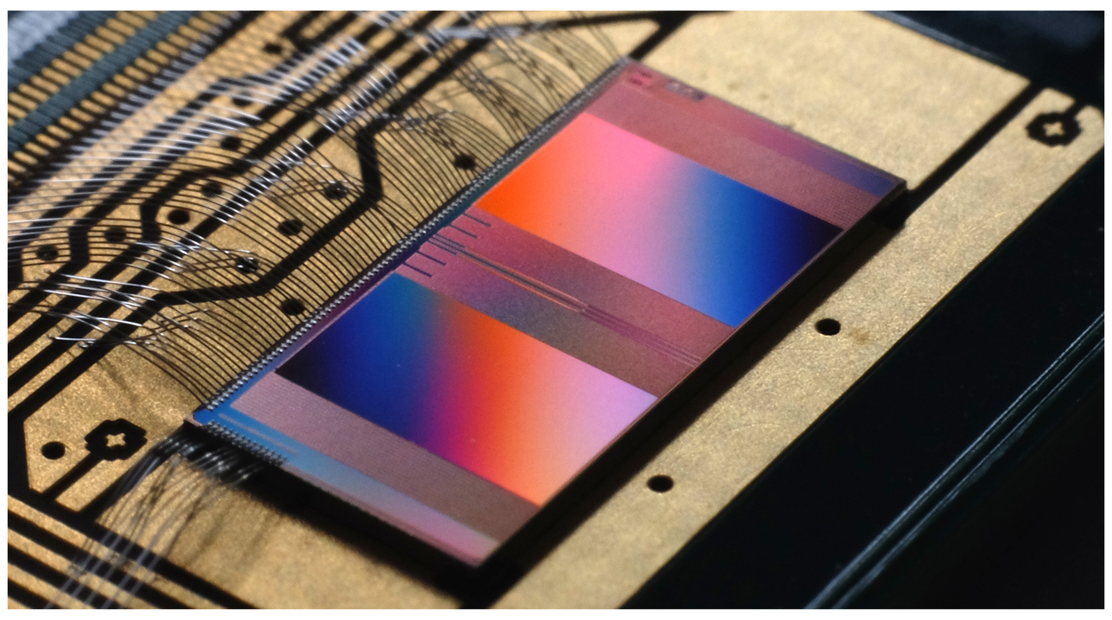
Structural plasticity on BrainScaleS-2

Sebastian Billaudelle*[1], Benjamin Cramer*[1], Mihai A. Petrovici[21], Korbinian Schreiber[1], Johannes Schemmel[1‡]

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

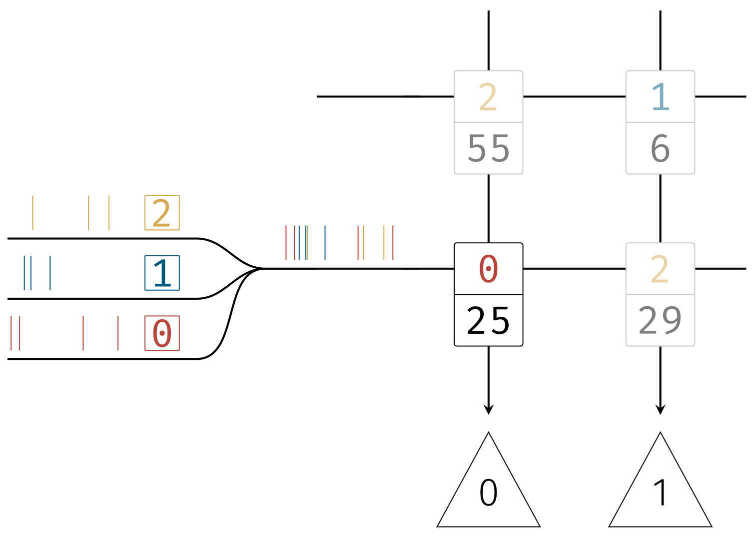
Introduction

In computational neuroscience, as well as in machine learning, neuromorphic devices promise an accelerated and scalable alternative to neural network simulations. Their neural connectivity and synaptic capacity depends on their specific design choices, but is always intrinsically limited. Here, we present a strategy to achieve structural plasticity that optimizes resource allocation under these constraints by constantly rewiring the pre- and postsynaptic partners while keeping the neuronal fan-in constant and the connectome sparse. In particular, we implemented this algorithm on the analog neuromorphic system BrainScaleS-2. It was executed on a custom embedded digital processor located on chip, accompanying the mixed-signal substrate of spiking neurons and synapse circuits. We evaluated our implementation in a simple supervised learning scenario, showing its ability to optimize the network topology with respect to the nature of its training data, as well as its overall computational efficiency.



