Stochastic computation on spiking neuromorphic hardware

D. Dold[1,2], Á. F. Kungl[1], A. Baumbach[1], J. Klähn[1], I. Bytschok[1], P. Müller[1], J. Schemmel[1], K. Meier[1] and M. A. Petrovici[2]
O. Breitwieser[1], A. Grübl[1], M. Güttler[1], D. Husmann[1], M. Kleider[1], C. Koke[1], A. Kugele[1], C. Mauch[1], E. Müller[1], S. Schmitt[1]

[1]Heidelberg University, Kirchhoff-Institute for Physics
[2]University of Bern, Department of Physiology

Electronic Vision(s)

The Manfred Stärk Foundation

Spiking neuromorphic hardware

Developed as part of the Human Brain Project's Neuromorphic Computing platform, the **BrainScaleS system** [1] consists of 20 integrated circuit wafer modules (Fig. 1-3). In the future, these will allow the implementation of large-scale spiking neural networks on neuromorphic hardware.

Fig. 1 *The BrainScaleS system, consisting of 5 racks with 20 modules and 2 racks with command and control hardware.*

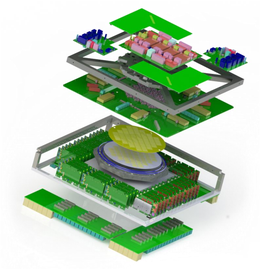

Fig. 2 *Schematic representation of a module including the support hardware (FPGAs, power supply, monitoring, networking, etc.) and the wafer.*

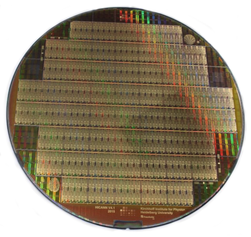

Fig. 3 *Photograph of a single silicon wafer, consisting of 384 HICANN chips.*

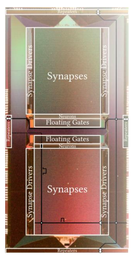

Fig. 4 *Photograph of a single HICANN-chip. Chip area is dominated by the synapse arrays, with the neurons and their analogue storage (floating gates) in the center. Surrounding area is used for communication.*

A single module contains 384 mixed-signal HICANNs (Fig. 4), each implementing up to 512 analogue neurons based on the **Adaptive Exponential Integrate-and-Fire model** [2]. The 112,640 possible synapses per chip allow a large variety of connectivity patterns.

Stochastic computing with spikes

It can be shown that networks of LIF neurons can approximately **sample from binary Boltzmann distributions** [4]. LIF neurons are associated to a binary state according to their refractory status (Fig. 6, top right). Synaptic connections represent the weights **W** of the Boltzmann distribution, while the biases **b** are implemented by a shift in the leak E_l.

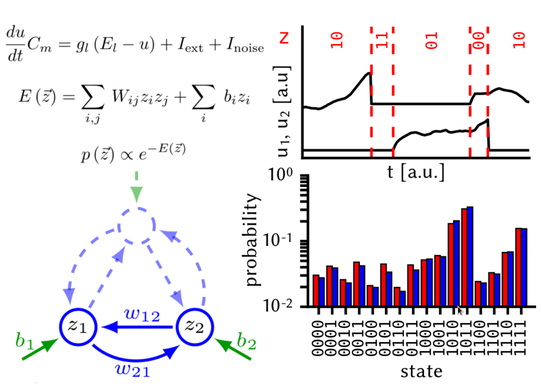

Fig. 6 *Framework used for sampling with LIF neurons. Individual neurons represent binary random variables. Weights W_{ij} and biases b_i of the Boltzmann distribution are implemented as synaptic connections and leak potentials, respectively.*

In order to perform sampling, the neurons receive high-frequency Poisson noise as input, elevating them to a high-conductance state [5] resulting in **sigmoidal response functions** (Fig. 7). These response functions can then be used to calculate the translation of weights and biases from the abstract Boltzmann domain to corresponding synaptic conductances and leak potentials (Fig. 8).

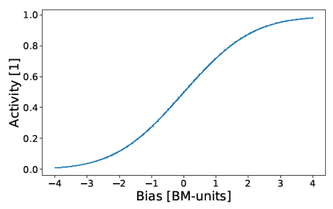

Fig. 7 *Response function of an LIF neuron with conductance-based synapses under high-frequency Poisson noise.*

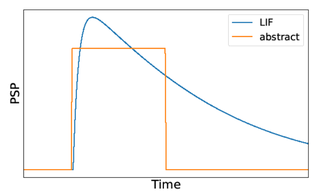

Fig. 8 *The synaptic interaction is chosen to approximate, on average, the corresponding Boltzmann weight.*

Noiseless computing

As an alternative to using external Poisson input, one can construct **ensembles of networks** where each neuron receives irregular input from neurons of adjacent networks (Fig. 12). This way, the **functional activity of each network can provide some of the noise required for computation by others.**

