Optimal spike-based signal representation, Part I: Single network

Julian Büchel

Date: 4. June 2019

1 Learning to optimally represent signals with spikes [2]

How is a signal represented in the brain? Most views on this topic include that larger neuronal populations represent information and that single neurons only contribute to the great cause, but are irrelevant when looking at them independently ([5], [4]). This raises several issues: In order to represent a signal this way, large amounts of neurons are necessary, which implies a high metabolic cost. Assuming that, on a general level, biological systems try to minimize their energy, while still being able to act, implies that also neuronal groups should minimize their energy to represent and propagate a signal. This however calls for a learning rule at the neuron level, which accomplishes low energy and low information loss. In the following section I will present a local learning rule to achieve these goals and also show experimental results obtained from simulations in brian2 [3].

1.1 Derivation of voltage dynamics from loss function

A general formula for the efficiency in terms of signal representation and number of spikes can be written as

$$E = ||x - \hat{x}||_2^2 + ||r||_2^2 \tag{1}$$

or as

$$E = ||\mathbf{x} - \hat{\mathbf{x}}||_2^2 + \mu||\mathbf{r}||_2^2 + \nu||\mathbf{r}||_1 \tag{2}$$

which is a linear combination of the sparsity enforcing *l1* and *l2* cost. In these equations $\hat{x}$ is the reconstructed signal, r is the population rate vector and x the true signal.
Given that the signal is reconstructed with

$$\hat{x} = Dr \tag{3}$$

for a decoder D, we can rewrite Eq. 13 to

$$E = (x - Dr)^T(x - Dr) + \mu||r||_2^2 + \nu||r||_1 \tag{4}$$

which has the derivative

$$\frac{\partial E}{\partial D} = -2(x - Dr)r^T = -2(x - \hat{x})r^T \tag{5}$$

Setting this to zero and solving for D yields

$$D = xr^T(rr^T)^{-1} \tag{6}$$

Since this only applies to vectors at one instance in time, we need to define the loss function over a time period and take the average. The derivative of the averaged loss function gives us:

$$\boxed{D = \left\langle xr^T \right\rangle \left\langle rr^T \right\rangle^{-1}} \tag{7}$$

where $\langle . \rangle$ denotes the average over time. In software simulations this is realized with a simple moving average.
However, there is a problem with this definition of the decoder: How does it enforce that not only one neuron encodes all of the signal? In particular, how can it make sure that this, over-dramatic, scenario does not occur:

$$r^T = [r_1 \; 0 \; ... \; 0]$$

and

$$D = \begin{pmatrix} D_{11} & .. & 0 \\ ... & ... & 0 \\ D_{1I} & ... & 0 \end{pmatrix}$$

meaning that the rate of one neuron describes the signal entirely. To prevent this, we can add a set of Lagrangians to the loss function:

$$L = ||\mathbf{x} - \hat{\mathbf{x}}||_2^2 + \sum_{n=0}^{N} \lambda_n (||D_n||_2^2 - a_n) \tag{8}$$

where a_n is the length of the $n-$th column of D. This has the desired effect that every neuron only partly contributes to the reconstruction of the signal. This results in a new, penalized, closed form solution for D:

$$\boxed{D = \left\langle xr^T \right\rangle \left\langle rr^T + \Lambda \right\rangle^{-1}} \tag{9}$$

where $\Lambda = I\lambda$ with λ being the vector of Lagrange multipliers.

Following the supplementary material (SI), we will now discuss how to optimize the network topology. In particular the synaptic weights and the thresholds under the assumption that the decoder is fixed. The main idea [1] behind the following derivation is that a neuron in the population should only spike if it contributes to the reduction of the loss function. This means that

$$\text{l}(\textit{neuron n spikes}) < \text{l}(\textit{neuron n does not spike}) \tag{10}$$

This can be formulated mathematically using the loss function from equation 15:

$$E = ||x - \hat{x}||_2^2 + \mu||r||_2^2 + \nu||r||_1$$

When neuron n spikes, r_n becomes $r_n + 1$ and thus the cost function becomes

$$\text{l}(\textit{neuron n spiked}) = ||x - \hat{x} - D_n|| + \mu||r + e_n||_2^2 + \nu||r + e_n||_1 \tag{11}$$

where e_n is the unit vector with $(e_n)_j = 1$ if j = n and $(e_n)_j = 0$ else. Since

$$||r + e_n||_2^2 = (\sum_{i,i\neq n} r_i) + (r_n + 1)^2 = ||r||_2^2 + 2r_n + 1$$

and

$$\nu||r + e_n||_1 = \nu(\sum_{i,i\neq n} |r_i|) + \nu|r_n + 1| = (*)\ \nu(\sum_{i,i\neq n} |r_i|) + \nu|r_n| + \nu = \nu||r||_1 + \nu$$

where (*) holds because $r \geq 0$ for all r.

