

# Training Integrate-and-Fire Neurons With the Informax Principle II

Jianfeng Feng, Yunlian Sun, Hilary Buxton, *Member, IEEE*, and Gang Wei

**Abstract**—We develop neuron learning rules using the Informax principle together with the input-output relationship of the integrate-and-fire (IF) model with Poisson inputs. The learning rule is then tested with constant inputs, time-varying inputs and images. For constant inputs, it is found that, under the Informax principle, a network of IF models with initially all positive weights tends to disconnect some connections between neurons. For time-varying inputs and images, we perform signal separation tasks called independent component analysis. Numerical simulations indicate that some number of inhibitory inputs improves the performance of the system in both biological and engineering senses.

**Index Terms**—Informax principle, integrate-and-fire (IF) model, learning rules, spiking independent component analysis.

## I. INTRODUCTION

**D**URING the past few decades, we have seen many illuminating publications on modeling single neurons, both at abstract and biophysical levels [18]. Many intriguing phenomena have been revealed such as how to ensure a single integrate-and-fire (IF) model to generate spike trains with a coefficient of variation between 0.5 and 1 [11], to carefully tune the noise level to exhibit stochastic resonance [4], to synchronize a network of spiking neurons with inhibitory inputs [17], [29]. Nevertheless, the majority are devoted to a “dead” neuron model: the model is not able to learn when it receives inputs.

In this paper, we study neuron models with a learning rule, i.e., the neuron is capable of updating its weights of inputs. The simplest and most general principle of learning rules is probably the Informax principle [21]. It has been demonstrated that the Informax principle is theoretically promising [3], [21], biologically plausible [12], and widely applicable in solving engineering problems [2], [5], [20].

By a simple combination of the Informax principle and the IF model, we want to address the following key issues.

- What are the implications of the Informax principle with a network of IF neurons? In other words, what is the outcome of the Informax principle learning? After learning, do the weights tend to self-organize themselves, represent inputs, or accomplish something else?
- What is the computational capacity of an IF neuron? Can the IF neuron be applied to solving practical problems? We have seen applications of the IF model, or more general

spiking neuron, to solving engineering problems, where the information contained in spike intervals is exploited. We test the computational capacity of the IF neuron in blind separation, using a rate coding assumption.

- What is the optimal value of the ratio between inhibitory and excitatory inputs? We all know there are inhibitory inputs in the neural system, but the functional role of these remains elusive.

Under the Informax principle, the learning rule developed for the IF model is complicated and so a complete theoretical treatment is impossible. For some ideal cases, we can understand its underpinning mechanisms [12]. For general cases, we have to resort to numerical simulations so that all the questions raised above are answered.

- The implication of the Informax principle is to disconnect some connections between neurons, i.e., a more sparse representation of inputs is achieved. With fully, randomly initialized connections, some weights will automatically die out after learning.
- We test the computational capacity of the IF model in blind separation tasks. The incoming signal for each single neuron is a Poisson process, implying a very low signal-to-noise ratio signal. Nevertheless, very crude simulations show that the computational capacity of the IF model is promising. It can successfully separate incoming, mixed signals.
- For neurons with both constant inputs and time-varying inputs, we find that the IF model behaves more reasonably when there are certain number of inhibitory inputs. Biologically, it produces spike trains with a coefficient of variation close to one, as in a Poisson process. In blind separation tasks, it better recovers the signals.

In classical neuronal network theory [16], we usually take into account the first-order statistics, i.e., the mean firing rates, as the input and output of a neuron. Hence classical neuron networks are essentially learning machines *computing mean*. Here, in our novel learning rules, the first-order statistics, as well as the second-order statistics (see Section IV), of the input and output are considered, resulting in a learning machine *computing mean and variance*. Even when the mean input vanishes (with exactly balanced excitatory and inhibitory inputs), a meaningless case in the classical setup, our setup, and learning rule are still fully functional.

## II. INFORMAX PRINCIPLE

We very briefly review some results on maximizing mutual information between input and output of a system and refer the

Manuscript received July 16, 2001; revised March 28, 2002 and August 21, 2002. The work of J. Feng was supported in part by EPSRC under Grant (GR/R54569) and a grant from the Royal Society.

The authors are with COGS, Sussex University, Brighton BN1 9QH, U.K. <http://www.cogs.susx.ac.uk/users/jianfeng>.

Digital Object Identifier 10.1109/TNN.2003.809419

reader to [5] for details. For a given input  $X$  and output  $Y$ , its mutual information  $M(Y, X)$  is defined by

$$M(Y, X) = H(Y) - H(Y|X)$$

where  $H(Y)$  is the entropy of the output  $Y$  and  $H(Y|X)$  is the conditional entropy. Under the assumption that the mapping between  $X$  and  $Y$  is deterministic, then maximizing the mutual information is reduced to maximizing the entropy  $H(Y)$  as pointed out in [5]. Note that maximizing entropy is, in general, not equivalent to maximizing mutual information. If we simply assume that  $Y = G(X) + N$ , where  $G$  is an invertible transformation and  $N$  is additive noise on the outputs. In this case, we see that [22]  $H(Y|X) = H(N)$ . Hence, maximizing the mutual information is equivalent to maximizing the entropy (we will return to this point in Section IV).

Suppose that the output firing frequency or interspike intervals (ISIs)  $y$  (realization of  $Y$ ) of a neuron is a function of input rate  $x$  (realization of  $X$ ), with synaptic weights  $w$ . Then the learning rule under the Informax principle is to maximize  $-\langle \log f(y) \rangle$  where  $f(y)$  is the distribution density of  $y$ . Equivalently we have [5, (6)]

$$\dot{w} \propto \left( \frac{\partial(-\langle \log f(y) \rangle)}{\partial w} \right) = \left( \frac{\partial y}{\partial x} \right)^{-1} \frac{\partial}{\partial w} \left( \frac{\partial y}{\partial x} \right) \quad (1)$$

which is the starting point of our development below. For a high-dimensional case we could simply replace  $\partial y / \partial x$  by the determinant of the Jacobian matrix, that is  $|J| = |\partial y_i / \partial x_j|_i$ .

### III. IF MODEL

Suppose that a neuron receives excitatory postsynaptic potentials (EPSPs) at  $n$  synapses and inhibitory postsynaptic potentials (IPSPs) at  $m$  inhibitory synapses. When the membrane potential  $V_t^{(i)}$  of  $i$ th neuron is between the resting potential  $V_{\text{rest}}$  and the threshold  $V_{\text{thre}}$ , it is given by

$$dV_t^{(i)} = -L \left( V_t^{(i)} - V_{\text{rest}} \right) dt + d\bar{I}_{\text{syn}}^{(i)}(t) \quad (2)$$

where  $L$  is the decay rate and synaptic inputs

$$\bar{I}_{\text{syn}}^{(i)}(t) = \sum_{j=1}^n w_{ij}^E E_j(t) - \sum_{j=1}^m w_{ij}^I I_j(t) \quad (3)$$

with  $E_i(t)$ ,  $I_i(t)$  as Poisson processes with rate  $\lambda_i^E$  and  $\lambda_i^I$ , respectively,  $w_{ij}^E > 0$ ,  $w_{ij}^I > 0$  being magnitude of each EPSP and IPSP. Note that the magnitude of each EPSP and IPSP is solely determined by weights  $w_{ij}^E$  and  $w_{ij}^I$  since the increment of  $E_j(t)$  and  $I_j(t)$  is unity (Poisson process).

Once  $V_t^{(i)}$  crosses  $V_{\text{thre}}$  from below a spike is generated and  $V_t^{(i)}$  is reset to  $V_{\text{rest}}$ . This model is termed the IF model [7], [10], [28].

Here we use the usual approximation to approximate the IF models, or more exactly the synaptic inputs of the models. The basic idea of the usual approximation is to replace the Poisson process of (3) by a diffusion process with the identical mean and variance. We do not check the approximation accuracy since this has been done by many authors [11], [24], [28]. Essentially the approximation is accurate when the input rates  $\lambda_i$ ,  $i = 1, \dots, n$

are high and signal magnitudes  $w_{ij}$ ,  $i, j = 1, \dots, n$  are not large.