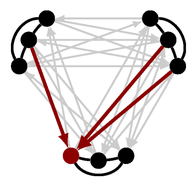

Fig. 12 *Neurons of adjacent networks provide noise to each other, replacing the Poisson sources. For instance, the red neuron receives noise from three adjacent neurons.*

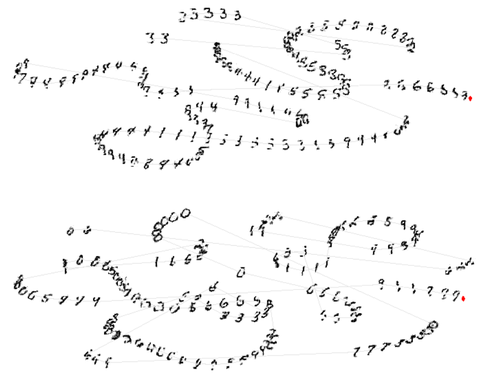

Fig. 13 *Hierarchical spiking networks generating handwritten digits without external noise. Only 2 out of 5 networks are shown here. The generated images are illustrated with t-SNE* [7].

Such networks are capable of **performing discriminative and generative tasks on the trained data spaces without external noise**. This is demonstrated in simulations (Fig. 13) for the case of five networks trained on the MNIST dataset [8].
The network parameters can either be **translated** from those of trained Boltzmann machines (Fig. 13), or can be **trained directly within the ensemble of networks** (Fig. 14).

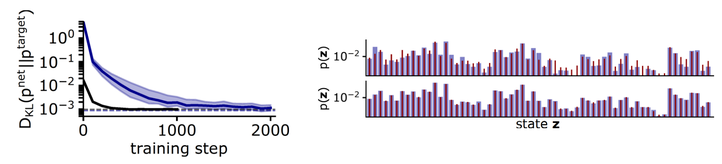

Fig. 14 *Training of an ensemble consisting of 100 6-neuron spiking networks that receive no external noise. Left: during training, the median D_{KL} (blue) approaches the one of networks with Poisson noise (black). Right: the sampled distribution (red) of a single network is compared to the target (blue) after 100 (top) and 2000 (bottom) training steps.*

Emulation rather than simulation

Instead of numerically integrating the ODEs that govern the dynamics of spiking networks, **BrainScaleS implements electric circuits** that obey these very equations. The faster time constants of these electric circuits lead to a **speedup of 10^4 over biological real time**, independent of the size of the emulated system.
Due to fixed-pattern noise on the manufactured transistors and the nature of the analogue parameter storage, **hardware settings vary both from component to component and from trial to trial**.

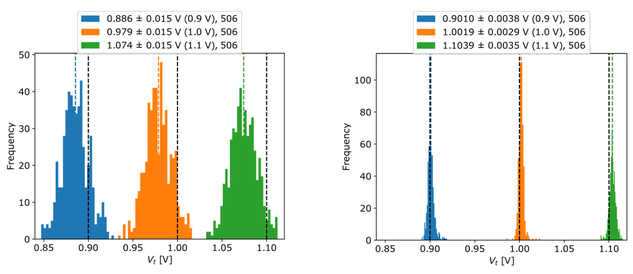

Fig. 5 *Calibration of the threshold potential (in hardware units) on a HICANN chip. Left: before calibration; right: after calibration. The black dashed lines represent the respective target values (with courtesy of A. Kugele).*

To compensate for variations between components, the relationship between neuron parameters, i.e., potentials and time constants, and hardware parameters is measured for every neuron [3].
This so-called **calibration** can reduce the component-to-component variation by more than one order of magnitude (Fig. 5). More than 12,000 adjustable analogue parameters per HICANN allow flexible control of the emulated network.

Note that **BrainScaleS is a continuously running system**, where the recorded start of the simulation is some arbitrary time after the configuration is completed. Aside from reconfigurations **the system continues to evolve according to its imprinted ODEs**.

Sampling on neuromorphic hardware

In order to circumvent the limited precision of analogue parameters (e.g., the leak potential E_l), biases are implemented as regular spike input. Additionally, to **preserve external bandwidth** (about 1.2kHz per HICANN), neurons with leak above threshold provide this input (Fig. 9). Synaptic weights are implemented as **4-bit digital values** driving analogue input circuitry. The response functions of sampling neurons are measured by varying the weight of the bias synapse (Fig. 10).

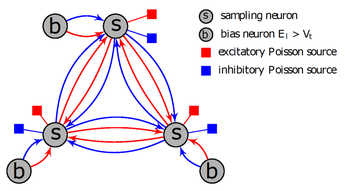

Fig. 9 *Illustration of the sampling network on hardware. To allow changes in synapse type during training, connections have to be implemented as excitatory (red) and inhibitory (blue) synapses.*

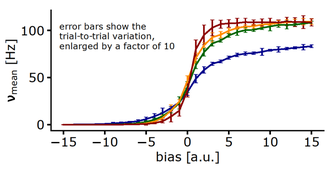

Fig. 10 *Response functions of four different hardware neurons, obtained by changing the weight of the bias neuron **b** to **s**.*

After training with a variant of the **wake-sleep algorithm** [6], the network approximately samples from the target distribution (Fig. 11). The final accuracy is limited by the 4-bit resolution of the weights.

