

A neuromorphic network for generic multivariate data classification

Michael Schmuker^{a,b,1}, Thomas Pfeil^c, and Martin Paul Nawrot^{a,b}

^aNeuroinformatics and Theoretical Neuroscience, Institute for Biology, Department of Biology Chemistry and Pharmacy, Freie Universität Berlin, 14195 Berlin, Germany; ^bBernstein Center for Computational Neuroscience Berlin, 10119 Berlin, Germany; and ^cKirchhoff-Institute for Physics, Heidelberg University, 69120 Heidelberg, Germany

Edited by Terrence J. Sejnowski, Salk Institute for Biological Studies, La Jolla, CA, and approved December 23, 2013 (received for review February 20, 2013)

Computational neuroscience has uncovered a number of computational principles used by nervous systems. At the same time, neuromorphic hardware has matured to a state where fast silicon implementations of complex neural networks have become feasible. En route to future technical applications of neuromorphic computing the current challenge lies in the identification and implementation of functional brain algorithms. Taking inspiration from the olfactory system of insects, we constructed a spiking neural network for the classification of multivariate data, a common problem in signal and data analysis. In this model, real-valued multivariate data are converted into spike trains using “virtual receptors” (VRs). Their output is processed by lateral inhibition and drives a winner-take-all circuit that supports supervised learning. VRs are conveniently implemented in software, whereas the lateral inhibition and classification stages run on accelerated neuromorphic hardware. When trained and tested on real-world datasets, we find that the classification performance is on par with a naïve Bayes classifier. An analysis of the network dynamics shows that stable decisions in output neuron populations are reached within less than 100 ms of biological time, matching the time-to-decision reported for the insect nervous system. Through leveraging a population code, the network tolerates the variability of neuronal transfer functions and trial-to-trial variation that is inevitably present on the hardware system. Our work provides a proof of principle for the successful implementation of a functional spiking neural network on a configurable neuromorphic hardware system that can readily be applied to real-world computing problems.

bioinspired computing | spiking networks | machine learning | multivariate classification

The remarkable sensory and behavioral capabilities of all higher organisms are provided by the network of neurons in their nervous systems. The computing principles of the brain have inspired many powerful algorithms for data processing, most importantly the perceptron and, building on top of that, multilayer artificial neural networks, which are being applied with great success to various data analysis problems (1). Although these networks operate with continuous values, computation in biological neuronal networks relies on the exchange of action potentials, or “spikes.”

Simulating networks of spiking neurons with software tools is computationally intensive, imposing limits to the duration of simulations and maximum network size. To overcome this limitation, several groups around the world have started to develop hardware realizations of spiking neuron models and neuronal networks (2–10) for studying the behavior of biological networks (11). The approach of the *Spikey* hardware system used in the present study is to enable high-throughput network simulations by speeding up computation by a factor of 10^4 compared with biological real time (12, 13). It has been developed as a reconfigurable multineuron computing substrate supporting a wide range of network topologies (14).

In addition to providing faster tools for neurosimulation, high-throughput spiking network computation in hardware offers the possibility of using spiking networks to solve real-world

computational problems. The massive parallelism is a potential advantage over conventional computing when processing large amounts of data in parallel. However, conventional algorithms are often difficult to implement using spiking networks for which many neuromorphic hardware substrates are designed. Novel algorithms have to be designed that embrace the inherent parallelism of a brain-like computing architecture.

A common problem in data analysis is classification of multivariate data. Many problems in artificial intelligence relate to classification in some way or the other, such as object recognition or decision making. It is the basis for data mining and, as such, has widespread applications in industry. We interact with classification systems in many aspects of daily life, for example in the form of Web shop recommendations, driver assistance systems, or when sending a letter with a handwritten address that is deciphered automatically in the post office.

In this work, we present a neuromorphic network for supervised classification of multivariate data. We implemented the spiking network part on a neuromorphic hardware system. Using a range of datasets, we demonstrate how the classifier network supports nonlinear separation through encoding by virtual receptors, whereas lateral inhibition transforms the input data into a sparser encoding that is better suited for learning.