This makes it possible to rewrite equation 10 to

$$\boxed{D_n^T x - D_n^T Dr - \mu r_n > \frac{1}{2}(||D_n||_2^2 + \mu + \nu)} \tag{12}$$

This equation is crucial in understanding how the voltage- and threshold equations and their derivatives are formed.
Earlier we said that this was the condition for neuron n to fire a spike, namely to reduce the loss. We notice that the r.h.s. is constant (remembering that the decoder is fixed for now) and the l.h.s. is time-dependent, since it depends on x and r. It is therefore natural to denote the voltage of neuron n as

$$\boxed{V_n(t) = D_n^T x - D_n^T Dr - \mu r_n} \tag{13}$$

and the threshold of the $n-$th neuron as

$$\boxed{T_n = \frac{1}{2}(||D_n||_2^2 + \mu + \nu)} \tag{14}$$

Following the SI, we take the temporal derivative and get

$$\dot{V}_n(t) = D_n^T \dot{x} - D_n^T D\dot{r} - \mu\dot{r}_n$$

using $\dot{r} = -\lambda r + o$ and $\dot{x} = -\lambda x + c$ (to clarify: c is the raw input and x is the filtered version)

$$\dot{V}_n(t) = D_n^T(-\lambda x + c) - D_n^T D(-\lambda r + o) - \mu(-\lambda r_n + o_n)$$
$$= -\lambda(D_n^T x - D_n^T Dr - \mu r_n) + D_n^T c - D_n^T Do - \mu o$$

using $V_n = D_n^T x - D_n^T Dr - \mu r_n$ from equation 13

$$\dot{V}_n(t) = -\lambda V_n + D_n^T c - (D_n^T D - \mu e_n)o$$

This gives us the chance to define the differential equation of the whole network:

$$\boxed{\dot{V} = -\lambda V + Fc - \Omega o} \tag{15}$$

where

$$F = D^T$$

and

$$\Omega = D^T D + \mu I = F F^T + \mu I$$

are the optimal weights. One can see that this is the classical differential equation describing a network of I&F neurons. Additionally, since $||D_n||_2^2 = (D^T D)_{n,n}$, we get

$$T = \frac{1}{2}(-\text{diag}(D^T D) + \mu + \nu)$$

$$\boxed{T = \frac{1}{2}(\text{diag}(\Omega) + \nu)} \tag{16}$$

1.2 Derivation of recurrent weight dynamics

After having defined the decoder and the voltage dynamics, it is time to derive the learning rules for the feedforward and recurrent weights.
We will design the learning rule of the recurrent weights in a way so that the excitatory and inhibitory signals cancel each other out. A good E/I balance is key to a good representation, but why?
Let us assume that we are in a state where we have learned a good decoder that can properly decode r so that

$$x - \hat{x} \approx 0$$

But since the network only has access to the weighted input it should be

$$Fx - F\hat{x} \approx 0$$

Using $\hat{x} = Dr$ we can rewrite this to

$$Fx - FDr \approx 0$$

which corresponds to the previously derived voltage function of the network. This also applies on the neuron level:

$$F_n^T x - F_n^T Dr \approx 0$$

meaning that excitatory inputs should be balanced by inhibitory inputs.

In order to achieve such a balance it is desirable that over the long run, the membrane voltages before and after a presynaptic spike cancel each other out:

$$L(t_j) = ||\frac{1}{2}(V^{\text{before}}(t_j) + V^{\text{after}}(t_j))||^2$$

Note that the "canceling out" happens because of negative voltages. After the arrival of a presynaptic spike at time t_j of neuron with index $k(j)$, the voltages are changed and the following relation is formed:

$$V^{\text{after}} = V^{\text{before}} + \Omega e_{k(j)}$$

where $e_{k(j)}$ is the unit vector with 1 at the position of the neuron that spiked. To summarize up to this point: Upon a presynaptic spike at time t_j of the $k(j)$-th neuron, all neuron-voltages change with respect to the influence the spiking neuron has. This influence is measured in the form of the recurrent weights of the connections this neuron has to the other neurons in the population.
Using this relation we can rewrite the loss function from above to

$$L = ||V^{\text{before}} + \frac{1}{2}\Omega e_{k(j)}||^2$$

this term has the derivative, with respect to $\Omega_{n,k}$:

$$\boxed{\Delta\Omega_{n,k} = -2V_n^{\text{before}} - \Omega_{n,k}} \tag{17}$$

if neuron k spiked. The connection $\Omega_{n,k}$ is the connection from the spiking, presynaptic neuron with index k to the n-th neuron.

