Supporting Information

Schmuker et al. 10.1073/pnas.1303053111

SI Results

Number of Glomeruli. We used 10 virtual receptors (VRs) and thus 10 glomeruli in the network. The specific choice of 10 was made as a compromise to use as many VRs as possible to encode the data while staying within the maximal neuron count of 192 on the present hardware system. In the following, we describe this circumstance in more detail. First, each VR requires one glomerulus. One glomerulus consists of 6 input channels [receptor neurons (RNs)] and 13 neurons [7 projection neurons (PNs) and 6 local inhibitory neurons (LNs)]. Ten glomeruli thus require 130 neurons and 60 additional synapse line drivers for input spikes (see ref. 1 for a detailed technical explanation of the hardware system with regard to synapse line drivers). In addition, each association neuron (AN) population consists of 16 neurons (8 excitatory and 8 inhibitory neurons). For the iris dataset, we require three AN populations, or a total of 48 neurons. Because each neuron requires a synapse line driver, we thus require a total of 130 + 60 + 48 = 238 synapse line drivers. An additional VR would require 6 RNs + 7 PNs + 6 LNs = 19 additional line drivers, totaling to 257 and exceeding the maximum number of 256 synapse line drivers. Although it might be possible to gain space for one or two additional glomeruli by tuning the network to use fewer neurons per glomerulus, we do not expect a significantly different network behavior from a small increase in glomerulus number (see also ref. 2 for an analysis of how VR count affects the performance of a naïve Bayes classifier). Large-scale neuromorphic hardware systems that are under development (e.g., refs. 3 and 4) will overcome this limitation and support thousands of neurons.

Per-Class Classification Performance. For a thorough examination of the classification outcome, we depict the confusion matrix produced by the classifier network (Table S2). The classifier only produced errors on the separation of *Iris versicolor* and *Iris virginica*, whereas it always succeeded to correctly separate *Iris setosa* ($R_K = 0.87$, $P^{20} = 0.85$, $P^{80} = 0.89$). *I. setosa* is well separated from the other classes in feature space. The classifier network achieved perfect separation in cross-validated training in all 50 repetitions. *I. versicolor* and *I. virginica* overlap in feature space, providing a harder challenge to the classifier that is reflected in the higher error rate for that particular separation.

Network Optimization for Robustness Against Neuronal Variability. Constructing a heterogeneous network under constraints of limited neuron count and bounded synaptic weights imposes a tradeoff in connectivity: The number of neurons in a population that project on postsynaptic neurons (the postsynaptic "fan-in") must be sufficiently large to be able to drive the postsynaptic neuron to spiking. At the same time, individual populations must be kept small to accommodate many populations on the chip. In a previous version of the network, we achieved the maximum possible postsynaptic fan-in by using all-to-all connectivity (connection probability p_{conn} = 100%) between all connected populations. In addition, that network contained only three LNs per glomerulus (instead of six in the current, optimized network). Because the fan-in of LNs on PNs is large anyway, this decision seemed a viable way to reduce the total neuron count. Achieving good classification performance with that network required a network-specific calibration routine (*SI Materials and Methods, Application-Specific Calibration of the Neuromorphic Hardware System*). The calibration improved the homogeneity of the transfer functions (Fig. S1*A*) and classification performance improved from R_K around 0.75 to values around 0.86 (Fig. S1*B*).

