Learning encoding and decoding filters for data representation with a spiking neuron

Learning encoding and decoding filters for data representation with a spiking neuron

Michael Gutmann Aapo Hyvärinen Kazuyuki Aihara

***Abstract*— Data representation methods related to ICA and sparse coding have successfully been used to model neural representation. However, they are highly abstract methods, and the neural encoding does not correspond to a detailed neuron model. This limits their power to provide deeper insight into the sensory systems on a cellular level. We propose here data representation where the encoding happens with a spiking neuron. The data representation problem is formulated as an optimization problem: Encode the input so that it can be decoded from the spike train, and optionally, so that energy consumption is minimized. The optimization leads to a learning rule for the encoder and decoder which features synergistic interaction: The decoder provides feedback affecting the plasticity of the encoder while the encoder provides optimal learning data for the decoder.**

I. Introduction

Learning a representation of the sensory input can be considered the fundamental functional task of a sensory neuron. We model in Figure 1a neural representation by means of a dynamical neural encoding system and a (hypothetical) decoder.

Mathematical data representation methods such as principal or independent component analysis (PCA or ICA) [1], [2] show the same encoding-decoding structure. This is illustrated in Figure 1b. The feedforward linear transform $\mathbf{y} = \mathbf{W}^\mathbf{T}\mathbf{x}$ models the neural encoding of the input $\mathbf{x}$ into the neural response $\mathbf{y}$. A further linear transform $\mathbf{Hy}$ implements the (hypothetical) decoder. These mathematical data representation methods lend themselves to the study of neural representation of natural stimuli. For the case of vision, the approach consists of taking samples of natural scenes for the input, learning the encoder-decoder pair, and comparing their properties with properties of the visual cortex. While PCA seems to be insufficient for learning relevant representations, ICA adds a sparseness constraint which has both computational (Bayesian as well as information-theoretic) and metabolic justifications. This approach led to important insight of how the receptive fields in the primary visual cortex could have been formed in order to represent natural stimuli [3], [4], [5]. Related approaches have been made earlier for the LGN or retina, see [6] for a review.

However, the linear encoding transformation in ICA corresponds to a rather abstract neuron model. This limits further investigations into neural representation on the cellular level. Furthermore, it makes it difficult to link theory to experiments on the single-cell level, see for example the study [7]. Also, the neural response $\mathbf{y}$ is usually assumed to correspond to the firing rate. There is however strong evidence that individual spike timings bear important information [8].

In this article, we propose data representation by means of a spiking neuron, for which a relatively detailed dynamical model is used. The encoding is done by firing single spikes, see Figure 1c. Data representation is based on the minimization of a squared reconstruction error, and optionally with an added penalty to minimize energy consumption during the encoding process. We derive an online learning rule based on minimization of the objective function. Learning of the encoder and decoder are synergistic: The encoder selects the learning data for the decoder by triggering spikes at the right time, while the decoder provides error feedback for the encoder affecting in that way its plasticity.

This paper is organized as follows. In section II, we formulate the data representation problem of Figure 1c as an optimization problem. In section III, we provide the solution in form of an online rule, and discuss the update rule. In section IV, we show simulation examples and contrast different ways to punish energy consumption. Section V discusses the relation to other work, and section VI concludes the paper.

II. Optimization problem

A. Encoding and decoding

The assumed neuron model, that is used for the encoding, is closely related to the SRM_0 model [9]. The equation for the membrane voltage u is