Results

We first outline our spiking neural network design and show examples of the network activity during operation in supervised classification of multivariate data. Then we analyze the temporal dynamics of the classification process and compare the network classification performance against the performance of a naïve Bayes (NB) classifier. We show that the network tolerates the

Significance

One primary goal of computational neuroscience is to uncover fundamental principles of computations that are performed by the brain. In our work, we took direct inspiration from biology for a technical application of brain-like processing. We make use of neuromorphic hardware—electronic versions of neurons and synapses on a microchip—to implement a neural network inspired by the sensory processing architecture of the nervous system of insects. We demonstrate that this neuromorphic network achieves classification of generic multidimensional data—a widespread problem with many technical applications. Our work provides a proof of concept for using analog electronic microcircuits mimicking neurons to perform real-world computing tasks, and it describes the benefits and challenges of the neuromorphic approach.

Author contributions: M.S. designed research; M.S. and T.P. performed research; M.S. analyzed data; and M.S., T.P., and M.P.N. wrote the paper.

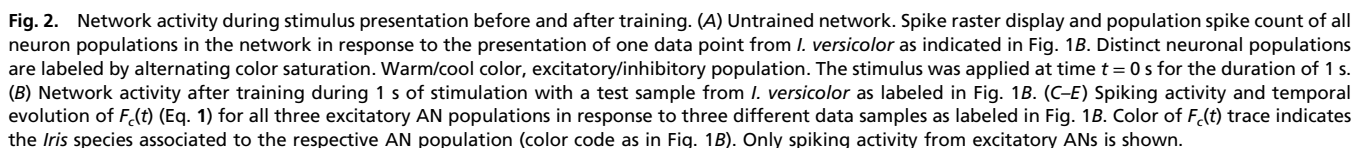
The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

Freely available online through the PNAS open access option.

¹To whom correspondence should be addressed. E-mail: m.schmuker@fu-berlin.de.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1303053111/-DCSupplemental.



separate *I. setosa*. This observation indicates that the classifier network is capable of delivering reliable classification not only of well-separable data, but also in cases where samples from different classes overlap in feature space.

Tolerance Against Neuronal Variability. The analog circuits used to represent neurons in the *Spikey* system exhibit inherent variability that the classifier network must tolerate to be useful in practice. Two sources of variability on the hardware system can be distinguished: “temporal noise” and “fixed-pattern noise.” Temporal noise (including thermal noise and other sources of stochastic variability) affects the circuits on short timescales in an unpredictable fashion. In contrast, fixed-pattern noise is caused by device mismatch. Device mismatch describes the deviance of an electronic component from its specification due to inevitable variations in the manufacturing process. The variations of neuron parameters due to device mismatch occur on much slower timescales and can be regarded as constant for our use case. They introduce heterogeneity across all analog components—neurons and synapses—according to a fixed pattern (hence the term “fixed-pattern noise”). The individual variation can be measured and calibrated for. The integrated development environment of the *Spikey* system contains calibration methods that reduce the amount of fixed-pattern variability. However, such generic calibration methods cannot account for all network configurations in an efficient manner, because calibration at the neuron level does not take into account network effects. This is particularly relevant for the *Spikey* system, which was designed to accommodate a wide variety of network topologies (14). In our case, the fixed-pattern variation that remains after built-in calibration manifests itself in variability of the neurons’ transfer functions that relate input rate to output rate. Both maximal output rate and slope of the transfer functions varied considerably across PNs and LNs (Fig. 4A).

Due to its stochastic nature, temporal noise cannot be avoided by systematic measures such as calibration of synaptic weights. We quantified the variation in spike count caused by temporal noise by measuring the variability of the spike count in all 192 hardware neurons across 50 repetitions with identical stimuli. For this purpose, we generated input spike trains only once and used them repeatedly as input to all 192 neurons (“frozen input”). We used six gamma processes of order five and mean rate of 25 spikes/s to mimic the inputs that PNs receive in the classifier network. We adjusted the weights of the neurons to yield a mean output frequency of 25.4 spikes/s. The neurons exhibited moderate trial-to-trial variability under these conditions. Fig. 4B shows the distribution of spike counts for one exemplary neuron that produced 25.3 spikes on average, with a variance of 1.0. This

amount of variability is also reflected when considering the total population (Fig. 4C). On average, the individual spike trains from the same neuron varied with a Fano factor (28) of 0.083, which is smaller than the variability inherent to the gamma process used for the generation of the RN spike trains ($\gamma = 5$, Fano factor = 0.2). Thus, the trial-to-trial variability due to temporal noise intrinsic to the neuromorphic hardware is small compared with those variations imposed by the biologically realistic stochastic generation of input spike trains.