Analyzing the operation of the fully connected network in detail, we found that neurons in PN, LN, and AN populations were highly synchronized (Fig. S2). This synchronization at the population level was a direct consequence of full connectivity, which entails that all neurons of a population receive the same input. For example, all PNs in a glomerulus received input from the same set of RNs, and their spiking activity was highly correlated as a consequence of the common input. Under these conditions, the assumption of a population rate code with independent neurons is violated—the whole population of n neurons behaves like a single neuron with n times the synaptic weight. Clearly, the postsynaptic interaction of excitatory and inhibitory inputs in this regime is impaired, because synchronous PN spikes lead to synchronized inhibitory LN spikes with a short delay. In contrast, if all n neurons fire independently, the postsynaptic cell receives the same total number of spikes, but distributed more evenly in time. Hence, the chance that excitatory and inhibitory postsynaptic potentials overlap is considerably higher in the asynchronous case. In the optimized network, we reduced the synchronization within relevant populations by sparsifying the connectivity from RNs to PNs and from PNs to LNs and ANs by 50% and readjusting the synaptic weights accordingly.

As an additional step to increase the robustness against transfer function variability, we increased the number of LNs from three to six. Because the individual transfer functions of LNs underlie variability on the hardware, the total transfer function of an LN population will vary according to σ^2/n, with n the population size and σ^2 the variance of the individual transfer functions. Thus, increasing LN population size decreases variability of LN population transfer functions. In consequence, the inhibition strength that a PN population receives from other glomeruli becomes more homogeneous. In other words, increasing LN population size decreases the likelihood that a particular glomerulus may exert significantly higher inhibition than the others and thus alleviates the impact of transfer function variability.

Taken together, we achieved robustness to transfer function variability by two measures: First, we improved population rate coding by making the connectivity sparser, thus alleviating strong coupling on the postsynaptic side. Second, we increased the size of LN populations, thus reducing the variance of the population transfer functions of the LN groups. These steps resulted in the present network that is robust against variability in the transfer functions of individual neurons.

Effect of Lateral Inhibition on Classification Performance. The result that the naïve Bayes classifier's performance increases if trained on the PN firing rates compared with training on the VR responses (Fig. 5*C* in the main text) points out the beneficial effect of lateral inhibition in the presented network. Lateral inhibition transforms the broad, overlapping receptive fields of VRs into localized and more selective receptive fields on the PN level. This step facilitates the "credit assignment problem," that is, the identification of the PNs (or more precisely the PN–AN synapses) that are most responsible for the classification outcome. This information is necessary to select the correct synapses to be potentiated or depressed during classifier training (the "credit assignment problem").

Fig. S3 shows a sketch to illustrate this circumstance. Consider the VR "R_2" in Fig. S3*A*. Because the distances d_1 and d_2 are equal, the response of R_2 to the respective points will be equal, since it depends linearly on these distances (Eq. **S1**). Thus, the response magnitude of this particular VR provides ambiguous information with regard to class adherence, which complicates the learning process. Moreover, because VR receptive fields are broad, there is considerable overlap in the receptive fields of R_1 and R_2 (Fig. S3*B*). One could now simply reduce the receptive field size of VRs (Fig. S3*C*). However, this approach would cause many data points not to be covered by any receptive field—the network would be "blind" toward these data points (Fig. S3*D*). They could neither be used for training, nor could the trained network achieve correct classification to any data point in the "blind" areas; these data points would simply produce no input to the network. Moreover, as the density of VRs in different regions of data space may be different, choosing one RF size for all VRs is clearly not optimal.

Lateral inhibition solves this problem in an elegant way: The response of a VR to a data point will be attenuated by lateral inhibition on the PN level if another VR is closer. Every PN thus has an "authoritative" region in data space where it provides the highest response and the responses of PNs in other glomeruli are attenuated. This region is equivalent to the Voronoi partitioning of input space with the VRs as generators (symbolized by the dotted lines in Fig. S3*E*). The resulting PN receptive fields become narrower in regions where there is overlap, but retain their full extent in regions where no other PN competes (Fig. S3*F*). Hence, lateral inhibition between PNs optimally and efficiently partitions data space on the PN level. Each PN thus represents a region in input space for which it is authoritative, considerably simplifying the credit assignment problem.