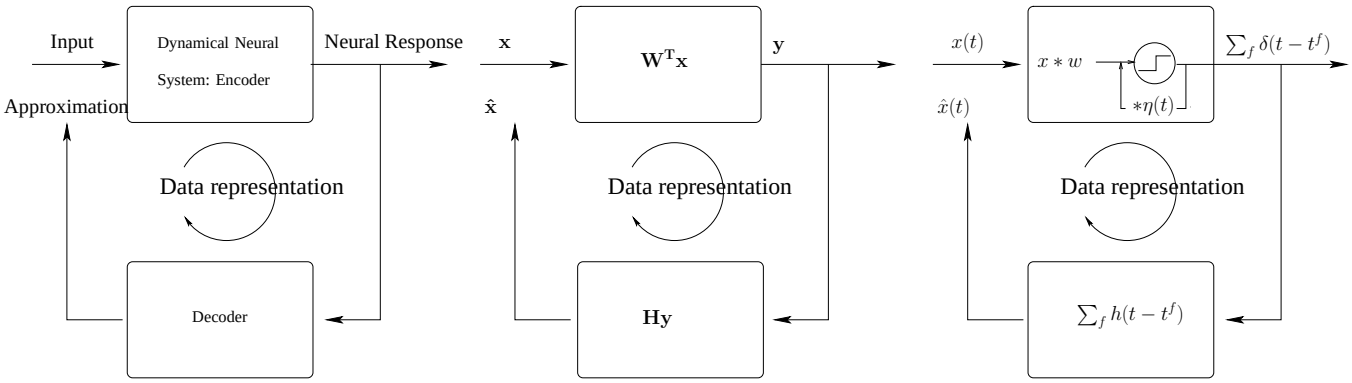
$$u(t) = \eta_0 \exp\left[-\frac{t-\hat{t}}{\tau}\right] + \int_0^{\min\{t,T_w\}} x(t-s)w(s)\mathrm{d}s + I_n(t), \quad (1)$$

where $I_n(t)$ is a sufficiently smooth, and optional, noise current, $\hat{t}$ the last spike time before time t, and w the unknown encoding filter, to be learned, of length T_w. The convolution of input x with w produces the input current I. Spike timings $\{t^f; f = 1, \ldots\}$ are defined by $u(t^f) = \theta$, where $\theta > 0$ is a fixed threshold. The remaining constants are the recovery time constant τ of the recovery current I_r and the reset amount $\eta_0 < 0$.

The reconstruction $\hat{x}$ is sought under the form

$$\hat{x}(t) = \sum_{f: t-T_p < t^f < t+T_d} h(t - t^f), \quad (2)$$

At time of writing, M. Gutmann was with the University of Tokyo (corresponding author, email: gutmann@sat.t.u-tokyo.ac.jp). At time of publication, he has moved to the Helsinki Institute for Information Technology, Dept. of Computer Science, University of Helsinki. A. Hyvärinen is with the Helsinki Institute for Information Technology, Dept. of Computer Science, University of Helsinki. K. Aihara is with the Institute of Industrial Science of the University of Tokyo, and the Aihara Complexity Modelling Project ERATO, JST.



(a) General scheme (b) PCA and ICA scheme (c) Scheme in this paper

Fig. 1. Modelling neural representation by means of data representation. (a) Input data x is transformed by a dynamical neural system. This encoding process should be such that the input can be decoded from the neural response. The energy consumption during the encoding process could also be required to be minimized. The encoder and decoder together provide data representation. (b) In PCA and ICA, the decoder and encoder are unknown linear transforms, and the transform is found by minimization of the the expected value of $||\mathbf{x}-\hat{\mathbf{x}}||_2^2$. ICA additionally maximizes the statistical independence between the elements of the vector $\mathbf{y}$, which is equivalent to maximizing the sparseness of the responses [2]. (c) Here, we encode by means of the spike response neuron model [9], and decoding is done from the spike timings t^f. Learning rules for the unknown encoding filter w and decoding filter h are derived from the minimization of the mean squared reconstruction error, optionally with an added penalty to minimize energy consumption during the encoding process.

which introduces a delay T_d. For the reconstruction at time t, spikes happening prior to $t-T_p$ are not considered. The decoding filter h is unknown and to be learned. The arguments for h are in the range $[-T_d \quad T_p]$. The role of $h(s)$ is different for $s>0$ and $s<0$. For $s>0$, the input at t is predicted from a spike event at $t^f<t$. On the other hand, for $s<0$, the input is reconstructed from a later spike event at $t^f>t$.

The neuron model in Equation (1) has been related to detailed biophysical quantities [9]. The encoding filter w can be considered to model a physical time-invariant system. The decoding filter h, however, is of more abstract nature. It is a hypothetical quantity and assigns meaning to each spike. It implements the "homunculus", which generates a running commentary on the spike train, see e.g. [8].

B. Cost functional

Our cost functional consists of two parts: reconstruction error and optionally, energy consumption.

The first part of the cost functional which is due to the reconstruction error is

$$J_e(h,w)=\frac{1}{2T}\int_0^T(\hat{x}(t)-x(t))^2\mathrm{d}t, \tag{3}$$

where T is a fixed time horizon.

