A Spiking Neuron as Information Bottleneck

Lars Buesing
lars@igi.tugraz.at
Wolfgang Maass
maass@igi.tugraz.at
Institute for Theoretical Computer Science, Graz University of Technology, A-8010 Graz, Austria

Neurons receive thousands of presynaptic input spike trains while emitting a single output spike train. This drastic dimensionality reduction suggests considering a neuron as a bottleneck for information transmission. Extending recent results, we propose a simple learning rule for the weights of spiking neurons derived from the information bottleneck (IB) framework that minimizes the loss of relevant information transmitted in the output spike train. In the IB framework, relevance of information is defined with respect to contextual information, the latter entering the proposed learning rule as a "third" factor besides pre- and postsynaptic activities. This renders the theoretically motivated learning rule a plausible model for experimentally observed synaptic plasticity phenomena involving three factors. Furthermore, we show that the proposed IB learning rule allows spiking neurons to learn a predictive code, that is, to extract those parts of their input that are predictive for future input.

1 Introduction

Information theory is a powerful theoretical framework with numerous important applications, including in the context of neuroscience, such as the analysis of experimental data. Information theory has also provided rigorous principles for learning in abstract and more biological realistic models of neural networks. Especially the learning objective of maximizing information transmission of single neurons and neural networks, a principle often termed InfoMax, has been intensively studied in Linsker (1989), Bell and Sejnowski (1995), Chechik (2003), Toyoizumi, Pfister, Aihara, and Gerstner (2005), and Parra, Beck, and Bell (2009). This learning principle has been shown to be a possible framework for independent component analysis; furthermore, it could successfully explain aspects of synaptic plasticity experimentally observed in neural tissue. However, one limitation of this learning objective for gaining a principled understanding of computational processes in neural systems is that the goal of numerous types

Neural Computation **22, 1961–1992** (2010)

of computations is not a maximization of information transmission (e.g., from sensory input neurons to areas in the brain where decision are made). Rather, a characteristic feature of generic computations (e.g., clustering and classification of data, or sorting a list of elements according to some relation) is that they remove some of the information contained in the input. Similarly, generic learning processes require the removal of some of the information originally available in order to achieve generalization capability.

Tishby, Pereira, and Bialek (1999) created a new information-theoretic framework, the information bottleneck (IB) framework, which focuses on transmitting the maximal amount of relevant information. This approach takes a step toward making computational and learning processes more amenable to information-theoretic analysis. We examine in this article whether the IB framework can foster an understanding of organizational principles behind experimentally verified synaptic plasticity mechanisms that involve a “third factor" (Sjöström & Häusser, 2006; Hee et al., 2007). These are plasticity effects where the amplitude of the synaptic weight change depends not only on the firing activity of the pre- and postsynaptic neuron, but also on a third signal that is transmitted, for example, in the form of neuromodulators or synaptic inputs from other neurons. Such third signals are known to modulate the amplitude of the backpropagating action potential, and thereby to critically influence the changes of synaptic weights elicited by spike-timing-dependent plasticity (STDP). Furthermore, we examine in this article whether one can derive from IB principles a rule for synaptic plasticity that establishes generic computation in neural circuits: the extraction of temporally stable (“slow”) sensory stimuli (see, e.g., Wiskott & Sejnowski, 2002).

The extraction of relevant features and the neglect of irrelevant information from given data is a common problem in machine learning, and it is also widely believed to be an essential step for neural processing of sensory input streams. However, which information contained in the input data is to be considered relevant is highly dependent on the context. In a seminal paper Tishby et al. (1999) proposed an information-theoretic definition of relevance with regard to a given context and also presented a batch algorithm for data compression minimizing the loss of relevant information. This framework, the IB method, is aimed at constructing a simple, compressed representation Y (relevant features, the IB) of the given input data X, which preserves high mutual information with a relevance (or target) signal R, which provides contextual or side information. In the IB framework, the amount of relevant information contained in a random variable is explicitly defined as the mutual information of this variable with the relevance signal R. Multiple algorithms rooted in the IB framework have been fruitfully applied to typical machine learning applications such as document clustering, document classification, image classification, and feature extraction for speech recognition (see Harremoes & Tishby, 2007).

Recently it has been conjectured that the IB framework might constitute one of the optimization principles underlying early neural processing of sensory input data in some organisms. Bialek, de Ruyter van Steveninck, and Tishby (2006) argued that biological agents maintain an internal representation of the external world that contains information important for their survival capabilities. More precisely, they hypothesize that only those parts of the sensory input X should be internally represented in some model Y that are predictive of the future state of the agent's environment, as only this information is relevant for the agent's future actions, which in turn increases its fitness. This learning paradigm was formalized as an IB optimization with the relevance signal R, defined as the future sensory stimuli. As an IB optimal internal representation, Y, called a predictive code, apparently depends strongly on the statistics of the environment, and as many organisms exhibit a remarkable ability to adapt to different environmental configurations, it is tempting to conjecture that the internal representation Y is (at least partially) learned during the agents lifetime. However, in the studies mentioned above, learning rules for developing this kind of internal representation in a biologically realistic setting, where the standard batch IB algorithms are implausible, are missing.

An attempt to fill this apparent gap has been made in Klampfl, Legenstein, and Maass (2009) on the level of single spiking neuron models. In this article, a single neuron is considered an information bottleneck, as it maps its high-dimensional input X to its one dimensional output spike train Y. Based on this interpretation, an online update rule has been proposed that adjusts the synaptic weights such that the neuron's output Y contains the maximal amount of relevant information with regard to a given relevance signal R, which was also modeled as a spike train. This learning rule has been shown to reliably solve numerous concrete IB optimization problems in a neural context. However the proposed learning rule, which was derived by stochastic gradient ascent on the IB objective function (essentially the amount of transmitted relevant information), has several drawbacks. The gradient of the transmitted relevant information (which determines the learning rule) was estimated using the correlation of the bottleneck neuron output Y and the relevance signal R within each single time step. This limits the "complexity" of IB problems that the neuron is able to solve, for example, this estimation cannot capture long delays between the input X and the relevance signal R or the impact of higher-order moments between the input X and the relevance variable R in the case of linear bottleneck neurons. Furthermore, the learning rule of Klampfl et al. (2009) is complicated, making it difficult to understand the learning dynamics. In addition, it contains nonlocal variables, which reduces its biological plausibility.

The goal of this article is to develop a simpler and more transparent approximate IB learning rule for spiking neurons. This new IB learning rule is based on a different estimation of the gradient of the relevant information contained in the neural output. The estimation is of a parametric nature, and it requires a given preprocessing of the relevance signal R. The main assumption is hence that the bottleneck neuron has access to a rich preprocessing of the relevance signal R. This preprocessing can be considered as a third factor, besides the presynaptic input X and the output Y, which modulates synaptic plasticity in order to implement an IB optimal coding of the inputs.

The outline of the article is as follows. In section 2, the IB framework is briefly revisited, the underlying spiking neuron model is defined, and the objective function for IB optimization for spiking neurons is introduced. We present the general IB learning rule for spiking neurons in section 3 and discuss a concrete, simple example IB task. Further, in this section, we propose an implementation of the relevance signal preprocessing using a generic recurrent neural network. In section 4, the proposed IB learning rule is used to model the learning of a predictive code. A detailed comparison to related work is presented in section 5. Furthermore, experimental results on synaptic plasticity with three factors that point out a possible implementation of the learning rule proposed in this article are discussed.

2 Neuron Model and Objective Function

In this section, the neuron model and the objective function for IB optimization with this model are defined. The model is formulated in discrete time of step size Δt. To introduce a biologically plausible timescale, we assume that a single time step corresponds to 1 millisecond: $\Delta t = 1$ ms. The value of a time-varying function f at time step t will be denoted as f^t. Further, the standard Euclidean dot product of two vectors $\boldsymbol{d} = (d_1, \ldots, d_N)$ and $\boldsymbol{e} = (e_1, \ldots, e_N)$ is written as $\boldsymbol{d} \cdot \boldsymbol{e} := \sum_{i=1}^{N} d_i e_i$. We start this section by briefly revisiting the IB method.

2.1 Information Bottleneck Method.

The IB method, originally introduced in Tishby et al. (1999), is a data compression technique that in its simplest version focuses on the following setup. Consider two random variables (RVs) a and b with a known joint distribution $p(a, b)$. The goal of the IB method is to construct an RV $\tilde{a}$, a compact, simple representation of a, via a stochastic mapping defined by the conditional probability $p(\tilde{a}\,|a)$ such that $\tilde{a}$ is still informative about b. The RV b will be called the *relevance* or *target* signal in the remainder of this article. This intuitive data compression task was formalized in Tishby et al. (1999) as a maximization problem of the objective function L_{IB}:

$$L_{\text{IB}} = I(\tilde{a}, b) - \gamma I(\tilde{a}, a).$$

Here $I(.,.)$ denotes the mutual information between the two arguments. The first term $I(\tilde{a}, b)$ of L_{IB} measures how informative the compressed representation $\tilde{a}$ is about b. The second term $I(\tilde{a}, a)$ with the Lagrange multiplier $\gamma > 0$ penalizes complex representations $\tilde{a}$ and can be regarded as an information-theoretic regularization term.[1] The IB method consists of finding a conditional probability distribution $p(\tilde{a}|a)$ that maximizes L_{IB} under the condition that $\tilde{a}, a$, and b form the Markov chain $b \to a \to \tilde{a}$.[2] The parameter $\gamma \in [0, 1]$ determines the degree of compression via the trade-off between the relevant information that $\tilde{a}$ carries about b and the complexity of $\tilde{a}$. For $\gamma = 0$, the representation $\tilde{a}$ is uncompressed, and all relevant information is preserved; that is, L_{IB} is maximal, for example, for the identity mapping and $I(\tilde{a}, b) = I(a, b)$. At the other extreme, for $\gamma = 1$, the variable $\tilde{a}$ is maximally compressed and always assumes a single value, resulting in $I(\tilde{a}, a) = 0$ and $I(\tilde{a}, b) = 0$.

An application in machine learning that illustrates the merits of the IB method is the feature selection for document classification as presented in Slonim and Tishby (2001). In this setup, the uncompressed input a corresponds to words, which occur in the documents, and the relevance variable b is chosen to be the class label (i.e., the document category, such as "sports" or "politics"); the joint distribution of words and document categories $p(a, b)$ is assumed to be known for a given training set. Via the IB method, it is possible to obtain a mapping $p(\tilde{a}|a)$ yielding a simple representation $\tilde{a}$ (word clusters instead of single words), which still carries most of the relevant information about the document class. These low-dimensional word clusters can then be conveniently used as features for document classification of test data.

2.2 Neuron Model. We consider a simple stochastic neuron model similar to the ones used in Toyoizumi et al. (2005) and Klampfl et al. (2009), however without taking a refractory mechanism into account. The neuron has N synapses with weights $\boldsymbol{w} = (w_1, \ldots, w_N)$, which we require to be nonnegative. It is driven by the input $X = (X_1, \ldots, X_N)$, consisting of N spike trains $X_j = (\ldots, x_j^{-1}, x_j^0, x_j^1, \ldots)$, formalized as left and right infinite sequences. We define $x_j^t = 1$ if there is a presynaptic spike at synapse j at time step t, and $x_j^t = 0$ otherwise. The spikes at synapse j from time step l up to t ($l < t$) are written as $X_j^{l,t} = (x_j^l, x_j^{l+1}, \ldots, x_j^t)$; further, the input history up to time step $t = 0$ of synapse j is denoted as $X_j^{-\infty} := (\ldots, x_j^{-1}, x_j^0) = X_j^{-\infty,0}$ and $X^{-\infty} = (X_1^{-\infty}, \ldots, X_N^{-\infty})$. The membrane potential u^t of the neuron at time t is given by the weighted sum of the synaptic activities

[1] The IB objective function in Tishby et al. (1999) was originally introduced with the opposite sign, and it was parameterized in terms of $\beta := \gamma^{-1}$.