Why does the naïve Bayes classifier benefit from lateral inhibition? This classifier estimates the mean μ and the variance σ^2 of each class along each dimension in its input space. Classification is then achieved by comparing the (naively) estimated probability of adherence to class 1 vs. class 2. These probabilities are computed from the multivariate normal distributions $N(\boldsymbol{\mu},\boldsymbol{\sigma}^2)$, with $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ the means and variances along each dimension of input space. Broad VR receptive fields entail high variance of VR responses; thus, the estimated variance of the multivariate response distribution will also be high. In contrast, PN responses exhibit smaller variance because their receptive fields are narrower. Thus, the estimated variance of the PN response distribution will be smaller, and in consequence the naïve Bayes estimate of class adherence will exhibit lower variance, allowing for a better discrimination of classes in data space.

SI Materials and Methods

Network Parameters. Each glomerulus was driven by six RNs and contained seven PNs and six LNs. Each population in the associative layer comprised eight excitatory and eight inhibitory neurons. Connectivity and synaptic weights are described in detail Table S1. For a schematic overview of the general network architecture, see Fig. 1*A* in the main text. Time constants in the table refer to the biological value they model. The actual values on the hardware are 10^4 times smaller, due to the 10^4 speedup factor at which the hardware operates (5, 6). The weights are specified as fractions of the maximal weight $w_{\max}^{\text{hw }\{\text{inh,exc}\}}$ for excitatory and inhibitory synapses in the hardware system, where $w_{\max}^{\text{hw inh}} \sim 4 \cdot w_{\max}^{\text{hw exc}}$. Neurons were implemented as standard integrate-and-fire models (see ref. 1 for details).

VRs. The response r of a VR with coordinates $\mathbf{p}$ to the stimulus $\mathbf{s}$ is given by Eq. **S1** as follows:

$$r = 1 - \frac{d(\mathbf{s},\mathbf{p}) - d_{\min}}{d_{\max} - d_{\min}}, \qquad \textbf{[S1]}$$

with $d(\mathbf{s},\mathbf{p})$ the Manhattan distance (Minkowski metric with $k = 1$, sum of absolute coordinate differences) between $\mathbf{s}$ and $\mathbf{p}$; $d_{\min}$ and $d_{\max}$ denote the minimum and maximum distance observed in the dataset. Hence, the receptor response is a value in [0, 1], and it is inversely proportional to the distance between stimulus and receptor.

The receptive fields implemented by Eq. **S1** are equivalent to linear radial basis functions representing cones. They extend over the entire space that is covered by the data ("broadly tuned"). Their receptive fields are largely overlapping. This guarantees that there are no "blind spots" in data space that are not covered by any receptive field.

VRs were placed in data space using a self-organizing process. In this study, we used the neural gas algorithm (7), as implemented in the MDP toolkit (8). The neural gas learns to represent the distribution of data in the original coordinate space, thus ensuring that the VRs cover data space appropriately. Each node in the neuronal gas corresponds to one VR. Using n VRs, a stimulus will thus evoke a response vector $\mathbf{r} = (r_1, \ldots, r_n)$. The elements of response vector r_i are then converted into firing rates ρ_i using Eq. **S2** as follows:

$$\rho_i = r_i \cdot (\rho_{\max} - \rho_{\min}) + \rho_{\min} \quad \text{for} \quad i = (1, \ldots, n), \qquad \textbf{[S2]}$$

with $\rho_{\min}$ and $\rho_{\max}$ the minimal and maximal firing rate, set to 20 and 70 spikes/s, respectively. Firing rates were transformed into spike trains using a gamma process of order five. The waiting time between stimulus onset and the first RN spike was drawn from the appropriate waiting time distribution, in our case a gamma distribution of order six, to prevent synchronization of RNs at stimulus onset. We chose a gamma process to generate spike times because its spiking statistics compares realistically to biological neurons (see, e.g., ref. 9). In addition, the increased regularity of a gamma process of order five [Fano factor (FF) = 0.2] compared with a Poisson process (FF = 1.0) reduces the spike count variability and thus yields a more reliable encoding of input firing rates.