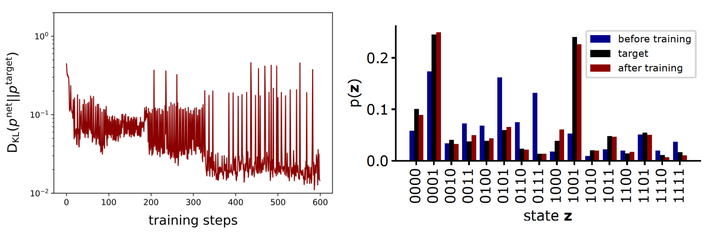

Fig. 11 *Training a network of four neurons on hardware towards a target Boltzmann distribution (right, black bars). During training, the D_{KL} between sampled and target distribution is reduced (left). After training, the network samples from a very good approximation of the sought target distribution.*

Autonomous hardware networks

First results on the BrainScaleS system demonstrate that ensembles of networks can be set up and trained on hardware **without any external noise**, reaching a similar performance as networks receiving independent Poisson noise (Fig. 9, 11). Here, we implemented a network of 15 Boltzmann machines, each consisting of 4 neurons.

After training, most networks are able to approximately sample from their target distribution, with some networks showing bad performance due to single hardware neuron deficiencies (Fig. 15).

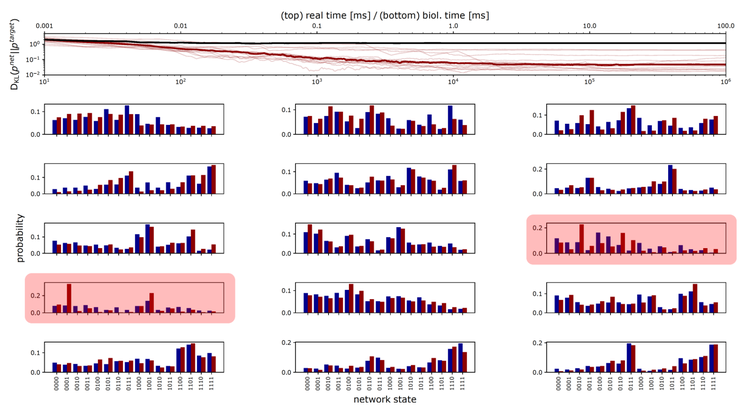

Fig. 15 *Sampling on hardware without external noise and post-training. Top: the median D_{KL} of the ensemble before (black) and after training (red) during a single network emulation. The opaque lines show the D_{KL} of the individual networks. Bottom: for every network, the sampled (red) and target (blue) distribution are shown.*

With a more carefully chosen network-to-hardware mapping, the incidence of such singular sources of disruption can be easily reduced. Once BrainScaleS becomes operational at full scale, **this approach will enable the emulation of large and computationally powerful spiking inference machines**.

This work has received funding from the European Union 7th Framework Programme under grant agreement 604102 (HBP), the Horizon 2020 Framework Programme under grant agreement 720270 (HBP) and the Manfred Stärk Foundation.

[1] J. Schemmel, D. Brüderle, A. Grübl, M. Hock, K. Meier, and S. Millner. A Wafer-Scale Neuromorphic Hardware System for Large-Scale Neural Modeling. Proceedings of the 2010 IEEE International Symposium on Circuits and Systems, 2010, 10.1109/ISCAS.2010.5536970
[2] R. Brette, W. Gerstner. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. Journal of Neurophysiology, 2005, 10.1152/jn.00686.2005
[3] C. Koke. Device Variability in Synapses of Neuromorphic Circuits, Dissertation, Heidelberg University, 2017
[4] M. A. Petrovici, J. Bill, I. Bytschok, J. Schemmel, and K. Meier. Stochastic inference with spiking neurons in the high-conductance state. Physical Review E, 2016, 10.1103/PhysRevE.94.042312
[5] M. A. Petrovici, J. Bill, I. Bytschok, J. Schemmel, and K. Meier. The high-conductance state enables neural sampling in networks of LIF neurons. In BMC Neuroscience 2015, 2015, 10.1186/1471-2202-16-S1-O2
[6] D. H. Ackley, G. Hinton, T. J. Sejnowski. A learning algorithm for Boltzmann machines. Cognitive Science, 1985, 10.1207/s15516709cog0901_7
[7] L. Maaten, G. Hinton. Visualizing data using t-SNE. Journal of Machine Learning Research, 2008
[8] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, 10.1109/5.726791

Email contacts: {dodo, fkungl, andreas.baumbach, kljohann}@kip.uni-heidelberg.de

hardware guidebook

publications