In the sections before we have seen that if we induce a small l2 cost on the rates, the Ω should ideally converge towards $-FD - \mu I$. This causes a minor change in the learning rule for the recurrent weights:

$$\boxed{\Delta\Omega_{n,k} = -2(V_n^{\text{before}} + \mu r_n) - \Omega_{n,k} - \mu\delta_{n,k}} \tag{18}$$

where $\delta_{n,k}$ is $[e_n]_j$ and therefore 1 if and only if $n = j$.

1.3 Derivation of feedforward weight dynamics

In order to understand how the weight update rule for the feedforward weights are derived, it is important to understand why D converges to

$$\boxed{D \to 2\left\langle x - \hat{x}\right\rangle}$$

This result has its origins in the vectorized weight update for the recurrent weights

$$\Delta\Omega = -2Ve_{k(j)}^T - \Omega e_{k(j)}e_{k(j)}^T$$

which, using $V = Fx + \Omega r$, can be rewritten as

$$\Delta\Omega = -2(Fx + \Omega r)e_{k(j)}^T - \Omega e_{k(j)}e_{k(j)}^T$$

$$= -2Fxe_{k(j)}^T - \Omega(2r + e_{k(j)})e_{k(j)}^T$$

We now assume that the recurrent weights were learned and we have reached a fixed point. Therefore $\langle\Delta\Omega\rangle_{spikes} = 0$ applies, where $\langle.\rangle_{spikes}$ stands for the average over many spikes. Using this we can rewrite the previous equation to

$$2F\left\langle xe_{k(j)}^T\right\rangle_{\text{spikes}} = -\Omega\left\langle(2r + e_{k(j)})e_{k(j)}^T\right\rangle_{\text{spikes}}$$

Since $\Omega \to -FD$ we can rewrite the equation to

$$2\left\langle xe_{k(j)}^T\right\rangle_{\text{spikes}} = D\left\langle(2r + e_{k(j)})e_{k(j)}^T\right\rangle_{\text{spikes}}$$

By looking at individual entries and using $\hat{x} = Dr$, one can see that

$$D_{i,n} = 2 \langle x_i - \hat{x}_i \rangle_n$$

For details see SI p.20 or derive using pen and paper. Note that since we made the substitution $\Omega \to -FD$, D is not initially equal to $2 \langle x - \hat{x} \rangle$, but rather converges towards it.

Another key assumption that is necessary for the feedforward weight update is that $x - \hat{x}$ is proportional to x. This may seem strange, but assuming that the networks firing rate is limited (imposed by the regularization on the rate vector) larger x generate a larger error since the rates are not able to "keep up". This leads to a quadratic relation, on average, between x and the error $x - \hat{x}$. This observation is key since the feedforward weights do not have access to the predicted output $\hat{x}$.

We want to minimize

$$L = F_n^T x - F_n^T \hat{x}$$

Remembering that when neuron n spikes, $r \to (r + e_n)$ we get

$$\begin{aligned} L &= F_n^T x - F_n^T (D(r + e_n)) \\ &= F_n^T (x - \hat{x}) - F_n^T F_n \end{aligned}$$

where we used $F \to D^T$. Now, we use the previously explained proportionality $x - \hat{x} \propto x$ to get

$$F_n^T x - F_n^T F_n$$

The resulting derivative is thus

$$\boxed{\Delta F_n = x(t_j) - F_n} \tag{19}$$

if neuron n spiked.

1.4 Scaling of recurrent and feedforward weights

Earlier we derived the condition

$$D_n^T x - D_n^T D r - \mu r_n > \frac{1}{2}(||D_n||_2^2 + \mu + \gamma)$$

for neuron n to spike. If this condition is met then a spike of neuron n will reduce the loss. From this condition we can derive the neuron threshold

$$T_n = \frac{1}{2}(||D_n||_2^2 + \mu + \gamma)$$

This leads to a constraint on the length of the n-th decoder vector:

$$||D_n||_2^2 = 2T_n - \mu - \gamma$$

Knowing the nature of the optimal feedforward and recurrent weights, we can derive following constraints: Since $F \to D^T$ we know that $||F_n^T||_2^2 \overset{!}{=} 2T_n - \mu - \gamma$

and because $-\Omega \rightarrow D^T D + \mu I$ we have $-\Omega_{n,n} \overset{!}{=} ||D_n||_2^2 + \mu$ and therefore $-\Omega_{n,n} \overset{!}{=} 2T_n - \gamma$.