VRs were implemented in software as a convenient approach to convert numerical data into a spiking format. The VR approach satisfies the need for dimensionality reduction due to limited neuron counts and provides a generic approach to convert real-valued data into bounded firing rate intervals.

Network Training and Supervised Learning Rule. The classifier network was trained using a supervised learning algorithm. Only synapses between PNs and excitatory association layer neurons were subject to learning.

After stimulus presentation, a synapse was eligible for weight update if it fulfilled a Hebbian eligibility constraint. A synaptic weight was eligible for updating if the target neuron υ_{target} was a member of the winner population Υ_{winner}, and if the firing rate ρ_{pre} of the presynaptic neuron during the previous stimulus presentation exceeded a threshold θ (fixed to 35 spikes/s in this study). The eligibility constraint ε can thus be formalized as follows:

$$\varepsilon = \begin{cases} 1, & \text{if } \rho_{\text{pre}} > \theta \text{ and } \upsilon_{\text{target}} \in \Upsilon_{\text{winner}}, \\ 0, & \text{otherwise}. \end{cases} \qquad \textbf{[S3]}$$

The change of the weight $\Delta w_{\text{PN}\to\upsilon}$ between any PN and target neuron υ in the association layer was governed by Eq. **S4** as follows:

$$\Delta w_{\text{PN}\to\upsilon} = \begin{cases} \varepsilon \cdot \text{c}, & \text{if classification was correct,} \\ -\varepsilon \cdot \text{c}, & \text{if classification was incorrect,} \end{cases} \qquad \textbf{[S4]}$$

with c a constant value determined by the granularity of synaptic weights on the hardware (1). The new weight w_{new} was computed from w_{old} as in Eq. **S5**:

$$w_{\text{new}} = w_{\text{old}} + \Delta w_{\text{PN}\to\upsilon}. \qquad \textbf{[S5]}$$

Synaptic weights were bounded in the interval $[w_{\min}, w_{\max}]$ by the constraints of the hardware. Thus, the final value of the synaptic weight was given by Eq. **S6** as follows:

$$w_{\mathrm{final}} = \begin{cases} w_{\max}, & \text{if } w_{\mathrm{new}} > w_{\max}, \\ w_{\min}, & \text{if } w_{\mathrm{new}} < w_{\min}, \\ w_{\mathrm{new}}, & \text{otherwise}. \end{cases} \qquad \textbf{[S6]}$$

Evaluation of Classifier Performance. Classifier performance was evaluated from fivefold cross-validation (CV). The data were split into five equal parts, and four parts were used in training and one part was used to test the classifier predictions in each CV run. After five runs, each data point was once in the test set, allowing computing a single performance value for all five CV runs. CV was repeated multiple times with different random splitting of the data into five equal parts.

Classifier performance (i.e., prediction accuracy) was assessed using Gorodkin's R_K correlation coefficient for discrete multicategory data (10). The aim is to compare a prediction $\mathbf{Y}_{\mathrm{pred}}$ to the true target values $\mathbf{Y}$, with $Y_{n,k} \in \{0,1\}$ for n predictions of k classes. The $K \times$ K confusion matrix $\mathbf{C}$ contains the number of correctly and falsely predicted data instances per class. $C_{k,k}$ contains the number of correctly predicted instances of class k, and off-diagonal elements contain the number of falsely predicted instances. For example, $C_{1,2}$ contains the number of instances predicted to belong to class 1, but actually belonging to class 2. The K-category correlation coefficient computes as in Eq. **S7**:

$$R_K = \frac{\sum_{klm} c_{k,k} c_{l,m} - c_{k,l} c_{m,k}}{\sqrt{\sum_k \left(\sum_l c_{k,l}\right)\left(\sum_{l',k' \neq k} c_{k',l'}\right)} \sqrt{\sum_k \left(\sum_l c_{l,k}\right)\left(\sum_{l',k' \neq k} c_{k',l'}\right)}}. \qquad \textbf{[S7]}$$

Compared with other frequently used performance measures like "percent correct," R_K is more sensitive to small performance differences when overall performance is already high and thus better suited for benchmarking. In addition, R_K is corrected for the bias introduced by skewed class proportions. For example, if 90% of the data are of one class and 10% the other class, we could yield "90% correct" classification by simply assigning all data samples to the first class. In contrast, R_K would report a value of zero, which is intuitively more accurate.