The input now reads

$$E_i(t) \sim \lambda_i^E t + \sqrt{\lambda_i^E} B_i^E(t)$$

and similarly

$$I_i(t) \sim \lambda_i^I t + \sqrt{\lambda_i^I} B_i^I(t)$$

where  $B_i^E(t)$  and  $B_i^I(t)$  are standard Brownian motions (a normally distributed random variable with mean zero and variance  $t$ ). The mean of  $E_i(t)$  is  $\lambda_i^E t$  and the variance of it is  $\lambda_i^E t$ ; so is the diffusion process  $\lambda_i^E t + \sqrt{\lambda_i^E} B_i^E(t)$ . Therefore the IF model can be approximated by

$$dv_t^{(i)} = -L \left( v_t^{(i)} - V_{\text{rest}} \right) dt + d\bar{i}_{\text{syn}}^{(i)}(t)$$

where

$$\bar{i}_{\text{syn}}^{(i)}(t) = \sum_{j=1}^n w_{ij}^E \lambda_j^E t - \sum_{j=1}^m w_{ij}^I \lambda_j^I t + \sum_{j=1}^n \sqrt{(w_{ij}^E)^2 \lambda_j^E} B_j^E(t) - \sum_{j=1}^m \sqrt{(w_{ij}^I)^2 \lambda_j^I} B_j^I(t). \quad (4)$$

Since the summation of Brownian motions (normal random variables) is again a Brownian motion we can rewrite the equation above as follows:

$$\bar{i}_{\text{syn}}^{(i)}(t) = \mu_i t + \sigma_i B_i(t) \quad (5)$$

where  $B_i(t)$  is a standard Brownian motion

$$\begin{cases} \mu_i = \sum_{j=1}^n \lambda_j^E w_{ij}^E - \sum_{j=1}^m \lambda_j^I w_{ij}^I \\ \sigma_i^2 = \sum_{j=1}^n \lambda_j^E (w_{ij}^E)^2 + \sum_{j=1}^m \lambda_j^I (w_{ij}^I)^2 \end{cases}. \quad (6)$$

In the sequel, for simplicity of notation, we assume that  $n = m$ ,  $w_{ij} = w_{ij}^E = w_{ij}^I$  and  $\lambda_j^I = r \lambda_j^E = r \lambda$  for  $r \in [0, 1]$ . Therefore the neuron has the same number of active excitatory and inhibitory synapses, with the same weights, but input rates might be different. When  $r = 0$ , the cell receives purely excitatory input and when  $r = 1$ , its inputs are exactly balanced. The ISI of efferent spikes is

$$T_i(r) = \inf \left\{ t: V_t^{(i)} \geq V_{\text{thre}} \right\}.$$

We only consider the case of rate coding since then a rigorous input-output relationship of firing rates is known for the IF model. By rate coding, we mean that the information is carried by the firing rate of a neuron. As shown in Fig. 1, filled circles or rectangles are the units of our network. Each represents a group of IF neurons. It is well known in the literature that the input-output relationship of a neuron takes a sigmoidal form and this is the basis of neural computations developed over the past few decades. The input-output relationship of an IF model (see Fig. 2) takes a sigmoidal function as well (not surprising at all), but it depends not only on the mean of inputs, but also on the variance of inputs. The latter feature enables us to derive novel

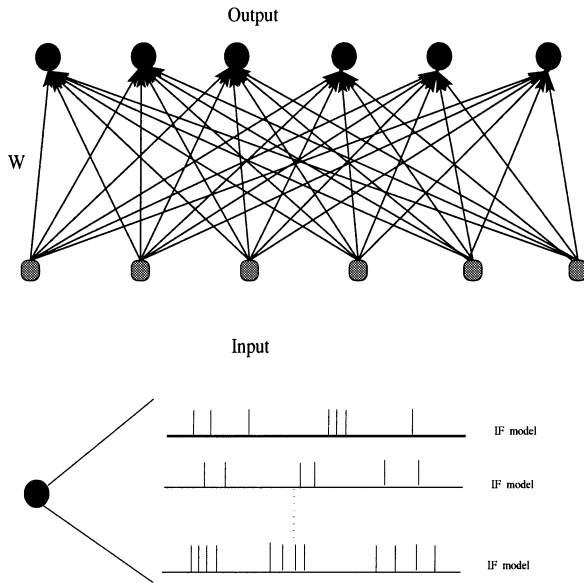


Fig. 1. Schematic input-output relationship. Each unit (circle) represents a group of IF neurons. The average of spikes over the group of neuron gives rise to the mean firing rate as defined in by (7). The activity of each IF neuron is described in Section III.

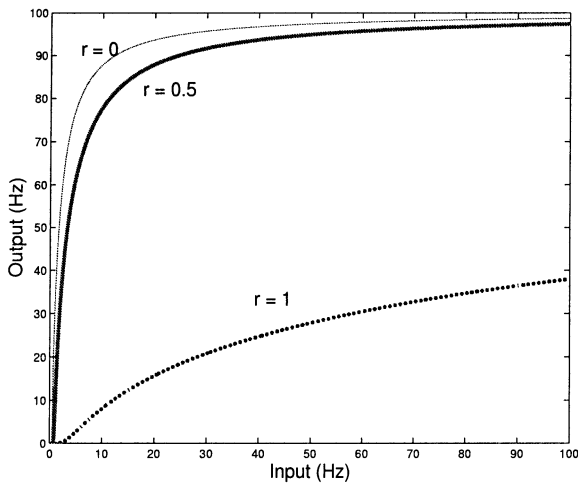


Fig. 2. Output  $\gamma_i$  (Hz) versus input  $\lambda_1 = \lambda_2 = \lambda_3$  (kHz) of the IF model with  $n = 3$ ,  $w_{ij} = 0.5$ ,  $i, j = 1, \dots, 3$ ,  $\lambda_1 = \lambda_2 = \lambda_3$ ,  $V_{\text{thre}} = 20$  mV,  $V_{\text{rest}} = 0$  mV and  $L = 1/20$ ,  $T_{\text{ref}} = 10$  ms.

learning rules which, to the best of our knowledge, have not been reported in the literature and exhibit some intriguing phenomena [see discussion after (7)]. The importance that a neuron might use higher order statistics in computation has been recognized early in the literature (see, for example, [3]).

#### IV. LEARNING RULE

Remember that the mean ISI of the IF model with Poisson inputs is given by [11]

$$\langle T_i(r) \rangle = \frac{2}{L} \int_{(V_{\text{rest}} L - \mu_i)/\sigma_i}^{(V_{\text{thre}} L - \mu_i)/\sigma_i} g(x) dx \quad (7)$$

where

$$g(x) = \left[ \exp(x^2) \int_{-\infty}^x \exp(-u^2) du \right]$$

and (6)

$$\begin{cases} \mu_i = \left( \sum_{j=1}^n w_{ij} \lambda_j \right) (1-r) \\ \sigma_i = \sqrt{\left( \sum_{j=1}^n w_{ij}^2 \lambda_j \right) L(1+r)}. \end{cases}$$

A detailed derivation of (7) is quite mathematically involved and we simply refer the reader to our review paper [11] for more details. In Fig. 2, we plot the input-output relationship, i.e.,  $\gamma_i$  versus  $\lambda = \lambda_1 = \dots = \lambda_n$ , described by

$$\gamma_i = \frac{1}{\langle T_i(r) \rangle + T_{\text{ref}}} \quad (8)$$

where  $T_{\text{ref}}$  is the refractory period. The relationship between mean ISIs and output firing rates is determined by (8). Therefore, for given input firing rates  $\lambda_j$ ,  $j = 1, \dots, n$ , weights  $w_{ij}$ ,  $j = 1, \dots, n$  and various parameters in the IF model, we can analytically calculate the mean ISIs via (7). From (8), we can determine output firing rates provided  $T_{\text{ref}}$  is given. In Fig. 2, we have scaled down the number of input neurons ( $n = 3$ ) and correspondingly scaled up the input rate of each synapses (in kHz), in comparison with typical parameters used in simulating biological systems [11]; see Section V for further explanation.

When  $r = 0$ , from Fig. 2, we see that the output firing rate is rapidly saturated. When  $r = 1$ , exactly balanced input case, we note that the mean input to each neuron  $\mu_i$  is zero. However, as long as the variance  $\sigma = \sigma_1 = \sigma_2 = \dots = \sigma_n$  is not zero ( $\sigma = 0$  is trivial), we have  $\langle T_i(r) \rangle < \infty$ , namely the neuron will fire. In Fig. 2, we see that the neuron fires with a frequency between 0 and 40 Hz. In most, if not all, classical neuron network theory [16], only the mean of input is taken into account. Therefore the case of  $r = 1$  is not discussed in neural-network literature. It has been recently pointed out that  $r \sim 1$  might be the most interesting case in a biological system [25], [11]. When  $r$  is close to zero, the output spike trains of a single neuron are simply too regular, in comparison with biological data.

From the activity of a single IF neuron, we can obtain spike trains. From the spike trains, the mean ISIs are calculated in two ways: averaging over a time window for a single spike train, the so-called temporal-average; averaging over a group of neurons within a short time window, the so-called spatio-average. The problem with temporal-average is its speed. To reliably obtain the mean ISIs we have to take a long time window. The shortcoming with spatio-average lies in the fact that neuron activities are usually inhomogeneous and so a simple average will not give us a correct mean ISI. Nevertheless, we prefer the second choice, i.e., the spatio-average. To get rid of the inhomogeneity in neuronal populations, we can average the ISIs using different weights for different neurons. To fully address the issue is outside the scope of the paper; we refer the reader to [15] for details.

Hence, in Fig. 1 we use a filled circle or rectangle to represent the spatio-average over a group of neurons.