[2] This condition is equivalent to requiring $\tilde{a}$ to be independent from b given a.

$\boldsymbol{v}^t = (v_1^t, \ldots, v_N^t)$:

$$u^t = \boldsymbol{w} \cdot \boldsymbol{v}^t = \sum_{j=1}^{N} w_j v_j^t$$

$$v_j^t = (\epsilon * X_j)^t = \sum_{l=-\infty}^{\infty} \epsilon^l x_j^{t-l}. \tag{2.1}$$

The kernel ϵ models the excitatory postsynaptic potential (EPSP) of a single spike, and $*$ denotes the discrete time convolution. Given the input X, the postsynaptic neuron spikes at time step t with the probability $p(y^t = 1|X^{-\infty,t})$, which is a function of the membrane potential:

$$p(y^t = 1|X^{-\infty,t}) = g(u^t) = g^t.$$

The function g is called the activation function. Its image is assumed to be in $[0, 1]$, and it is assumed to be continuously differentiable with a derivative $g'(u^t) =: g'^t$. The postsynaptic spike train is denoted as $Y = (\ldots, y^{-1}, y^0, y^1, \ldots)$ with $y^t = 1$ if an output spike occurs at time step t, and 0 otherwise. In simulations, the EPSP kernel ϵ was chosen to be a nonanticipating, decaying exponential with a time constant of 10 time steps, and the activation function $g(u^t) = \sigma(u^t - u_0)$ was chosen as the logistic function $\sigma(x) := (1 + \exp(-x))^{-1}$ with an offset $u_0 = -2$.

Furthermore, according to the IB framework, another external signal besides the input X is given, namely, the relevance or target signal R. We consider situations where R is given by a stochastic process denoted by the sequence $R = (\ldots, R^{-1}, R^0, R^1, \ldots) \in \mathbb{R}^{\mathbb{Z}}$. It is not restricted to spike trains, and it may be given by a more general real-valued sequence. It is straightforward to extend the results presented below to multidimensional relevance variables. We assume that R does not directly influence the activity of the neuron, but that it takes part in the process of the synaptic plasticity only in order to ensure that $R \to X \to Y$ is a Markov chain as required by the IB framework. For simplicity, we assume that the processes X and R are stationary.

2.3 Applying the IB Framework to Spiking Neurons. Following the approach taken by Klampfl et al. (2009), we apply the IB framework to a single neuron as illustrated in Figure 1. At any given time step t, without loss of generality, we may assume $t = 0$, the neuron under consideration, which we call the bottleneck neuron from now on, maps its input history $X^{-\infty}$ to an output $y^0 \in \{0, 1\}$. Hence the neuron can be regarded as an information bottleneck, which compresses its high-dimensional input history $X^{-\infty}$ (corresponding to a in the notation introduced in section 2.1) to its one-dimensional binary output y^0 (corresponding to $\tilde{a}$). This mapping is parameterized by the weight vector $\boldsymbol{w}$ for which we want to find

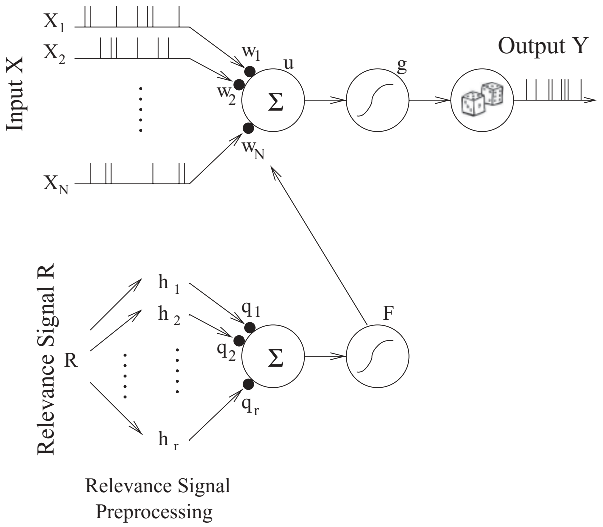

Figure 1: General setup for IB optimization with a spiking neuron. The neuron receives input spike trains X_i for $i = 1, \ldots, N$ and emits the output Y. Furthermore, a second signal, the relevance signal R, is given, which allows introducing the notion of relevance of information. The weights $\boldsymbol{w}$ should be learned such that the relevant information $I(y^0, R)$ contained in the neuron output is maximal (under regularization constrains). In order to carry out this optimization in an online manner, an estimation of the gradient of $I(y^0, R)$ with regard to $\boldsymbol{w}$ is required, which is based on the quantity F^0. The latter is a parameterized function (with parameters $\boldsymbol{q}$) of a given preprocessing $h = (h_1, \ldots, h_r)$ of the relevance signal. The parameters $\boldsymbol{q}$ are adapted such that F^0 optimally predicts the neural output y^0 given the relevance signal R.

the configuration giving rise to the output of the neuron that is maximally informative about the relevance signal R (corresponding to b in section 2.1). There are multiple possible ways of formalizing this setup in the IB framework, more precisely of choosing the IB objective function.

2.3.1 The Choice of the IB Objective Function for Spiking Neurons. We define the amount of relevant information transmitted by the neuron per time step as the mutual information $I(y^0, R)$ between the current output y^0 and the whole relevance signal R. Hence, following the IB framework, we are looking for the synaptic weight $\boldsymbol{w}$ that maximizes the IB objective function L_{IB} with a regularization term L_{reg}:

$$\begin{aligned} L_{\mathrm{IB}} &= I(y^0, R) - \gamma L_{\mathrm{reg}} \\ &= \left\langle \log\left(\frac{p(y^0|R)}{p(y^0)}\right)\right\rangle - \gamma L_{\mathrm{reg}}, \end{aligned} \tag{2.2}$$

where the brackets $\langle . \rangle$ denote the expected value over the input spike trains X, the output spike train Y, and the relevance signal R. Further, $p(y^0)$ and $p(y^0|R)$ denote the unconditioned spiking probability and the spiking probability conditioned on the relevance signal, respectively.

Our definition of the relevant information as $I(y^0, R)$ can be interpreted as the limit of the mutual information $I(y^0, R^{-T,T})$ between y^0 and the relevance signal $R^{-T,T}$ in a time window of length $2T + 1$ for $T \to \infty$ (i.e., $I(y^0, R) = \lim_{T\to\infty} I(y^0, R^{-T,T})$). This choice eliminates "cut-off" artifacts like the following. If the relevant information contained in the input X arrived at the bottleneck neuron with a delay of $T + 1$ relative to the relevance signal R, the objective function $I(y^0, R^{-T,T})$ would be insensitive to this statistical relation. One might reckon that the choice to maximize $I(y^0, R)$ introduces anticipatory effects, for example, that for adapting its weights $\boldsymbol{w}$, the bottleneck neuron would need information that will be available only in the future. Such effects will, however, not show up in our approach, as it is explicitly designed to take into account only information for the IB optimization that is currently available to the neuron, as outlined in the next section.

Alternative definitions of the relevant information are also possible, of course. It might be argued that the mutual information $I(y^{-T,T}, R^{-T,T})$ between the neuron output and the relevance signal in some time window is a more natural definition of the relevant information. However, it turned out that the online optimization of $I(y^{-T,T}, R^{-T,T})$ is considerably more difficult than the one of $I(y^0, R)$ due to accounting for relations between multiple output spikes of the bottleneck neuron. These technical difficulties motivated the choice of optimizing $I(y^0, R)$.

As stated in section 2.1 the regularization L_{reg} in the original IB formulation from Tishby et al. (1999) was given by the mutual information between the input and the output of the IB mapping, that is, in this setup $L_{\mathrm{reg}} = I(y^0, X^{-\infty})$. This choice is also possible in the neural context considered here. However, simulation results indicate that for this definition of L_{reg}, sensible values of the trade-off parameter γ—those values of γ that neither "unregularize" nor "overregularize" (basically the $\boldsymbol{w} = 0$) the IB optimization—are confined to a small interval and are thus hard to be determined numerically. Therefore, we replace the original regularization with a conventional quadratic regularization of the weights $\boldsymbol{w}$:

$$L_{\mathrm{reg}} = \frac{1}{2}\boldsymbol{w}^2.$$

With this choice of L_{reg}, it is considerably easier to determine sensible values of γ. Other choices of L_{reg} are also possible, such as penalizing deviations from an average target firing rate. It is straightforward to incorporate such a regularization into the objective function presented here.

2.3.2 Online Estimation of the Relevant Information. Eventually we wish to maximize L_{IB} defined in equation 2.2 via a stochastic gradient ascent with regard to the weights $\boldsymbol{w}$ yielding an online update rule. However, this maximization requires explicit knowledge of the conditional distribution $p(y^0|R)$ of the output y^0 given the relevance signal R, as can be seen from equation 2.2. Most IB algorithms resolve this issue by estimating the joint distribution of the input data and the relevance signal (here, $p(X^{-\infty}, R)$) from the whole batch data set and subsequently evaluating the conditional distribution of the compressed output variable given the relevance signal (here, $p(y^0|R)$) using the fact that $R \to X \to Y$ is a Markov chain. However, this approach seems plausible only in an offline, batch IB optimization task where the entire data set is available at all times. In the neural setup considered here, we do not want to, assume that the neuron has all information about the joint distribution of $X^{-\infty}$ and R; rather, it should estimate $p(y^0|R)$ and the relevant information $I(y^0, R)$ online. As this can be arbitrarily difficult (depending on the "complexity" of $p(X^{-\infty}, R)$), we can only hope to solve the neural IB optimization task approximately under some simplifying assumptions.

A possible strategy that addresses the problem outlined above is the following. As its output y^0 is binary, the neuron has to estimate $p(y^0 = 1|R)$ only in order to determine an approximation of $I(y^0, R)$ (and its gradient). We therefore assume the neuron has access to a parametric estimation $F^t \approx p(y^t = 1|R)$ with r parameters $\boldsymbol{q} = (q_1, \ldots, q_r)$. For simplicity, we restrict ourselves to the case where F^t is of the form $F^t = \sigma(\boldsymbol{q} \cdot \boldsymbol{h}^t)$, with σ denoting the logistic function that ensures $F^t \in [0, 1]$. The quantities $h = (h_1, \ldots, h_r)$ are r given filters[3] operating on the relevance sequence R and $\boldsymbol{h}^t = (h_1^t, \ldots, h_r^t)$ denotes their values at time step t. These filters h are a preprocessing of the relevance signal R that is currently available to the neuron. In a neural system, it might be implemented by some neural circuitry that carries out transformations of the sequence R. In simulations, h can be modeled, for example, by moving averages or Volterra series of the relevance signal R; concrete examples for such a preprocessing as well as for a preprocessing with a simulated neural circuitry are given below. The concrete choice of the form of the estimator F^t, a linear model in the parameters $\boldsymbol{q}$ followed by the logistic function, results in a simple online learning rule (due to similarities with logistic regression). Other forms of F^t that also allow simple gradient ascent could potentially be worth studying.