Application-Specific Calibration of the Neuromorphic Hardware System. The network-specific calibration for the previous version of the network with 100% connectivity (*SI Results, Network Optimization for Robustness Against Neuronal Variability*) consisted of two steps. We first calibrated the PNs for homogeneous rate response, before calibrating the LNs. Calibration was carried out with the weight of all inhibitory synapses set to zero. We first measured PN firing rates in response to a 1-s stimulation with nominal intensity, formed the median from all PN rates and used this as target firing rate. The "fitness" of the rate distribution was assessed by mean square deviation (MSD) of PN firing rates from the targeted PN firing rate as follows:

$$\mathrm{MSD} = \frac{1}{n} \sum_{1}^{n} \left(\rho_{\mathrm{goal}}^{\mathrm{PN}} - \rho_{\mathrm{i}}^{\mathrm{PN}}\right)^2, \qquad \textbf{[S8]}$$

with n the number of PNs, $\rho_{\mathrm{i}}^{\mathrm{PN}}$ the firing rate of the ith PN, and $\rho_{\mathrm{goal}}^{\mathrm{PN}}$ the targeted firing rate. The weights w_i from the RNs to the i^{th} PN were then updated according to the following:

$$w_i^{\mathrm{new}} = w_i \cdot \frac{\rho_{\mathrm{goal}}^{\mathrm{PN}}}{\rho_{\mathrm{i}}^{\mathrm{PN}}}. \qquad \textbf{[S9]}$$

In this case, we relied on the automatic conversion of the *Spikey* control software that mapped the weight values into the discrete distribution required by the hardware (1).

When the MSD failed to decrease over five iterations, optimization was terminated and the set of weights that yielded the best MSD until then was used. After the weights from RNs to PNs were optimized, we adjusted the weights between PNs and LNs using the same algorithm.

Speed Considerations for the Neuromorphic Hardware System. The execution of the network on the accelerated hardware happens extremely fast: A simulation lasting for 150-s biological time is executed in 15 ms (a 10^4 speedup factor). However, the total run time of the classifier network is mainly determined by other factors, which we describe in the following.

A typical CV run requires 150 stimulus presentations of 1-s duration. Before starting such a simulation session, generic calibration data must be loaded and applied. The network connectivity as well as synaptic weights must be encoded and transferred, and subsequently be mapped from their specification in biologically realistic physical units to the appropriate hardware parameters. In addition, for each of the 150 simulations, spike data need to be sent to and received from the hardware, including transfer, encoding, and decoding of spike times and neuron IDs. During the training phase of the classifier, synaptic weights also have to be updated before every stimulus presentation.

The absolute duration of these additional factors depends heavily on the efficiency of the software interface that links the hardware with the host system. Because it is a prototype system, this software interface is constantly developed and improved. It is therefore difficult to state an absolute number for the effective speedup achieved by offloading network simulations to the hardware. To give the reader the opportunity of an informed estimate, we analyzed how much time is required by each of the above steps (Fig. S4).