We introduce two ways to measure the energy consumption. First, we use the average power P_a,

$$P_a=\frac{1}{T}\int_0^T I(t)^2\mathrm{d}t. \tag{4}$$

It is the electrical power that is consumed in a unit resistance through which the current $I(t)$ flows. A second measure for energy consumption stems from the idea that $I(t)\mathrm{d}t$ is proportional to the ion load that must be pumped out, or in, to restore the ion gradients in a neuron. This, together with the propagation of the action potential, is a dominant energy cost which accrues during signaling [10]. The total postsynaptic ion load per time T is

$$P_p=\frac{1}{T}\int_0^T|I(t)|\mathrm{d}t. \tag{5}$$

The output $I(t)$ of the convolution between x and w satisfies $I(t)^2\le||w||_2^2||x||_2^2$. Hence, $P_a\le||w||_2^2||x||_2^2$, and in order to minimize the average power consumption P_a during the encoding, we seek to minimize additionally to J_e

$$J_2(w)=\int_0^{T_w}w(t)^2\mathrm{d}t. \tag{6}$$

Alternatively, we can use the relation $P_a\le||w||_1^2||x||_2^2/T$, where the squared L_1 norm of w indicates the amplification gain, i.e. the ratio between output and input power. Hence, in order to punish amplification, we could minimize additionally to J_e

$$J_{1s}(w)=\left(\int_0^{T_w}|w(t)|\mathrm{d}t\right)^2. \tag{7}$$

For the second measure of energy consumption P_p, we have due to properties of the convolution $P_p\le||w||_1||x||_1/T$. Hence, in order to punish postsynaptic ion load, we minimize additionally to J_e

$$J_1(w)=\int_0^{T_w}|w(t)|\mathrm{d}t. \tag{8}$$

Alternatively, we could also take P_p directly as additional quantity to be minimized, i.e.

$$J_p(w)=\frac{1}{T}\int_0^T|I(t)|\mathrm{d}t. \tag{9}$$

In the following, we refer to the energy cost by J_E , which can be J_2, J_{1s}, J_1, or J_p.

The total cost to be minimized, due to the reconstruction error and the energy consumption, is given by

$$J = J_e + \alpha J_E, \tag{10}$$

where α weights the influence of the energy constraint.

III. Online learning rule

A. Encoding filter w

Key to the update rule for w is the calculation of the functional derivative $\delta J/\delta w(s)$. In [11], we dealt in detail with the mathematical derivation of $\delta J_e/\delta w(s)$[1]. Here, we summarize the approach. It goes via variational calculus: The encoder $w(s)$ is perturbated to $w(s) + \delta w(s)$, and the resulting change δJ_e is calculated: The perturbation $\delta w(s)$ leads to the perturbation of the spike timings δt^f which in turn changes the reconstruction $\hat{x}(t)$ to $\hat{x}(t) + \delta\hat{x}(t)$. This allows for the calculation of the functional derivative $\delta J_e/\delta w(s)$. It amounts to [11]

$$\frac{\delta J_e}{\delta w(s)} = -\frac{1}{T}\sum_f \bar{e}(t^f) y_f(s), \tag{11}$$

where

$$\bar{e}(t^f) = \int_{t^f - T_d}^{t^f + Tp} (\hat{x}(t) - x(t))\dot{h}(t - t^f)\mathrm{d}t \tag{12}$$

$$y_f(s) = \frac{-x(t^f - s)}{\dot{u}(t^f)} + \Gamma(t^f, t^{f-1}) y_{f-1}(s) \tag{13}$$

$$\Gamma(t^f, t^{f-1}) = \frac{-\eta_0}{\tau \dot{u}(t^f)} \exp\left[-\frac{t^f - t^{f-1}}{\tau}\right]. \tag{14}$$