The classifier network achieved the reported performance despite transfer function variability caused by fixed-pattern noise and trial-to-trial variability caused by temporal noise and by the stochasticity of the input. This robustness is the result of considerable efforts to optimize network topology. Essentially, the key to achieve robustness in our network was to leverage population coding. Two network properties proved essential to ensure a valid population code. First, synchronization of neurons within a population should be avoided because it violates the rate code assumption of independent neurons within each population. We achieved this by sparsifying the input to individual neurons, i.e., using 50% connection probability instead of full connectivity. Second, population sizes must be sufficiently large to reduce the variance of the population transfer function. We provide a detailed explanation of how these properties affect network operation in *SI Results* (Figs. S1 and S2, and *SI Results, Network Optimization for Robustness Against Neuronal Variability*).

General Applicability to Other Datasets. As a demonstration for the ability of the network to solve nonlinear problems, we applied the network to classification of a 2D “Ring” dataset. This simple dataset consists of two classes, one class situated in a cluster centered at the origin and a second class surrounding it (Fig. 5A). It has skewed class proportions with sevenfold more data points in the surround than in the center class. In addition, the arrangement of data points requires a nonlinear separation between the center and surround classes. Our network achieves this separation through the VR trick: By using 10 VRs to represent a 2D dataset, we transform the data into a higher-dimensional space in which linear separation is possible. The classifier network running on the *Spikey* system achieved an average performance of $R_K = 0.96$ on the Ring dataset (NB: $R_K = 0.98$; Fig. 5C, Left).

The Mixed National Institute of Standards and Technology (MNIST) database is a commonly used high-dimensional benchmark problem with practical relevance (<http://yann.lecun.com/exdb/mnist/>). The database contains images of handwritten digits from 0 to 9, digitized to 28×28 pixels. Hence, each observation has $28 \times 28 = 768$ dimensions. The dataset is divided into a training and a test set to enable reproducible benchmarking. We picked a subset of this dataset consisting of the digits “5” and “7,” using 2,000 samples from the training set and 1,920 samples from the test set (Fig. 5B). On the MNIST dataset, the spiking network outperformed the NB classifier by a large margin (hardware network: mean $R_K = 0.94$; NB: 0.82; Fig. 5C, Right). Interestingly, when training the NB classifier on the spike counts produced by PNs in the network (that is, after the lateral inhibition stage in the decorrelation layer), its performance increases to similar levels as obtained with the classifier network (mean $R_K = 0.96$). This observation is in line with a previous study which demonstrated that lateral inhibition increases classifier performance on a 184-dimensional odor dataset (17).

The reason for this effect lies in the fact that lateral inhibition transforms the broad, overlapping receptive fields of VRs (and in consequence RNs) into more localized representations of input space. In other words, receptive fields of PNs are narrower than those of VRs, and they overlap to a lesser degree. As a result, PN activity is also sparser than VR activity, that is, only few PN populations respond to a particular stimulus. This behavior can be observed, for example, when comparing the spike counts of RNs and PNs in Fig. 2A and B. Sparser activity and more local receptive fields simplify the training process, because it

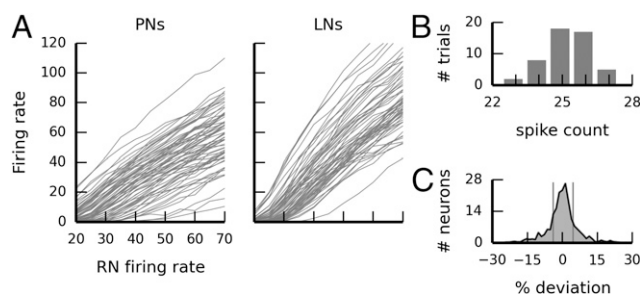
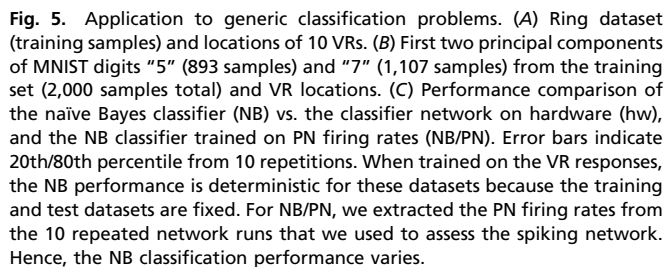