Several of these steps still bear potential for optimization. For example, the time required for weight update could be drastically shortened by differential configuration, i.e., updating only those hardware weights that have changed, instead of overwriting all weights as in the current implementation. In addition, on the current system all spike times produced in the network are being transferred back to the host system during training and testing phases of the classifier network. The interface can be improved to only transfer those spikes that are necessary for the off-chip calculation of the weight change, namely PNs and excitatory ANs, and not transferring spike times from LNs and inhibitory ANs. When the network is completely trained, only the spike times from excitatory ANs are needed, further reducing the overhead due to handling spike data. We plan to implement these optimizations in future versions of the software interface.

1. Pfeil T, et al. (2013) Six networks on a universal neuromorphic computing substrate. *Front Neurosci* 7:11.
2. Schmuker M, Schneider G (2007) Processing and classification of chemical data inspired by insect olfaction. *Proc Natl Acad Sci USA* 104(51):20285–20289.
3. Furber SB, et al. (2013) Overview of the SpiNNaker System Architecture. *IEEE Trans Comput* 62(12):2454–2467.
4. 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.
5. 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* (IEEE, Piscataway, NJ), pp 2784–2787.
6. Schemmel J, Gruebl A, Meier K, Mueller E (2006) *Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN)* (IEEE, Vancouver), pp 1–6.
7. Martinetz T, Schulten K (1991) A "neural-gas" network learns topologies. *Artificial Neural Networks*, eds Kohonen T, Mäkisara K, Simula O, Kangas J (Elsevier B.V., North-Holland, Amsterdam), pp 397–402.

8. Zito T, Wilbert N, Wiskott L, Berkes P (2008) Modular toolkit for Data Processing (MDP): A Python data processing framework. *Front Neuroinform* 2:8.

9. Nawrot MP, et al. (2008) Measurement of variability dynamics in cortical spike trains. *J Neurosci Methods* 169(2):374–390.

10. Gorodkin J (2004) Comparing two K-category assignments by a K-category correlation coefficient. *Comput Biol Chem* 28(5-6):367–374.

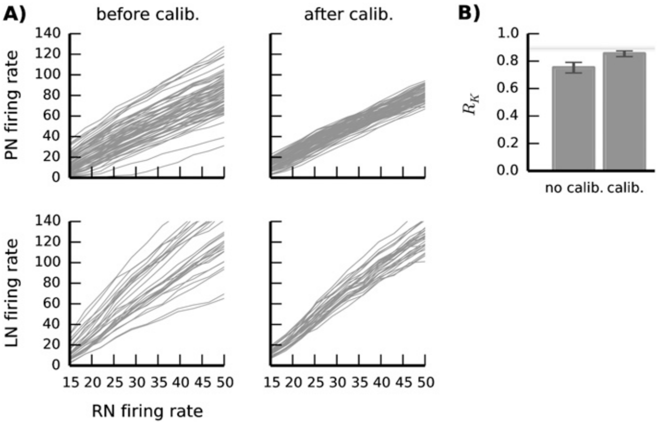

Fig. S1. Neuronal variability on the hardware system and impact of calibration on classifier performance using a previous version of the network with 100% connectivity. (*A*) Rate-response functions of the hardware neurons, before (*Left*) and after (*Right*) calibration (5-s stimulation duration). Upper row, PNs; lower row, LNs. (*B*) Classifier performance in the iris benchmark before and after network specific calibration. Error bars denote P^{20} and P^{80}. The horizontal gray bar indicates naïve Bayes performance.