We can now rewrite the recurrent weight update with a scaled factor:

$$\boxed{\Delta\Omega_{n,k} = -\beta_n(V_n + \mu r_n) - \Omega_{n,k} - \mu\delta_{n,k}} \tag{20}$$

If neuron k spiked. But what is β_n?
By solving $-\Omega_{n,n} = 2T_n - \gamma \leftrightarrow F_n\beta_n \left\langle x - \hat{x} \right\rangle_n = 2T_n - \gamma$ for β_n, we obtain

$$\boxed{\beta_n = \frac{2T_n - \mu - \gamma}{T_n + \mu \left\langle r_n \right\rangle}} \tag{21}$$

For details on this see the SI or use pen and paper with the identities: $V_n = D_n^T x - D_n^T Dr - \mu r_n$ and $V_n = T_n$ (before the spike the voltage per definition equals the threshold).

Similar scaling is applied to the feedforward weights F:

$$\boxed{\Delta F_n = \alpha_n x - F_n} \tag{22}$$

This scaled version of the update corresponds to a scaled fixpoint $F_n \rightarrow \alpha_n \left\langle x \right\rangle_n$. In order to fulfill the constraint $||F_n||_2^2 = 2T - \mu - \gamma$, α_n needs to be

$$\boxed{\alpha_n = \frac{2T - \mu - \gamma}{F_n^T \left\langle x \right\rangle_n}} \tag{23}$$

For a derivation see the SI or use pen and paper with the identity $||F_n||_2^2 = F_n^T F_n$ and $\left\langle x \right\rangle_n^T \left\langle x \right\rangle_n = F_n^T \left\langle x \right\rangle_n$.

In the main paper, α_n and β_n are the same for all neurons and are simply treated as free parameters. This leaves us with the final updates:

$$\boxed{\Delta F_n = \alpha x(t_j) - F_n} \tag{24}$$

$$\boxed{\Delta\Omega_k = -\beta(V + \mu r) - \Omega_k - \mu e_k} \tag{25}$$

1.5 Experiments

In order to test the learning rules, I implemented a network of $N = 200$ spiking neurons using brian2. In the following figure one can observe the two signals x_1 and x_2 (red) and the reconstructed signals $\hat{x}_1$ and $\hat{x}_2$ (green). Below that, one can see the spike trains of the neurons. As expected, the spike trains sparsify. Despite the increasing sparsity, the reconstruction error is reduced over the course of training, as can be seen in the 4th row. In order to observe the increasing importance of individual spikes, we randomly shifted a few spikes (red dots) in the spike train and reconstructed the signal using the previously learned decoder. As expected, the signal is off and in regions of increased sparsity, shifts of the spikes cause a greater reconstruction error.

1.6 Difficulties

In the main paper, the authors used a time constant of 6ms for the filter of the white noise signal. When I used 6 as the sigma for the Gaussian filter, the signal was still very "white-noise like" and the assumption that $x - \hat{x} \propto x$ does not hold anymore because the sparsifying constraint on the rates does not allow for encoding of rapidly changing signals. I therefore used a higher sigma in order to produce a smoother signal that allows for better reconstruction.
Furthermore I used 200 instead of 20 neurons for the simulation. When I use 20 neurons, one can still observe sparsifying spike trains at more or less the same, but not improving, reconstruction accuracy.

1.7 Outlook

The paper also designed learning rules for the case of separated excitatory and inhibitory neuron populations following Dales law, which I have not implemented yet. If we can reproduce the results found in the paper, we could integrate a signal representation unit in a novel chip design, which takes as input a smoothed signal x and produces metabolic-cost-efficient spike trains, which can optimally encode any correlated and uncorrelated signal.
Furthermore, this way of representing signals allows for cost-efficient classification through sparse coding. If these sparse spike trains are used for classification, another problem would occur: The neurons do not "know", which part of the signal is noise and which part is valuable for classification. The network therefore needs feedback to guide the signal representation to the right direction. This problem is known from Machine Learning and one of many solutions is the Ladder Network [6], which combines an unsupervised Autoencoder approach with standard backpropagation-based supervised learning. It would be interesting to see if a completely local learning rule can be developed that incorporates the current autoencoder-based learning rule with some supervised component.

