Predicting voluntary movements from motor cortical activity with neuromorphic hardware

Iulia-Alexandra Lungu, Alexa Riehle, Martin Paul Nawrot*, Michael Schmuker*
* corresponding authors.

Keywords (for review): Brain-Machine Interfacing, Neuromorphic Hardware, Primate Motor Cortex, Spiking Neural Network

Abstract

Neurons in mammalian motor cortex encode physical parameters of voluntary movements during planning and execution of a motor task. Brain-machine interfaces can decode limb movements from the activity of these neurons in real time. The future goal is to control prosthetic devices in severely paralyzed patients or to restore communication if the ability to speak or make gestures is lost. Here, we implemented a spiking neural network that decodes movement intentions from the activity of individual neurons recorded in the motor cortex of a monkey. The network runs on neuromorphic hardware and performs its computations in a purely spike-based fashion. It incorporates an insect-brain-inspired, three-layer architecture with 176 neurons. Cortical signals are filtered using lateral inhibition, and the network is trained in a supervised fashion to predict two opposing directions of the monkey's arm reaching movement before the movement is carried out. Our network operates on the actual spikes that have been emitted by motor cortical neurons, without the need to construct intermediate non-spiking representations. Using a pseudo-population of 12 manually-selected neurons, it reliably predicts the movement direction with an accuracy of 89.32 % on unseen data after only 100 training trials. Our results provide a proof of concept for the first-time use of a neuromorphic device for decoding movement intentions.

Introduction

Brain-machine interfacing (BMI) refers to a neuro-engineering approach that couples brain activity to external devices. Individual cortical neurons in the motor and premotor areas of the primate neocortex accurately encode the initiation and kinematic parameters (e.g. speed and direction) of voluntary limb movements, and these signals can be recorded intra-cranially (i.e. from inside the skull) using invasive approaches [1]–[6]. Previous studies have shown that decoding this information efficiently allows for the real-time control of a computer screen cursor or technical devices with several degrees of freedom in humans [7], [8] and non-human primates [9]–[12]. Some of the possible medical applications of brain-computer interfacing are the control of prosthetic devices or the restoration of communication in severely paralyzed patients [13].

There are two distinct classes of invasive brain-machine interfacing approaches. The first uses composite neurophysiological signals such as local or epicortical field potentials that represent the spatially averaged synaptic input to neurons within a small volume around the electrode tip [3], [14]–[18]. We focus here on the second class that interfaces with extracellularly recorded action potentials (spikes) of multiple single neurons. The state of the art is to record from several tens to hundreds of single units simultaneously from the mammalian cortex (e.g. [19]). Virtually all current brain machine interfacing approaches do not operate on the recorded spikes, but rather transform these time series of discrete events into a time-averaged, continuous signal, i.e. neuronal firing rates. Conventional decoding algorithms operating on rate signals have employed model-based decoding (e.g. [4], [10], [20]), spatio-temporal filtering (e.g. [11]), or machine learning

methods for classification (e.g. [3], [5], [16]). Yet, since communication in the brain is performed via discrete events, so-called *spikes*, it is intuitive to use the same protocol in a realistic closed-loop brain-computer interfacing system. The most appropriate classifier to be used under such a paradigm is a spiking neural network, which draws inspiration from biological brains and uses spikes to relay information between its neurons (units). The use of a spiking neural network decoder has previously been suggested by Dethier and colleagues [21], [22], who have successfully simulated a spiking neural network that mimicked a Kalman filter to decode firing rates of multi-unit activity in a closed-loop scenario.

Simulating spiking networks on conventional computers comes with substantial computational overhead that annihilates a large portion of the conceptual benefit of coherently using a spiking representation in the BMI scenario. A more effective solution is provided by so-called neuromorphic hardware, which has been developed in recent years to overcome the computational penalty associated with neuronal simulations. These hardware systems achieve a speedup e.g. by using analog circuits to accelerate neuronal computations, or dedicated digital circuits that support massively parallel data exchange in a brain-inspired fashion, or a combination of both. Several platforms for neuromorphic computing have emerged, starting with the pioneering work of Carver Mead [23], up to rather recent developments like e.g. SpiNNaker [24], NeuroGrid [25], ROLLS [26], IBM's TrueNorth [27], or the systems developed at the University of Heidelberg [28], [29] (see [30] for a review). Aside from testing principles of neural computation [28], [31], [32], initial applications of neuromorphic hardware have been demonstrated for generic pattern recognition [27], [33]. These applications highlight the potential of this new technology to solve various widely investigated computing problems. Moreover, since neuromorphic architectures differ fundamentally from conventional computers, they are thought of as one of the potential solutions to upcoming challenges in computing as the integration density of digital processors will cease to grow – commonly termed "the end of Moore's law" [34].

Certain neuromorphic platforms are geared towards lower power consumption and/or more compact size than conventional computing solutions for spiking networks, e.g. Spikey, ROLLS, or the small (4-chip) SpiNNaker systems. Therefore, they are in principle well suited for embedded and portable applications. Despite these advantages, the use of spiking neuromorphic systems to decode single neuron activity in real-time has only rarely been considered in the literature (i.e., in [22]), and, to our knowledge, no actual implementation has yet been published.

Here, we propose and test an approach to decode and predict voluntary movements from motor cortical signals that uses a spiking neural network implemented on the Spikey neuromorphic hardware. Since all computations are carried out with spiking neurons, our on-chip network does not require an intermediate rate-based representation, and it operates completely on spike events.

This paper is organized as follows: the methods part describes the Spikey neuromorphic platform, the experimental paradigm in which the neuronal data was recorded, the architecture of the spiking neural network, and our process for training and testing the network. In the results section we show the time-resolved generalization performance of the spiking network, trained and tested on single unit activity recorded from the primary cortex of a monkey performing a delayed arm-reaching task [5], [35]. Finally, we discuss our approach in comparison with previous efforts that have suggested spiking or artificial neural networks for the decoding of motor cortical activity and possible future steps.

Methodological background

The Spikey neuromorphic hardware system

We used the Spikey neuromorphic system that has been developed at the Kirchhoff-Institute for Physics at Heidelberg University (Germany) [28]. Spikey contains a mixed-signal neuromorphic chip with analog neuron circuits and digital routing for spike events. Spikey's distinctive feature is the high speedup factor of 10^4 compared to biological real time. That is, a network that runs on Spikey for 1 s wall-clock time will have performed 10,000 s of spike-based computing in that time. Since Spikey also supports on-chip short-term plasticity (STP) and spike-timing dependent synaptic plasticity (STDP), it is predestined for high-performance spike-based computing. Spikey is the predecessor of a large, wafer-scale system currently developed in Heidelberg [29], which is aimed at large-scale neuromorphic computing for accelerated biological simulation and data mining.

Figure 1 shows a schematic overview of the Spikey system. The neuromorphic chip hosts analog neuron models (leaky integrate-and-fire with conductance-based synapses). A detailed account of the operation of Spikey (including several examples) is provided in [28]. Therefore, we only provide an overview on a typical workflow for running a neuromorphic network on Spikey.

**** FIGURE **1** ABOUT HERE.

On the host computer, the network is defined using PyNN, a Python library for spiking network modelling. The communication with the neuromorphic system is achieved using custom control software that converts the network specification and input data into an appropriate format and sends it to the system.

Since the bandwidth of the interface between host and Spikey is limited, the speed-up factor of 10^4 over real-time complicates synchronous data exchange. Therefore, Spikey operates in an asynchronous fashion, meaning, the network to be run is prepared on the host computer, then uploaded to spikey for emulation. After the emulation has finished, the results of the emulation are transferred back to the host computer.