Based on the estimator F^t (see Figure 1 for a visualization of this approach), we propose the following objective function L to be maximized

[3] We define a filter $h_i : \mathbb{R}^{\mathbb{Z}} \to \mathbb{R}^{\mathbb{Z}}$ as a mapping from left-right infinite sequences to left-right infinite sequences with $R \mapsto h_i[R] = (\ldots, h_i[R]^{-1}, h_i[R]^0, h_i[R]^1, \ldots)$. F^t is precisely defined as $F^t = \sigma(\boldsymbol{q} \cdot \boldsymbol{h}[R]^t)$. For simplicity in the main text, the shorter notation is used.

with regard to $\boldsymbol{w}$ and $\boldsymbol{q}$,

$$L = L_F - \gamma L_{\text{reg}} = \left\langle \log\left(\frac{F(y^0, R)}{p(y^0)}\right)\right\rangle - \frac{\gamma}{2}\boldsymbol{w}^2, \tag{2.3}$$

where $F(y^0, R) = (F^0)^{y^0}(1 - F^0)^{1-y^0} \approx p(y^0|R)$. The term γL_{reg} represents the regularization term, which remains unchanged from the objective function L_{IB} given in equation 2.2. The term L_F is the approximation of the relevant information $L_F \approx I(y^0, R)$ based on F^0, and it can easily be shown that the following relation holds:

$$L_F = \left\langle \log\left(\frac{F(y^0, R)}{p(y^0)}\right)\right\rangle = I(y^0, R) - \langle D_{\text{KL}}(p(y^0|R) \| F(y^0, R))\rangle$$

$$\text{with} \quad D_{\text{KL}}(p(y^0|R) \| F(y^0, R)) = \sum_{y^0=0}^{1} p(y^0|R) \log\left(\frac{p(y^0|R)}{F(y^0, R)}\right), \tag{2.4}$$

where $D_{\text{KL}}(P \| Q)$ denotes the Kullback-Leibler divergence between P and Q. It can be seen from equation 2.4 that optimizing L with regard to $\boldsymbol{q}$ for fixed weights $\boldsymbol{w}$ amounts to minimizing the Kullback-Leibler divergence between the estimation $F(y^0, R)$ and the "true" conditional distribution $p(y^0|R)$. The divergence $\langle D_{\text{KL}}(p(y^0|R) \| F(y^0, R))\rangle$ assumes its unique minimum at $F(y^0, R) = p(y^0|R)$. On the other hand, maximizing L with regard to $\boldsymbol{w}$ for fixed $\boldsymbol{q}$ (i.e., for fixed F^0) can be interpreted as maximizing an estimation L of the "true" IB objective function L_{IB}. It can easily be shown that this is equivalent to maximizing the difference of the transmitted information $I(y^0, X^{-\infty})$ and the Kullback-Leibler divergence $\langle D_{\text{KL}}(p(y^0|X^{-\infty}) \| F(y^0, R))\rangle$:

$$L_F = I(y^0, X^{-\infty}) - \langle D_{\text{KL}}(p(y^0|X^{-\infty}) \| F(y^0, R))\rangle.$$

The objective function L also has the following pleasant property:

$$L \leq L_{\text{IB}}.$$

This ansatz can hence be understood as maximizing a lower bound L of the "true" IB objective function L_{IB}. In a batch setting, this optimization problem could be solved by an algorithm with two alternating steps that are iterated, reminiscent of the expectation-maximization algorithm. In the first step, minimize $\langle D_{\text{KL}}(p(y^0|R) \| F(y^0, R))\rangle$ with regard to $\boldsymbol{q}$ for fixed $\boldsymbol{w}$. In the second step, minimize $\langle D_{\text{KL}}(p(y^0|X^{-\infty}) \| F(y^0, R))\rangle - I(y^0, X^{-\infty}) + \gamma L_{\text{reg}}$ with regard to $\boldsymbol{w}$ for fixed $\boldsymbol{q}$. These two steps are iterated until a termination condition is fulfilled.

In the remainder of the article, we investigate an online optimization scheme for $\boldsymbol{w}$ and $\boldsymbol{q}$ that is obtained by a stochastic gradient ascent on L. For deriving this online learning rule, it is advantageous to rewrite L in the following form:

$$L = \langle \log F(y^0, R) \rangle + H(y^0) - \frac{\gamma}{2}\boldsymbol{w}^2, \tag{2.5}$$

where $H(y^0)$ denotes the entropy of y^0.

3 IB Learning Rule for Spiking Neurons

3.1 Online Learning Rule. From the objective function L defined in equation 2.3, an online learning rule for $\boldsymbol{w}$ can be obtained by performing a stochastic gradient ascent with a learning rate η_w:

$$\Delta \boldsymbol{w}^t = \boldsymbol{w}^{t+1} - \boldsymbol{w}^t = \eta_w L_w, \quad \text{with} \quad \langle L_w \rangle = \frac{\partial L}{\partial \boldsymbol{w}},$$

and analogously for the parameters $\boldsymbol{q}$ with a learning rate η_q. As shown in appendix A, this leads to the following equations:

$$\begin{aligned} \Delta \boldsymbol{w}^t &= \eta_w g'^t \boldsymbol{v}^t \left(\sigma^{-1}(F^t) - \sigma^{-1}(\langle g^t \rangle)\right) - \eta_w \gamma \boldsymbol{w} \\ \Delta \boldsymbol{q}^t &= \eta_q \boldsymbol{h}^t (y^t - F^t), \end{aligned} \tag{3.1}$$

where σ^{-1} is the inverse logistic function and $\langle g^t \rangle$ is the average firing rate of the neuron. Further, g'^t denotes the derivative of the activation function at time step t, and $F^t = \sigma(\boldsymbol{q}^t \cdot \boldsymbol{h}^t)$ denotes the estimator of $p(y^t = 1|R)$. For online learning, $\langle g^t \rangle$ is estimated by a running average $\hat{g}^t$ of g^t over an exponential time window of width η_g^{-1}:

$$\hat{g}^{t+1} = (1 - \eta_g)\hat{g}^t + \eta_g g^t.$$

Apart from the multiplicative term g'^t, which modulates the amount of weight change $\Delta \boldsymbol{w}^t$ with the sensitivity (i.e., the derivative) of the activation function, learning rule 3.1 consists basically of three additive terms. These terms correspond to the gradients of the estimation $\langle \log F(y^0, R) \rangle \approx -H(y^0|R)$, of the entropy $H(y^0)$ and of the regularization L_{reg}, which stem from the three additive terms of L in the form of equation 2.5. The first term $\boldsymbol{v}^t \sigma^{-1}(F^t)$, being proportional to the gradient of $\langle \log F(y^0, R) \rangle$, increases those weights w_j whose synaptic activity v_j^t correlates with the estimator F^t (as σ^{-1} is just a monotonic rescaling). Thus, those weights are potentiated whose activity can be well predicted by the estimator F^t and hence carry relevant information. The second term, which is proportional to $-\sigma^{-1}(\langle g^t \rangle) = \log((1 - \langle g^t \rangle)/\langle g^t \rangle)$, stemming from the gradient of $H(y^0)$, changes the weights in order to achieve a high entropy of the output y^0 (which is maximal at $\langle g^t \rangle = 1/2$). This term can be interpreted as a homeostatic control on a long time-scale, as the average $\langle g^t \rangle$ is slowly changing due to changes of the weights. It pushes the activity of the neuron toward a working regime of optimal information transmission. The last term $-\boldsymbol{w}$ from equation 3.1 represents the gradient of the regularization $-\boldsymbol{w}^2/2$ and yields a conventional weight decay term.

The update rule for the parameters $\boldsymbol{q}$ is proportional to the difference of the neuron output y^t and F^t. The parameters $\boldsymbol{q}$ assume stationary values if, for example, the estimation F^t fulfills $F^t = \langle y^t \rangle_{Y|X,R}$.[4]

3.2 A Simple Example. In this section, the IB learning rules 3.1 for adapting the weights $\boldsymbol{w}$ and parameters $\boldsymbol{q}$ are applied to a simple IB optimization task. The inputs to the neuron, as well as the relevance signal, consist of (discrete-time) Poisson spike trains. Some of the input spike trains exhibit a statistical dependence on the relevance signal on the level of precise spike times, and hence carry relevant information. When learning rule 3.1 is used, the neuron should learn to exclusively potentiate the weights of these input channels and neglect the remaining inputs.

Consider the following setup, which is shown in Figure 2. Let the $N = 100$ inputs $X = (X_1, \ldots, X_{100})$ to the bottleneck neuron be arranged into three groups $G_1,\ G_2,\ G_3$ consisting of 25, 25, and 50 neurons, respectively. The inputs X_i, as well as the relevance signal R, are given by spike trains (i.e., binary sequences in $\{0, 1\}^{\mathbb{Z}}$) of constant rate[5] $\nu_X = 0.02$ and $\nu_R = 0.06$ (corresponding to 20 Hz and 60 Hz for a time step $\Delta t = 1$ ms). Spike trains from different input groups are statistically independent. Furthermore, the inputs are generated such that spike trains from the input groups G_1 and G_2 exhibit a correlation coefficient (CC) with the relevance spike train R of $c_1 = 0.1$ and $c_2 = 0.075$ due to coincident spikes of X_i and R within one time step. Spike trains of G_3 are highly correlated with each other with a CC of $c_3 = 0.2$. Here the CC c between two spike trains $x_i(t)$ and $x_j(t)$ is defined as $c = \langle (x_i(t) - \langle x_i(t) \rangle)(x_j(t) - \langle x_j(t) \rangle) \rangle / (\mathrm{var}(x_i(t))\mathrm{var}(x_j(t)))^{1/2}$, where $\mathrm{var}(x_i(t))$ denotes the variance of $x_i(t)$. In this setup, the inputs of the groups G_1 and G_2 carry relevant information, whereas inputs of G_3 have interesting statistics (i.e., high correlation) but nevertheless are irrelevant due to the definition of R. Further simulation parameters and details can be found in section 5.3. The preprocessing $\boldsymbol{h}^t = (h_1^t, h_2^t)$ was chosen in the following way. The first element $h_1^t = 1$ is a constant bias, whereas

[4] Here $\langle f \rangle_{Y|X} = \sum_Y f \cdot p(Y|X)$ denotes the expected value of f over Y conditioned on X.

[5] We define the rate of a spike train X_j at time step t as the current probability to spike $p(x_j^t = 1)$.

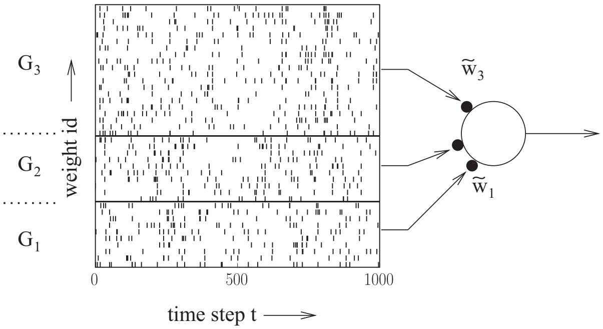

Figure 2: Setup of the simple IB task described in section 3.2. The synapses of the neuron are arranged in three groups G_1 to G_3 whose average weights are denoted as $\tilde{w}_1$ to $\tilde{w}_3$. The inputs to the neuron, illustrated here by a spike raster plot (notice that only 2/5 of the spike trains of each group are shown), as well as the relevance signal are modeled as spike trains. The different groups convey different amounts of relevant information in their precise spike timings, parameterized by the correlation coefficient between the input and the relevance signal. IB learning rule 3.1 adapts the weights $\boldsymbol{w}$ such that eventually the output of the neuron is most informative about the relevance signal (with regularization).