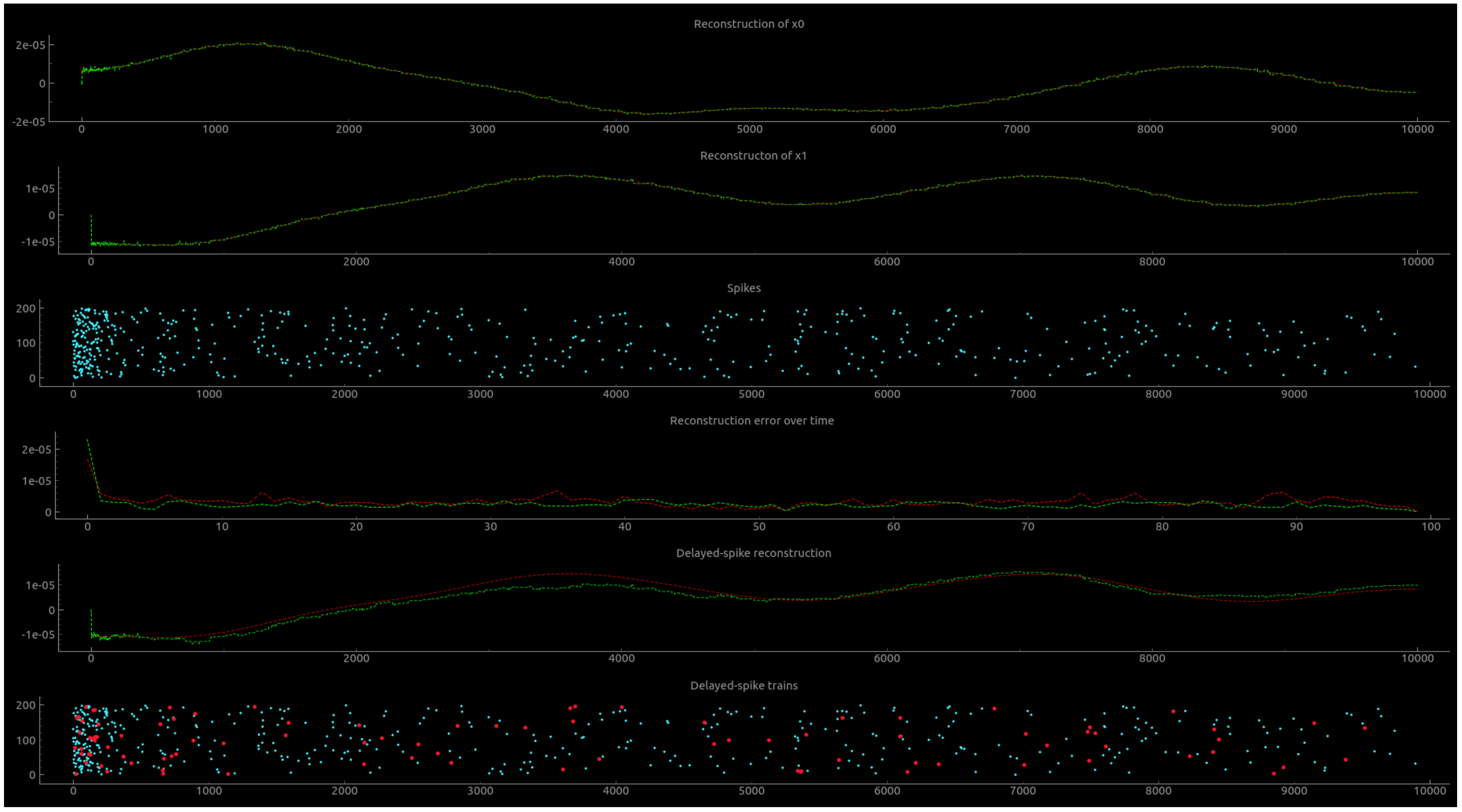

Figure 1: **Top two rows**: True signals (red) and reconstructed signals (green). **Third row**: Spike trains of the neurons. **Fourth row**: Reconstruction error over time. The error was taken over a period of 10ms. **Fifth row**: Reconstruction after shifting neurons in the spike train. **Sixth row**: The perturbed spike trains of the neurons.

References

[1] Ralph Bourdoukan, David G. T. Barrett, Christian K. Machens, and Sophie Denève. Learning optimal spike-based representations. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'12, pages 2285–2293, USA, 2012. Curran Associates Inc.

[2] Wieland Brendel, Ralph Bourdoukan, Pietro Vertechi, Christian K. Machens, and Sophie Denéve. Learning to represent signals spike by spike. *arXiv e-prints*, page arXiv:1703.03777, Mar 2017.

[3] Dan Goodman and Romain Brette. The brian simulator. *Frontiers in Neuroscience*, 3:26, 2009.

[4] J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

[5] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput.*, 14(11):2531–2560, November 2002.

[6] Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. Semi-supervised learning with ladder network. *CoRR*, abs/1507.02672, 2015.