The functional derivatives of J_E are, depending on the measurement of the energy consumption,

$$\frac{\delta J_E}{\delta w(s)} = 2w(s) \quad \text{for} \quad J_E = J_2, \tag{15}$$

$$\frac{\delta J_E}{\delta w(s)} = 2||w||_1 \mathrm{sign}(w(s)) \quad \text{for} \quad J_E = J_{1s}, \tag{16}$$

$$\frac{\delta J_E}{\delta w(s)} = \mathrm{sign}(w(s)) \quad \text{for} \quad J_E = J_1, \tag{17}$$

and

$$\frac{\delta J_E}{\delta w(s)} = \frac{1}{T}\int_0^T \mathrm{sign}(I(t)) x(t - s)\mathrm{d}t \tag{18}$$

for $J_E = J_p$. We propose the following online rule: After spike k at t^k, update the encoder by

$$w_k(s) = w_{k-1}(s) + \mu\left(\bar{e}(t^k) y_k(s) - \alpha\frac{\delta J_E}{\delta w(s)}\right), \tag{19}$$

where $\mu > 0$ is the step size and α weights the influence of the energy constraint. The algorithm is initialized with $y_0 = 0$ and e.g. $w_0 = 0$. If $J_E = J_p$, we integrate in the update rule not from zero till T but from t^{k-1} till t^k.

[1]In contrast to the present article, the focus in [11] is on the functional derivative: it includes neither simulations not the complete online rule with its discussion.

B. Decoding filter h

For the learning of h, we form from the spike timings a binary vector $\rho(n)$ by binning the spike timings into containers of size $\triangle t$. If there is a spike in the bin centered at $t = n\triangle t$, then $\rho(n)$ equals one, otherwise zero. We discretize $h(s)$ with the same bin size. The reconstruction in Equation (2) at $t = n\triangle t$ becomes then

$$\hat{x}(n) = \mathbf{h}\boldsymbol{\rho}(n), \tag{20}$$

where $\mathbf{h} = [h(-N_d)\ldots h(N_p)]$ and $\boldsymbol{\rho}(n) = [\rho(n + N_d)\ldots\rho(n - N_p)]^T$. We may further search for $h(n)$ in the form of $h(n) = \sum_k c_k \Psi_k(n)$, for some given Ψ_k and unknown c_k. The Ψ_k can for example be the Daubechies's D6 wavelet basis on Z (see e.g. [12]). Omitting wavelets located in the highest frequency bands in that representation allows for a reduction in the parameters to be learned for the decoder h. With reference to [12], we are looking in that case for $h(n)$ in V_{-j}, with e.g. $j = 2$.

Denoting by $\boldsymbol{\Psi}$ the matrix with rows $[\Psi_k(-N_d)\ldots\Psi_k(N_p)]$, we obtain for the reconstruction $\hat{x}(n) = \mathbf{c}\mathbf{y}(n)$, where

$$\mathbf{y} = \boldsymbol{\Psi}\rho. \tag{21}$$

The row vector $\mathbf{c}$ is to be determined such that J_e is minimized. Calculation of the derivative of J_e with respect to the row vector $\mathbf{c}$, i.e. after discretization, gives

$$\frac{\delta J_e}{\delta \mathbf{c}} = \frac{1}{N}\sum_{n=1}^{N}(\hat{x}(n) - x(n))\mathbf{y}^{\mathbf{T}}(n). \tag{22}$$

Using the stochastic gradient, the following least mean square (LMS) like learning rule is obtained

$$\triangle\mathbf{c}(n) = -\mu_h\left[\hat{x}(n) - x(n)\right]\mathbf{y}^{\mathbf{T}}(n), \tag{23}$$

where μ_h is the step size. The step size can be chosen optimally in each update step by using a recursive least squares algorithm, see e.g. [13], for the learning of $\mathbf{c}$.

C. Interpretation

We discuss here the mutual influence between the encoding and decoding filters w and h during learning.

The decoding filter h enters into the update rule for w in Equation (19) via $\bar{e}$, defined in Equation (12). Let us assume that the input x is positive valued. The parameter Γ in Equation (14) is also positive so that y_f in Equation (13) is < 0. The quantity $\bar{e}$ can be positive or negative. In the regime where the energy cost does not matter (because for example $\alpha = 0$), it is the sign of $\bar{e}$ which decides whether w increases ($\bar{e} < 0$) or decreases ($\bar{e} > 0$). Figure 2 illustrates that $\bar{e}$ is an indicator for the reliability of the spike, which is calculated with the aid of the decoding filter h.

The encoding filter exerts influence on the learning of the decoding filter h in Equation (23). It produces the spike timings which define the vector $\mathbf{y}$ of Equation (21). Noise triggered spikes provide thus unstructured learning data for h while while spikes which were triggered over input current I by a characteristic feature in the input provide good learning data.

IV. Simulations

The online rule was initialized with $w = 0$ and $h = 0$. Figure 3 shows an overview of the setup of the simulation and the result for $J_E = J_2$.