$h_2^t = \sum_{s=0}^{\infty} \exp(-s/\tau) R^{t-s}$ is a low-pass filter of the relevance spike train with an exponential window of size $\tau = 10$.

In Figure 3 the results of a simulation of this setup with a trade-off parameter $\gamma = 8 \cdot 10^{-6}$ are plotted. Figure 3A shows the temporal evolution of the average group weights $\tilde{w}_a = |G_a|^{-1} \sum_{i \in G_a} w_i$ of group G_a for $a = 1, 2, 3$ with group size $|G_a|$. It can be observed that the average group weights $\tilde{w}_a$ converge to a value roughly proportional to the CC between the corresponding input spike trains and the relevance signal; for example, the weights of G_1 are strongest after the learning. The inputs of G_3 do not carry relevant information by construction, and therefore the weights decay toward zero due to the regularization term of the objective function 2.3. The dynamics of the parameters $\boldsymbol{q}$ are plotted in Figure 3B. They eventually assume stationary values that are (possibly locally) optimal values for estimating the output probability $p(y^t = 1|R)$ conditioned on the relevance signal R by the estimator $F^t = \sigma(\boldsymbol{q}^t \cdot \boldsymbol{h}^t)$. An estimation of the IB objective function L_{IB} (for details, see section B.1) and of the lower bound L are shown in Figure 3C. Both measures increase over time due to the stochastic gradient ascent learning. Furthermore, L is quite "tight" for this task and provides the neuron with a good estimation of the "true" value of the objective function L_{IB} as well as its gradient with regard to $\boldsymbol{w}$ and $\boldsymbol{q}$. Additionally the regularization L_{reg} is plotted separately to illustrate its contribution to the total objective function L.

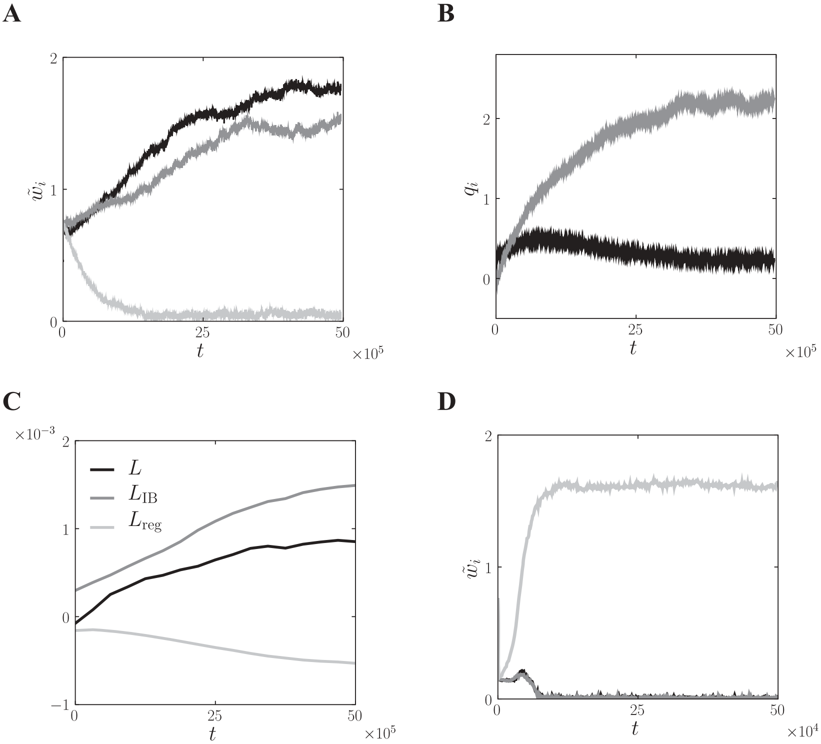

Figure 3: Results of the simulation described in section 3.2. (A) Trajectories of the average group weights $\tilde{w}_i$ for $i = 1, \ldots, 3$ as a function of the time step t. The weights of G_1 (black) and G_2 (gray) are increased as they have nonvanishing mutual information with the relevance signal R. The weights of the remaining group G_3 (light gray) decay to zero as the corresponding inputs are independent of R. (B) The trajectories of the parameters q_1 (black) and q_2 (gray). They evolve such that $\sigma(\boldsymbol{q}^t \cdot \boldsymbol{h}^t)$ optimally estimates the conditional probability $p(y^t = 1|R)$ (in terms of the Kullback-Leibler divergence). (C) Numerical estimations of the IB objective function L_{IB} (gray) and of the lower bound L (black) are plotted as functions of time step t. One sees that the lower bound gives a good estimation of L_{IB} in this example task. Furthermore, the regularization term L_{reg} is plotted (light gray). (D) Results of applying an InfoMax learning rule to the same setup. InfoMax does not take the relevance signal R into account, and therefore weights of group G_3 get potentiated (color coding as in panel A).

In Figure 3D we show results of a simulation using the same setup as described above, with the only difference that the weights are learned not with the IB learning rule but with an InfoMax learning rule. InfoMax aims at maximizing the amount of transmitted information $I(y^0, X^{-\infty})$ between the input and output of the neuron without taking the relevance signal R

into account. It can be seen that in contrast to the results of IB learning, InfoMax potentiates the weights of G_3 as their inputs exhibit the strongest correlation with each other. The InfoMax rule is given in section A.2, and a more general comparison to IB learning can be found in section 5.

3.3 Neural Implementation of the Relevance Signal Preprocessing. In section 3.2 a simple IB optimization task was solved assuming that the filters $h = (h_1, \ldots, h_r)$ provide a suitable preprocessing of the relevance signal R (in that case, a low-pass filter of the relevance signal and a constant bias). In this example the preprocessing was quite specific for the given IB task (i.e., specific for the distribution $p(X^{-\infty}, R)$), and one might argue that the neuron could not have solved other IB tasks with this preprocessing. In this section, we address this point by proposing the implementation of the relevance signal preprocessing by a generic neural circuit, which is not tailored for a single IB task but allows the bottleneck neuron to solve a larger class of IB tasks with the same preprocessing filters. These tasks may also feature more statistically complex dependencies between the neural input X and the relevance signal R, in particular in the temporal domain.

In the approach for IB optimization presented above, it is essential for the bottleneck neuron to have a reasonable approximation F^t of the conditional probability $p(y^t = 1|R)$ in order to optimize the weights $\boldsymbol{w}$. The quality of the lower-bound L, which is optimized (i.e., the difference $|L_{\text{IB}} - L|$), is given by the Kullback-Leibler divergence between the estimation $F^t = \sigma(\boldsymbol{q} \cdot \boldsymbol{h}^t)$ and the "true" value $p(y^t = 1|R)$. Hence $|L_{\text{IB}} - L|$ critically depends on the given preprocessing h of R. Ideally the preprocessing would be powerful enough such that $F^t = p(y^t = 1|R)$ for some parameters $\boldsymbol{q}$, and optimizing L would then be equivalent to optimizing the "true" IB objective function L_{IB}. Maass, Natschlaeger, and Markram (2002) investigate the general problem of approximating with a fixed set of preprocessing filters (here $h = (h_1, \ldots, h_r)$) and a fixed class of memoryless readout functions (here, the linear maps parameterized by $\boldsymbol{q}$ followed by the logistic function σ) for any given target filter (here $p(y^t = 1|R)$).[6] A largely positive result is given for approximating target filters that are time invariant (TI) and have the fading memory (FM) property (for exact definitions and results, see Maass et al., 2002) under suitable assumptions concerning the set of filters and the set of readout maps. Roughly speaking, a filter has fading memory if it becomes asymptotically insensitive to the remote history of its input. A further specific result given in Boyd and Chua (1985) states that TI-FM filters can be approximated with arbitrary precision by a finite-dimensional, linear dynamical system implementing the filters and a polynomial readout map.

[6] The considerations in Maass et al. (2002) focus on the case of continuous time, but similar results also hold for discrete time.

Based on the theoretical results of Maass et al. (2002), in a series of publications (for a review see Buonomano & Maass, 2009; for a similar approach, see Jäger, 2001), it was observed that various TI-FM filters can be efficiently approximated using a fixed generic neural network implementing the filters h and exclusively learning a memoryless linear readout function. This approach exploits that sufficiently large recurrent networks of nonlinear neurons provide a sufficiently generic nonlinear preprocessing. Hence, it often suffices to use linear, rather than polynomial, adaptive readouts. More precisely, in this approach, the filters h are implemented by a sufficiently complex recurrent neural network that is generated randomly (in particular, the network is not designed for approximating a specific filter) and receives an external input given by the signal on which the target filter operates on (here, the relevance signal R). The value $\boldsymbol{h}^t$ of the filters h at time step t is then defined, for example, as the vector of neuron activations (for continuous networks) or the output spikes of the network units at time step t. The readout map in these studies was restricted to linear maps, $\boldsymbol{q}^t \cdot \boldsymbol{h}^t$, and only the parameters $\boldsymbol{q}$ are learned in order to approximate the given specific target filter. This neural architecture poses a sensible implementation of the preprocessing h in the IB setup if the target filter $\sigma^{-1}(p(y^t = 1|R))$ can be assumed to have the FM property (it is guaranteed to be TI if R and X are stationary processes). The FM property amounts to assuming that input spikes x_i^t become asymptotically independent from the relevance signal $R^{t\pm\tau}$ for large delays τ. In the following paragraph, an example IB task is discussed that illustrates this approach. We show that in this example, a generic recurrent network with a trainable linear readout provides the bottleneck neuron with a sufficiently accurate estimation $F^t = \sigma(\boldsymbol{q} \cdot \boldsymbol{h}^t)$ of $p(y^t = 1|R)$, allowing it to solve a given IB task.

Consider the following setup. Let the relevance signal R be a piece-wise constant, real-valued stochastic process that assumes every 30 time steps a new value that is identically and independently distributed in $[-0.5, 0.5]$, (see Figure 4A). The input spike trains X_i are arranged in four subgroups G_1 to G_4 similar to the setup of the example given in section 3.2. The inputs of the group G_j were generated as spike trains with a time-varying rate λ_j^t at time step t. The rate λ_1^t was defined as

$$\lambda_1^t = a\, R^{t-\tau_1} R^{t-\tau_2} + b, \tag{3.2}$$

with delays $\tau_1 = 10$ and $\tau_2 = 50$ time steps and coefficients $a,\ b$. The remaining rates λ_2^t, λ_3^t, λ_4^t were generated with the same statistics as λ_1^t, but they are independent from R. By construction, only inputs of G_1 contain relevant information, whereas the remaining inputs do not. The preprocessing of the relevance signal was implemented by a recurrent network of $r = 200$ sigmoidal rate neurons, which receives the relevance signal R as input; that is, the values of the filters $\boldsymbol{h}^t$ at time step t were chosen as the network activity at time step t. According to the approach proposed above, the estimation F^t was given by $\sigma(\boldsymbol{q} \cdot \boldsymbol{h}^t)$, and the parameters $\boldsymbol{q}$ were

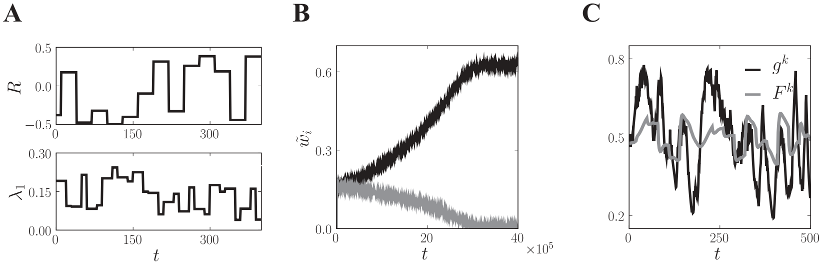

Figure 4: Numerical results for an IB optimization task with a preprocessing of the relevance signal R implemented by a generic recurrent network as described in section 3.3. (A) Shown is the relevance signal R (top) as well as the spiking probability λ_1 (bottom) for the inputs of group G_1. By design, only λ_1 is statistically dependent on R, and hence only the inputs of G_1 carry relevant information. (B) The trajectories of the mean weights $\tilde{w}_1$ (black) of G_1 and $\tilde{w}_\gamma$ (gray) of the remaining groups G_2, G_3, G_4 are plotted as functions of time step t. As only the inputs of G_1 have a nonvanishing mutual information with R, $\tilde{w}_1$ is exclusively potentiated. The average $\tilde{w}_\gamma$ of the remaining weights decays due to regularization. (C) Trajectories of the activation function $g^t = p(y^t = 1|X^{-\infty,t})$ (black) and of the estimator F^t (gray) are shown for an interval of 500 time steps after the weights $\boldsymbol{w}$ and the parameters $\boldsymbol{q}$ have been learned.

learned by equation 3.1. The quantity F^t can be interpreted as the activity of a logistic readout neuron with input $\boldsymbol{h}^t$ from the recurrent network and weights $\boldsymbol{q}^t$. All further details and parameter can be found in section B.2.