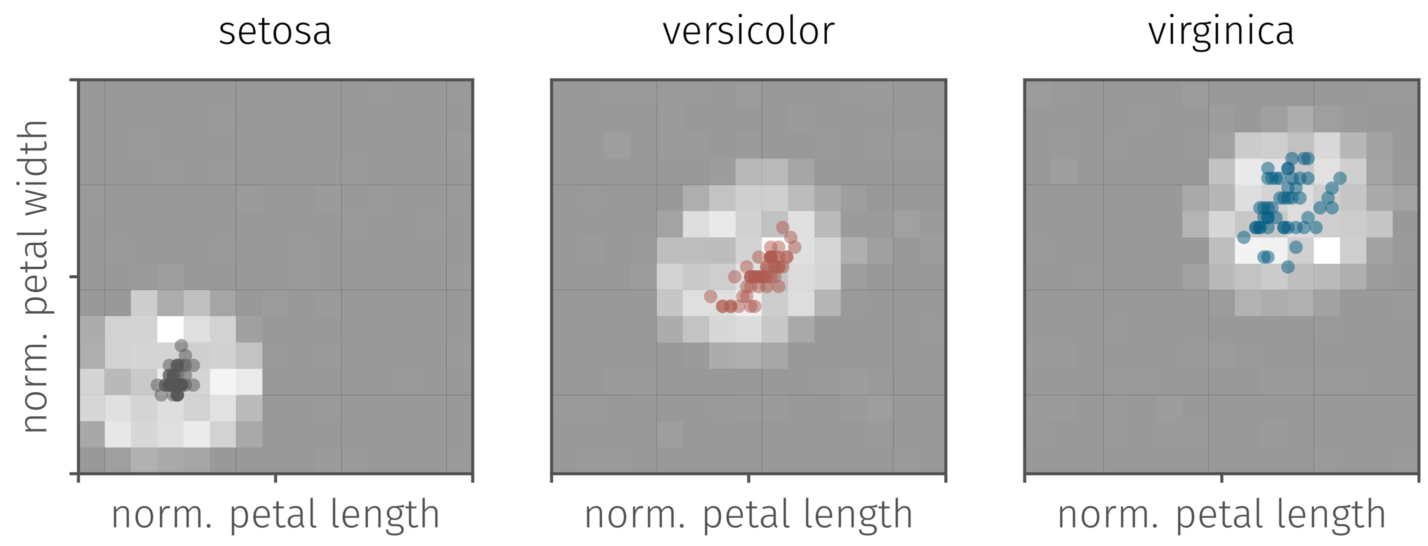
Synaptic event filtering enables structural plasticity

```
for row in 0 ... 31 do
    w ← synram_weights_read(row)
    w ← w + alpha *
    min(f_max,correlation_read(row))
    w ← w - beta * w * rates_read()
    w ← w + gamma * rng(-1,1)
    if w < theta_w then
        w ← w_init
        a ← rng(0,k)
        synram_labels_write(row,a)
    end if
    synram_weights_write(row,w)
end for
```



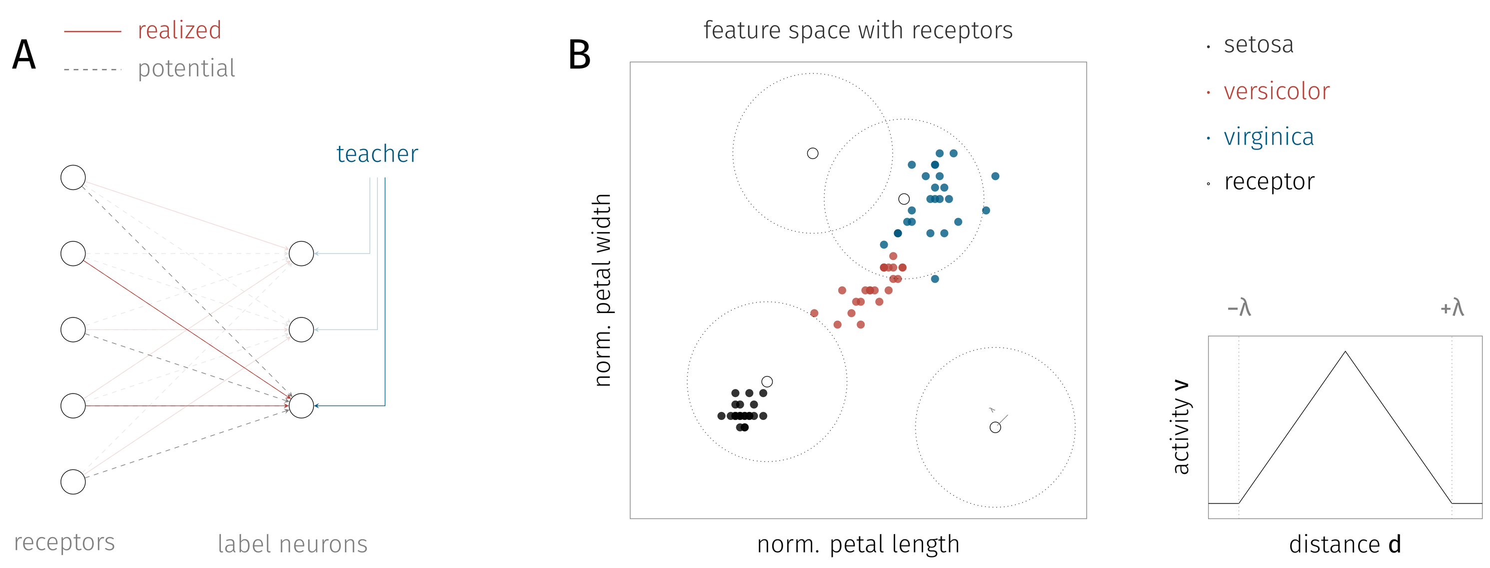
Events are identified with an address denoting their source. Spike trains from different origins can be overlayed and injected into a single synapse row. Synapses filter afferent events by comparing the source address to a label stored in their local SRAM and forward only matching spikes. Addresses and labels can be reconfigured by the PPU to implement weight dynamics and structural changes.