The learning rule under the Informax principle is

$$\dot{w}_{ij} \propto \frac{\partial |J|}{\partial w_{ij}}$$

where

$$J = \begin{pmatrix} \frac{\partial \gamma_1}{\partial \lambda_1} & \cdots & \frac{\partial \gamma_1}{\partial \lambda_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \gamma_n}{\partial \lambda_1} & \cdots & \frac{\partial \gamma_n}{\partial \lambda_n} \end{pmatrix}$$

The matrix  $J$  could be rewritten as

$$J = \begin{pmatrix} -\frac{1}{(\langle T_1(r) \rangle + T_{\text{ref}})^2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -\frac{1}{(\langle T_n(r) \rangle + T_{\text{ref}})^2} \end{pmatrix} \cdot A$$

where

$$A = \begin{pmatrix} \frac{\partial \langle T_1(r) \rangle}{\partial \lambda_1} & \cdots & \frac{\partial \langle T_1(r) \rangle}{\partial \lambda_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \langle T_n(r) \rangle}{\partial \lambda_1} & \cdots & \frac{\partial \langle T_n(r) \rangle}{\partial \lambda_n} \end{pmatrix}.$$

Therefore

$$|J| = (-1)^n \cdot \left[ \prod_{i=1}^n \frac{1}{(\langle T_i(r) \rangle + T_{\text{ref}})^2} \right] \left[ \sum_{j=1}^n (-1)^{j+i} \frac{\partial \langle T_i(r) \rangle}{\partial \lambda_j} |A_{ij}| \right]$$

where  $A_{ij}$  is the  $(n-1) \times (n-1)$  matrix obtained by deleting the  $i$ th row and  $j$ th column of the matrix  $A$  and

$$\frac{\partial |J|}{\partial w_{ij}} = \frac{-2 \frac{\partial \langle T_i(r) \rangle}{\partial w_{ij}}}{\langle T_i(r) \rangle + T_{\text{ref}}} |J| + (-1)^n \prod_{i=1}^n \frac{1}{(\langle T_i(r) \rangle + T_{\text{ref}})^2} \cdot \left[ \sum_{k=1}^n (-1)^{k+i} \frac{\partial^2 \langle T_i(r) \rangle}{\partial \lambda_k \partial w_{ij}} |A_{ik}| \right]$$

which yields

$$\frac{\partial |J|}{\partial w_{ij}} = \frac{-2 \frac{\partial \langle T_i(r) \rangle}{\partial w_{ij}}}{\langle T_i(r) \rangle + T_{\text{ref}}} + \frac{\sum_{k=1}^n (-1)^{k+i} \frac{\partial^2 \langle T_i(r) \rangle}{\partial \lambda_k \partial w_{ij}} |A_{ik}|}{\sum_{k=1}^n (-1)^{k+i} \frac{\partial \langle T_i(r) \rangle}{\partial \lambda_k} |A_{ik}|}. \quad (10)$$

Defining

$$\begin{cases} \Sigma_i^2 = \left( \sum_{j=1}^n w_{ij}^2 \lambda_j \right) L(1+r) \\ \Theta_i = \left( \sum_{j=1}^n w_{ij} \lambda_j \right) (1-r) \end{cases}$$

letting  $V_{\text{rest}} = 0$

$$(9) \quad \begin{cases} \xi_{ij}(x) = \frac{2w_{ij}(1-r)\Sigma_i^2 + w_{ij}^2 L(1+r)(xL - \Theta_i)}{2\Sigma_i^3} \\ \eta_{ij}(x) = \frac{\lambda_j(1-r)\Sigma_i^2 + (xL - \Theta_i)w_{ij}\lambda_j L(1+r)}{\Sigma_i^3} \end{cases}$$

and

$$\zeta_{ijk}(x) = \begin{cases} \frac{2(1-r)\Sigma_i^2 + 3w_{ij}^2 \lambda_j L(1-r^2)}{2\Sigma_i^3} + \frac{w_{ij} L(1+r)(xL - \Theta_i)}{\Sigma_i^3} - \frac{3w_{ij} \lambda_j L(1+r)w_{ij}^2 L(1+r)(xL - \Theta_i)}{2\Sigma_i^5} \\ - \frac{3w_{ij} \lambda_j L(1+r)w_{ij}(1-r)\Sigma_i^2}{\Sigma_i^5}, & \text{if } k = j \\ \frac{w_{ik} \lambda_j L(1-r^2)(4w_{ij} - w_{ik})}{2\Sigma_i^3} - \frac{3w_{ij} \lambda_j L(1+r)w_{ik}^2 L(1+r)(xL - \Theta_i)}{2\Sigma_i^5} \\ - \frac{3w_{ij} \lambda_j L(1+r)w_{ik}(1-r)\Sigma_i^2}{\Sigma_i^5}, & \text{if } k \neq j \end{cases}$$

we arrive at

$$\begin{aligned} \frac{\partial \langle T_i(r) \rangle}{\partial \lambda_j} &= -\frac{2}{L} g \left( \frac{V_{\text{thre}} L - \Theta_i}{\Sigma_i} \right) \xi_{ij}(V_{\text{thre}}) \\ &\quad + \frac{2}{L} g \left( -\frac{\Theta_i}{\Sigma_i} \right) \xi_{ij}(0) \\ \frac{\partial \langle T_i(r) \rangle}{\partial w_{ij}} &= -\frac{2}{L} g \left( \frac{V_{\text{thre}} L - \Theta_i}{\Sigma_i} \right) \eta_{ij}(V_{\text{thre}}) \\ &\quad + \frac{2}{L} g \left( -\frac{\Theta_i}{\Sigma_i} \right) \eta_{ij}(0) \end{aligned} \quad (11)$$

and

$$\begin{aligned} \frac{\partial}{\partial w_{ij}} \left( \frac{\partial \langle T_i(r) \rangle}{\partial \lambda_k} \right) &= -\frac{2}{L} g \left( \frac{V_{\text{thre}} L - \Theta_i}{\Sigma_i} \right) \zeta_{ijk}(V_{\text{thre}}) \\ &\quad + \frac{2}{L} \left[ 2g \left( \frac{V_{\text{thre}} L - \Theta_i}{\Sigma_i} \right) \frac{V_{\text{thre}} L - \Theta_i}{\Sigma_i} + 1 \right] \\ &\quad \cdot \xi_{ik}(V_{\text{thre}}) \eta_{ij}(V_{\text{thre}}) + \frac{2}{L} g \left( -\frac{\Theta_i}{\Sigma_i} \right) \zeta_{ijk}(0) \\ &\quad - \frac{2}{L} \left[ 2g \left( -\frac{\Theta_i}{\Sigma_i} \right) \frac{-\Theta_i}{\Sigma_i} + 1 \right] \xi_{ik}(0) \eta_{ij}(0). \end{aligned} \quad (12)$$

Combining (11) and (12) with (10), we obtain a novel learning rule based upon the IF model. The first term in (10) represents how the weight should be updated according to its input-output relationship; the second term relies on the derivatives of  $T_i(r)$ .

To fully understand the learning rule presented here is a tough issue, nevertheless for a special case we can grasp a complete picture. Let us consider the ideal case of  $w_{ij} = w$ ,  $\lambda_i = \lambda$  and

$n = 1$ , i.e., the neuron only has one input and one output or all inputs are uniform. Now (10) is reduced to

$$\frac{\partial |J|}{\partial w} = \frac{-2\partial \langle T(r) \rangle}{\partial w} + \frac{\partial^2 \langle T(r) \rangle}{\partial \lambda \partial w} \quad (13)$$

and we have  $\eta(0) = \eta_{ij}(0) = 0$ . After taking some further calculations, we obtained the learning rule developed in [12]

$$\frac{\partial |J|}{\partial w} = \frac{[(2Ug(U) + 1)V + g(U)] \frac{V_{\text{thre}} \sqrt{L}}{w^2 \lambda \sqrt{\lambda(1+r)}}}{-g(U)V + g\left(\frac{-\lambda(1-r)}{\sqrt{\lambda L(1+r)}}\right) \frac{(1-r)}{\sqrt{\lambda(1+r)L}}} + \frac{V_{\text{thre}}}{\sqrt{(1+r)L}} \frac{4\gamma g(U)}{w^2 \sqrt{\lambda}}$$

with

$$U = \frac{V_{\text{thre}}L - \lambda(1-r)w}{w\sqrt{\lambda L(1+r)}}, \quad V = \frac{V_{\text{thre}}L + \lambda(1-r)w}{w\sqrt{\lambda L(1+r)}}.$$

For the ideal case, we know that there is a unique stable point for the learning rule, and the weight is automatically restricted in the regions of  $(0, \infty)$ . The derived learning rule is similar to the BCM learning rule [12], [18].

For the general case, the learning rule presented here is too complex to be explored theoretically, nevertheless we can simulate it numerically, as presented in the next section.