Spikes are relayed to Spikey as time-stamped events, where they are cached in local random-access memory (RAM) before the network emulation is started. The RAM also holds configuration data. On the digital part of the chip, spikes are processed in discrete time steps and transformed into digital pulses. Input spikes are supplied by custom logic implemented in a field programmable gate array (FPGA), which also provides configuration data like analog parameter voltages, that are applied through digital-to-analog converters (DACs). Input spike trains are represented as external spike sources and not as actual neurons, unlike the rest of the network neurons which are built as analog circuits in silicon.

During the emulation, propagation of pulses between on-chip network neurons happens in continuous time, at the accelerated rate 10^4 times faster than real time. Spikes produced during the emulation are stored in RAM, as well as selected voltages that are read from the chip by analog-to-digital converters (ADCs). When the network emulation is finished, the control software retrieves the recorded data (e.g. spikes, analog voltages) and makes it available on the host for analysis.

Experimental paradigm

Our study is based on experimental data recorded from the motor cortex of one male monkey

(*Macaca mulatta*) performing a delayed arm-reaching task. Experimental details are provided in [5]. The animal was seated in front of a vertical panel displaying seven light-emitting touch sensitive buttons: a central one and six others positioned around it in a circle (arrangement shown in **Figure 2**). The trial started (TS) when the central position lit up and the animal placed its right arm at this location. After a delay of 500 ms, either one, two or three adjacent target buttons lit up in green (preparatory signal, PS). After a one-second delay, only one of the green light-emitting diodes (LEDs) turned red (response signal, RS), signaling the monkey to leave the central position and reach out to touch the red target. The time point of movement onset (MO) was recorded. The time interval between PS and RS will henceforth be referred to as the preparatory phase. After RS, the execution phase followed, which corresponds to the monkey moving its arm to perform the task.

**** FIGURE **2** ABOUT HERE.

The possible preparatory target combinations in the two and three-position cases were (1 2), (3 4), or (5 6), and (6 1 2) or (3 4 5), respectively. Only LEDs in the indicated combinations could light up during the preparatory phase. In the one-target condition, the same LED that was illuminated in green during the preparatory phase turned red with the response signal.

On each experimental day, blocks of trials corresponding to each experimental condition (one, two or three targets during preparatory phase) were presented to the monkey in random order. During each block, the target directions were chosen at random with equal probability. Although the monkey performed movements in six directions during the experiment, we only distinguish movement to the left or right side in this study. In other words, correct trials in directions 6, 1 and 2 (3, 4 and 5) were pooled for right-hand (left-hand) side movements. For our analysis, we only used the 1-target and the 3-target conditions, as the 2-target condition contained an ambiguous pair (5, 6) located between the right and left panes. We only used data from one monkey, indicated as "monkey 1" in [5].

On each experimental day neuronal signals were recorded using seven quartz-insulated platinum–tungsten electrodes (outer diameter = 80 μm; impedance = 2–5 MOhm at 1 kHz; Reitböck System, Thomas Recording, Germany), which were transdurally inserted at the beginning of each recording day. Up to seven simultaneously recorded single units were isolated, one per electrode. On each electrode, the spikes belonging to a single neuron were defined by their amplitude, and their waveforms were controlled online on the screen. Only the signals which passed a user-defined threshold were recorded. The threshold was observed and adjusted online to compensate for long term changes in signal amplitude. The time stamps of spike occurrences along with behavioral events such as the preparatory signal, the response signal, movement onset, and movement end were stored at a time resolution of 1 ms. In total, the spiking activity of 111 neurons was selected for further analysis.

As shown in [5], these 111 single neurons encode different information across distinct parts of the trial. For example, individual neurons were tuned for movement direction only during the preparatory phase, or only during movement execution, or during both. Due to restrictions on the size of the network that can be accommodated on the Spikey system, we were limited to use only 12 neurons as input for our decoder network. Therefore, we manually selected 12 neurons with directional tuning phase by computing their firing rate for each direction over the duration of the trial.

Since we were interested in decoding movement intention, we then selected neurons that exhibited noticeably higher firing rates during the preparation phase for a specific experimental direction.

A different set of neurons was recorded each day, and the number of simultaneously recorded neurons was limited by the number of electrodes. Therefore, the best-tuned neurons among the 111 units total were rarely recorded simultaneously, but rather acquired during separate sessions. In fact, none of the 12 selected neurons were recorded in the same session. While this may be seen as a restricting factor when reproducing our results in BMI applications, we wish to stress that more recent recording techniques achieve much higher electrode densities and thus can record from substantially more units in parallel, increasing the chance that well-tuned neurons are available from the same recording session [19]. Since our aim in this study was to demonstrate the feasibility of a spiking neuromorphic decoder as a proof of concept, we accepted the restriction to using already available, albeit limited, data.

Spiking neural network

The classifier spiking neural network used in this work modifies the network in [33]. Its three-layer architecture comprising an input layer, a decorrelation layer, and an association layer, is inspired by the insect olfactory system [36] (**Figure 3**). Neurons in each layer are organized as small neuronal populations, for two reasons: First, single spike sources are too weak to generate activity in downstream neurons due to hardware-constrained bounds on the synaptic weights. Second, due to their analog implementation, individual neurons on the neuromorphic hardware have slightly different physical parameters and are affected by electronic noise. In consequence, individual neurons of a population that receive similar input can vary considerably in their output. Grouping neurons in small populations mitigates both these effects. The populations sizes that we adopted in this study have been chosen as a trade-off to minimize neuron count while counteracting these adverse effects as much as possible (see [33] and supplemental material therein for a detailed analysis).

Two preprocessing steps were performed on the recorded spike times prior to feeding them into the network. First, since each of the 12 input populations that correspond to the 12 selected motor cortical cells consists of 6 neurons, we copied each of the 12 spike trains six times to supply all input neurons. Second, since simultaneous spikes would lead to undesired synchronous bursts in activity (see [33] and supplement therein for a detailed analysis), we introduced a small random delay, fixed for each spike train within a population, drawn from a uniform distribution with values between 0 ms and 100 ms. We wish to stress that the computational overhead on these operations is very low, essentially consisting of drawing one random number per input neuron, and executing 6 additions/delays per spike. Therefore these steps could easily be performed on live data in a real-time processing scenario, even with very modest low-power hardware.

The decorrelation layer is organized in so-called glomeruli, i.e. functional groups of neurons that exist in the insect olfactory system [36]. Each glomerulus consists of a population of 6 excitatory projection neurons (PN) and a population of 6 local inhibitory neurons (LN). Each input neuron population is sparsely connected to the projection neurons of one glomerulus (see **Table 1**). Projection neurons make sparse excitatory connections to the inhibitory population in the same glomerulus. Local inhibitory neurons in the decorrelation layer have inhibitory connections to all projection neurons in all other glomeruli but not to projection neurons in the same glomerulus.

This so-called lateral inhibition reduces input correlation and thereby improves classification performance [33], [36]. In the association layer there are two excitatory association neuron (ANe) populations that receive input from the projection neurons via plastic synapses. After training (outlined below), activity in the first (second) of these populations indicates an upcoming leftward (rightward) movement. The contrast between the two ANe populations is increased through the inhibitory association neuron (ANi) populations.

Both excitatory and inhibitory connection weights are bounded between 0 - 0.025 and 0 - 0.15 respectively. These weights are unit-less because they correspond to software-defined parameters with an arbitrary reference [28]. Connection weights per population are detailed in Table 1. Weights are fixed for all synapses except the projection to association layer connections, which are modified during learning using a Hebbian-type three factor learning rule, explained in the next section.

*** FIGURE 3 ABOUT HERE.

*** TABLE 1 ABOUT HERE.

Network training and testing

Tuning a classifier typically involves three stages: training, testing and validation. Testing is performed at the end of training on a subset of the training data, whereas validation requires novel input data on which the classifier hasn't been trained. The accuracy of the classifier on the validation data is referred to as validation or generalization performance.