Self-organized formation of receptive fields



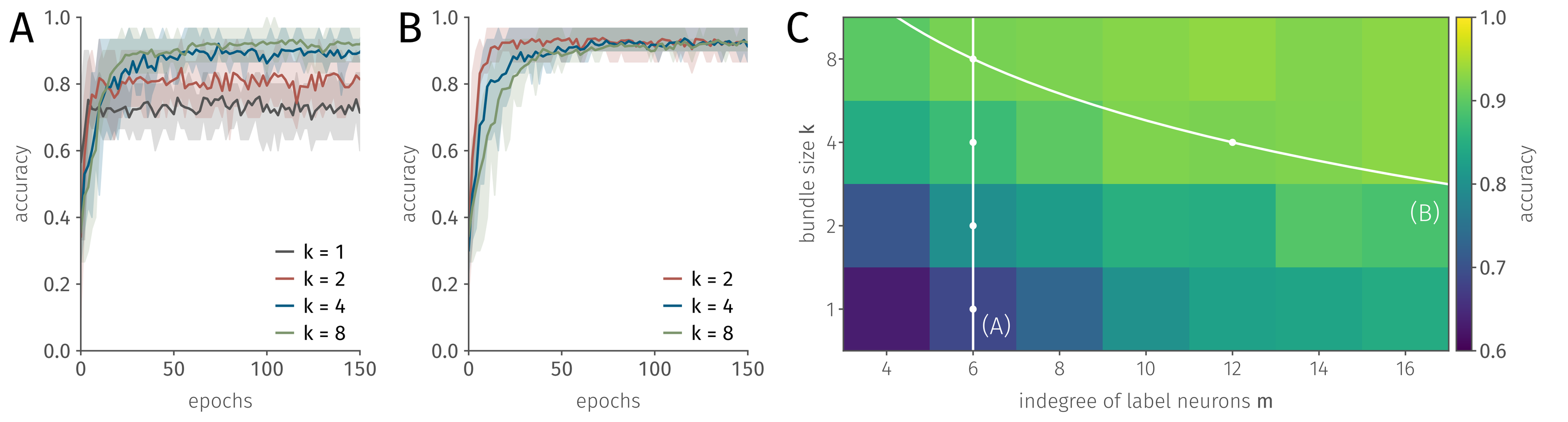
Structural plasticity leads to the self-organized formation of receptive fields. The probability of synapse expression depends on the location of receptors in the feature space and the class of label neurons. The size of the three emerging clusters is determined by the receptor radius λ.

Learning receptive fields in a feed-forward network



The two-layer network consists of a group of receptors and a label population. One teacher per label neuron ensures excitation of the correct labels during learning. The inputs project onto the label layer with a potential all-to-all connectivity (gray), but only a subset of synapses is realized (red).