Fig. 4. Neuromorphic hardware variability. (A) Variability of transfer functions across all PNs and LNs on the hardware chip. Each line corresponds to one neuron. (B) Hardware trial-to-trial variability: Histogram of spike counts for one example neuron across 50 repeated stimulations with identical “frozen input” of 1-s duration (average spike count, 25.3; variance, 1.0). (C) Histogram of per-neuron spike counts relative to the individual average spike count emitted by each of the 192 neurons across 50 repeated identical 1-s stimulations. The vertical lines indicate 20th/80th percentile ($P^{20} = 3.99$, $P^{80} = 4.58$).



Speed Considerations. The major advantage in using accelerated neuromorphic hardware for spiking neuronal simulations is its potentially fast execution time. On the neuromorphic hardware system used in this study, simulations run with a speedup factor of 10^4 . Hence, presenting all 150 iris data points for 1 s (biological time) each to the hardware network takes $150 \text{ s}/10^4 = 15 \text{ ms}$ pure network run time. Practical applications require data transfer for spikes and synaptic weights to and from the system as well as the parameterization of the hardware network, which adds to the pure network run time (for details, see [SI Materials and Methods](#) and [Fig. S4](#)). These factors depend on the efficiency of the software interface for the hardware system. Because we are working with a prototype setup, its interface is under constant development and improvement. At the time of writing, the hardware system effectively achieved an overall 13-fold speedup compared with biological real time. We want to stress that this number may improve as the software interface is continuously optimized.

We demonstrated the implementation of a spiking neuronal network for classification of multidimensional data on a neuromorphic hardware system. The network is capable of separating data in a nonlinear fashion through encoding by VRs. The transformation by lateral inhibition increases classification performance. It performed robustly in the presence of stochastic trial-to-trial variability inherent to the hardware system. The network is not restricted to any specific kind of data, but is capable of classifying arbitrary real-valued, multidimensional data, and hence universally suited for all kinds of classification tasks. It achieved performance values comparable to a standard machine learning classifier, which points out the network's wide applicability to real-world problems. The present network implementation is a proof of concept that can serve as a building block for classifier tasks on neuromorphic hardware. Together with the high speedup factor of the neuromorphic hardware system, our universal classification network is an important step toward high-performance neurocomputing.

separation is possible. The network we presented contains a linear classifier, with the additional constraint that the separating hyperplane must pass through the origin (29). As such, it is limited to separating linear problems. We overcame this limitation through the VR approach, which provides a higher-dimensional representation of the data. Our results on the MNIST dataset point out that the lateral inhibition step is crucial for successful classification of real-world, high-dimensional datasets. Although more complex machine learning algorithms like support vector machines or restricted Boltzmann machines may allow for better classification performance directly on the VR data, the strength of our approach lies in the simplicity of a linear classifier combined with appropriate filtering of input data through the lateral inhibition step, which is very efficiently carried out in a massively parallel neuromorphic hardware network.

Lateral inhibition provides a soft partitioning of input space that facilitates classifier training. Note that this circumstance also points out a limitation of the presented classifier network, because class boundaries in data space can only be optimally represented if they coincide with partition borders. A straightforward way to deal with this problem is to increase the number of VRs and glomeruli, resulting in a more fine-grained partitioning of data space. Such an approach will be possible using emerging large-scale neuro-morphic hardware systems supporting tens of thousands of neurons (8, 30).