The simulation results of the average weights $\tilde{w}_1 = \frac{1}{25}\sum_{j\in G_1} w_j^t$ of G_1 and $\tilde{w}_\gamma = \frac{1}{75}\sum_{j\notin G_1} w_j^t$ of the remaining groups G_2, G_3, G_4 are presented in Figure 4B. In agreement with the learning goal, only the weights from G_1 are potentiated, while the other weights decay to zero due to the regularization term in the objective function L defined in equation 2.3. In Figure 4C the spiking probability $g^t = p(y^t = 1|X^{-\infty,t})$ and the estimation $F^t \approx p(y^t = 1|R)$ are plotted for 500 time steps after learning of $\boldsymbol{w}$ and $\boldsymbol{q}$. Although the rates of G_1 are related to the relevance signal R by a second-order Volterra series defined in equation 3.2, which involves temporal delays of 10 and 50 time steps, the estimation F^t is sufficiently accurate for learning an approximate IB optimal coding. Hence, this task is an example where the preprocessing of the relevance signal R via a generic untrained recurrent network with a trainable readout enables the neuron to extract statistical dependencies between $X^{-\infty}$ and R and to solve the given IB task.

4 Application: Predictive Coding

In Bialek et al. (2006), the H1 neuron of the blowfly, an extensively studied cell of the fly sensory-motor control system, is proposed as a possible example for a biological system providing an IB optimal coding. It is hypothesized that the output of this neuron is maximally informative about future external stimuli; hence this coding paradigm is termed *predictive coding*. In the following section, we show how such a predictive coding scheme can be learned by a neuron using a variant of the presented IB learning rule, equation 3.1.

It is has often been hypothesized that biological agents maintain an internal representation, denoted here as X_{int}, of the external world that is obtained and updated via previous sensory stimuli X_{past} (as sensing is a causal process that takes time). The representation X_{int} allows the agent to adapt its behavior to the state of the environment and plan future actions. The hypothesis that this representation X_{int} is optimal in some information-theoretic sense (for a given amount of invested resources L_{reg}) has drawn much attention and served as a guideline for many intriguing studies (see the references in Bialek et al., 2006). Bialek et al. (2006), however, argue that not all sensory information contained in X_{past} is equally important for the behavior and the survival capabilities of the agent, and hence the entire sensory information should not be represented internally in X_{int}. More precisely, it is hypothesized that only such external stimuli are worth being represented that are informative about the future sensory stimuli X_{future}, which encodes the future state of the environment. Only this information can be used by the agent to plan behavior and eventually improve its fitness. This predictive coding paradigm was formalized as an IB optimization problem. The mapping from the past sensory stimuli X_{past} to the representation X_{int} should be chosen such that it maximizes the predictive information $I(X_{\text{int}}, X_{\text{future}})$ about the future sensory stimuli X_{future} at fixed costs L_{reg}. Following this train of thought, the agent should hence maximize the following IB objective function $L_{\text{predictive}}$:

$$L_{\text{predictive}} = I(X_{\text{int}}, X_{\text{future}}) - \gamma L_{\text{reg}}.$$

Bialek et al. (2006) also discuss a concrete example of this predictive coding paradigm: the H1 neuron of the blowfly. This neuron is part of the optomotor control loop, and it is known to approximately code logarithmically for the horizontal angular velocity of the fly. Bialek et al. (2006) argue that this specific coding of the H1 neuron of the external stimuli could be optimal with regard to the objective function $L_{\text{predictive}}$.

Here we show that a single neuron can learn to extract predictive information from its inputs and establish a predictive coding scheme, similar to the one described in Bialek et al. (2006), using a slightly modified version the IB learning rule, equation 3.1. At any time step t, we identify the sensory input X_{past} with the input history $X^{-\infty,t}$ of the bottleneck neuron and identify the internal representation X_{int} with its output y^t. Furthermore, we define the relevance signal X_{future} as the future input $X^{t,t+\delta}$ to the neuron in the time interval $[t, t + \delta]$, which extends δ time steps into the future for a given parameter $\delta \in \mathbb{N}$. For simplicity, we also assume in this section that the activation function $g = \sigma$ of the bottleneck neuron is the logistic function.[7]

One possible approach to learn a predictive code is the following. If we assume that the synaptic kernel ϵ (see equation 2.1) is nonanticipating and that its support is shorter than δ time steps, the future activation $g^{t+\delta}$ is exclusively a function of the future input $X^{t,t+\delta}$. Hence, $g^{t+\delta}$ can be interpreted as a preprocessing $\boldsymbol{h}^t$ of the relevance signal $X^{t,t+\delta}$. Based on this observation, we make the ansatz $F^t = g^{t+\delta}$ for the estimator F^t; that is, we hypothesize that the neuron uses its own future spiking probability $g^{t+\delta}$ to estimate the amount of predictive (i.e., relevant) information contained in its input at time step t. The objective function L to maximize resulting from this approach reads:

$$L = \left\langle \log \left(\frac{(g^{\delta})^{y^0}(1 - g^{\delta})^{y^0}}{p(y^0)} \right) \right\rangle - \frac{\gamma}{2}\boldsymbol{w}^2. \tag{4.1}$$

Due to the ansatz $F^t = g^{t+\delta}$, the objective function 4.1, and consequently its gradient at time step t, contains the spiking probabilities g^t and $g^{t+\delta}$. Performing a straightforward stochastic gradient ascent (analogous to the procedure that leads to rule 3.1) would result in an anticipating learning rule; the weight update $\Delta\boldsymbol{w}^t$ would involve the future spiking probability $g^{t+\delta}$. This can be circumvented by shifting the time step index on the right-hand side of the learning rule by $-\delta$, which is allowed in stochastic gradient ascent as this does not change the expected value of the learning rule. This leads to the following update equation for the weights:

$$\eta_w^{-1}\Delta\boldsymbol{w}^{t+1} = g'^{t-\delta}\boldsymbol{v}^{t-\delta}\Big(\sigma^{-1}(g^t) - \sigma^{-1}(\langle g^{t-\delta}\rangle)\Big) - \gamma\boldsymbol{w}^t + \boldsymbol{v}^t(g^{t-\delta} - g^t). \tag{4.2}$$

The above learning rule is nonanticipating, but it is still not local in time as it contains the terms $g^{t-\delta}$, $g'^{t-\delta}$, and $\boldsymbol{v}^{t-\delta}$. Therefore, the values of the activity g^t as well as the synaptic activity $\boldsymbol{v}^t$ have to be buffered by the neuron for δ time steps in order to learn a predictive code with rule 4.2. Although an exact implementation of this buffering seems rather implausible, approximate implementations of the predictive coding learning rule might be biologically achievable. Assuming that the time parameter δ is not exactly defined but is rather given by a more diffuse parameter range,

[7] Similar learning rules can also be derived for more general activation functions, but they turn out to be slightly more complex.

running averages of g and ν with appropriate window sizes might prove to be sufficiently informative in order to learn an approximate predictive code. These averages could possibly be encoded in the signaling cascades that are triggered by pre- and postsynaptic spike events.

The structure of learning rule 4.2 is similar to the one of the general IB rule, equation 3.1. The first term, which is proportional to $\boldsymbol{\nu}^{t-\delta}\sigma^{-1}(g^t)$, potentiates those synapses whose input at time $t-\delta$ is correlated with the output rate g^t at time step t, that is, those synapses are potentiated whose inputs are predictive for the future neural output. The next term, which is proportional to $\sigma^{-1}(\langle g^{t-\delta}\rangle)$, remains unchanged from the original rule, equation 3.1. The last term $\boldsymbol{\nu}^t(g^{t-\delta}-g^t)$ stems from the fact that the estimator $F^t := g^{t+\delta}$ depends now on $\boldsymbol{w}$ itself. This term replaces the learning rule for $\boldsymbol{q}$ form, equation 3.1 (second line), and it drives the weights such that the past activity $g^{t-\delta}$ can be well estimated by the present activity g^t.

We want to point out that the simple choice for the estimator $F^t = g^{t+\delta}$ made above limits the power of rule 4.2 for learning a predictive code. Only those weights w_j are potentiated whose input X_j is positively correlated at time t with the output at time $t+\delta$. Negative correlations or higher-order statistical dependencies cannot be extracted with this choice of the estimator F. In order to achieve this, a more sophisticated ansatz for F with a more diverse preprocessing of $X^{t,t+\delta}$ would be required (e.g., the ansatz proposed in section 3.3).

We illustrate the behavior of learning rule 4.2 by a simple numerical example for a delay parameter $\delta = 25$. Consider the following setup where the synapses are again divided into four groups $G_1, \ldots, G_4$. Synapses from subgroup G_1 receive spike trains with a rate that is determined by a (discrete time) Ornstein-Uhlenbeck[8] (OU) process with a time constant $\tau_1 = 50$, mean $\mu_1 = 0.2$ and standard deviation (SD) $\sigma_1 = 0.3 \cdot \mu_1$. The inputs for group G_2 are generated in a similar way, however, with a time constant $\tau_2 = 25$ and $\sigma_2 = 0.5 \cdot \mu_1$. A (discrete time) telegraph process with mean μ_1 and SD σ_1 and a time constant $\tau_3 = 20$ determine the rate for the spike train of group G_3. Spike trains of G_4 are generated with a constant rate μ_1. Additional parameters and details can be found in section B.3. The results of the simulation are plotted in Figure 5. As expected the weights of G_4 rapidly decay as they transmit no relevant information. Further, due to the long autocorrelation time constant τ_1, the weights of G_1 are exclusively potentiated, while the weights for G_2 and G_3 decay. If, however, the values of the time constants τ_1 and τ_3 are switched ($\tau_1 = 20$, $\tau_3 = 50$), the results are reversed: weights of G_3 grow over time, while those of G_1 decay (results not shown). This example illustrates that the specialized version (see equation 4.2) of the IB rule (see equation 3.1) enables the neuron to extract predictive information from its input in simple setups.