The spiking classifier in this study was trained to distinguish between actions that were performed when response signal LEDs on the left versus right side of the circle lit up. For network training we used spikes generated between 650 ms and 1400 ms after trial start, which translates to 150 ms after the preparatory signal and 100 ms before the response signal – in other words, the network was trained on neuronal activity obtained before the actual movement was carried out. The 150 ms time margin after PS was excluded from training as this corresponds to the average latency until the information presented to the monkey activates neurons in the primary motor cortex [5]. Since the monkey could already initiate the movement before the response signal arrived in the 1-target condition, we discarded 100 ms at the end of the preparatory phase in order to prevent contaminating the preparatory neuronal signals [5]. All times are stated in "biological" time – on Spikey, these times are effectively 10^4 times shorter, due to the 10,000-fold speedup compared to real-time. For a detailed analysis of the execution speed of the system as a whole, please refer to [33] and the supplemental material therein.

We calculated the time-resolved generalization performance on the entire spike train (from 0 to 3000 ms) using a sliding window of 500 ms length (*cf.* **Figure 4)**. At each time point *t*, the window included spikes within the interval *t* - 500 ms to *t*. The generalization capability of the network was tested using percent-correct as performance measure, in a five-fold cross-validation paradigm. The number of trials in each of the two artificially-created (left vs. right) directions for each cortical neuron was limited (in the range of 29 to 148 trials per direction). Consequently, at each cross-validation run we used sampling with replacement to create our training and test set, insuring an equal number of trials for left- vs. rightward movement. To achieve this, the complete dataset, containing on average 130 spike trains for each neuron, was first split into five equal parts. For

each cross-validation run, four of these parts were concatenated to form the training data, while the fifth part was set aside for validation. 100 spike trains per direction and per input neuron were randomly sampled from the pool of training data. Since we employed 2 directions, a total of 200 combinations of 12 spike trains (i.e. one for each of the 12 input neuron populations) were used as input. At the end of the training phase, one quarter of the training set was used as sampling population to draw another 200 x 12 spike trains to test the training performance. Validation was performed on 200 x 12 spike trains extracted from the fifth subset, which was not used during training.

The classification decision of the network corresponds to the label ("left" vs. "right") of the association neuron population having the highest spike count in the time interval of 500 ms length used for the sliding window analysis (see Results). Since PN to ANe connections are initialized with random values at the beginning of the training phase, by chance one of the two association populations will produce a slightly higher spike count (the winner population). Lateral inhibition via ANi populations leads to a soft winner-take-all behavior. If the label of the winner population matches the one of the input, the classification of the input spike trains is deemed correct and the weights between active projection neurons and the winner ANe population are strengthened. Otherwise, the weights of active connections are reduced. This Hebbian-type learning rule is essentially identical to the perceptron rule and is formally described by the following:

$w_{new} = w_{old} + \Delta w$, where
$\Delta w = +\eta \cdot c$, if classification was correct
$\Delta w = -\eta \cdot c$, if classification was incorrect.

c is a constant which corresponds the minimal weight change supported by the hardware, and η is the learning rate. We used $c = 0.0004$ and $\eta = 10$. During the training phase, the weight changes were computed on the host computer from the spiking activity that has been recorded on the chip.

A PN-ANe connection is eligible for modification only if the PN is considered "active", i.e., number of emitted spikes during the previous training interval exceeded a certain threshold (30 spikes in this study). The ANe to which the eligible PN is connected must be part of the winner population for the weight update to take place.

After computing the new weights w_{new}, we verify if w_{new} is within the minimum and maximum weights allowed on the hardware. The weight w_{post} that is set after a training interval is obtained by:

$$w_{post} = \begin{cases} w_{max}, & \text{if } w_{new} > w_{max}, \\ w_{min}, & \text{if } w_{new} < w_{min}, \\ w_{new}, & \text{otherwise.} \end{cases}$$

An in-depth analysis of the function of the network as a generic classifier for multivariate data and the role of the various parameters is provided in [33] and the supplemental material therein.

Results

We tested the performance of our spiking neural network decoder implemented on the Spikey neuromorphic hardware in a time-resolved fashion. The results are shown in **Figure 4** where we used a sliding window of 500 ms length to compute the percentage of correct prediction across

validation trials. The curve represents the average performance across 10 cross-validated network iterations. Initially, the performance is at chance level (50%). This changes as the target information presented with the preparatory signal (PS, at t=500 ms) takes effect in directionally tuned responses of the motor cortical neurons. This leads to a gradual increase in performance as the sliding window comprises an increasing number of informative spikes. The highest performance was achieved around the time interval that corresponded to the training interval (650ms, 1400ms] within the preparatory phase. The independent results for the 1-target and 3-target conditions are highly consistent. The maximum performance on the validation set was 89.32% (0.63) (1 target) and 84.79% (0.57) (3 targets) average correct classification (standard deviation) across 10 network iterations. For comparison, a Naïve Bayes classifier trained on the firing rates alone will yield a maximum performance 98.0% ±0.3% (1 target), resp. 97.1% ± 0.8% (3 targets) (red traces in **Figure 4**).

As explained in the *Experimental paradigm* section, the cortical neurons chosen had preferential tuning for the preparatory phase. This explains the drop in performance when testing the network on a trial segment outside this phase, i.e., after t=1500 ms. The apparent inversion of prediction performance during the late phase of the trial around t=2500 ms originated from the monkey's movement back to the central position (i.e., in the opposite direction) as it was preparing to start the next trial.

*** FIGURE 4 ABOUT HERE.

To further explore the spiking neural network behavior, we recorded the spiking activity from all neurons on the Spikey chip during one individual validation run. **Figure 5** shows the spike trains of all neurons in the network during a single trial of rightward movement. The activation of the two populations of excitatory association neurons (ANe) predict a movement either to the left (upper population) or the right (lower population). In the time interval (0ms, 500ms] there was no directional information available to the monkey. The twelve input neuron populations, identical with the motor cortical neurons, show typical cortical baseline activity. This activity propagates through the network. The spiking activity in the two excitatory association neuron populations is very similar, indicating equal probability for both movement directions consistent with the initial 50% chance level in **Figure 5**. At t=500 ms the preparatory signal (PS, vertical dashed line) was presented to the monkey indicating three possibly rightward movement targets (*cf.* **Figure 2**). The input neurons change their activity in response to the visually presented target information with a latency in the order of about 150 ms to 200 ms. In effect, the activity of the association neuron populations shows an activity distribution that is strongly biased towards the population that (correctly) predicts a rightward movement during the preparation phase. Around the response signal there is again little distinction between the two ANe populations.

*** FIGURE 5 ABOUT HERE.

In additional experiments, we tested whether the selection of the neurons that we made prior to training plays an important role for the performance of the network. One of the main limitations of our approach is probably that due to experimental constraints the 12 neurons we selected for network were not recorded simultaneously. The largest number of simultaneously recorded neurons with useful tuning was 7. Trained with these 7 neurons, the network still performed above chance, even if only with a thin margin (57.3% ± 0.9% correct for 1 target, 56.8% ± 1.8% for 3

targets). When we used neurons from separate sessions but reduced the neuron count, we found that the classifier still performed well with 6 neurons (82.7% ± 1.2% for 1 target, 77.6% ± 1.0% for 3 targets), however performance degraded (yet still being above chance level) when using only 3 neurons (54.0% ± 1.7%; 52.7% ± 1.3%).

Besides the neuron selection, the length of the integration window during which spikes are counted to assess the classifier outcome plays a role. Longer windows typically reach better results because they are less susceptive to short-term fluctuations and irregularities of the neuronal spike trains, although at the expense of temporal resolution. The above numbers were all obtained using an 500 ms integration window. Performance declined slightly using a 300 ms window, and a bit more so using a 150 ms window (300 ms: 79.9% ± 2.7% 1 target, 77.4% ± 2.7 3 targets; 150 ms: 73.2% ± 2.0% 1 target, 66.0% ± 1.0% 3 targets).

