 16–25, 2001.

[4] D.C. Knill and A. Pouget, "The Bayesian brain: the role of uncertainty in neural coding and computation", **TRENDS in neurosciences**, vol. 27, no. 12, pp. 712–719, 2004.

[5] M.O. Ernst and M.S. Banks, "Humans integrate visual and haptic information in a statistically optimal fashion", **Nature**, vol. 415, no. 6870, pp. 429–433, 2002.

[6] W.J. Ma, J.M. Beck, P.E. Latham, and A. Pouget, "Bayesian inference with probabilistic population codes", **Nature neuroscience**, vol. 9, pp. 1432–1438, 2006.

[7] C.M. Bishop and SpringerLink (Online service), **Pattern recognition and machine learning**, vol. 4, Springer New York:, 2006.