Figure 4 shows characteristic stages in the learning process of w and h, the associated currents and the reconstructions. In *stage 1* and *stage 2*, the encoding filter w is so small that the spikes are noise triggered. Compared to stage 1, the decoding filter h shows in stage 2 some structure which supports the learning of w. *Stage 3* shows the situation where w has strongly developed. The input current I drives now the neuron, and structured training data is provided for the learning of h. Therefore, in *stage 4*, h has reached a good decoding performance. From stage 3 on, the energy constraint on w takes effect and the encoder converges to the attractor shown as *final stage*. The final form of w reflects the trade-offs in the learning. The main peak needs be large enough to provide spikes in case of an input feature, but it cannot be too large due to the energy constraint mediated by α.

We have performed simulations to assess the influence of α, which weights the influence of the energy cost J_E on the total cost J in Equation (10). The amplitude of w, and also h, becomes smaller for values of α larger than in the present case, where $\alpha = 10^{-4}$. For $\alpha = 10^{-3}$, for example, we have $w < 10$ and $h < 0.5$ (results not shown). The general shape of both kernels is however related.

Figure 5, upper left, shows the special case where $\alpha = 0$, i.e. the case where energy consumption is not punished. The main difference to the results with $J_E = J_2$ lies in the larger main peak and sidelobes. The shape is however the same: There is also a negative primary as well as a secondary sidelobe. The remaining subfigures show the encoding filters which are obtained after learning with different measures of energy consumption J_E. The additional punishment of the energy consumption leads to encoding filters with different characteristics. The case without energy constraint shows however that this additional punishment is not needed to have stability in the development of w. The value of $\bar{e}$ decreases with increasing accuracy of the reconstruction, and makes the update rule stable.

V. Relation to other work

We have related our work to ICA and PCA in the introduction and Figure 1. Here, we discuss the relation with a method which, as our method but unlike ICA, includes time structure in the representation. The method works for a population of neurons while the presented results in this article deal only with a single neuron. In [14], a decomposition that resembles our decoding formula in Equation (2) is done for natural sounds. The researchers iteratively optimized the function dictionary which is used to decompose input x by means of the matching pursuit algorithm [15]. Important differences to our work are that in our approach, we are working with spike timings only, while in [14], a further scalar weighting of each shifted $h(t - t^f)$ is needed in the reconstruction. The weighting was called "analogue spike" and measured the strength of the spike happening at t^f. In our case, we are working with all-or-none spikes: either a spike is happening or not. Furthermore, the decomposition via matching pursuit is acausal and does not correspond to a typical neuron model while here, we have presented a data representation method that uses a standard, causal neuron model for the encoding.

VI. Conclusion

In this article, we presented and discussed an online rule for data representation with a formal spiking neuron. First, we formulated the data representation problem as an optimization problem in which reconstruction error and an optional energy cost are minimized. This enables learning of the encoder and decoder. Then, we used variational calculus to derive an online rule for the learning of encoder and decoder. Simulations showed that the online rule can learn the general shape of the input distribution.

Acknowledgment

This research is partially supported by Grant-in-Aid for Scientific Research on Priority Areas - System study on higher-order brain functions - from the Ministry of Education, Culture, Sports, Science and Technology (MEXT) of Japan (17022012). MG is further supported by MEXT grant 040680.

References

[1] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Networks*, vol. 2, no. 1, pp. 53–58, 1989.

[2] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, Jun. 2000.

[3] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, Jun. 1996.

[4] J. van Hateren and D. Ruderman, "Independent component analysis of image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex," *Proc.R.Soc.Lond. B*, vol. 265, pp. 2315–2320, 1998.

[5] A. Hyvärinen and P. Hoyer, "Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces." *Neural Computation*, vol. 12(7), pp. 1705–1720, 2000.

[6] E. P. Simoncelli and B. A. Olshausen, "Natural image statistics and neural representation," *Annual Review of Neuroscience*, vol. 24, no. 1, pp. 1193–1216, 2001.

[7] J. L. Puchalla, E. Schneidman, R. A. Harris, and M. J. Berry, "Redundancy in the population code of the retina," *Neuron*, vol. 46, no. 3, pp. 493–504, May 2005.

[8] F. Rieke, D. Warland, R. d. R. van Steveninck, and W. Bialek, *Spikes: Exploring the neural code*. MIT Press, 1997.

