Learning Data Representation with a Population of Spiking Neurons as Encoder

Michael Gutmann[1], Aapo Hyvärinen[2], and Kazuyuki Aihara[13]

[1]The University of Tokyo, [2]University of Helsinki, [3]ERATO, JST

The neural representation of sensory input remains an incompletely understood topic in neuroscience. On the theoretical side, neural representation has been modeled by data representation. In, for example, [1], an extension of ICA was used to find a linear encoding transform to represent movies. In [2], an efficient representation of sound was learned where matching pursuit [see reference within [2]] was used for encoding.

The modeling of neural representation by means of data representation could be improved by using established, and more detailed neuron models for the encoding. This would facilitate theoretical investigations into neural representation on a cellular level.

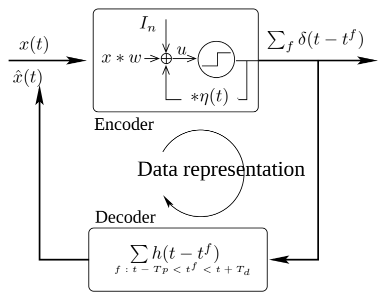

Figure 1: Only a single neuron is shown to highlight the encoding process. The convolution of input x with causal w defines the input current I. Voltage u is the sum of I and the recovery current η, and an optional noise current I_n. If u passes threshold θ at $t = t^f$, the spike timing t^f is recorded for the reconstruction, the voltage is reset to $-\theta$, and a new recovery current is triggered. A delay is allowed in the decoding. Encoding filter w and decoding filter h are unknown and learned from the input so that the mean squared reconstruction error is minimized.

We derived learning rules for data representation where a population of spiking neurons acts as encoder. Each neuron was modeled by the spike response model [3]. Figure 1 explains our approach. In case of a population of neurons, for each neuron m, an encoding filter w_m and decoding filter h_m is learned. Figure 2 left, shows an example where the input (blue) is after learning accurately represented by a population of spiking neurons (red). We used initially five neurons for the representation. Learning silenced two neurons, and the other neurons differentiated to code for different aspects of the input (Figure 2 right). As an extension, we have also derived learning rules for synaptic connections between the neurons, and included punishment of energy consumption.

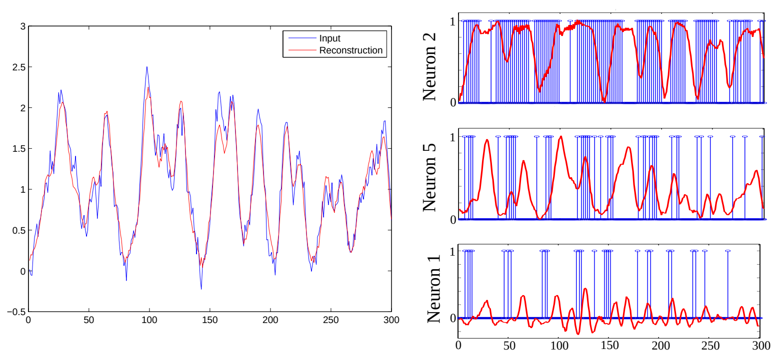

Figure 2: Left: Input and reconstruction. The reconstruction is obtained by decoding all the spike trains: $\hat{x}(t) = \sum_m \sum_f h_m(t - t_m^f)$. The input was artificially created by summing two randomly shifted bump functions of different width. Right: Spike timings (blue) and partial approximation (red) for each neuron. The partial approximation shows for fixed neuron m its contribution $\sum_f h_m(t - t_m^f)$ to the reconstruction. Neuron 2 codes for the local mean, neuron 5 for an intermediate resolution, and neuron 1 for fine details.

Acknowledgments

This research is partially supported by Grant-in-Aid for Scientific Research on Priority Areas - System study on higher-order brain functions - from MEXT (17022012). MG is supported by MEXT grant 040680.

References

[1] Hyvärinen, Hurri and Väyrynen, *J.Opt. Soc. Am. A, OSA* 20:1237-1252, 2003.

[2] Smith and Lewicki, *Nature* 439:978-982, 2006.

[3] Gerstner and Kistler, Spiking Neuron Models *Cambridge University Press* 2002.