Now, we make a comment on the relationship between the maximizing entropy and maximizing mutual information for the IF model. Using the notation in Section II, for the IF model, if we have  $Y = G(X) + N$  and  $G = 1/(T_{\text{ref}} + \langle T(r) \rangle)$ , where  $T(r)$  is the ISI. In this case, we see that [22]  $H(Y|X) = H(N)$  is independent of  $w$ . Hence, maximizing the mutual information is equivalent to maximizing the entropy. Nevertheless, in general, changing weights and the ratio between inhibitory inputs and excitatory inputs, will both lead to a change of the output distribution of the ISIs (see, for example, [11]) and, therefore,  $H(Y|X)$  depends on  $w$ . To develop an exact learning rule based upon the information maximization then requires an exact expression of the distribution of the ISIs, which is lacking at moment [11].

## V. NUMERICAL RESULTS

### A. Supervised Learning

Recall that in (10), we have

$$\frac{\partial |J|}{\partial w_{ij}} = \frac{-2\partial \langle T_i(r) \rangle}{\partial w_{ij}} \gamma_i + \frac{\sum_{k=1}^n (-1)^{k+i} \frac{\partial^2 \langle T_i(r) \rangle}{\partial \lambda_k \partial w_{ij}} |A_{ik}|}{\sum_{k=1}^n (-1)^{k+i} \frac{\partial \langle T_i(r) \rangle}{\partial \lambda_k} |A_{ik}|} \quad (14)$$

where  $\gamma_i$  is the efferent firing rate (in unit of 1/ms) of the  $i$ th neuron. Therefore, if we fix (clamp) the efferent firing rate, we have a version of supervised learning. The desired output firing rate is  $\gamma_i, i = 1, \dots, n$ . We train the network according to (14) and investigate whether the actual output satisfies (14) or not.

We simulate the learning rule with the following parameters:  $n = 6$ ,  $\gamma_i = 50$  Hz,  $\lambda_i = 500$  Hz for  $i = 1, 2, 3$ ,  $\lambda_i = 2000$  Hz for  $i = 4, 5, 6$ ,  $r \in [0, 1]$ ,  $V_{\text{thre}} = 20$ ,  $V_{\text{rest}} = 0$  and  $L = 1/20$ , using Matlab. The total excitatory input is of

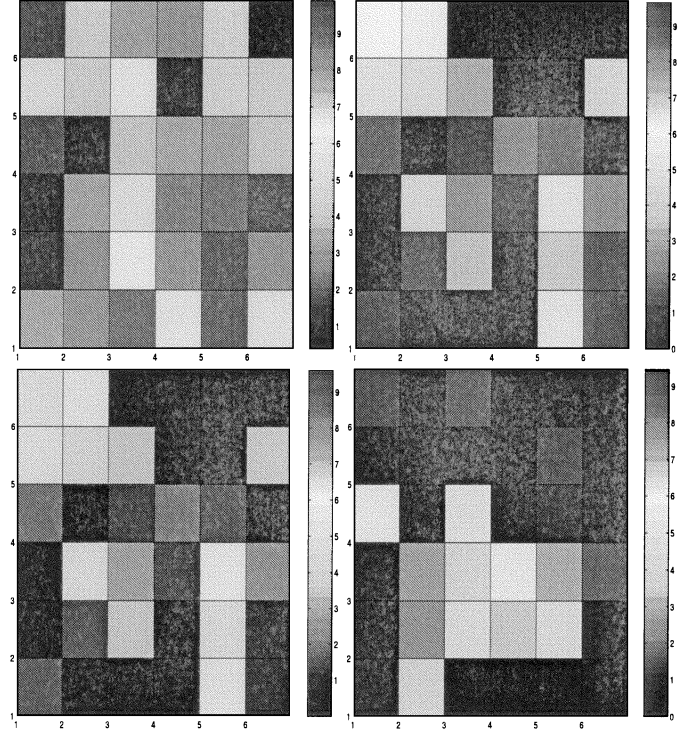


Fig. 3. Weights of the network we considered. The color level of right square of each grid, say  $(i, j)$  represents the value of  $w_{ij}$ . For example, the color level of the most left bottom square is  $w_{11}$ . Upper left is the initial states of weight. Upper right ( $r = 0$ ) and bottom panel From left to right, ( $r = 0.5$  and  $r = 1$ ) show weights when the learning is finished (800 iterations).

7500 Hz. The input is equivalent to each neuron receiving 100 active synaptic inputs, each with 75 Hz, which are physiologically plausible parameters [11]. Note that for the purpose of visualizing results, here we scale own  $n$  and scale up  $\lambda$  correspondingly. The initial weights are random variables from  $U(0, 9)$ , the uniform distribution over  $[0, 9]$ , and step size of learning is 0.05. After 800 time steps of learning, the synaptic weights are stable, for all cases we considered here.

We carry out simulations for  $r = 0, 0.5$  and one. Fig. 3 shows that weights become more specific after learning. For example when  $r = 0$  we have

$$(w_{11}(0), w_{12}(0), w_{13}(0), w_{14}(0), w_{15}(0), w_{16}(0)) = (7.8, 2.9, 2.2, 5.5, 9.1, 5.2)$$

and

$$(w_{11}(800), w_{12}(800), w_{13}(800), w_{14}(800), w_{15}(800), w_{16}(800)) = (9.5, 0, 0, 0, 5.3, 1.4)$$

namely the connections

$$w_{12}(800), \quad w_{13}(800), \quad w_{14}(800)$$

die out. More precisely, before learning the number of weights near to zero is three, and after learning it is 13, 14, and 15 for  $r = 0, 0.5$ , and  $r = 1$ , respectively. Hence, in this case (also see Section V-B), some connections of an IF model network, under the Infromax learning, become disconnected.

To achieve successful learning, the ISI of output of each neuron should be 20 ms, the desired output. Nevertheless, in

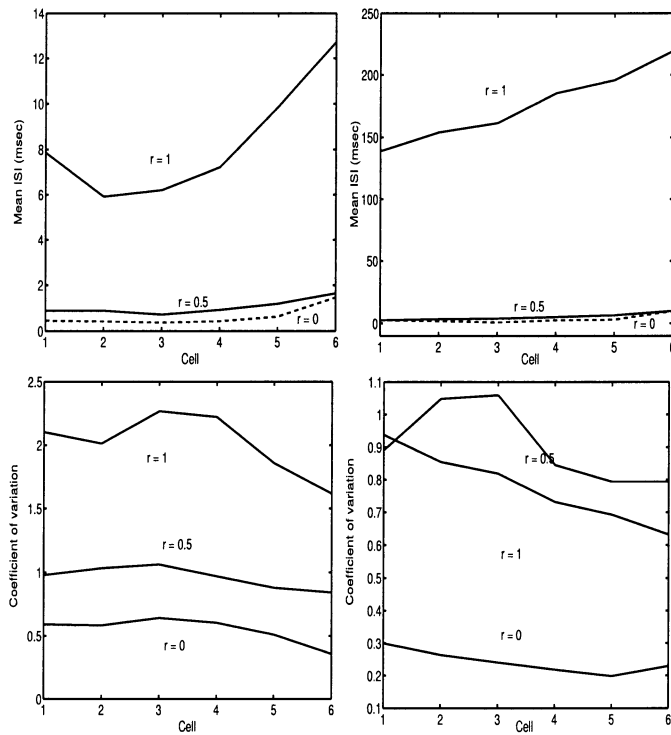


Fig. 4. Mean and CV of output ISIs. Left are initial values and right are values after learning.

the learning rule, we still have one parameter  $T_{\text{ref}}$  which is free. Hence, for an ISI smaller than 20 ms, we could always add an appropriate refractory period so that the output ISI is 20 ms. The appropriate refractory period could be chosen via a learning rule, say the gradient descent method. In other words, the learning rule is *potentially* successful if output ISI is less than 20 ms. Fig. 4 tells us that in all cases of  $r = 0$  and  $r = 0.5$ , the learning rule is potentially successful, i.e. all mean ISI values are less than 20 ms. The conclusion is not true for  $r = 1$  where too big an ISI is obtained. Fig. 4 also shows the coefficient of variation ( $\text{CV} = \text{Standard deviation of ISI} / \text{mean of ISI}$ ) of ISI. It is interesting to see that the coefficient of variation of the model with  $r = 0$  is generally smaller than 0.5. For a recent discussion on the CV of ISI of the IF model, we refer the reader to [11]. Briefly speaking, it is found in experiments that the output of a single neuron is very irregular. The CV of the output ISIs is between 0.5 and 1, similar to that of the Poisson process with CV being one. These results are very interesting, since it is obvious that a neuronal system is required to encode and then decode information as reliably as possible. A large CV, i.e., a very noisy system, simply contradicts the requirement.

In summary, only when  $r = 0.5$  the IF model learns potentially successfully and generate spikes trains with a coefficient of variation inside  $[0.5, 1]$  [25], “optimally behaving” in the biological sense.