[8] More precisely the firing rate $p(x_i(t)=1)$ is defined as $p(x_i(t)=1) = \min\{1, \max\{0, \mathrm{O}(t)\}\}$ in terms of the OU process $\mathrm{O}(t)$.

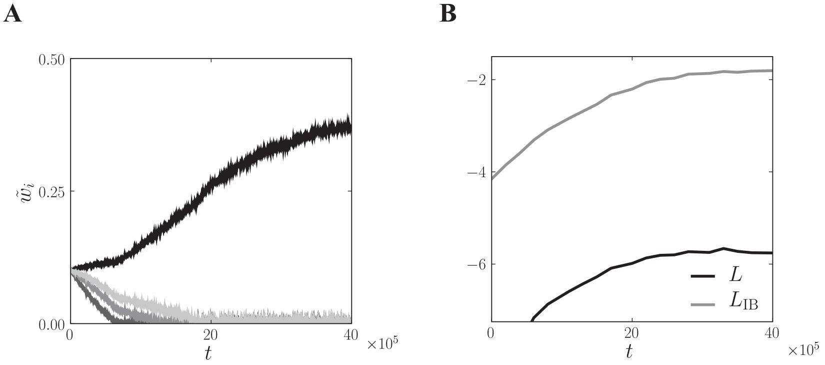

Figure 5: Numerical results for the predictive coding application described in section 4. (A) Shown are the average group weights $\tilde{w}_i$ for G_1 (top curve in black), G_2 (dark gray), G_3 (gray), and G_4 (light gray). Group G_1 transmits the largest amount of predictive information due to the long autocorrelation time constant τ_1 of its input, and hence the average $\tilde{w}_1$ is exclusively increased, whereas the remaining weights decay to zero. (B) The lower bound L (black) is plotted as a function of the time step t. Also shown is an estimation of the "true" IB objective function $L_{\text{predictive}}$ (gray), for which the mutual information $I(y^t, X^{t,t+\delta})$ was approximated by $\sum_{j=1}^{N} I(y^t, X_j^{t,t+\delta})$ (causing the large offset between L and $L_{\text{predictive}}$). The trajectories indicate that the neural output becomes more predictive for the future input $X^{t,t+\delta}$.

5 Discussion

5.1 Relation to Existing Work. Here we briefly discuss existing work that is related to the IB learning rules proposed in this contribution. Additionally, in the first paragraph, the differences between the approach presented in this article and other IB algorithms are described.

5.1.1 Other IB Algorithms. Most IB algorithms determine nonparametric mappings from the input to the output RVs (Tishby, Pereira, & Bialek, 2000; Slonim & Tishby, 1999). Hence, the approach presented here, which determines an IB optimal mapping via gradient ascent with regard to the model parameters $\boldsymbol{w}$, $\boldsymbol{q}$ might be regarded to be against the spirit of the IB framework. However, we argue that in the neural setup considered here, where the global structure of the IB mapping is fixed (e.g., the dimensionality of the output and the class of transformations that can be used), such a parametric approach is nevertheless justified. Another difference is that most IB algorithms operate in batch mode on the complete input data (with the notable exceptions of predictive coding described below), whereas the setup we propose maps input sequences onto an output sequence online. This approach reflects the fact that neurons naturally operate in the temporal domain, which also requires an online algorithm for learning the IB optimal mapping. Furthermore, most IB algorithms assume that the joint distribution $p(X, R)$ of the input RV X and the relevance RV R is known; hence, they require a beforehand estimation of $p(X, R)$ based on finite samples (for an in-depth analysis of this procedure, see Ohad Shamir, Sabato, & Tishby, 2008). In contrast to this procedure, our approach (based on the objective function 2.3) directly unifies this estimation process and learning of the IB mapping. This unification, however, comes at the expense of not optimizing the "true" IB objective function L_{IB} but a lower bound L of the latter.

The work presented here directly builds on Klampfl et al. (2009). There, the authors derived an online IB learning rule by gradient ascent for a quite sophisticated stochastic neuron model, assuming that the relevance signal sequence is given by a spike train of the same neuron model. The gradient of the relevant information (the mutual information between the output Y and the relevance signal R) was estimated by measuring the correlation $\langle y^t R^t \rangle$, where $R^t \in \{0, 1\}$ is the relevance spike train at time step t. The resulting learning rule is sensitive only to instantaneous correlations between the neural output and the relevance signal, a fact that limits the applicability of the learning rule. In contrast, we propose in this study the more general approach of a parametric estimation of the gradient of $I(y^0, R)$ based on a given preprocessing $\boldsymbol{h}^t = (h^t_1, \ldots, h^t_r)$. The resulting rule, equation 3.1, is as powerful as the preprocessing that is available to the neuron. Given that neurons are strongly interconnected and receive many recurrent inputs resulting in highly nontrivial transformations of the external input, it seems reasonable to assume that the preprocessing of the relevance signal, which is potentially available to the bottleneck neuron, is diverse and rich enough to carry out a large class of IB optimization tasks. Further, a considerable simplification of learning rule 3.1 compared to the one presented in Klampfl et al. (2009) was achieved by choosing to maximize the mutual information $I(y^0, R)$ instead of the more complex quantity $I(Y, R)$, where $Y = (\ldots, y^{-1}, y^0, y^1, \ldots)$. [9]

5.1.2 InfoMax and Imax. The learning goal of maximizing the mutual information between the input and output of individual neurons or neural networks, so-called InfoMax learning, has served as a fruitful theoretical principle for learning with artificial and more biologically realistic neural models (see, e.g., Linsker, 1989; Bell & Sejnowski, 1995; Chechik, 2003; Toyoizumi et al., 2005; Parra et al., 2009). More precisely InfoMax is defined as learning a neural mapping $X \rightarrow Y$ of some input X to the output Y, which maximizes the amount of transmitted information defined

[9] This simplification can apparently be made without reducing the power of the learning rule, as all numerical IB tasks presented by Klampfl et al. (2009) can also be solved by rule 3.1 even when assuming only a simple preprocessing (data not shown).

as the mutual information $I(X, Y)$ (with some regularization constraints). While InfoMax shares a common theoretical foundation with the IB method, namely, information theory, there are differences with regard to the specific learning goals and their biological interpretation. InfoMax is an unsupervised learning principle; its formulation involves only the input X and the output Y. There is no external guideline of how the input X is to be transformed into the output Y except for maximizing the scalar mutual information $I(X, Y)$. InfoMax can be interpreted as a possible approach to dimensionality-reduction techniques, to clustering as well as to blind source separation (Bell & Sejnowski, 1995). The IB method also aims at constructing a mapping $X \rightarrow Y$ of the input X to the output Y, which exhibits certain information-theoretic properties. In contrast to InfoMax, however, the IB method is not an unsupervised learning framework. In the IB framework, it is assumed that the environment offers information about what can be considered relevant in the input via the given relevance signal R. It has to be emphasized that IB mapping, once learned, maps the input to the output $X \rightarrow Y$; it is not a mapping from the input and the relevance signal to the neural output $X \times R \rightarrow Y$. Thus, relevant information given by R that is not present in the input X will not be encoded in the output Y. Further, it should be noted that the relevance signal R has to be present only during learning of the IB mapping. After this learning phase, the IB optimal mapping $X \rightarrow Y$ can be carried out by the neural architecture without the presence of the relevance signal R. In a biologically plausible setting, the distinction between learning and operation phase could, for example, be implemented with a learning rate $\eta(\|R^t\|)$ that detects the presence of the relevance signal by monitoring some measure of its intensity $\|R^t\|$ over time and stops learning if the relevance signal is absent.

To further illustrate the difference between IB and InfoMax, consider the setup of the simulation presented in section 3.2. The input to the IB neuron consists of three groups: G_1, G_2, and G_3. The results presented in section 3.2 show that if the relevance signal R is statistically dependent on the input G_1 and G_2 (i.e., G_1, G_2 convey relevant information), the corresponding weights are potentiated (see Figure 3A). If, however, the synapses are updated with an InfoMax learning rule (for details, see section A.2), only the weights of group G_3 are potentiated, while all other weights decay (see Figure 3D). This weight configuration maximizes the transmitted mutual information $I(X, Y)$ since group G_3 subsumes the most afferents and its inputs exhibit the strongest spike-spike correlations. This is an example where the learning results of IB and InfoMax differ.

Becker (1996) and Becker and Hinton (1992) propose a learning principle called Imax that is similar to IB learning and can be interpreted as a special case of the latter. The objective of Imax is to maximize the mutual information between the outputs of two (or more) networks that receive disjoint but statistically related inputs. It is therefore different from InfoMax, which aims at maximizing the mutual information between input and output.

More precisely, in Becker (1996), two (multilayer) feedforward networks are considered, whose two inputs X_1, X_2 are given by neighboring patches of visual input. The learning objective was defined as maximizing the mutual information $I(\hat{X}_1, \hat{X}_2)$ between the activations $\hat{X}_1$, $\hat{X}_2$ of the output layers of the networks. Partial derivatives of $I(\hat{X}_1, \hat{X}_2)$ are propagated back into the hidden layers of the network to maximize $I(\hat{X}_1, \hat{X}_2)$. In contrast to the work presented in this article, the architecture of Becker (1996) and Becker and Hinton (1992) does not operate in the temporal domain; it implements an instantaneous mapping from input to output (which simplifies drastically the evaluation of the objective function $I(\hat{X}_1, \hat{X}_2)$). An interesting topic for future research is to port the architecture proposed in Becker (1996) and Becker and Hinton (1992) to neurons operating in time by using the learning rule presented here. This can possibly be achieved by adopting ideas from the symmetric IB setup presented in Friedman, Mosenzon, Slonim, and Tishby (2001), where two disjoint input streams are mapped to simpler representations while preserving as much mutual information between each other.

5.1.3 Predictive Coding. Predictive coding, which was formalized in the IB framework in Bialek et al. (2006), has been studied for linear mappings and gaussian noise in Creutzig and Sprekeler (2008), revealing an intriguing relation to slow feature analysis (for an introduction, see Wiskott & Sejnowski, 2002). The solutions to this past-future bottleneck are explicitly given. Furthermore, the analysis of predictive coding was expanded to linear dynamical systems in Weiss (2007), also resulting in a complete characterization of the IB optimal systems assuming a linear dependence of the input RV X on the relevance RV R with additive gaussian noise. These studies provide strong, exact results to the considered restricted setups. The spirit of the approach presented here is quite different. We provide an iterative scheme for IB optimization, which is possibly prone to local minima, focusing on a neural mapping while making only a few assumptions about the input and relevance processes X and R.

Predictive coding as a learning goal for sensory processing with neural architectures was motivated in Bialek et al. (2006) by arguing that this paradigm allows learning a "useful" (with regard to the agent's fitness) internal representation or model of the environment. It can therefore be considered to be closely related to learning a generative model of the environment (see Slonim & Weiss, 2003, for a relation between IB and generative models). However it can be argued that learning a sufficiently accurate model of the environment may consume too many resources and may require too many data to be a suitable strategy for adapting to the environment. An alternative would be a discriminative approach; one might hypothesize that it is more appropriate for an agent to directly learn a mapping from environmental configurations to behavioral decisions, without the need for an explicit representation of the environment. Which of these two approaches, generative versus discriminative, is the better theoretical model depends, among others, on the structure and amount of data that the agent learns from. In a machine learning context (Hinton, 2007) argues that a combination of generative learning, making use of unlabeled data, and discriminative learning is a powerful and promising approach. This indicates that such a combination of discriminative and generative approaches might also be a powerful model for sensory processing in biological agents.