Discussion

Here we successfully demonstrated the off-line classification of the direction of an arm reaching movement (left vs. right) from single unit recordings in the motor cortex of a behaving monkey using a spiking neural network on neuromorphic hardware. To our knowledge, at the time of writing this is the first demonstration of cortical spike train decoding with spiking neuromorphic hardware. Our model here modifies a spiking model for generic data classification that we had previously developed and tested on the same hardware [33]. The model architecture is directly inspired by the brain architecture that underlies associative learning in insects and mimics biological information processing. Using the activity of only 12 single motor cortical neurons selected from a pool of 111 neurons and a realistic number of 100 training trials per direction we achieved useful classification performance with more than 89 % of correctly predicted movements on unseen data in the 1-target condition (~85% in the three-target condition). It should be noted, though, that these numbers are still lower than what can be obtained with a Naïve Bayes decoder, pointing out the potential of further improvements, e.g. through fine-tuning the network architecture.

The Spikey system on which we carried out this study was limited to a maximum of 192 neurons. This constrained the number of neuron populations in all layers, i.e. the total number of input neurons as well as the number of association populations (number of directions) that could be tested. As recent versions of the Spikey system provide a higher neuron count (384 neurons), future studies could use larger networks that may enable classification of more directions, and potentially also more complex movements.

Two previous approaches have used neural network models for the decoding of movement parameters from motor cortical spiking activity. In [21] the authors employed the neural engineering framework [37] in order to design a spiking neural network with 2,000 neurons that implements a Kalman filter algorithm. The conventional Kalman filter has repeatedly been shown to be successful in decoding smooth hand movement trajectories from motor cortical activity. The authors simulated their network on a computer fast enough to achieve decoding in real time and they successfully tested this approach in an experiment where the hand movement trajectories were accurately decoded in real time [22] (a closed-loop scenario). As input to their network the authors used the multi-unit spike count (or firing rate) estimated in 50 ms time bins from each channel recorded with a so-called Utah array (a 96-channel electrode system). Therefore, besides using a much higher neuron count, they did not feed the recorded action potentials to the neurons in their computer simulated network, but used an intermediate rate-based representation.

Sussillo and co-authors [38] trained a so-called echo state network for the decoding of firing rates. This network type belongs to the class of non-spiking artificial neural networks and operates with 1200 to 1500 rate neurons. Connectivity is random and recurrent. The training is achieved by tuning a limited number of feedback connections. The temporal resolution of the simulation is 15-25 ms. As input to the rate neurons the authors provided the spike count (or firing rate) estimated in bins that use resolution of the simulation (15-25 ms) from multi-unit activity of 96 channels simultaneously recorded with a Utah array in a closed-loop scenario. These experiments were performed in the same lab and with the same monkeys as in [22].

We wish to emphasize that our approach presents a proof of concept for decoding cortically recorded spikes using neuromorphic hardware, rather than a full-fledged BMI solution. En route to practical closed-loop applications of a spiking neuromorphic decoder several future steps are necessary. Foremost, our analysis was based on 12 manually selected neurons that were recorded at different times. This was done in order to obtain a sufficient number of "good" neurons (i.e., with useful motor tuning) from the limited number of simultaneously recorded neurons in the original experiments. More recent recording techniques, such as the aforementioned Utah arrays, could potentially deliver a higher number of well-tuned neurons in simultaneous recordings [19]. This should be addressed in future research.

Moreover, recorded brain activity should be processed in real-time for a BMI solution to be practical, so a tight coupling between spikes produced in the brain and the actual computation needs to be established. The Spikey hardware runs at a 10^4 speedup factor compared to real time (see Methodological background), which enables accelerated neuromorphic computing. However, due to this speedup, it is not possible to feed spikes in real time from the brain to the chip in a closed-loop scenario (at least not without creating intermediate firing-rate representations). If the spikes were fed as they are recorded, the effective spike rate on chip would be 10^4 times lower than in the biological system, thus practically reaching zero and preventing interaction between individual spikes. Rather, spikes would have to be buffered and processed on the chip in batches. While this asynchronous mode of operation complicates a closed-loop application, due to the high speedup factor of the Spikey platform it also provides the possibility to perform multiple classification runs, e.g. with different network parameters, to be performed on each batch of recorded spikes. Such an approach may even provide higher robustness against noise than fully synchronous operation, and should be investigated in a future study.

Nonetheless, other hardware platforms have been developed that are designed for real-time applications, such as e.g. SpiNNaker [24], or the low-power analog ROLLS neuromorphic processor [26]. These platforms would be more straightforward to use in a closed-loop scenario, because networks on these chips will operate at exactly the same time scale as the brain, and recorded spikes can be injected into the network in real-time. Since these platforms also support on-chip learning, they could in theory be trained without any interaction with a host machine, therefore enabling a truly portable and compact solution. It should be noted that the network that we used here for classification has already been successfully implemented on the SpiNNaker system [39] with full exploitation of the on-chip plasticity facilities, convincingly demonstrating the feasibility of such an endeavor for future studies.

Regarding the input spike signals, it might be of practical advantage to use multi-unit instead of

single-unit activity. Multi-unit activity represents the pooled spiking activity of many neurons recorded on a single electrode. It can be generated simply by thresholding extracellular voltage signals, and thus avoids computationally costly spike sorting algorithms [40]. It has been shown that in the motor cortex multi-unit activity of a single channel may carry a similar amount of information about a movement as the single units extracted from the same recording channel [3], [41].

In our demonstration the spiking network operates on real spike-timing input. For predicting the single trial movement direction, however, we interpreted the association neurons' output spike count within a fixed-width time window. In future biomedical applications, it might become feasible to directly use the spiking output to activate body muscles [42], potentially obviating the need for counting spikes in a time window to control an actuator. Other potential biomedical applications of spiking neuromorphic computing include restoring spinal cord function [43] and the application in closed-loop deep brain stimulation devices [44], [45].

Conclusion

In this paper, we have shown for the first time a neural network decoder of voluntary movement intentions from motor cortical activity that runs on neuromorphic hardware. Our implementation used a biologically inspired spiking neural network with only 176 artificial neurons in total. As input to the decoder network we used modified spiking activity recorded *in vivo* from the primary motor cortex of a behaving monkey. In our offline test paradigms, the decoder successfully predicted the left-right direction of a delayed hand reaching movement during the movement preparation period (i.e., in the phase before the actual hand movement was executed) with high accuracy from the spiking activity of a comparably small number of 12 manually-selected cortical neurons. Our results provide a proof of principle for the potential use of neuromorphic computing in a future biomedical application.

Acknowledgements

This work was supported by Deutsche Forschungsgemeinschaft (DFG grant SCHM2474/1-2 to MS) and the German Israeli Foundation (grant I-1224-396.13/2012 to MN). MS received support by a Marie Curie Intra-European Fellowship (EU FP7 grant 331892, BIOMACHINELEARNING). The Spikey Neuromorphic hardware was kindly provided by Prof. Karlheinz Meier, Kirchhoff-Institute for Physics, University of Heidelberg, Germany, as a loan within EU FP 7 grant no. 604102 (Human Brain Project). Author contributions: MS and MPN designed research; AR provided data; IL and MS performed research; IL, MPN, and MS wrote the paper.

References

[1] A. P. Georgopoulos, J. F. Kalaska, R. Caminiti, and J. T. Massey, "On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex.," *J. Neurosci.*, vol. 2, no. 11, pp. 1527–37, Nov. 1982.