### B. Unsupervised Learning

Now we turn our attention to unsupervised learning which means that the output firing rate is determined by the input rate

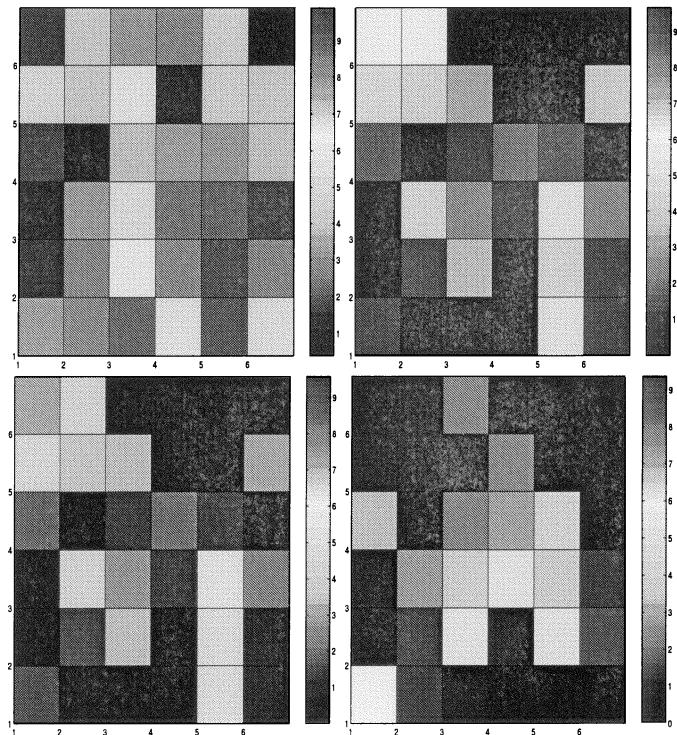


Fig. 5. Weights of the network we considered. The color level of right square of each grid, say  $(i, j)$  represents the value of  $w_{ij}$ . For example, the color level of the most left bottom square is  $w_{11}$ . Upper left is the initial states of weight. Upper right ( $r = 0$ ) and bottom panel, from left to right,  $r = 0.5$  and  $r = 1$ , show weights when the learning is finished (800 iterations).

via (8), rather than fixed as in the previous section. We fix  $T_{\text{ref}} = 10$  ms in all simulations.

We carry out simulations for  $r = 0, 0.5$  and  $1$ . Fig. 5 shows weights after learning with initial states identical to these in the previous subsection. After learning with the Informax principle, some connections become disconnected. Note that both in Fig. 3 and Fig. 5, weights for  $r = 0$  and  $r = 0.5$  are very similar from one experiment to the other. This can be easily understood. As pointed out in [13, p. 1656 and Fig. 2], the output firing rate of the IF neuron is not sensitive to its input when the input is superthreshold. In fact this can also be observed from Fig. 2 where the output firing rates of  $r = 0$  and  $r = 0.5$  are close to each other.

Furthermore, Fig. 6 depicts the efferent ISIs and the coefficient of variation. Again as in the case of supervised learning, we note that when  $r = 0.5$ , the most reasonable behavior is observed: the output CV of ISIs is roughly close to one, as observed in experiments [25].

Note that the input and output relationship of a neuron depends not only on its input weights and inputs, but also on  $r$ , the ratio between inhibitory and excitatory inputs. With identical weights and inputs but different  $r$ , the output of a neuron could be very different [see, for example, (7) and Fig. 2].

### C. Signal Separation

To test the computational capacity of the IF model, we take into account a toy model of time-varying (sources) inputs, as in [16]. For general background on blind separations, we refer the reader to [2], [5], [20].

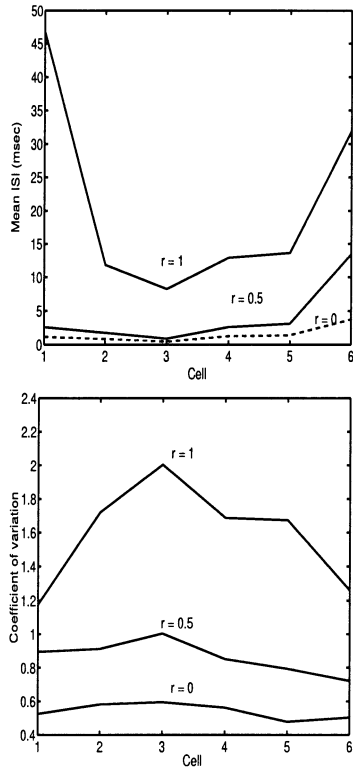


Fig. 6. Mean and CV of output ISIs after unsupervised learning. The initial mean and CV are as shown in Fig. 4, left column.

The input signals are

$$\begin{cases} a(t) = 2 \sin(400t) \cos(30t) \\ b(t) = 2 \operatorname{sign}(\sin(500t + 9 \cos(40t))) \\ c(t) = 2(\operatorname{rand} - 0.5) \end{cases} \quad (15)$$

for  $t \geq 0$  and  $\operatorname{rand}$  is the uniform random numbers from  $[0, 1]$ . The input signals are mixed with the matrix

$$M = \begin{pmatrix} 0.3985 & -0.1966 & 0.0021 \\ -0.0801 & 0.2436 & -0.0072 \\ 0.0070 & -0.0715 & 0.1134 \end{pmatrix}$$

and a constant vector  $(2, 2, 2)$  is added (to ensure that input signals are positive) i.e., the received signals are

$$(\lambda_1(t), \lambda_2(t), \lambda_3(t))' = M(a(t), b(t), c(t))' + (2, 2, 2)'$$

and inverse matrix of  $M$  is

$$\begin{pmatrix} 3.0000 & 2.4500 & 0.1000 \\ 1.0000 & 5.0000 & 0.3000 \\ 0.4450 & 3.0000 & 9.0000 \end{pmatrix}.$$

The weights are updated with a learning rate of  $1/(t + 100)$ , with an initial state of randomly generated number from  $U(0, 10)$ , the uniform distribution over  $[0, 10]$ . It is well known from stochastic approximation theory [31] that a learning rate of  $O(1/t)$  will ensure that the algorithm converges

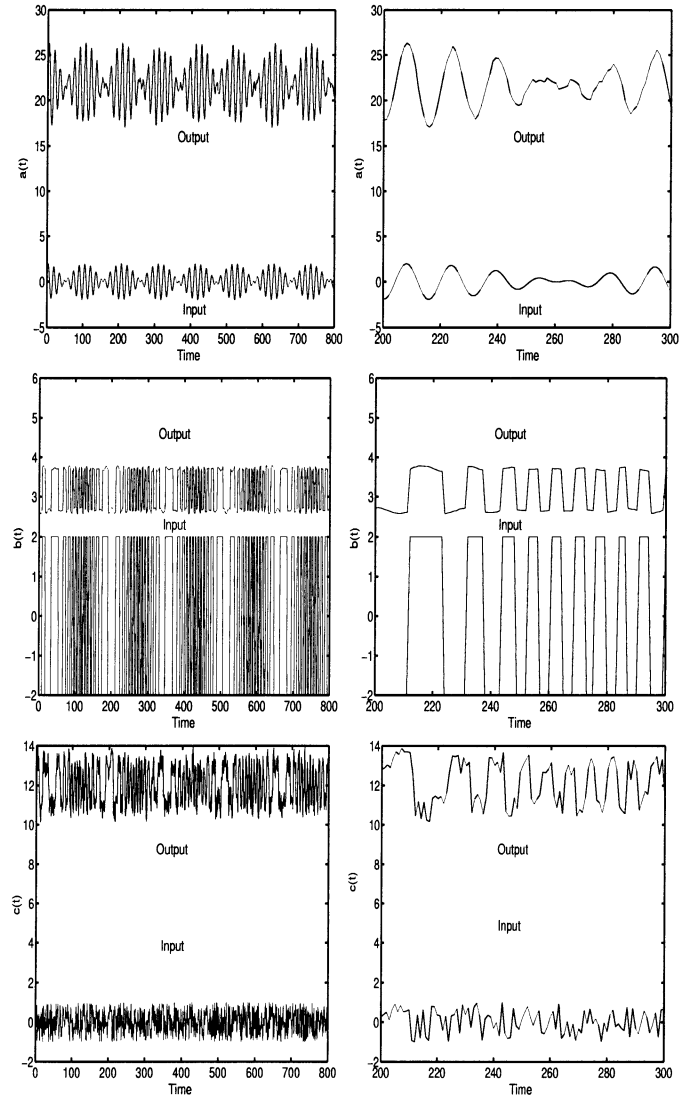


Fig. 7. Input signals (bottom trace of each figure) and output signals (upper trace) after blind separations with  $r = 0.5$ . Right column is the blow up of left column in time interval  $[200, 300]$ .

to the local minima of its cost function. After 800 times of iterations we obtain

$$w(800) = \begin{pmatrix} 0.3679 & 0.0052 & 5.6465 \\ 0.1533 & 1.2619 & 0.1755 \\ 6.3680 & 4.4491 & 0.0023 \end{pmatrix}$$

which gives us the results as shown in Fig. 7. The output signals are obtained via

$$\text{output} = w(800)(\lambda_1(t), \lambda_2(t), \lambda_3(t))'.$$

We also run the same simulations with  $r = 0$  and  $r = 1$ . It is interesting to find that when  $r = 1$ , i.e., with exactly balanced excitatory and inhibitory input, input signals are difficult to recover. In all simulations we carried out, the weights quickly diverge to large numbers. In contrast, when  $r$  is small ( $r = 0, r = 0.5$ ), input signals are reasonably recovered.