5.2 A Possible Biological Implementing of IB Optimization with Spiking Neurons. The experimental investigation of synaptic plasticity has made significant advances in the past decade (for reviews, see Caporale & Dan, 2008; Sjöström, Rancz, Roth, & Häusser, 2008). The classical picture of synaptic plasticity, as postulated by Hebb (1949) and later experimentally described by others, which exclusively depends on the pre- and postsynaptic activity, had to be considerably expanded over the years due to accumulating experimental evidence. It is now known that many additional factors modulate synaptic weight changes e.g., neuromodulators (Hee et al., 2007), details of neural morphology (Sjöström et al., 2008), and extracellular subthreshold stimulation (Sjöström, Turrigiano, & Nelson, 2001), for example). This large body of experimental literature poses a huge challenge for theoretical work about the underlying functions of these mechanisms for neural information processing. The fact that synaptic plasticity is determined by additional quantities besides pre- and postsynaptic activity might be a suitable mechanism allowing synaptic weight changes to fulfill complex optimization goals like IB optimization. In the following, we show that the proposed IB learning rule 3.1 fits well with recent experimental results concerning the influence of dendritic depolarization on synaptic plasticity.

The plasticity of synapse j described by rule 3.1 depends on the synaptic activity ν_j^t, in agreement with experimental findings. Further, a measure of the average postsynaptic activity $\langle g^t \rangle$ influences the weight change. This may be interpreted in a biological context as a homeostatic control of the weights. The central claim of learning rule 3.1 is, however, that a "third" factor $\sigma^{-1}(F^t)$, which quantifies the influence of the relevance signal, shapes synaptic plasticity. This contribution crucially determines the sign and amplitude of the weight change in this plasticity model. A biological mechanism termed dendritic switch, which has recently been uncovered by Sjöström and Häusser (2006), is a plausible candidate for a third factor modulating plasticity as required by IB learning. It is known that plasticity of dendritic synapses depends on the backpropagating action potential (bAP) in the dendrite (Caporale & Dan, 2008). Further, it has been shown that the bAP amplitude and the reliability of the bAP are shaped by the active and passive conductance properties of the dendrite (see Stuart & Häusser, 2001). These conductance properties can in turn be considerably modulated by local de- or hyperpolarization of the dendrite as demonstrated in Stuart and Häusser (2001). Hence it can be assumed that properly timed EPSPs

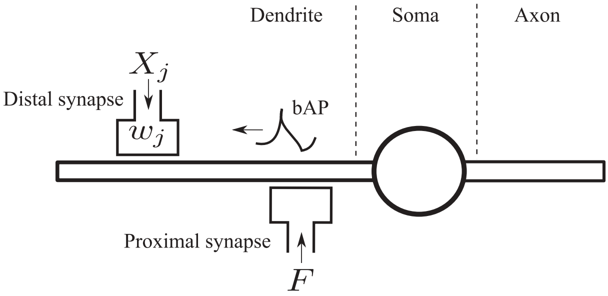

Figure 6: A possible biological mechanism implementing IB optimization in a single neuron. As shown in experiments, the activity of proximal synapses can act as "dendritic switches" that critically influence the amplitude and the sign of weight changes of distal synapses. The IB learning rule could be implemented in the distal synapses assuming that the proximal synapses convey the influence of the relevance signal R via F^t.

and IPSPs in the dendrite shape synaptic plasticity by influencing the bAP amplitude. According to Sjöström and Häusser (2006), these mechanisms indeed enable proximal synapses to act as "dendritic switches," which modulate the weight changes at distal synapses by boosting or shunting the bAP. These "dendritic switches" were shown to be able to change the amplitude as well as the sign of the weight change at distal synapses by modulating the bAP with EPSPs and IPSPs that de- or hyperpolarize the proximal part of the dendrite. In the IB model, the proximal synapses, the dendritic switches, would convey the influence of the relevance signal $\sigma^{-1}(F^t)$ (see Figure 6). The weights $\boldsymbol{w}$, which obey rule 3.1, would correspond to more distal synapses whose plasticity is controlled by the relevance signal. With this correspondence, the IB plasticity model predicts a boosting (shunting) of bAPs leading to potentiation (depression) at active distal synapses (those with $\boldsymbol{\nu}^t > 0$) whenever the weighted, preprocessed relevance signal $\sigma^{-1}(F^t)$ (representing the input at the proximal synapses) is high (low). As the dendritic switches act on a millisecond timescale, this mechanism would provide a sufficiently high temporal resolution for the relevance signal in contrast to other factors modulating plasticity (e.g., neuromodulators).

In spite of this possible correspondence with the experimental data discussed above, we wish to point out the limitations of IB learning rule 3.1 as a theoretical model for experimental findings. It has to be noted that the IB learning rule presented here cannot account, for example, for the effect of spike-timing-dependent plasticity (STDP) as reported in Bi and Poo (1998), for example. The weight change given by IB rule 3.1 does not exhibit a dependence on the postsynaptic spike times, only on a long-term average of the postsynaptic firing rate $\langle g \rangle$. This is in contrast to the experimental results on STDP, which report a strong dependence of plasticity on the precise timing of pre- and postsynaptic spikes. Furthermore, numerous more subtle aspects of plasticity, such as postsynaptic voltage dependence and weight dependence of plasticity, are not reflected by the IB rule. A topic of current research is whether the IB approach in conjunction with more realistic neuron models or other constraints can reproduce experimental data more faithfully.

5.3 Summary. In this article, we presented an online learning rule for IB optimization with a simple, idealized spiking neuron model. The neuron was regarded as an information bottleneck that maps its high-dimensional input sequence on a one-dimensional output sequence of spikes. With the help of the proposed IB learning rule, the neuron can adapt its weights such that its output contains the maximal amount of relevant information, that is, its output is maximally informative (possibly locally optimal) about a relevance signal also given by a sequence of RVs in time. This learning rule was derived assuming that the neuron has access to an estimation of its currently transmitted amount of relevant information (more precisely, the gradient), which is based on a given preprocessing of the relevance signal and an adaptable set of parameters that are learned simultaneously with the weights. This approach extends previous studies on neural IB optimization (Klampfl, Legenstein, & Maass, 2007; Klampfl et al., 2009) that were based on correlations between the output of the bottleneck neuron and the relevance signal.

We also addressed the question of how a suitable and sufficiently general preprocessing of the relevance signal may be implemented in a biological neural system. Motivated by previous theoretical, numerical, and experimental studies (see Buonomano & Maass, 2009), we argued that a generic recurrent neural circuit, which is not learned for a certain IB task and hence looks randomly structured from the perspective of the bottleneck neuron, can be considered a plausible candidate for such an implementation of the preprocessing. Simple models of recurrent neural networks were shown in previous studies to provide a considerable amount of memory and nonlinearity and, hence, render themselves to be a suitable preprocessing of the relevance signal enabling the bottleneck neuron to carry out a class of IB tasks.

Further, we have discussed a biological mechanism that can in principle resolve the problem of spatial nonlocality encountered in previous IB learning rules for spiking neurons—the problem of how the current values of quantities that are essential to the learning rule can be made available at the location of the synapse. The recently discovered mechanism of dendritic switches (Sjöström & Häusser, 2006) seems well suited as an implementation of a learning process that is modulated by an external "third" factor (the relevance signal in in addition to the two factors given by pre- and postsynaptic signals) as required by IB learning.

Predictive coding has been proposed as an unsupervised learning goal for single neurons in Bialek et al. (2006), a hypothesis that seems to be in agreement with experimental findings. As an application of IB learning with spiking neurons, we have shown that a variant of the proposed IB learning rule enables the neuron to learn a predictive code assuming simple input statistics. It has to be emphasized that several neural learning rules exist that extract temporal regularities from the input. In a recent study (Creutzig & Sprekeler, 2008), a close relation between IB optimization and learning of temporal invariances in a more machine-learning-oriented setting was pointed out. The approach presented here shows that also on a single neural level, the well-known learning goals related to temporal invariance can be motivated and a viable learning rule can be derived from the IB framework.

The proposed learning rules are based on idealized assumptions especially with regard to the neuron model. The applied neuron model neglects several characteristics observed in experiments, most prominently a refractory mechanism, complex voltage dynamics (e.g., bursting, rebound spikes), and spatial extension and morphology of a neuron. We argue that studying learning in a highly simplified system is nevertheless sensible, as it possibly provides a baseline architecture (a possible learning strategy and its essential functional building blocks) that is not (we hope) cluttered by unimportant contingent details of the neural dynamics.

Appendix A: Derivation of the Learning Rules

A.1 IB Learning Rule. Here we calculate the gradient of the objective function L with regard to $\boldsymbol{w}$ and $\boldsymbol{q}$. The following relations are useful:

$$\begin{aligned}\left\langle \log F(y^0, R)\right\rangle &= \left\langle y^0 \log(F^0) + (1-y^0)\log(1-F^0)\right\rangle \\ &= \left\langle g^0 \log(F^0) + (1-g^0)\log(1-F^0)\right\rangle \\ \left\langle \log p(y^0)\right\rangle &= \left\langle g^0\right\rangle \log\left\langle g^0\right\rangle + (1-\left\langle g^0\right\rangle)\log(1-\left\langle g^0\right\rangle).\end{aligned}$$

When these identities are used, the objective function L can be written as

$$\begin{aligned}L = \Big\langle & g\log F + (1-g)\log(1-F) - g\log\langle g\rangle \\ & - (1-g)\log(1-\langle g\rangle)\Big\rangle - \frac{\gamma}{2}\boldsymbol{w}^2.\end{aligned}$$

For simplicity the time step index was left out as no confusion can occur (e.g., $F = F^0$). From this form of L, the gradient with regard to $\boldsymbol{w}$ is straightforward to calculate:

$$\begin{aligned}\frac{\partial L}{\partial \boldsymbol{w}} &= \left\langle \left(\log\left(\frac{F}{1-F}\right) - \log\left(\frac{\langle g\rangle}{1-\langle g\rangle}\right)\right)\frac{\partial g}{\partial \boldsymbol{w}}\right\rangle - \gamma\boldsymbol{w} \\ &= \left\langle (\sigma^{-1}(F^t) - \sigma^{-1}(\langle g\rangle))\frac{\partial g}{\partial \boldsymbol{w}}\right\rangle - \gamma\boldsymbol{w}.\end{aligned}$$

Here we used the fact that the expected value $\langle\cdot\rangle$ in the above equation is taken only over the joint distribution $p(X^{-\infty}, R)$, which is independent of $\boldsymbol{w}$ (and $\boldsymbol{q}$), and hence the gradient $\frac{\partial}{\partial \boldsymbol{w}}$ (and $\frac{\partial}{\partial \boldsymbol{q}}$) commutes with the average operator $\langle\cdot\rangle$. We notice that $\log(x/(1-x))$ is the inverse function of the logistic function $\sigma(x) = 1/(1+\exp(-x))$. Further, the gradient of g yields $\partial g/\partial \boldsymbol{w} = g'\boldsymbol{v}$, where g' is the derivative of g. This results in the $\boldsymbol{w}$-part of learning rule equation 3.1.

The gradient of L with regard to $\boldsymbol{q}$ is even simpler, as only F depends on $\boldsymbol{q}$. Hence, only the following term has to be calculated:

$$\frac{\partial}{\partial \boldsymbol{q}} \langle g \log F + (1-g)\log(1-F)\rangle = \left\langle \frac{F'}{F(1-F)} \boldsymbol{h}(g-F) \right\rangle .$$

Using the relation $\sigma' = \sigma(1-\sigma)$, which holds for the logistic function σ, yields the final learning rule for the parameters $\boldsymbol{q}$.