VRs depend on a self-organizing process that is trained in data space. A particularly interesting prospect is to implement this process on the neuronal substrate. Spiking self-organizing maps have been described in the literature (31–33), suggesting that, in principle, it is possible to implement a self-organizing process on a neuromorphic hardware system. However, the learning rule used in these studies would require sophisticated control logic, which makes it difficult to implement them on the *Spikey* system. A more straightforward and mathematically well-founded approach has recently been put forward by Nessler et al. (34). They suggested a probabilistic, self-organizing mechanism to learn prototypes in feature space using spike timing-dependent plasticity (STDP) and a winner-take-all circuit, which is suited to represent the VR encoding. An integrated implementation of this encoding together with the classifier network we present here will likely require a much higher neuron count and more flexible plasticity mechanisms compared with what is available on the *Spikey* system (13). On-chip implementations may become feasible considering the BrainScaleS wafer-scale hardware system that extends the number of available neurons by up to several orders of magnitude and provides more sophisticated plasticity mechanisms (35, 36). In that system, multiple identical neuromorphic modules may be implemented on a single silicon wafer and communicate through high-bandwidth connections. Moreover, advanced control logic for on-chip implementation of elaborate STDP rules is under development (36), which is designed to be compatible with the self-organized prototype learning mechanisms described by Nessler et al. (34). In addition, the deterministic connectivity structure of the glomerular classifier network presented here facilitates splitting the network across different neuromorphic modules. The increased neuron count available in a large-scale system would allow for a larger number of VRs to solve more complex problems and enables scaling the network to larger population sizes to support robustness against noise.

Analysis of the dynamic network activation in response to the onset of a stimulus presentation revealed a fast decision time where the average performance reached its maximum within less than 100 ms in biological time (Fig. 34). This is in good agreement with recent measurements in insects. In the honeybee, a prominent animal model for studying learning and memory, it was shown that the encoding of the identity of an olfactory stimulus at the level of PNs evolved rapidly within tens of milliseconds (37, 38). Neuronal populations at the output of the mushroom body encode odor–reward associations. These neuronal populations fulfill a similar function like the ANs in our network. In a classical conditioning paradigm, they indicated the

classification of the conditioned stimulus (an odor that was previously paired with a sugar reward) within less than 200 ms (39).

Our network proved to be robust against neuronal variability, which is an important factor in the design of neuromorphic algorithms. Biological neuronal networks face a similar challenge. The study of neuronal variance is an integral part of today's neuroscience ever since the seminal study by Mainen and Sejnowski (40). Many neural properties are stochastic in nature, like neurotransmitter release or spike initiation, so a certain amount of variability is inevitable in biological neuronal networks (41). In the same vein, the analog nature of the circuits in the hardware enables the massive speedup and integration density, but unavoidably entails variability. In our case, we achieved tolerance against variability by using a population code. Generally, accelerated analog neurocomputing requires models that can cope with and, ideally, make use of variability. The design of these models will benefit greatly from a deep understanding of biological circuits, interpreted in the light of variability. Likewise, creating functional networks on an analog neuromorphic substrate provides insight into critical properties that networks must possess to operate under noisy conditions.

Materials and Methods

Stimuli were presented to the classifier network in a sequential manner. For each stimulus i , the corresponding feature vector \mathbf{x}_i was obtained from the observation matrix \mathbf{X} , converted into a firing-rate presentation with VRs, from which spike trains were generated by a gamma point process (42). Each

stimulus was presented for 1 s of biological time. Synaptic weights that fulfilled a Hebbian eligibility constraint were updated after each stimulus presentation. A synaptic weight was eligible for updating if the target neuron was a member of the winner population, and if the spike count emitted by the presynaptic neuron during the previous stimulus presentation exceeded a threshold (fixed to 35 spikes in the 1-s stimulus interval). Eligible synapses were potentiated by a fixed amount if classification was correct, or depressed by a fixed amount if classification was incorrect. A formal description of the training algorithm is available in *SI Materials and Methods, Network Training and Supervised Learning Rule*.

Network training was implemented in an interactive chip-in-the-loop fashion: Stimuli were processed by the network on the chip. After each stimulus, the network response was evaluated on the host computer where the weight changes are calculated. The network was then reconfigured and the next stimulus presented.