Comparing Fig. 7 with Fig. 8, we find that a better blind separation of input signal is achieved when  $r = 0.5$  [see  $b(t)$ ]. For signal  $a(t)$  the quality of recovered signals are very similar, at

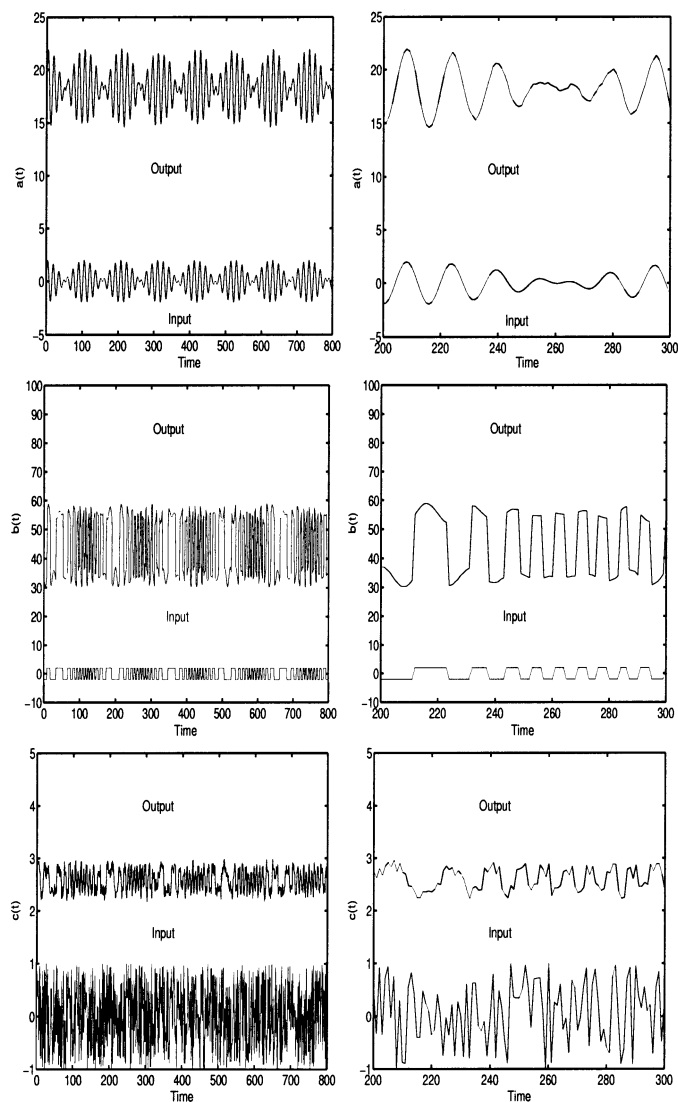


Fig. 8. Input signals (bottom trace of each figure) and output signals (upper trace) after blind separations with  $r = 0$ . Right column is the blow up of left column in time interval  $[200, 300]$ .

least to a visual inspection. Since  $c(t)$  is noise, we pay less attention to it. For  $b(t)$ , we perform a more detailed comparison of recovered signals between  $r = 0$  and  $r = 0.5$  in Fig. 10. Recovered signals  $b(t)$  are plotted using identical scales. Note that input signals are flat when they reach the maximum i.e., two or minimum i.e.,  $-2$ . Hence for recovered signals we could use the flatness of maximum and minimum as a criterion to assess the quality of separations. Two horizontal, thick lines are added in Fig. 10 to help us visually assess the flatness of separated signals. It is easily seen from Fig. 10 that  $b(t)$  obtained with  $r = 0.5$  is better than that with  $r = 0$ . We thus conclude that, at least in the numerical example we considered here, the IF model has a better performance when certain number of inhibitory inputs are added in the inputs. Finally, in Fig. 9, the output firing rates of three neurons during learning period are shown with  $r = 0$ .

We can have a comparison of the approach presented here with the conventional one [5] where only the mean of input is taken into account. Going to the extremal case when  $r = 1$ , we

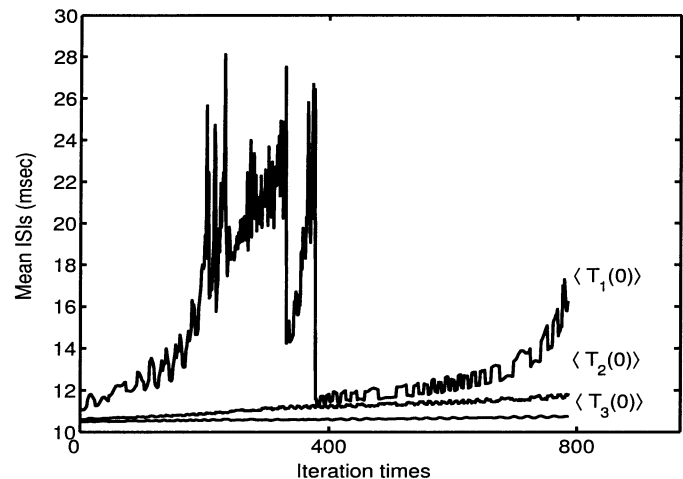


Fig. 9. Efferent ISIs versus iteration times with  $r = 0$ , see Fig. 8.

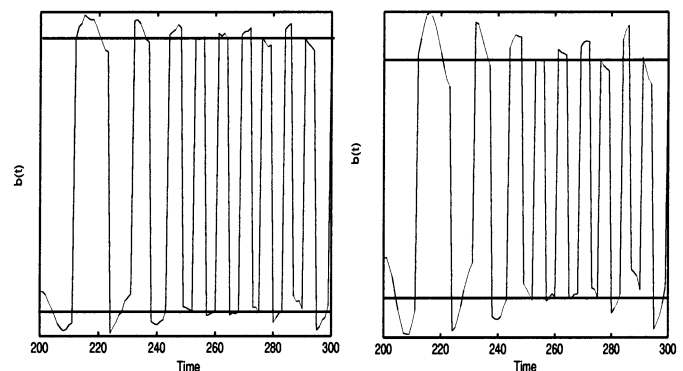


Fig. 10. Detailed comparison between recovered signals with  $r = 0.5$  (left), which is the same as in Fig. 7, middle panel, and  $r = 0$  (right), which is the same as in Fig. 8, middle panel. Note that for comparison, we have rescaled two figures. Two horizontal, thick lines are added to help us assess the quality of separated signals. It is easily seen that with  $r = 0.5$ , a more flat signal is obtained.

see that the mean input signal is zero and so it is impossible to discuss the issue of recovering signals in the classical setup. As we mentioned before, we have difficulties in separating signals as well. However, we can simply modify the learning algorithm by setting an upper bound for the weights, which is biologically plausible and widely employed in neural networks. Fig. 11 shows the actual outcome after 200 iterations, with all parameters as before. Fig. 12 is the ISIs of all three cells during the learning period. Note that data are truncated at 50 ms. Finally we emphasize that it is not easy to have a quantitative comparison between the ICA carried out here and the conventional ICA [16]. The signals employed in the ICA here is a mixture of conventional signals (i.e., the mean term in Section III) and conventional noise (i.e., the variance term in Section III). The learning updates the weights, implying simultaneous change to the mean and variance. Furthermore the input signal in the mean term is the original signal  $\lambda(t)$ , but in the standard deviation term  $\sigma$  it is  $\sqrt{\lambda(t)}$  [see (6)], i.e., the input signal is distorted.

From Figs. 7, 8, and 11, we see that the recovered signals are shifted in positions and rescaled in magnitude. This is a well known feature of blind separation, i.e., the recovered signals are not unique [2], [5].



