

# The Best Spike Filter Kernel Is a Neuron

D. van den Broek<sup>1</sup> (Dick.vandenBroek@mpi.nl), M. Uhlmann<sup>1</sup>, R. Duarte<sup>3</sup>, H. Fitz<sup>1,2</sup>, P. Hagoort<sup>1,2</sup>, K.M. Petersson<sup>1,2,4</sup>

<sup>1</sup>Max-Planck Institute for Psycholinguistics, Department Neurobiology of Language, Wundtlaan 1, 6525 XD Nijmegen, The Netherlands

<sup>2</sup>Donders Centre for Cognitive Neuroimaging, Radboud University, Kapittelweg 29, 6525 EN Nijmegen, the Netherlands

<sup>3</sup>Institute of Neuroscience and Medicine, Forschungszentrum Jülich, Wilhelm-Johnen-Strasze, 52428 Jülich, Germany

<sup>4</sup>Center for Biomedical Research, Department of Biomedicine, University of Algarve, Portugal

## Abstract

**A common approach to extracting information from simulated spiking neural networks is to train readouts on a spike-rate variable obtained through convolution of output spike-trains with a filter. Here we argue that best practice is to use neurons as spike filters. We describe how neural circuits consist of stock and flow variables that co-determine each other and argue that membrane potentials provide access to the information contained in the circuit in a more natural and unbiased way than filtered spike-trains. We compare the two different approaches to readout calibration in a classification task.**

**Keywords:** Spiking neural networks; decoding; spike-train filter; membrane potential; stock-flow duality

## Spiking neural network models

Spiking neural networks (SNN) are used to simulate brain dynamics at the level of single cells. A SNN includes reduced models of neurons and synapses, and connectivity structure. For instance, a linear leaky integrate-and-fire neuron describes how the membrane potential integrates synaptic currents and models spiking by resets at a threshold. More detailed models add additional dynamic variables, resulting in more complex and realistic dynamics. Synaptic models translate spike input into post-synaptic currents and can be absorbed into the neuron model equations. Connectivity determines where neuronal axons terminate. A general SNN model of  $N$  neurons is a coupled system of dynamic variables:

$$\begin{aligned} \frac{d\vec{x}_i}{dt} &= \vec{f}(\vec{x}_i) + \mathbf{G}(\vec{x}_i) \text{Inp}_i(t), \\ \text{Inp}_i(t) &= \mathbf{W}\vec{S}(t), \quad S_i = \sum_{\{t_i^f\}} \delta(t - t_i^f). \end{aligned} \quad (1)$$

Neuron  $i$  is described by state vector  $\vec{x}_i$ , which contains the somatic membrane potential. Additionally, it may contain input conductances, adaptive currents, moving thresholds, membrane potentials of dendritic compartments, etc. (Gerstner, Kistler, Naud, & Paninski, 2014). When the potential of neuron  $i$  reaches a (dynamic) threshold it is reset and a spike is recorded at time  $t_i^f$ . The spike train vector  $\vec{S}$  keeps track of all spikes in the network and the connectivity matrix  $\mathbf{W}$  distributes spike-trains over pre-synaptic sites. The vector  $\text{Inp}_i$  denotes the spike input to the synapses of neuron  $i$ . The function  $\vec{f}$  describes the autonomous dynamics of the system and matrix  $\mathbf{G}$  describes how it is forced by spike input in a state-dependent manner. External input can be provided, e.g., by bias currents, background noise, or spiking activity on some input channels.

## Decoding

Input-driven networks generate time-series of spike-trains and state-variables. Since the neural code used by various brain systems is not known, these time-series are analyzed to track information processing and memory. One approach is to train a linear classifier, or readout, on some set of dynamic variables in the network (Maass, Natschläger, & Markram, 2002). It is common to calibrate the readout on spike-trains which are discrete events in continuous time (point processes). Since methods for analyzing continuous signals are better developed, spike-trains are transformed into continuous variables such as the instantaneous rate of neurons. Generally a spike filter takes the form:

$$h(t) = \int_{-\infty}^t k(t,s)S(s)ds, \quad (2)$$

where  $k$  is a convolution kernel. Often, the kernel is an exponential decay  $\exp(-(t-t^f)/\tau)$  with time-constant  $\tau$ . Depending on  $\tau$ , filtered signals can look similar to the spike-train or a local estimate of the spike-rate. Typically,  $\tau$  is chosen to match the decay of post-synaptic currents (PSC), to have the signal correspond to a biologically available signal. In the remainder of this abstract we argue that readouts can and should be adapted directly on the membrane potential instead. The two approaches are illustrated in Figure 1.

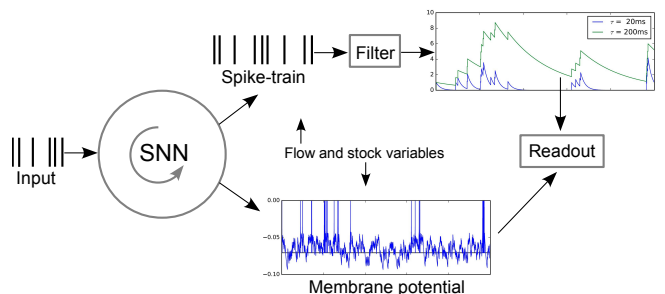


Figure 1: Input-driven, recurrent SNN. Decoding on filtered spike-trains versus decoding on membrane potentials.