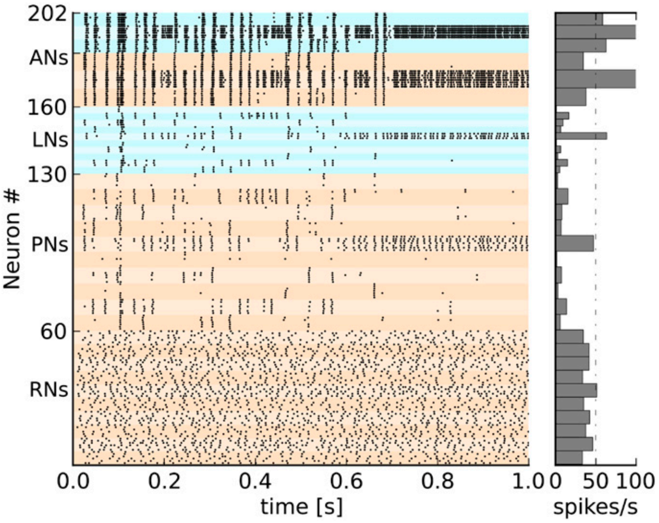

Fig. S2. Synchronized spiking activity in PNs, LNs, and ANs in a previous version of the network with p_{conn} = 100%. The total neuron count in the previous network is lower than in the version presented in the main text due to different per-population neuron counts for LNs (three in the previous network vs. six in the main text) and inhibitory ANs (six vs. eight).

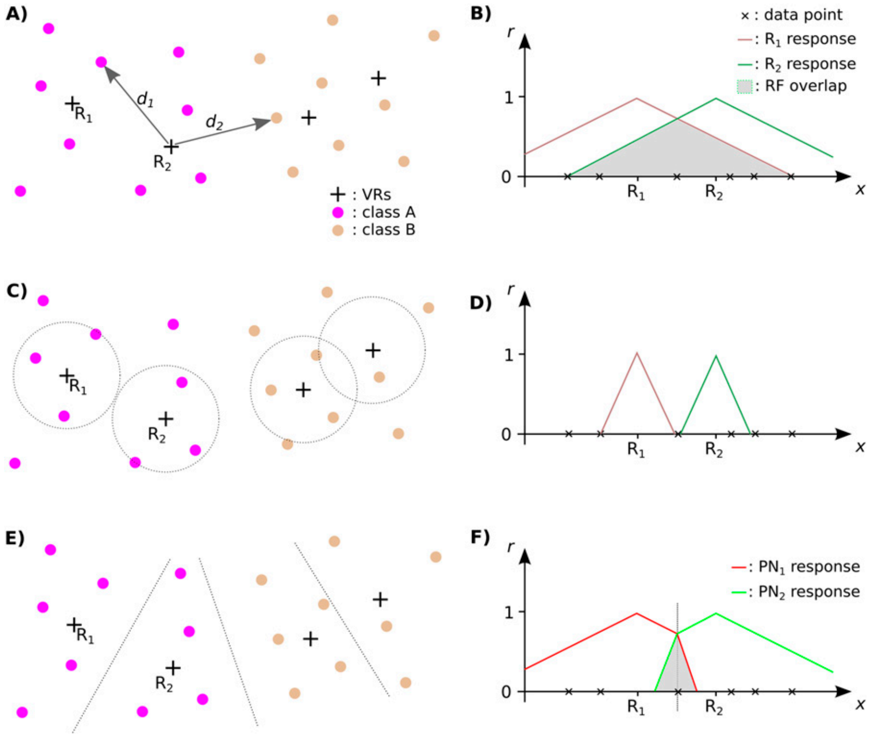

Fig. S3. Illustration of the credit assignment problem and the effect of lateral inhibition. (*A*) Cartoon of a hypothetical two-class, 2D classification problem with VRs. The distances d_1 and d_2 are equal. (*B*) One-dimensional sketch of the response profile of the two VRs, R_1 and R_2. (*C*) Effect of reducing VR receptive field size in data space. (*D*) Effect of reducing VR receptive field size on the response profiles. (*E*) Voronoi partitioning of input space with VRs as generators. (*F*) Effect of lateral inhibition on PN receptive field size.