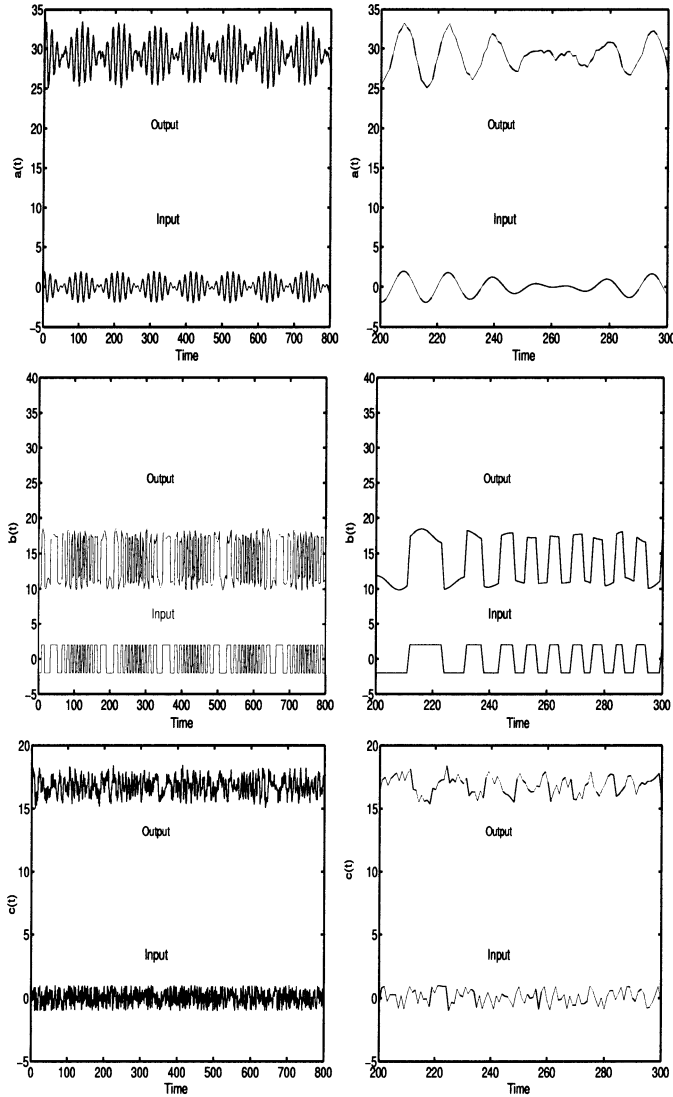


Fig. 11. Input signals (bottom trace of each figure) and output signals (upper trace) after blind separations with  $r = 1$ . An upper bound of weights of ten is introduced in the algorithm. Right column is the blow up of left column in time interval [200, 300].

Next we test our algorithms on people faces. Each face in Fig. 13 is a file of  $287 \times 384$  pixels, denoting it as  $\bar{\lambda}_k(i, j)$ ,  $k = 1, 2, 3$  with  $i = 1, \dots, 287$ ,  $j = 1, \dots, 384$ . For  $t = 1, \dots, 287 + 384$ , we then transform the matrix  $\bar{\lambda}_k$  into vectors by defining

$$\begin{aligned}\bar{a}(t) &= \bar{\lambda}_1(i + j), & i = 1, \dots, 287, j = 1, \dots, 384 \\ \bar{b}(t) &= \bar{\lambda}_2(i + j), & i = 1, \dots, 287, j = 1, \dots, 384 \\ \bar{c}(t) &= \bar{\lambda}_3(i + j), & i = 1, \dots, 287, j = 1, \dots, 384.\end{aligned}$$

Using the mixed matrix  $M$  as before, we have the mixed images as shown in Fig. 13, bottom panel. Input signals from  $(i, j) \in [100, 200] \times [100, 200]$  are used and we recover the faces as shown in Fig. 13.

## VI. DISCUSSION

We have presented a theoretical and numerical approach to derive novel learning rules based upon spiking neurons.

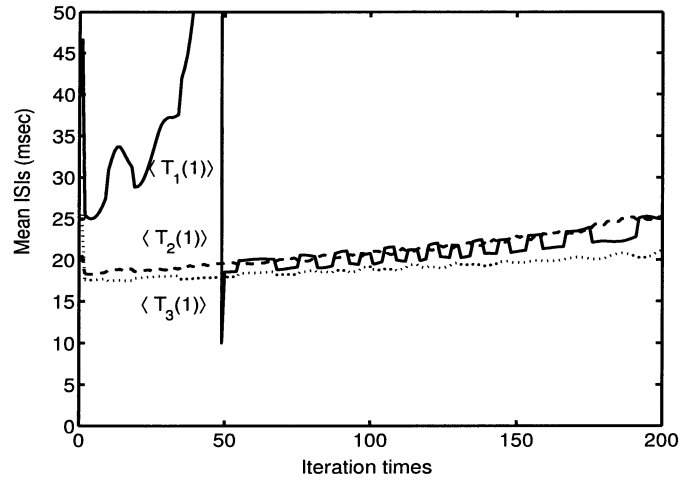


Fig. 12. Efferent ISIs (ms) versus iteration times with  $r = 1$  (see Fig. 11).



Fig. 13. Upper panel is the original faces, each with  $287 \times 384$  pixels. Middle panel is obtained after 10000 iterations and bottom panel is mixed faces.  $r = 0.5$

In particular, for the IF model and both supervised learning and unsupervised learning, we show that the model under the Infomax principle tends to disconnect some of its connections and some number of inhibitory inputs is required to optimize its performance. Here optimization means both biological and engineering senses. The conclusions are tested both with constant rate inputs and time-varying inputs.

To find a learning rule which is biologically plausible and is applicable to solving engineering problems has been a long

endeavour in neural-network research [30], [27]. The typical learning rule in biology is the Hebbian learning rule, which has been widely used in past few decades, for rate coding-based models [16]. In recent years, the Hebbian type learning rule in the time domain has been found experimentally and applied to developing models mimicking the development of visual cortex [32], [26], [6]. However, to the best of our knowledge, the application of the Hebbian type learning rule in time domain to engineering problems is less developed. The typical learning rule in engineering applications is probably the gradient descent method. To apply the gradient descent method to spiking neuron networks is not an easy task due to the nonlinear dynamics of a spiking neuron. Interestingly, the same idea, to apply the Informax principle to derive a learning rule for spiking neuron network, has been reported in the literature [9]. Unfortunately the spiking neuron used in [9] is not a genuine spiking neuron, it is a linear unit *per se* and the author terms it “perfect leaky integrator.” The obtained results are thus less interesting than a learning rule based upon a truly spiking neuron such as the IF model, as developed here.

We have only considered the IF model under the rate coding assumption here. Is the learning rule developed here applicable to time coding? The answer is affirmative. Remember the learning rule developed in the previous sections

$$\frac{\partial J}{\partial w_{ij}} = \frac{-2\langle T_i(r) \rangle}{|J|} \gamma_i + \frac{\sum_{k=1}^n (-1)^{k+i} \frac{\partial^2 \langle T_i(r) \rangle}{\partial \lambda_k \partial w_{ij}} |A_{ik}|}{\sum_{k=1}^n (-1)^{k+i} \frac{\partial \langle T_i(r) \rangle}{\partial \lambda_k} |A_{ik}|} \quad (16)$$

if we simply use  $\gamma_i$  and  $\lambda_i$  as instantaneous firing rate, i.e. the inverse of ISI, a learning rule based upon the time is obtained [6], [14], [19], [26], [32]. We will explore it in further publications. In the literature, there are increasing publications using spiking time to deal with engineering problems. Typically, they use synchronization of a group of neurons as a meaningful event [8], [27].

To summarize, the differences between our approach here and the approach used in, for example, [26], are the following.

- 1) The learning rule used here is derived from the Informax principle, but the spike-timing-dependent plasticity used in [26] is directly from experiments. Nevertheless, the learning rules are applied to spiking neurons and spiking neuronal networks.
- 2) As mentioned above, rate coding assumptions are used here, but in [26] time coding assumptions are employed. Again, as we emphasized above, we intend to carry out our experiments using the time coding assumptions as well.
- 3) Correlations between neurons are taken into account in [26], but we have simply ignored correlations in our developments. However, it is quite straightforward to generalize our results to correlated inputs [11]. It is definitely worth looking at the issue.

The approach presented here is quite general. In principle, once we know the relationship of the input-output of a neuron, we can obtain the learning rule of the neuron. As we have mentioned in the Introduction, we have accumulated many results

on the input-output relationship of a single neuron, for example the IF-FHN model [11]. We expect our approach can also shed more light onto the coding problem [14].

It is also very interesting to note that for all cases we considered, the best separation is achieved for  $a(t)$ , which is supposed to be a more natural signal than the other two. Whether it is an intrinsic property of the IF model, or more generally, a biologically realistic model, is not clear at moment.

As aforementioned, we consider a small network of neurons due to the constraints of both visualizing results and computational considerations: it is very time-consuming to run a simulation with 100 neurons. Nevertheless we are going to try simulations of large networks in the near future. It is reported that a neural system employs spatio-temporal patterns to process information. To gain a better understanding on how a neural system tackles incoming signals and then apply it to solving engineering problems, we have to mimic a biological system at a relatively detailed level.