## Choice of kernel

Properties of filtered spike-trains depend on the choice of the filter kernel. This is problematic if SNNs are used to study cognitive functions as performance might depend on kernel characteristics more than on network dynamics. To illustrate this, consider a SNN that is driven by a random sequence of symbols (50ms each) and the task of the readout is to recall the previous symbol. An exponential kernel with  $\tau = 200\text{ms}$  preserves transients of spiking from the previous input. Even

though there might be no trace of this information left in the network, the readout can succeed at the task. For  $\tau = 20\text{ms}$  (i.e., the filter mimics PSCs), on the other hand, the readout might fail. In this case, the filtered spike train may not contain a trace of the previous input, but this information might still be available in network state-variables with longer time-scales. Again, we are unable to assess the network’s capabilities.

### Stock-flow duality

Network dynamics can be characterized in terms of stock- and flow-variables ( $S$  and  $F$ , respectively). Stock-variables describe the state of the network nodes, while flow-variables describe the quantities that are transferred between nodes. In a SNN, spike-trains are flow-variables and neuronal state-vectors are stock-variables. According to equation (1), neuronal input evolves the state-variables in time. If all input spike-trains  $F$  are known, we can determine all state-variables  $S$ . Conversely, the state-variable responsible for generating spikes is the membrane potential. If the membrane potential time-series are known, we can reconstruct the spike-trains. Thus,  $S$  and  $F$ -variables co-determine each other, they are duals. They do not hold the same information at each point in time, but a full time-series of either  $S$  or  $F$  describes the entire system. Therefore, readout calibration on  $S$  or  $F$  is, in principle, equivalent. The choice of readout variable should aim to minimize artifacts and biases associated with each method.

### Decoding on state-variables

The arbitrariness of choosing a filter for spike-trains can be problematic (see example above). Reading out from membrane potentials,  $V_m$ , instead, ensures a continuous signal and no free parameters have to be chosen. When the readout succeeds at the recall task, we know that relevant information was available in the network at the time of decoding. Should it fail, the information could still be located in other state-variables such as, e.g., synaptic conductances or adaptive currents. When these variables are coupled to the somatic membrane potential, they can ‘write’ stored information into it. Thus,  $V_m$  often provides a good, unbiased estimate of the information present in the SNN.

### Neurons as spike filtering kernels

In large-scale brain networks, readouts are implemented as neural circuits (Rigotti et al., 2013) and information from the functional SNN needs to be transmitted to these readouts. If relevant information is available in  $S$ , but not in  $F$ , how could a downstream network have access to it? A readout neuron transforms input spikes into PSCs that integrate with local state-variables to determine the membrane potential at the soma. The filter kernel that best captures this transformation is the coupled equation for the neuronal-state dynamics itself! In contrast, choosing PSC-like time-scales in the spike filter is insufficient since in conductance-based models PSCs depend on the neuron state as well. Hence, the most natural filter for a spike-train is a neuron, and this justifies readout adaptation directly on membrane potentials.

## Experimental results

We illustrate the difference between the two decoding methods in a simple identity mapping task. Different input symbols, each represented by a unique set of excitatory spike-trains, were forcing a SNN of 10,000 neurons. The readout was adapted on a sub-sample of either membrane potentials  $V_m$ , or exponentially filtered spike-trains  $h$  with  $\tau = 20\text{ms}$ . The task was to produce a vector representation identifying the input symbol. The number of distinct symbols  $N_u$  varied between 1 and 500. Figure 2 shows the squared error between readout and binary target vectors. Error was substantially lower for the

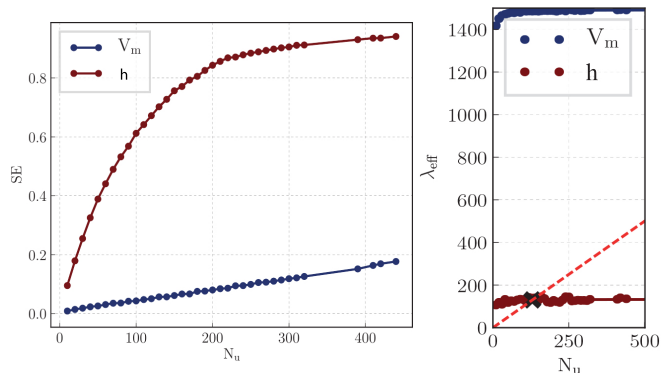


Figure 2: Left: Squared error against number of input symbols when reading out from membrane potentials (blue) or filtered spike-trains (red). Right: Effective dimensionality  $\lambda_{\text{eff}}$  of the two signals.

$V_m$ -readout and scaled linearly in  $N_u$ . Thus, the  $V_m$ -readout was better able to distinguish symbolic representations within the SNN. Effective dimensionality  $\lambda_{\text{eff}} = 1/\sum_{j=1}^M \lambda_j^2$ , where  $\lambda_j$  are normalized PCA eigenvalues, measures how well the available SNN state-space is used for encoding. For both readouts,  $\lambda_{\text{eff}}$  was largely insensitive to  $N_u$ . However,  $V_m$  provided a higher-dimensional signal than  $h$ , and is thus capable of carrying more information.

## Conclusion

We have argued that the natural way to interpret a spike signal is to filter it through a neuron. Across time, neuronal state-variables carry the same information as spike-trains. Adapting a readout directly on neuronal state-variables, such as the membrane potential, avoids arbitrary choice of a filter kernel. This makes a state-variable readout less biased and potentially improves SNN performance in modeling cognitive tasks.

## References

Gerstner, W., Kistler, W. M., Naud, R., & Paninski, L. (2014). *Neuronal dynamics*. Cambridge University Press.  
 Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states. *Neural Computation*, 14, 2531–2560.  
 Rigotti, M., Barak, O., Warden, M., Wang, X.-J., Daw, N., Miller, E., & Fusi, S. (2013). The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497, 585–590.