[9] W. Gerstner and W. K. Kistler, *Spiking Neuron Models*. Cambridge University Press, 2002.

[10] P. Lennie, "The cost of cortical computation," *Current Biology*, vol. 13, no. 6, pp. 493–497, Mar. 2003.

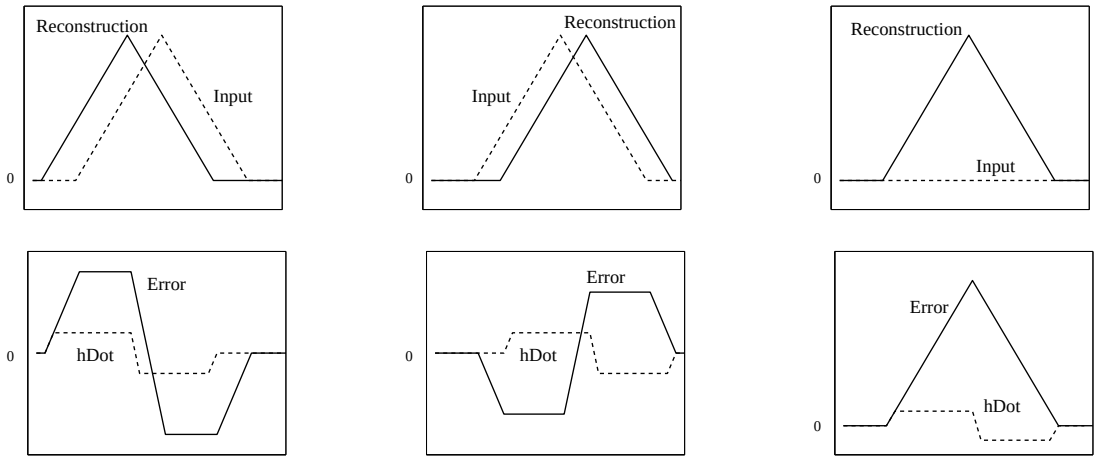
[11] M. Gutmann and K. Aihara, "Toward data representation with formal spiking neurons," *Artificial Life and Robotics*, vol. 12, 2008, in press.

[12] M. Frazier, *An introduction to wavelets through linear algebra*. Springer, 2001.

[13] S. Haykin, *Adaptive Filter Theory*. Prentice Hall, 2001.

[14] E. C. Smith and M. S. Lewicki, "Efficient auditory coding," *Nature*, vol. 439, no. 7079, pp. 978–982, Feb. 2006.

[15] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.



(a) $\bar{e} > 0 : w \downarrow$ (b) $\bar{e} < 0 : w \uparrow$ (c) $\bar{e} \approx 0 : w \leftrightarrow$

Fig. 2. The role of $\bar{e}$ in case of perfect decoder h. (a) The reconstruction $\hat{x}$ leads the input x. This happens if the spike is triggered too early. The resulting error $e = \hat{x} - x$ is weighted with $\dot{h}$ and integrated to yield $\bar{e}$ (Equation (12)). In the regime where the energy cost does not matter, $\bar{e} > 0$ leads to a decrease in w. The decrease causes the next spike to happen later and the reconstruction matches the input better. (b) The opposite case where the reconstruction lags behind the input. (c) The spike is noise triggered: $\bar{e}$ is small so that w is not affected by x to a large extent.