[2] A. Riehle and J. Requin, "Monkey primary motor and premotor cortex: single-cell activity related to prior information about direction and extent of an intended movement.," *J. Neurophysiol.*, vol. 61, no. 3, pp. 534–49, Mar. 1989.

[3] C. Mehring, J. Rickert, E. Vaadia, S. Cardosa de Oliveira, A. Aertsen, and S. Rotter, "Inference of hand movements from local field potentials in monkey motor cortex.," *Nat. Neurosci.*, vol. 6, no. 12, pp. 1253–1254, 2003.

[4] W. Truccolo, G. M. Friehs, J. P. Donoghue, and L. R. Hochberg, "Primary motor cortex tuning to intended movement kinematics in humans with tetraplegia.," *J. Neurosci.*, vol. 28, no. 5, pp. 1163–78, Jan. 2008.

[5] J. Rickert, A. Riehle, A. Aertsen, S. Rotter, and M. P. Nawrot, "Dynamic encoding of movement direction in motor cortical neurons.," *J. Neurosci.*, vol. 29, no. 44, pp. 13870–13882, Nov. 2009.

[6] K. V Shenoy, M. Sahani, and M. M. Churchland, "Cortical control of arm movements: a dynamical systems perspective.," *Annu. Rev. Neurosci.*, vol. 36, pp. 337–59, 2013.

[7] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue, "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature*, vol. 442, no. 7099, pp. 164–71, 2006.

[8] P. R. Kennedy and R. A. E. Bakay, "Restoration of neural output from a paralyzed patient by a direct brain connection," *Neuroreport*, vol. 9, no. 8, pp. 1707–1711, Jun. 1998.

[9] J. Wessberg, C. R. Stambaugh, J. D. Kralik, P. D. Beck, M. Laubach, J. K. Chapin, J. Kim, S. J. Biggs, M. a Srinivasan, and M. a Nicolelis, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates.," *Nature*, vol. 408, no. 6810, pp. 361–365, 2000.

[10] D. M. Taylor, S. I. H. Tillery, and A. B. Schwartz, "Direct cortical control of 3D neuroprosthetic devices.," *Science*, vol. 296, no. 5574, pp. 1829–32, Jun. 2002.

[11] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue, "Instant neural control of a movement signal.," *Nature*, vol. 416, no. 6877, pp. 141–2, Mar. 2002.

[12] M. Velliste, S. Perel, M. C. Spalding, a S. Whitford, and a B. Schwartz, "Cortical control of a robotic arm for self-feeding," *Nature*, vol. 453, no. June, pp. 1098–1101, 2008.

[13] M. A. L. Nicolelis and M. A. Lebedev, "Principles of neural ensemble physiology underlying the operation of brain-machine interfaces.," *Nat. Rev. Neurosci.*, vol. 10, no. 7, pp. 530–40, 2009.

[14] C. Mehring, M. P. Nawrot, S. C. De Oliveira, E. Vaadia, A. Schulze-Bonhage, A. Aertsen, and T. Ball, "Comparing information about arm movement direction in single channels of local and epicortical field potentials from monkey and human motor cortex," *J. Physiol. Paris*, vol. 98, no. 4–6 SPEC. ISS., pp. 498–506, 2004.

[15] T. Pistohl, T. Ball, A. Schulze-Bonhage, A. Aertsen, and C. Mehring, "Prediction of arm movement trajectories from ECoG-recordings in humans.," *J. Neurosci. Methods*, vol. 167, no. 1, pp. 105–14, Jan. 2008.

[16] T. Milekovic, W. Truccolo, S. Grün, A. Riehle, and T. Brochier, "Local field potentials in

primate motor cortex encode grasp kinetic parameters.," *Neuroimage*, vol. 114, pp. 338–55, Jul. 2015.

[17] U. Mitzdorf, "Current source-density method and application in cat cerebral cortex: investigation of evoked potentials and EEG phenomena.," *Physiol. Rev.*, vol. 65, no. 1, pp. 37–100, Jan. 1985.

[18] E. C. Leuthardt, G. Schalk, J. R. Wolpaw, J. G. Ojemann, and D. W. Moran, "A brain-computer interface using electrocorticographic signals in humans.," *J. Neural Eng.*, vol. 1, no. 2, pp. 63–71, 2004.

[19] A. Riehle, S. Wirtssohn, S. Grün, and T. Brochier, "Mapping the spatio-temporal structure of motor cortical LFP and spiking activities during reach-to-grasp movements.," *Front. Neural Circuits*, vol. 7, no. March, p. 48, 2013.

[20] M. Nawrot, A. Aertsen, and S. Rotter, "Single-trial estimation of neuronal firing rates: From single-neuron spike trains to population activity," *J. Neurosci. Methods*, vol. 94, no. 1, pp. 81–92, 1999.

[21] J. Dethier, P. Nuyujukian, C. Eliasmith, T. C. Stewart, S. A. Elasaad, K. V Shenoy, and K. A. Boahen, "A brain-machine interface operating with a real-time spiking neural network control algorithm," *Adv. Neural Inf. Process. Syst.*, pp. 2213–2221, 2011.

[22] J. Dethier, P. Nuyujukian, S. I. Ryu, K. V Shenoy, and K. Boahen, "Design and validation of a real-time spiking-neural-network decoder for brain-machine interfaces.," *J. Neural Eng.*, vol. 10, no. 3, p. 036008, Jun. 2013.

[23] C. A. Mead, *Analog VLSI and Neural Systems*. Reading, MA: Addison Wesley, 1989.

[24] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown, "Overview of the SpiNNaker System Architecture," *IEEE Trans. Comput.*, vol. 62, no. 12, pp. 2454–2467, Dec. 2013.

[25] B. V. Benjamin, Peiran Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J.-M. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. A. Merolla, and K. Boahen, "Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations," *Proc. IEEE*, vol. 102, no. 5, pp. 699–716, May 2014.

[26] N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, and G. Indiveri, "A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses," *Front. Neurosci.*, vol. 9, no. April, pp. 1–17, 2015.

[27] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, a. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science (80-.).*, vol. 345, no. 6197, pp. 668–673, Aug. 2014.

[28] T. Pfeil, A. Grübl, S. Jeltsch, E. Müller, P. Müller, M. A. Petrovici, M. Schmuker, D. Brüderle, J. Schemmel, and K. Meier, "Six networks on a universal neuromorphic computing substrate," *Front. Neurosci.*, vol. 7, no. February, p. 11, Jan. 2013.

[29] J. Schemmel, D. Brüderle, A. Grübl, M. Hock, K. Meier, and S. Millner, "A Wafer-Scale Neuromorphic Hardware System for Large-Scale Neural Modeling," in *Proceedings of the 2010 International Symposium on Circuits and Systems (ISCAS)*, 2010, pp. 1947–1950.

[30] E. Neftci, C. Posch, and E. Chicca, "Neuromorphic Engineering," in *Computational Intelligence - Volume II*, Hisao Ishibuchi, Ed. Encyclopedia of Life Support Systems (EOLSS), 2015, pp. 278–307.

[31] E. O. Neftci, J. Binas, U. Rutishauser, E. Chicca, G. Indiveri, and R. J. Douglas, "Synthesizing cognition in neuromorphic electronic systems.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 110, no. 37, pp. E3468–76, Sep. 2013.

[32] T. Rost, H. Ramachandran, M. P. Nawrot, and E. Chicca, "A neuromorphic approach to auditory pattern recognition in cricket phonotaxis," in *2013 European Conference on Circuit Theory and Design (ECCTD)*, 2013, vol. 1, pp. 1–4.

[33] M. Schmuker, T. Pfeil, and M. P. Nawrot, "A neuromorphic network for generic multivariate data classification.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 111, no. 6, pp. 2081–2086, Feb. 2014.

[34] M. M. Waldrop, "The chips are down for Moore's law," *Nature*, vol. 530, no. 7589, pp. 144–147, Feb. 2016.