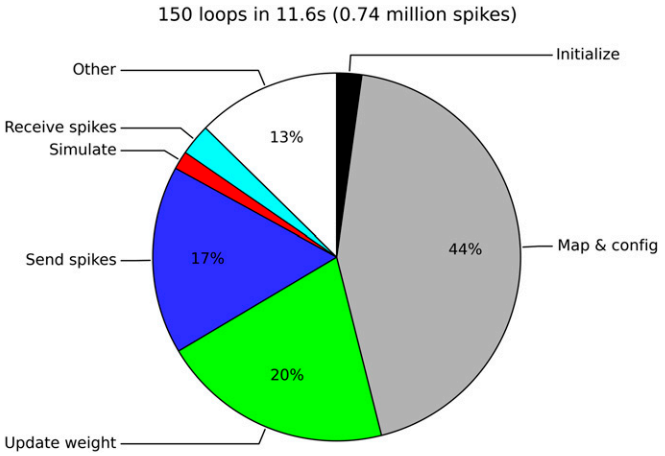

Fig. S4. Simulation time for one CV run (150 simulations) broken down into discrete steps. The largest fraction of the time is required by mapping the simulation parameters to hardware-compatible values and configuring the hardware network. Some of these tasks have to be repeated for every simulation, adding up to a substantial amount of total time. The second largest chunk is taken up by updating weights. The actual simulation requires less than 2% of the total time. "Other" encompasses numerous small tasks like handling of spike data and network configuration in the PyNN interface code. All numbers are subject to change as the software interface evolves.

Table S1. Network parameters

Type of neuron	Parameters
Receptor neurons (RNs)	
Type	Gamma process ($\gamma = 5$)
Count	Six RNs per VR
Outgoing connectivity	Each RN projects on the PNs in one glomerulus; connection probability $p_{\mathrm{conn}} = 50\%$
Outgoing weights	RN to PN: $0.5 \cdot w_{\mathrm{max}}^{\mathrm{hw\ exc}}$
Projection neurons (PNs)	
Type	Leaky integrate-and-fire
Count	Seven PNs per glomerulus
Outgoing connectivity	Excitatory synapses on LNs in the same glomerulus ($p_{\mathrm{conn}} = 50\%$) and on excitatory ANs ($\mathrm{p}_{\mathrm{conn}} = 50\%$)
Outgoing weights	PN to LN: $0.7 \cdot w_{\mathrm{max}}^{\mathrm{hw\ exc}}$ PN to AN: initially random between $0.2 \cdot w_{\mathrm{max}}^{\mathrm{hw\ exc}}$ and $0.66 \cdot w_{\mathrm{max}}^{\mathrm{hw\ exc}}$ (adjusted in training)
Local inhibitory neurons (LNs)	
Type	Leaky integrate-and-fire
Count	Six LNs per glomerulus
Outgoing connectivity	Inhibitory synapses on all PNs in all other glomeruli ($p_{\mathrm{conn}} = 100\%$)
Outgoing weights	LN to PNs: $0.133 \cdot w_{\mathrm{max}}^{\mathrm{hw\ inh}}$
Excitatory neurons in association layer (ANs)	
Type	Leaky integrate-and-fire
Count	Eight per association population
Outgoing connectivity	Excitatory synapses on adjoint inhibitory population ($p_{\mathrm{conn}} = 50\%$)
Outgoing weights	AN to adjoint inhibitory population: $0.5 \cdot w_{\mathrm{max}}^{\mathrm{hw\ exc}}$
Inhibitory neurons in association layer	
Type	Leaky integrate-and-fire
Count	Eight per association population
Outgoing connectivity	Inhibitory synapses on excitatory neurons of all other association populations ($p_{\mathrm{conn}} = 100\%$)
Outgoing weights	Inhibitory neuron to ANs in different association populations: $1.0 \cdot w_{\mathrm{max}}^{\mathrm{hw\ inh}}$

Table S2. Average count of predicted vs. actual class adherence (columns vs. rows) obtained across 50 repetitions of fivefold CV

	I. setosa	*I. versicolor*	*I. virginica*
I. setosa	50.0	0.0	0.0
I. versicolor	0.0	47.1	10.7
I. virginica	0.0	2.9	39.3