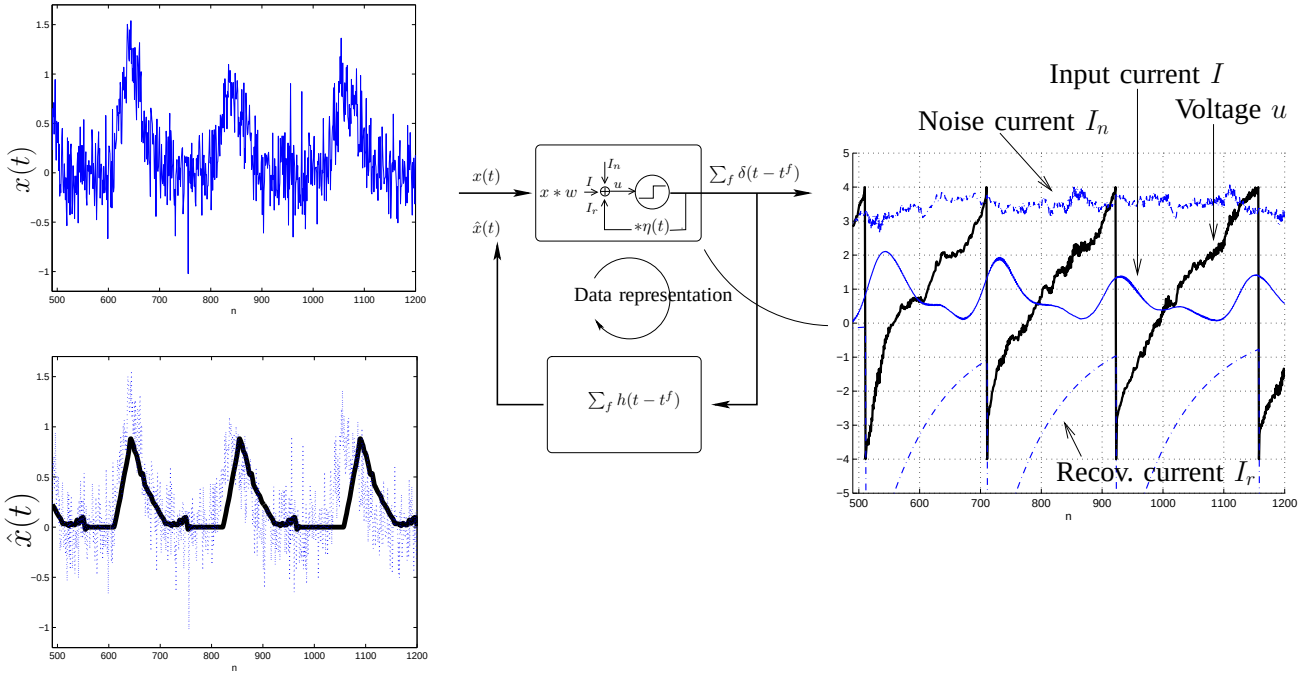


Fig. 3. Simulation setup and resulting representation of the input after learning (for $J_E = J_2$). The time varying input $x(t)$ is encoded with a formal spiking neuron such that decoding of the neural response yields an accurate reconstruction of the input: The input x yields via convolution with the encoding filter w the input current I. It forms together with the noise current I_n and the recovery current I_r the membrane voltage u. If u reaches the threshold $\theta = 4$ at $t = t^f$, the spike timing t^f is recorded for the reconstruction and the voltage is reset to $-\theta$. The spike causes a recovery current I_r. The decoding filter h implements the hypothetical homunculus which generates a running commentary $\hat{x}$ of the spike train. Comparison of $\hat{x}$ with x shows that the input is well represented by the neural spike timings. *Simulation parameters:* In total, we have run 300 rounds, where x had in each round length $50 \cdot 1024$. Discretization and integration step size was 10^{-3} time units. Integration was done with a Simpson scheme. The noise current I_n was obtained by convolution of an i.i.d. Gaussian random process (mean 11, standard deviation 8) with an exponential kernel (time constant $\tau_m = 0.05$ time units). Recovery time constant τ was 0.1 time units, step size μ was 1. The decoder h was searched in V_{-2}: 69 coefficients needed to be learned. For the encoder w 200 points were learned. The energy punishment was weighted with $\alpha = 10^{-4}$.