[35] A. Bastian, G. Schöner, and A. Riehle, "Preshaping and continuous evolution of motor cortical representations during movement preparation," *Eur. J. Neurosci.*, vol. 18, no. 7, pp. 2047–2058, 2003.

[36] M. Schmuker and G. Schneider, "Processing and classification of chemical data inspired by insect olfaction," *Proc. Natl. Acad. Sci.*, vol. 104, no. 51, pp. 20285–20289, 2007.

[37] C. Eliasmith and C. H. Anderson, *Neural Engineering*. Cambridge, MA, USA: MIT Press, 2004.

[38] D. Sussillo, P. Nuyujukian, J. M. Fan, J. C. Kao, S. D. Stavisky, S. Ryu, and K. Shenoy, "A recurrent neural network for closed-loop intracortical brain-machine interface decoders.," *J. Neural Eng.*, vol. 9, no. 2, p. 026027, 2012.

[39] A. Diamond, T. Nowotny, and M. Schmuker, "Comparing Neuromorphic Solutions in Action: Implementing a Bio-Inspired Solution to a Benchmark Classification Task on Three Parallel-Computing Platforms," *Front. Neurosci.*, vol. 9, no. January, p. 491, 2016.

[40] H. G. Rey, C. Pedreira, and R. Quian Quiroga, "Past, present and future of spike sorting techniques," *Brain Res. Bull.*, vol. 119, pp. 106–117, 2015.

[41] E. Stark and M. Abeles, “Predicting movement from multiunit activity.,” *J. Neurosci.*, vol. 27, no. 31, pp. 8387–8394, 2007.

[42] C. T. Moritz, S. I. Perlmutter, and E. E. Fetz, “Direct control of paralysed muscles by cortical neurons,” *Nature*, vol. 456, no. 7222, pp. 639–642, 2008.

[43] R. Jung, E. J. Brauer, and J. J. Abbas, “Real-time interaction between a neuromorphic electronic circuit and the spinal cord,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 9, no. 3, pp. 319–326, 2001.

[44] C. M. Thibeault, “A role for neuromorphic processors in therapeutic nervous system stimulation.,” *Front. Syst. Neurosci.*, vol. 8, no. October, p. 187, Jan. 2014.

[45] M. Rosa, G. Giannicola, S. Marceglia, M. Fumagalli, S. Barbieri, and A. Priori, “Neurophysiology of Deep Brain Stimulation,” in *International Review of Neurobiology*, 1st ed., vol. 107, Elsevier Inc., 2012, pp. 23–55.

Author Biographies

Iulia-Alexandra Lungu, Bernstein Centre for Computational Neuroscience Berlin, Technical University and Humboldt University, Berlin (iulialexandralungu@gmail.com). Ms. Lungu is an M.S. graduate in Computational Neuroscience. She received her B.Sc. degree in Bioinformatics in 2013 from Claude Bernard University Lyon. Previous work includes recurrent neural networks, brain-computer-interfaces and neuromorphic engineering, in collaboration with the Max Planck Institute for Brain Research in Frankfurt, Technical University Berlin, Freie Universität Berlin and the University of Sussex.

Alexa Riehle, Institut de Neurosciences de la Timone (INT), Centre National de la Recherche Scientifique (CNRS) – Aix-Marseille Université, Marseille, France, (alexa.riehle@univ-amu.fr). Dr. Riehle is Research Director at the CNRS and studies higher cortical processes involved in movement preparation and execution and visuomotor integration by using massively parallel multi-electrode recording techniques ("Utah" arrays implanted in multiple cortical areas) in the behaving monkey. She investigates the temporal dynamics of cooperative, distributed cortical networks involved in higher cognitive (motor) processes. She combines approaches from cognitive and theoretical neuroscience. She is the head of the CoMCo (Cognitive Motor Control) team at the INT. She has an honorary position as "Visiting Scientist" at both the RIKEN Brain Science Institute, Wakoshi, Japan, and the Research Centre Jülich, Jülich, Germany. She is the co-founder of the CNRS International Associated Laboratory "Vision-for-Action", a joint initiative by the CoMCo team at INT and the INM-6 at Research Centre Jülich.

Martin P. Nawrot, Computational Systems Neuroscience, Institute for Zoology, Biocenter, University of Cologne, Cologne, 50674, Germany (martin.nawrot@uni-koeln.de). Dr. Nawrot is head of the Computational Systems Neuroscience group at the University of Cologne, Germany. He studied physics, political sciences and history at the University of Freiburg, Germany and at the University of Kent at Canterbury, UK. He received a Diplom (M.S. degree) in Physics and a

PhD in Natural Sciences from the University of Freiburg in 1998 and 2003, respectively. He subsequently obtained a fellowship with the Heidelberg Academy of Sciences and Humanities where he worked on invasive Human Brain Machine Interfacing and became co-founder of the Bernstein Center for Computational Neuroscience Freiburg. In 2005 he moved to Berlin working as a postdoc with Prof. Dr. Sonja Grün before he became a group leader at the Bernstein Center for Computational Neuroscience Berlin in 2007. In 2008 he accepted a call as Assistant Professor for Neuroinformatics and Theoretical Neuroscience at the Freie Universität Berlin, Germany. Since 2015 he is Full Professor for Computational Systems Neuroscience and Animal Physiology at the University of Cologne, Germany. His research focuses on principles of neural computation at the level of functional neural networks. Current research topics include memory formation and recall in the insect brain, spiking neural network models for motor control and robotic control, and physiology and models of neuronal variability in the mammalian cortex.

Michael Schmuker, School of Computer Science, University of Hertfordshire, Hatfield AL10 9AB, UK (m.schmuker@biomachinelearning.net). Dr. Schmuker is Senior Lecturer (Assistant Professor) in the Biocomputation group at the University of Hertfordshire in Hatfield, UK. He received diploma in Biology from the University of Freiburg, Germany in 2003, and a PhD in Chemistry from Goethe-University Frankfurt, Germany in 2007. His postdoctoral work focused on computational neuroscience of Olfaction, and olfaction-inspired pattern recognition using neuromorphic hardware (Freie Universität Berlin, Germany, 2007-2014). In 2014, he joined Sussex University as a Marie Curie Research Fellow to work on bio-inspired signal processing for electronic gas sensors. Current research topics include research into efficient methods for processing of chemical sensor data, image processing for neuroscience, and neuromorphic signal processing and pattern recognition.