A.2 An InfoMax Learning Rule. In close analogy to the derivation of the IB learning rule presented above, one can derive an InfoMax learning rule starting from the objective function L_{InfoMax}:

$$\begin{aligned} L_{\text{InfoMax}} &= I(y^0, X^{-\infty}) - \frac{\gamma}{2}\boldsymbol{w}^2 \\ &= \langle g \log g + (1-g)\log(1-g) - g\log\langle g\rangle \\ &\quad - (1-g)\log(1-\langle g\rangle)\rangle - \frac{\gamma}{2}\boldsymbol{w}^2. \end{aligned}$$

The yields the following InfoMax learning rule:

$$\Delta \boldsymbol{w} = \eta_w g' \boldsymbol{v}(\sigma^{-1}(g) - \sigma^{-1}(\langle g\rangle)) - \eta_w \gamma \boldsymbol{w}.$$

Appendix B: Details of the Numerical Examples

B.1 Example of Section 3.2. Weights $\boldsymbol{w}$ were initialized with 0.15 and the parameters $\boldsymbol{q}$ with 0 and $\hat{g}$ with 0.02. The learning rates were set to $\eta_w = 0.075$, $\eta_{q_1} = 4.25 \cdot 10^{-4}$, $\eta_{q_2} = 4.25 \cdot 10^{-3}$, and $\eta_g = 2 \cdot 10^{-3}$. Correlated spike trains are generated using techniques described in Gütig, Aharonov, Rotter, and Sompolinsky (2003). The values of L and L_{IB} shown in Figure 3C were estimated with the `pyentropy` software package described in Ince, Petersen, Swan, and Panzeri (2009) using sophisticated bias-correcting methods. Every point is an average of 50 independent trials, each estimated from sequences Y and R of length $5 \cdot 10^5$ with frozen $\boldsymbol{w}$ and $\boldsymbol{q}$.

B.2 Example of Section 3.3. Weights $\boldsymbol{w}$ were initialized with 0.05, $\hat{g}$ with 0.02 and the initial values of the components of $\boldsymbol{q}$ were set to 0. The learning rates were set to $\eta_w = 2 \cdot 10^{-3}$, $\eta_q = 10^{-3}$, $\eta_g = 2.5 \cdot 10^{-3}$, and the parameters $a,\ b$ were set to $a = 1/2,\ b = 1/8$. The trade-off parameter was set to $\gamma = 6 \cdot 10^{-5}$. For this example, a recurrent network of $r = 200$ sigmoidal rate neurons was used (as a LSM). The state vector $\boldsymbol{s}^t = (s_1^t, \ldots, s_r^t) \in \mathbb{R}^r$ obeys the equation

$$\boldsymbol{s}^{t+1} = \boldsymbol{s}^t \cdot (1 - \alpha) + \beta f \left(W_s \boldsymbol{s}^t + W_{in}(R^t - 0.5) * 2 \right),$$

with the parameters $\alpha = 0.4$ and $\beta = 0.44$. The activation function $f : \mathbb{R}^r \rightarrow \mathbb{R}^r$ is given by applying the hyperbolic tangent component-wise. The elements of the recurrent weight matrix $W_s \in \mathbb{R}^{r^2}$ are generated in the following way. The probability of two neurons to be connected was set to 1/2; the weight for a connected pair was drawn from a normal distribution $\mathcal{N}(0, 1)$. Finally W_s was rescaled by a scalar such that its spectral radius was equal to 0.8. The elements of $W_{in} = ((W_{in})_i, \ldots, (W_{in})_r) \in \mathbb{R}^r$ were drawn i.i.d. from $\{0, 1\}$ with $p((W_{in})_i = 1) = 0.3$. The filter bank h was then chosen to equal the state vector, $\boldsymbol{h}^t = \boldsymbol{s}^t$.

B.3 Details of the Predictive Coding Application. Weights $\boldsymbol{w}$ were initialized with 0.1 and $\hat{g}$, as well as all elements of the history of g and g' with 0.01. The learning rates were set to $\eta_w = 2 \cdot 10^{-4}$ and $\eta_g = 2.5 \cdot 10^{-4}$. The trade-off parameter was set to $\gamma = 10^{-3}$. Furthermore, the values of L and $L_{\text{predictive}}$ were evaluated using the `python` module `pyentropy` based on spike trains $X,\ Y$ of length $5 \cdot 10^5$ with frozen weights $\boldsymbol{w}$. For $L_{\text{predictive}}$, the term $I(y^0, X^{0,\delta})$ was approximated by $I(y^0, X^{0,\delta}) \approx \sum_{j=1}^{100} I(y^0, X_j^{0,\delta})$ to avoid the undersampling problem occurring in the evaluation of the mutual information for high-dimensional variables. This approximation introduced a large error; however, the results still give intuition of the evolution of the "true" IB objective function $L_{\text{predictive}}$.

Acknowledgments

This article was written under partial support by project FP6-015879 (FACETS), project FP7-216593 (SECO), project FP7-506778 (PASCAL2), and project FP7-231267 (ORGANIC) of the European Union.

References

Becker, S. (1996). Mutual information maximization: Models of cortical self-organization. *Network: Computation in Neural Systems, 7*(1), 7–31.

Becker, S., & Hinton, G. E. (1992). Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature, 355*, 161–163.

Bell, A. J., & Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation, 7*(6), 1129–1159.

Bi, G., & Poo, M. (1998). Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neuroscience, 18*(24), 10464–10472.

Bialek, W., de Ruyter van Steveninck, R. R., & Tishby, N. (2006). Efficient representation as a design principle for neural coding and computation. In *IEEE International Symposium on Information Theory*. Piscataway, NJ: IEEE Press.

Boyd, S., & Chua, L. O. (1985). Fading memory and the problem of approximating nonlinear oparators with Volterra series. *IEEE Trans. on Circuits and Systems, 32*, 1150–1161.

Buonomano, D., & Maass, W. (2009). State-dependent computations: Spatiotemporal processing in cortical networks. *Nature Reviews in Neuroscience, 10*(2), 113–125.

Caporale, N., & Dan, Y. (2008). Spike timing-dependent plasticity: A Hebbian rule. *Annu. Rev. Neurosci., 31*, 25–46.

Chechik, G. (2003). Spike-timing-dependent plasticity and relevant mutual information maximization. *Neural Computation, 15*(7), 1481–1510.

Creutzig, F., & Sprekeler, H. (2008). Predictive coding and the slowness principle: An information-theoretic approach. *Neural Computation, 20*(4), 1026–1041.

Friedman, N., Mosenzon, O., Slonim, N., & Tishby, N. (2001). Multivariate information bottleneck. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence* (pp. 152–161). San Francisco: Morgan Kaufmann.

Gütig, R., Aharonov, R., Rotter, S., & Sompolinsky, H. (2003). Learning input correlations through non-linear temporally asymmetric Hebbian plasticity. *Journal Neurosci., 23*, 3697–3714.

Harremoes, P., & Tishby, N. (2007). The information bottleneck revisited or how to choose a good distortion measure. In *Proceedings of the IEEE International Symposium on Information Theory, 2007 (ISIT 2007)*, (pp. 566–570). Piscataway, NJ: IEEE Press.

Hebb, D. O. (1949). *The organization of behavior*. New York: Wiley.

Hee, S. G., Ziburkus, J., Huang, S., Song, L., Kim, I. T., Takamiya, K., et al. (2007). Neuromodulators control the polarity of spike-timing-dependent synaptic plasticity. *Neuron, 55*, 919–929.

Hinton, G. (2007). To recognize shapes, first learn to generate images. *Progress in Brain Research, 165*, 535–547.

Ince, R. A. A., Petersen, R. S., Swan, D. C., & Panzeri, S. (2009). Python for information theoretic analysis of neural data. *Frontiers in Neuroinformatics, 3*, 4.

Jäger, H. (2001). *The "echo state" approach to analyzing and training recurrent neural networks* (GMD Rep. 148). St. Augustin, Germany: German National Research Center for Information Technology.

Klampfl, S., Legenstein, R., & Maass, W. (2007). Information bottleneck optimization and independent component extraction with spiking neurons. In B. Schölkopf, J. Platt, & T. Hoffman (Eds.), *Advances in neural information processing systems, 19* (pp. 713–720). Cambridge, MA: MIT Press.

Klampfl, S., Legenstein, R., & Maass, W. (2009). Spiking neurons can learn to solve information bottleneck problems and to extract independent components. *Neural Computation, 21*(4), 911–959.

Linsker, R. (1989). How to generate ordered maps by maximizing the mutual information between input and output signals. *Neural Computation, 1*(3), 402–411.

Maass, W., Natschlaeger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation, 14*(11), 2531–2560.

Ohad Shamir, S., Sabato, S., & Tishby, N. (2008). Learning and generalization with the information bottleneck. In *International Symposium on AI and Mathematics (ISAIM)-2008*. Heidelberg: Springer-Verlag.

Parra, L., Beck, J., & Bell, A. (2009). On the maximization of information flow between spiking neurons. *Neural Computation, 21*, 1–19.

Sjöström, P. J., & Häusser, M. (2006). A cooperative switch determines the sign of synaptic plasticity in distal dendrites of neocortical pyramidal neurons. *Neuron, 51*, 227–238.

Sjöström, P. J., Rancz, E. A., Roth, A., & Häusser, M. (2008). Dendritic excitability and synaptic plasticity. *Physiol. Rev., 88*, 769–840.

Sjöström, P. J., Turrigiano, G. G., & Nelson, S. B. (2001). Rate, timing and cooperativity jointly deterimine cortical synaptic plasticity. *Neuron, 32*, 1149–1164.

Slonim, N., & Tishby, N. (1999). Agglomerative information bottleneck. In S. A. Solla, T. K. Leen, & K.-R. Muller (Eds.), *Advances in neural information processing systems, 12*. Cambridge, MA: MIT Press.

Slonim, N., & Tishby, N. (2001). The power of word clustering for text classification. In *Proceedings of the European Colloquium on IR Research, ECIR 2001*. Heidelberg: Springer-Verlag.

Slonim, N., & Weiss, Y. (2003). Maximum likelihood and the information bottleneck. In S. Thrün, L. Saul, & B. Schölkopf (Eds.), *Advances in neural information processing systems, 16* (pp. 351–358). Cambridge, MA: MIT Press.

Stuart, G. J., & Häusser, M. (2001). Dendritic coincidence detection of EPSPs and action potentials. *Nature, 4*(1), 63–71.

Tishby, N., Pereira, F. C., & Bialek, W. (1999). The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing* (pp. 368–377). Piscataway, NJ: IEEE Press.

Tishby, N., Pereira, C. F., & Bialek, W. (2000). *The information bottleneck method.* Available online at http://www.citebase.org/abstract?id=oai.arXiv.org.physics/0004057.

Toyoizumi, T., Pfister, J.-P., Aihara, K., & Gerstner, W. (2005). Generalized Bienenstock-Cooper-Munro rule for spiking neurons that maximizes information transmission. *Proc. Natl. Acad. Sci. USA, 102*, 5239–5244.

Weiss, R. (2007). *Predictive information as a criterion to linear dynamical systems reduction*. Unpublished master's thesis, Racah Institute of Physics, Hebrew University, Jerusalem, Israel.

Wiskott, L., & Sejnowski, T. J. (2002). Slow feature analysis: Unsupervised learning of invariances. *Neural Computation, 14*(4), 715–770.

Received August 11, 2009; accepted January 13, 2010.