Finally, we want to emphasize that although it is generally accepted that neurons in the cortex receive and send out Poisson process (or more general, renewal process) spike trains [1], [23], in the literature, to the best of our knowledge, it has not been rigorously tested what is the functional role of Poisson process input and output. On the contrary, it is widely accepted that the stochastic part of input signal is simply noise and harmful. Our approach here, as a first step toward exploring the truly computational function of Poisson process, reveals some primary and interesting properties. Our approach is very different from the well known stochastic resonance phenomena. In stochastic resonance, the noise level is finely tuned and kept small [4]. In our setup, as we mentioned before, the noise level is proportional to the square root of the signal and is not artificially modified.

#### ACKNOWLEDGMENT

The authors are grateful to the referees for their constructive comments on this paper and one of them for bringing [9] to our attention.

#### REFERENCES

- [1] T. D. Albright, T. M. Jessell, E. R. Kandel, and M. I. Posner, “Neural science: A century of progress and the mysteries that remain,” *Cell*, vol. 100, pp. s1–s55, 2000.
- [2] S. Amari, “Natural gradient learning for over- and under-complete bases in ICA,” *Neural Comput.*, vol. 11, pp. 1875–1883, 1999.
- [3] H. Barlow, “Perception: What quantitative laws govern the acquisition of knowledge from the senses?,” in *Functions of the Brain*, C. Coen, Ed. Oxford, U.K.: Clarendon, 1986.
- [4] A. R. Bulsara, T. C. Elston, C. R. Doering, S. B. Lowen, and K. Lindenberg, “Cooperative behavior in periodically driven noisy integrate-and-fire models of neuronal dynamics,” *Phys. Rev. E*, vol. 53, pp. 3958–3969, 1996.
- [5] A. J. Bell and T. J. Sejnowski, “An information maximization approach to blind separation and blind deconvolution,” *Neural Comput.*, vol. 7, pp. 1129–1159, 1995.
- [6] G. Q. Bi and M. M. Poo, “Activity-induced synaptic modifications in hippocampal culture: Dependence on spike timing, synaptic strength and cell type,” *J. Neurosci.*, vol. 18, pp. 10464–10472, 1998.
- [7] D. Brown, J. Feng, and S. Feerick, “Variability of firing of Hodgkin-Huxley and FitzHugh-Nagumo neurons with stochastic synaptic input,” *Phys. Rev. Lett.*, vol. 82, pp. 4731–4734, 1999.
- [8] S. R. Campbell, D. L. Wang, and C. Jayaprakash, “Synchrony and desynchrony in integrate-and-fire neurons,” *Neural Comput.*, vol. 7, pp. 1595–1619, 1999.

- [9] G. Chechik and N. Tishby, "Temporally dependent plasticity: An information theoretic account," in *Advances in Neural Information Processing Systems*, D. S. Touretzky, Ed. San Mateo, CA: Morgan-Kaufmann, 2000.
- [10] J. Feng, "Behaviors of spike output jitter in the integrate-and-fire model," *Phys. Rev. Lett.*, vol. 79, pp. 4505–4508, 1997.
- [11] —, "Is the integrate-and-fire model good enough?—A review," *Neural Networks*, vol. 14, pp. 955–975, 2001.
- [12] J. Feng, H. Buxton, and Y. C. Deng, "Training the integrate-and-fire model with the Infixmax principle I," *J. Phys. A*, vol. 35, pp. 2379–2394, 2002.
- [13] J. Feng and G. Li, "Neuronal models with current inputs," *J. Phys. A*, vol. 24, pp. 1649–1664, 2001.
- [14] W. Gerstner, A. K. Kreiter, H. Markram, and A. V. M. Herz, "Neural codes: Firing rates and beyond," *Proc. Nat. Acad. Sci. USA*, vol. 94, pp. 12740–12741, 1997.
- [15] W. Gerstner and W. M. Kristler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge, U.K.: Cambridge Univ. Press, 2002.
- [16] S. Haykin, *Neural Networks*. Upper Saddle River, NJ: Prentice-Hall, 1999.
- [17] J. J. Hopfield and A. V. M. Herz, "Rapid local synchronization of action-potentials-toward computation with coupled integrate-and-fire neurons," *Proc. Nat. Acad. Sci. USA*, vol. 92, pp. 6655–6662, 1995.
- [18] C. Koch, *Biophysics of Computation*. Oxford, U.K.: Oxford Univ. Press, 1999.
- [19] R. Kempter, W. Gerstner, and J. L. van Hemmen, "Hebbian learning and spiking neurons," *Phys. Rev. E*, vol. 59, pp. 4498–4514, 1999.
- [20] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," *Neural Comput.*, vol. 12, pp. 337–365, 2000.
- [21] R. Linsker, "An application of the principle of maximum information preservation to linear systems," in *Advances in Neural Information Processing Systems 1*, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1989.
- [22] J. P. Nadal and N. Parga, "Nonlinear neurons in the low noise limit: A factorial code maximizes information transfer," *Network*, vol. 5, pp. 565–581, 1994.
- [23] F. Rieke, D. Warland, R. d. R. van Steveninck, and W. Bialek, *SPIKES: Exploring the Neural Code*. Cambridge, MA: MIT Press, 1997.
- [24] L. M. Ricciardi and S. Sato, "Diffusion process and first-passage-times problems," in *Lectures in Applied Mathematics and Informatics*, L. M. Ricciardi, Ed. Manchester, U.K.: Manchester Univ. Press, 1990.
- [25] M. N. Shadlen and W. T. Newsome, "Noise, neural codes and cortical organization," *Curr. Opin. Neurobiol.*, vol. 4, pp. 569–579, 1994.
- [26] S. Song, K. D. Miller, and L. F. Abbott, "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neurosci.*, vol. 3, pp. 919–926, 2000.
- [27] *Neural Networks*, vol. 14, 2001.
- [28] H. C. Tuckwell, *Introduction to Theoretical Neurobiology*. Cambridge, U.K.: Cambridge Univ. Press, 1988, vol. 2.
- [29] C. van Vreeswijk, L. F. Abbott, and G. B. Ermentrout, "When inhibition not excitation synchronizes neural firing," *J. Computat. Neurosci.*, vol. 1, pp. 313–321, 1994.
- [30] S. Wermter, J. Austin, and D. Willshaw, *Emergent Neural Computational Architectures Based on Neuroscience, Toward Neuroscience-Inspired Computing*. New York: Springer-Verlag, 2001.
- [31] G. Yin and K. Yin, "Asymptotically optimal rate of convergence of smoothed stochastic recursive algorithms," *Stochast. Rep.*, vol. 47, pp. 21–46, 1994.

- [32] L. I. Zhang, H. W. Tao, C. E. Holt, W. A. Harris, and M. M. Poo, "A critical window for cooperation and competition among developing retinotectal synapses," *Nature*, vol. 395, pp. 37–44, 1998.

**Jianfeng Feng** received the B.Sc., M.Sc., and Ph.D. degrees from the Department of Probability and Statistics, Peking University, Beijing, China.

He is a Reader in informatics with the University of Sussex, Brighton, U.K. His research interests include novel variants of spiking neuron and neuron-network models, and analysis of their properties at both biophysical and abstract levels, bioinformatics, and neuroinformatics.

**Yunlian Sun** received the B.Sc. degree from Peking University, Beijing, China, in 1984, the M.Sc. degree from Wuhan University, P.R. China, in 1989, and the Ph.D. degree from Wuhan University of Hydraulic and Electrica, P.R. China, in 1999.

She has been with the school of Power Electric Engineering, Wuhan University, P.R. China, since 1984. Her research interests are adaptive and optimal motor control, signal processing, MIS, and E-G.

**Hilary Buxton** (M'XX) received the B.Sc. degree in cognitive science from the University of Bristol, Bristol, U.K., and the Ph.D. degree from Cambridge University, Cambridge, U.K.

She is Professor of visual intelligence with the University of Sussex, Brighton, U.K. Her projects have focused on biological motion understanding, Bayesian and neural learning, plus visual reasoning and control in advanced vision systems and temporal aspects of perception.

**Gang Wei** received the B.S. degree in mathematics from the Peking University, Beijing, China, in 1985, the M.S. degree in statistics from the Institute of Applied Mathematics, Academia Sinica, Beijing, in 1988, and the D.Phil. degree in mathematical science from the University of Oxford, Oxford, U.K., in 1995.

From 1987 to 1991, he was a Research Assistant with the Institute of Applied Mathematics, Academia Sinica, Beijing, and from 1995 to 1997, he was a Postdoctoral Research Assistant at the Department of Statistics, Oxford, U.K. He is currently an Assistant Professor at the Department of Mathematics, Hong Kong Baptist University. His D.Phil. dissertation establishes the equivalence framework of the point processes embedded in the linear population network and the Poisson processes driven by interacting diffusions. A fast algorithm for the maximum likelihood estimator and other related statistical inference schemes for the Cox process with Feller diffusion intensity was developed. His postdoctoral research concentrated on the probability analysis of two-way diffusions. His current research interests include the probability modeling of single neuron firing pattern, the wavelet analysis and the statistical inference for the EMG and basal ganglia records, the design and implementation of Monte Carlo and quasi-Monte Carlo algorithms.