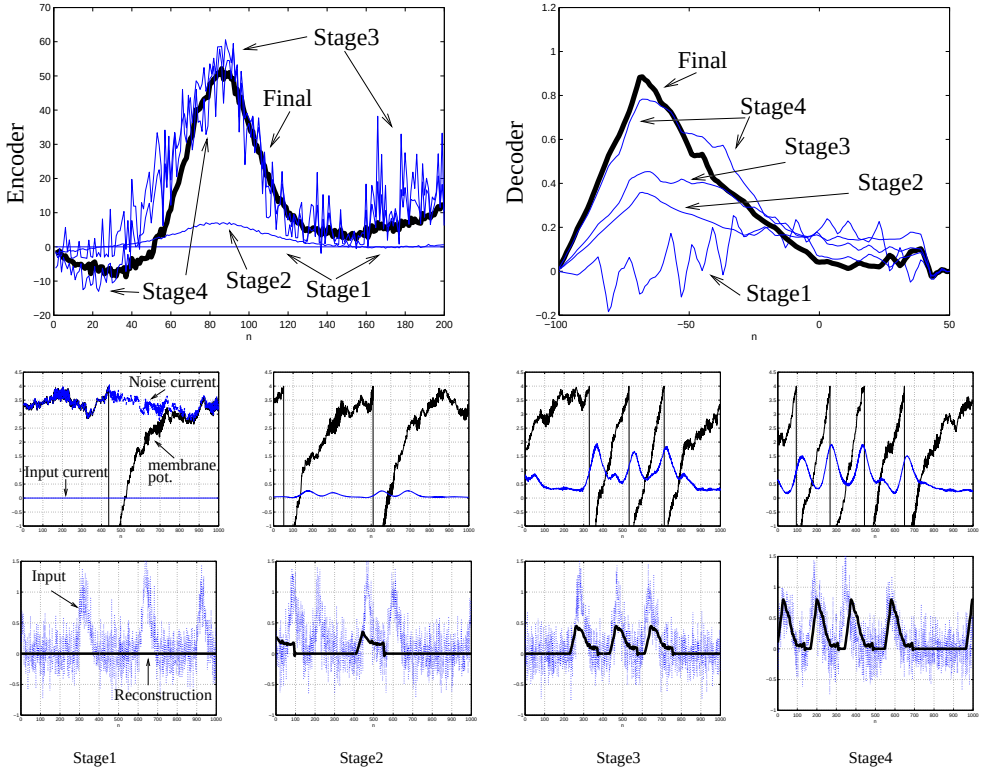


Fig. 4. The learning process for $J_E = J_2$. It can qualitatively be separated into different stages. Learning happening during stage 1 and 2 is noise driven. As explained in Figure 2, $\bar{e}$ is in that case small and the encoding filter w grows slowly. Noise triggered spikes provide unstructured learning data for the decoding filter h so that the growth of the decoder goes slowly as well. In stage 2, w is still small but the input current can have influence on the spike timings by providing, given the right amount of noise current, the additional amount of current needed to reach the threshold. Then, w develops strongly providing from stage 3 on good training data for h which develops nearly to the final form (see stage 4 curve). The difference between w shown as stage 3 and 4 is that a negative front lobe develops and that the secondary lobe is reduced. The encoder attains then a smooth form (final stage). The role of the negative front lobe of w is to prevent too early spiking. The secondary sidelobe helps to overcome the refractory current for closely space input features.

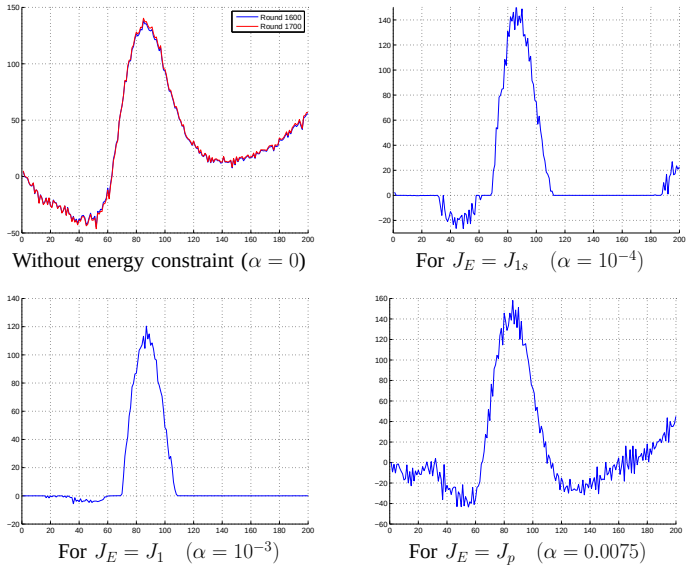


Fig. 5. Comparison of encoding filters for different energy costs. The upper left subfigure shows that the additional energy constraint is not needed for stable learning. Punishment of energy consumption via J_{1s} or J_1 (Equations (7) and (8)) leads to filters of "ungraded" shape with clear zeros. J_1 has a stronger suppressive effect in the initial phase of the learning since a vector that is not related to the scale of w is subtracted in each update (compare Equations(16) and (17)). For $J_E = J_p$, defined in Equation (9), the subtraction vector becomes input dependent, see Equation (18). Compared to the other cases, the learned w features a new, negative, secondary sidelobe after the main peak.