Figures

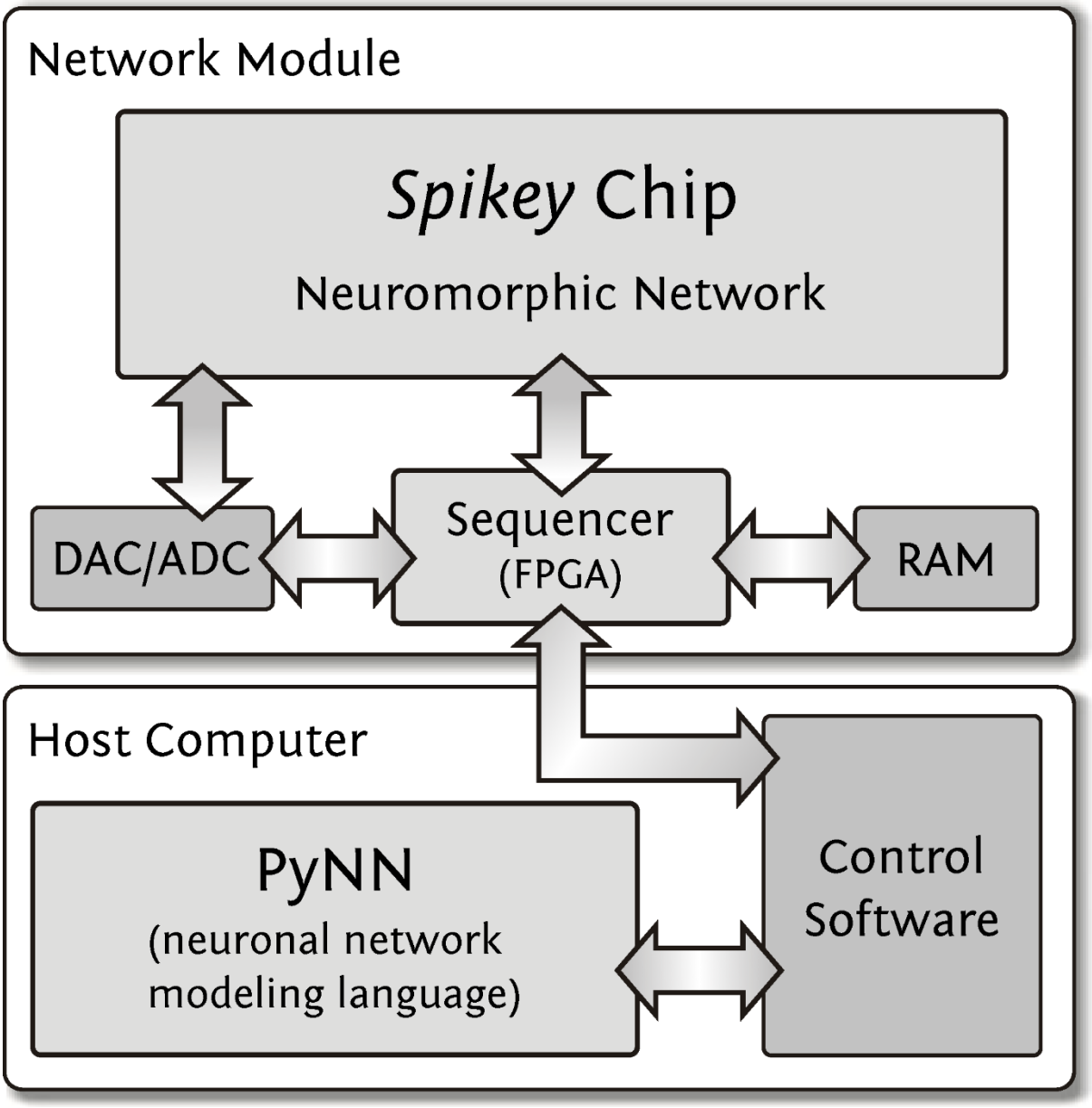

Figure 1: Schematic of the neuromorphic hardware system. Figure adapted from [28].
The Spikey chip hosts analog neuron models (leaky integrate-and-fire with conductance-based synapses). Network models are designed using PyNN (Python Neural Networks package) on a host computer. The network model, along with the input are cached in local random-access memory (RAM) on Spikey before the network emulation is started. Custom logic implemented in a field programmable gate array (FPGA) supplies the neuromorphic chip with input (spikes) and configuration data. Digital-to-analog converters (DACs) supply analog parameter voltages, while analog-to-digital converters (ADCs) read selected voltages from the chip.

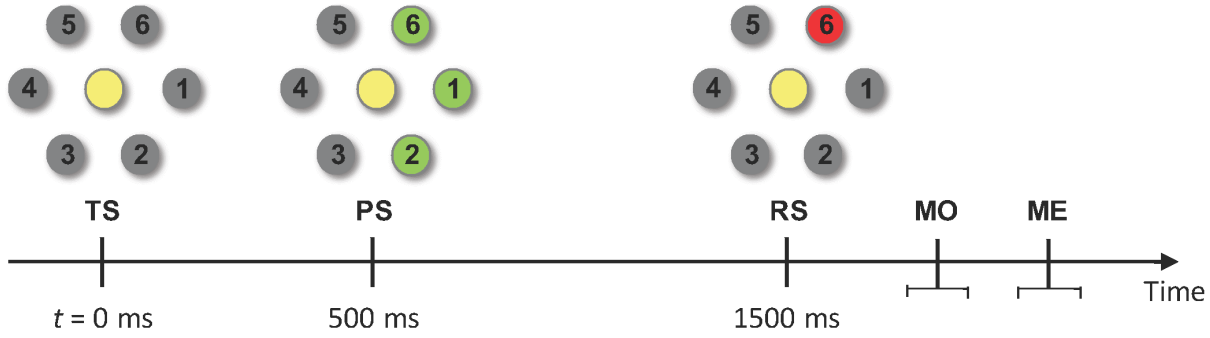

Figure 2: Schematic of the behavioral task, three-target condition. The yellow cue light lit up at $t = 0$ ms, indicating trial start (TS). The preparatory signal (PS) was given at $t = 500$ ms when the targets lit up in green. The preparatory phase ended when the response signal (RS) indicated the correct target in red and signaled to the monkey that it was now allowed to execute the movement. After RS the onset (MO) and end (ME) of the movement were observed. In the one-target condition, only one of the targets would light up during PS.

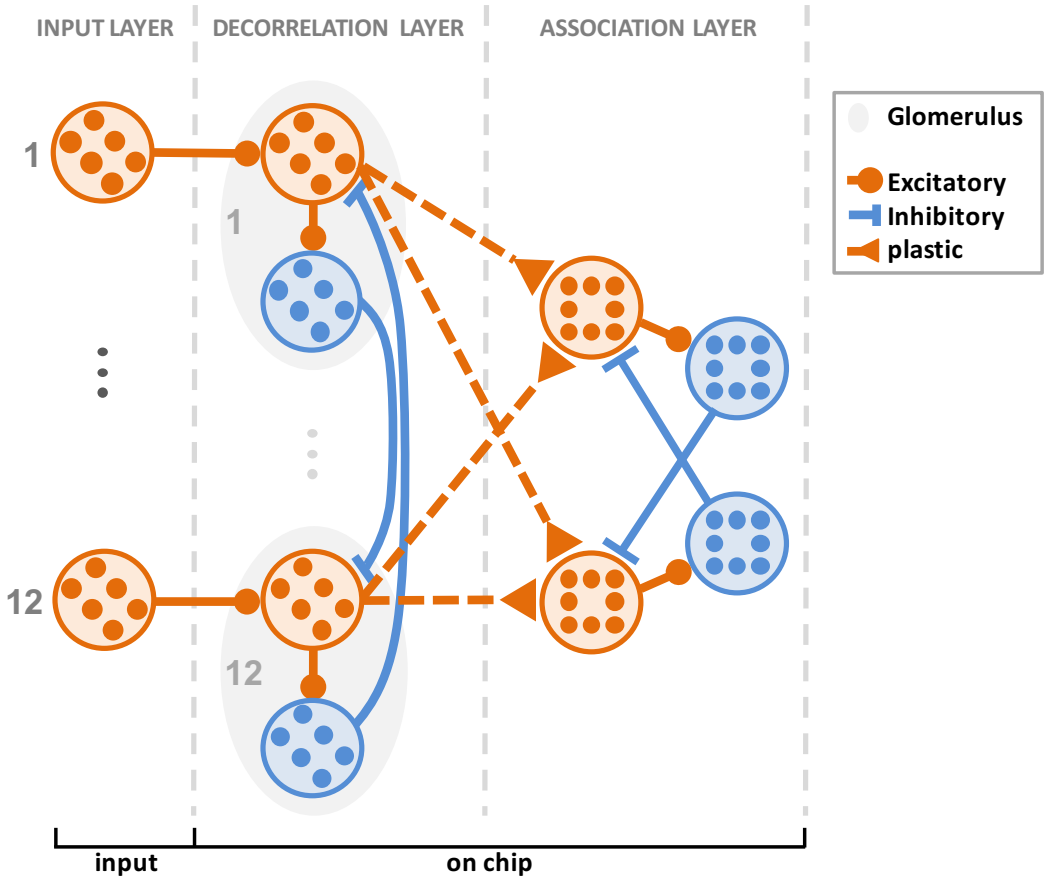

Figure 3. Schematic of the network. Spike trains are generated in neurons in the input layer, propagate to the decorrelation layer where projection neuron populations (orange) receive recurrent lateral inhibition from local inhibitory neurons (blue). The weights from projection neuron populations to the excitatory association neurons (orange in the association layer) are adapted by a supervised learning mechanism during training. Dots represent neuron counts.