ACKNOWLEDGMENTS. We are deeply indebted to Daniel Brüderle for his support in using the *Spike* system, and we thank Eric Müller for technical assistance and Mihai A. Petrovici and Emre Neftci for comments and discussion. We also thank the two anonymous reviewers for insightful comments that stimulated new experiments that led to great improvements of the manuscript. This work was supported by Deutsche Forschungsgemeinschaft Grants SCHM2474/1-1 and 1-2 (to M.S.), a grant from the German Ministry of Education and Research [Bundesministerium für Bildung und Forschung (BMBF)], Grant 01GQ1001D [to M.S. (Bernstein Center for Computational Neuroscience Berlin, Project A7) and M.P.N. (Project A9)], and a grant from BMBF [Bernstein Focus Neuronal Learning: Insect Inspired Robots; Grant 01GQ0941 (to M.P.N.)]. T.P. has received funding by the European Union Seventh Framework Programme under Grant 243914 (Brain-i-Nets) and Grant 269921 (BrainScale5).

- Hertz JA, Krogh AS, Palmer RG (1991) *Introduction to the Theory of Neural Computation* (Addison-Wesley, Redwood City, CA), Vol 1.
- Mahowald M, Douglas RJ (1991) A silicon neuron. *Nature* 354(6354):515–518.
- Indiveri G, Chicca E, Douglas R (2009) Artificial cognitive systems: From VLSI networks of spiking neurons to neuromorphic cognition. *Cognit Comput* 1:119–127.
- Yu T, Sejnowski TJ, Cauwenberghs G (2011) Biophysical neural spiking, bursting, and excitability dynamics in reconfigurable analog VLSI. *IEEE Trans Biomed Circuits Syst* 5(5):420–429.
- Indiveri G, et al. (2011) Neuromorphic silicon neuron circuits. *Front Neurosci* 5:73.
- Renaud S, et al. (2010) PAX: A mixed hardware/software simulation platform for spiking neural networks. *Neural Netw* 23(7):905–916.
- Brüderle D, et al. (2011) A comprehensive workflow for general-purpose neural modeling with highly configurable neuromorphic hardware systems. *Biol Cybern* 104(4-5):263–296.
- Furber SB, et al. (2013) Overview of the SpiNNaker System Architecture. *IEEE Trans Comput* 62(12):2454–2467.
- Choudhary S, et al. (2012) Silicon neurons that compute. *Artificial Neural Networks and Machine Learning—ICANN 2012*, eds Villa AEP, Duch W, Érdi P, Masulli F, Palm G (Springer, Berlin), pp 121–128.
- Merolla P, et al. (2011) A digital neurosynaptic core using embedded crossbar memory with 45pJ per spike in 45nm. *Custom Integrated Circuits Conference (CICC), 2011 IEEE* (IEEE, Piscataway, NJ), pp 1–4.
- Imam N, et al. (2012) Implementation of olfactory bulb glomerular-layer computations in a digital neurosynaptic core. *Front Neurosci* 6:83.
- Brüderle D, et al. (2010) Simulator-like exploration of cortical network architectures with a mixed-signal VLSI system. *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCCS)* (IEEE, Piscataway, NJ), pp 2784–2787.
- Schemmel J, Gruebl A, Meier K, Mueller E (2006) Implementing synaptic plasticity in a VLSI spiking neural network model. *Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN)* (IEEE, Vancouver), pp 1–6.
- Pfeil T, et al. (2013) Six networks on a universal neuromorphic computing substrate. *Front Neurosci* 7:11.
- Davison AP, et al. (2008) PyNN: A common interface for neuronal network simulators. *Front Neuroinform* 2:11.
- Brüderle D, et al. (2009) Establishing a novel modeling tool: A Python-based interface for a neuromorphic hardware system. *Front Neuroinform* 3:17.
- Schmuker M, Schneider G (2007) Processing and classification of chemical data inspired by insect olfaction. *Proc Natl Acad Sci USA* 104(51):20285–20289.
- Huerta R, Nowotny T, García-Sánchez M, Abarbanel HDI, Rabinovich MI (2004) Learning classification in the olfactory system of insects. *Neural Comput* 16(8):1601–1640.
- Huerta R, Nowotny T (2009) Fast and robust learning by reinforcement signals: Explorations in the insect brain. *Neural Comput* 21(8):2123–2151.