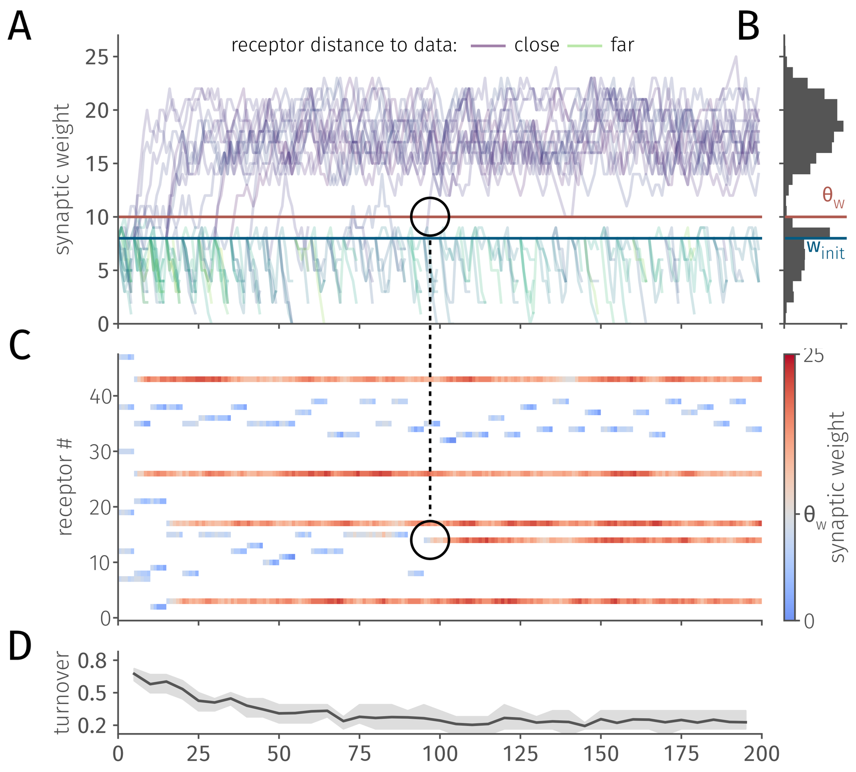
Structural plasticity improves learning in sparse networks



For a constant indegree **m** of the label neurons (equivalent with the number of synapse rows on the hardware), classification accuracy improves with larger **k** (number of receptors per row), as the neurons gain access to an increasing number of receptors **n = km**. For a constant number of receptors **n**, structural plasticity can compensate for increased sparsity (reduced indegree **m** induced by a larger bundle size **k**) up to a certain degree.

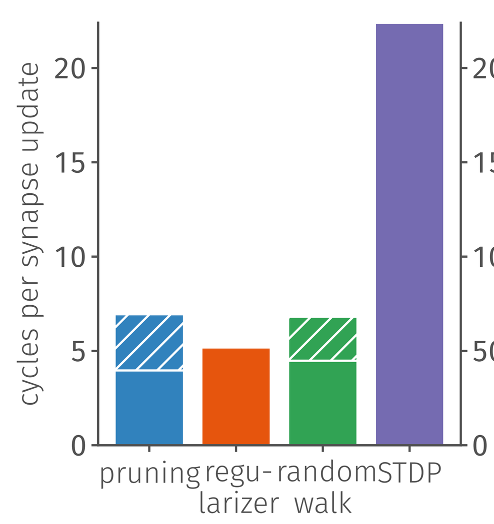
Informative synapses emerge during training

Exemplary evolution during the course of a single experiment. Synapses that receive inputs from relevant receptors (i.e., those lying close to the features that are relevant for their postsynaptic label neuron) are strengthened towards values that lie above the pruning threshold θ_w. All other, less informative synapses remain below θ_w and are pruned at regular intervals of five epochs.



Efficient implementation of structural plasticity

Duration of a synapse update broken down into its four individual contributions, including structural reconfiguration. The hatched areas indicate the time spent on pseudo-random number generation. Contributions of the individual terms to the overall update duration, taking into consideration that pruning and reassignment are executed five times less often than synaptic weight updates.



As compared to other synaptic pruning and reassignment strategies, our algorithm and implementation of structural plasticity requires a particularly low overhead. Our implementation scales well with growing system sizes, since it is fully based on synapse-local quantities.

References

[1] S. Friedmann and J. Schemmel et al., "Demonstrating hybrid learning in a flexible neuromorphic hardware system," IEEE Transactions on Biomedical Circuits and Systems, vol. 11, no. 1, pp. 128–142, 2017.

[2] S. Billaudelle, Y. Stradmann, K. Schreiber, B. Cramer, A. Baumbach, D. Dold, J. Göltz, A. F. Kungl, T. C. Wunderlich, A. Hartel et al., "Versatile emulation of spiking neural networks on an accelerated neuromorphic substrate," in 2020 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2020, pp. 1–5.

[3] S. Billaudelle, B. Cramer, M. A. Petrovici, K. Schreiber, D. Kappel, J. Schemmel, and K. Meier, "Structural plasticity on an accelerated analog neuromorphic hardware system," Neural Networks, vol. 133, pp. 11–20, 2021.

This research has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nos. 720270, 785907 and 945539 (HBP) and the Manfred Stärk Foundation. * equal contribution