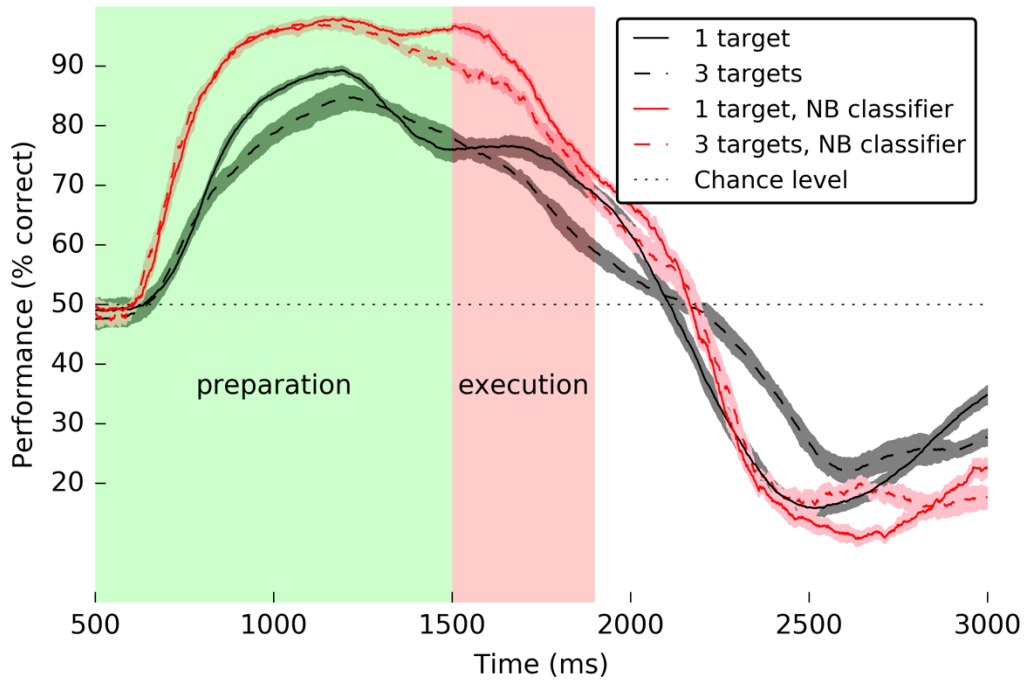

Figure 4. Time-resolved generalization performance of the spiking classifier for predicting movement direction. Black traces: Spiking network performance, shaded areas: standard deviation over cross-validation. The network was trained on spikes recorded in the time interval (650 ms, 1400 ms], i.e. in the preparatory phase of the experiment (highlighted in green). Highlighted in pink is the execution phase of the experiment, when the monkey reached out to touch the red target. Performance of a time-causal prediction was computed in a 500 ms sliding window, averaged over 10 network iterations, each implementing 5-fold cross-validation. Performance at t=500 ms thus refers to spikes in the association populations obtained in t=(0ms, 500ms]. Red traces show the performance of a Naïve Bayes classifier for comparison.

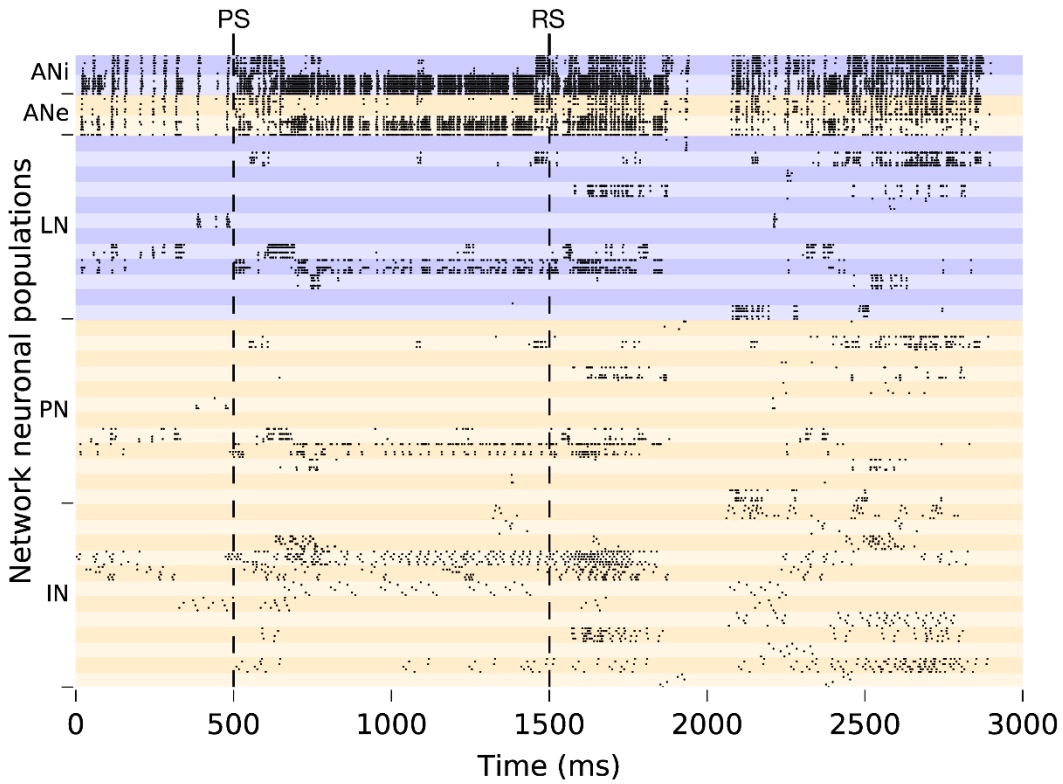

Figure 5. Spiking activity of all network neurons during a single validation trial. Excitatory populations are shown in orange, whereas inhibitory populations are blue. IN: Input neurons, PN:Projection neurons, LN: local inhibitory neurons, :AN{e,i}: excitatory/inhibitory association neurons. Vertical dashed lines define the preparatory phase from 500 (preparatory stimulus, PS) to 1500 ms (response signal, RS). The present trial belongs to the 3-target condition and required a movement to the right (*cf.* Figure 2). The two excitatory ANe populations predict rightward (lower) or leftward (upper) movement, respectively. The network had been trained on the time interval (650 ms, 1400 ms].

Tables

Table 1: Network parameters. Unit-less parameters refer to the values used in the PyNN front-end.

	Population size	**Outgoing connections**	**Connection probability**	**Connection weights**
Input layer				
Input neurons	6	Excitatory to projection neurons	50%	0.75 · max. excitatory weight
Decorrelation layer				
Projection neurons	6	Excitatory to local inhibitory in the same glomerulus	50%	0.7 · max. excitatory weight
		Excitatory to all association populations	50%	randomly initialized between 0.2 and 0.666 · max. excitatory weight
Local inhibitory neurons	6	Inhibitory to projection neurons in other glomeruli	100%	0.5 · max. inhibitory weight
Association layer				
Excitatory neurons	8	Excitatory to inhibitory association neurons	50%	0.8 · max. excitatory weight
Inhibitory neurons	8	Inhibitory to excitatory association neurons	100%	0.9 · max. inhibitory weight