- Martinetz T, Schulten K (1991) A “neural-gas” network learns topologies. *Artificial Neural Networks*, eds Kohonen T, Mäkiäsa K, Simula O, Kangas J (Elsevier B.V., North-Holland, Amsterdam), pp 397–402.
- Linstner C, Sachse S, Galizia CG (2005) Computational modeling suggests that response properties rather than spatial position determine connectivity between olfactory glomeruli. *J Neurophysiol* 93(6):3410–3417.
- Wick SD, Wiechert MT, Friedrich RW, Riecke H (2010) Pattern orthogonalization via channel decorrelation by adaptive networks. *J Comput Neurosci* 28(1):29–45.
- Schmuker M, Yamagata N, Nawrot MP, Menzel R (2011) Parallel representation of stimulus identity and intensity in a dual pathway model inspired by the olfactory system of the honeybee. *Front Neuroeng* 4:17.
- Brader JM, Senn W, Fusi S (2007) Learning real-world stimuli in a neural network with spike-driven synaptic dynamics. *Neural Comput* 19(11):2881–2912.
- Soltani A, Wang X-J (2010) Synaptic computation underlying probabilistic inference. *Nat Neurosci* 13(1):112–119.
- Fisher R (1936) The use of multiple measurements in taxonomic problems. *Ann Eugen* 7(2):179–188.
- Gorodkin J (2004) Comparing two K-category assignments by a K-category correlation coefficient. *Comput Biol Chem* 28(5-6):367–374.
- Nawrot MP, et al. (2008) Measurement of variability dynamics in cortical spike trains. *J Neurosci Methods* 169(2):374–390.
- Häusler C, Nawrot MP, Schmuker M (2011) A spiking neuron classifier network with a deep architecture inspired by the olfactory system of the honeybee. *2011 5th International IEEE/EMBS Conference on Neural Engineering (IEEE, Piscataway, NJ)*, pp 198–202.
- Schemmel J, et al. (2010) A wafer-scale neuromorphic hardware system for large-scale neural modeling. *Proceedings of the 2010 International Symposium on Circuits and Systems (ISCAS)* (IEEE, Paris), pp 1947–1950.
- Ruf B, Schmitt M (1998) Self-organization of spiking neurons using action potential timing. *IEEE Trans Neural Netw* 9(3):575–578.
- Choe Y, Mikkulainen R (1998) Self-organization and segmentation in a laterally connected orientation map of spiking neurons. *Neurocomputing* 21(1-3):139–158.
- Behi T, Arous N (2011) Word recognition in continuous speech and speaker independent by means of recurrent self-organizing spiking neurons. *Signal Processing* 5(5):215–226.
- Nessler B, Pfeiffer M, Buesing L, Maass W (2013) Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. *PLoS Comput Biol* 9(4):e1003037.
- Pfeil T, et al. (2012) Is a 4-bit synaptic weight resolution enough?—constraints on enabling spike-timing dependent plasticity in neuromorphic hardware. *Front Neurosci* 6:90.
- Friedmann S, Frémaux N, Schemmel J, Gerstner W, Meier K (2013) Reward-based learning under hardware constraints—using a RISC processor embedded in a neuromorphic substrate. *Front Neurosci* 7:160.
- Krofczik S, Menzel R, Nawrot MP (2008) Rapid odor processing in the honeybee antennal lobe network. *Front Comput Neurosci* 2:9.
- Strube-Bloss MF, Herrera-Valdez MA, Smith BH (2012) Ensemble response in mushroom body output neurons of the honey bee outpaces spatiotemporal odor processing two synapses earlier in the antennal lobe. *PLoS One* 7(11):e50322.
- Strube-Bloss MF, Nawrot MP, Menzel R (2011) Mushroom body output neurons encode odor-reward associations. *J Neurosci* 31(8):3129–3140.
- Mainen ZF, Sejnowski TJ (1995) Reliability of spike timing in neocortical neurons. *Science* 268(5216):1503–1506.
- Boucsein C, Nawrot MP, Schnepel P, Aertsen A (2011) Beyond the cortical column: Abundance and physiology of horizontal connections imply a strong role for inputs from the surround. *Front Neurosci* 5:32.
- Tuckwell HC (1988) *Introduction to Theoretical Neurobiology: Volume 2, Nonlinear and Stochastic Theories* (Cambridge Univ Press, Cambridge, UK).