Object Recognition Model Network (SGA2 Milestone 9.5.3)

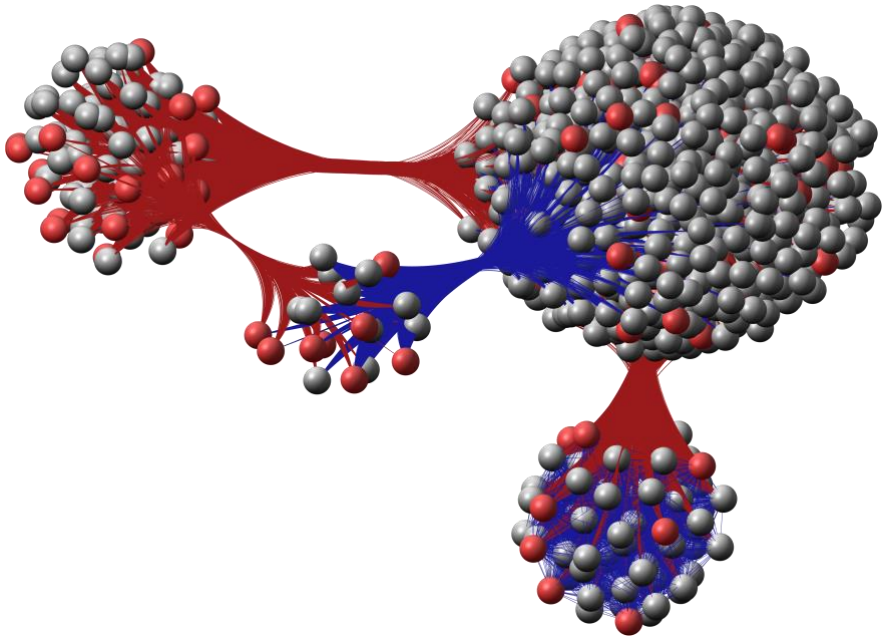

Figure 1: Schematic of the insect mushroom body model underlying the neuromorphic object recognition network described in this milestone report

The model populations represent antennal lobe (left), lateral horn (front), mushroom body Kenyon cells (right top) and mushroom body output neurons (right bottom), as described in detail in [4]. Functionally, the network is essentially a feedforward network not unlike a multi-layer perceptron, albeit we needed to augment it to function properly within the BrainScaleS I feature set (see Figure 2).

Project Number:	785907	Project Title:	Human Brain Project SGA2

Document Title:	Object Recognition Model Network (SGA2 Milestone 9.5.3)
Document Filename:	HBP_SGA2_Milestone_9.5.3_v2.docx
Milestone Number:	SGA2 MS9.5.3 (MSXXX)
Deliverable Type:	Milestone Report
Work Package(s):	WP9.5
Dissemination Level:	CO = Confidential
Planned Delivery Date:	SGA2 M12 / 31 Mar 2019
Actual Delivery Date:	SGA2 M12 / 29 Mar 2019

Authors:	Garibaldi PINEDA GARCIA, UoS (P106) Thomas NOWOTNY, UoS (P106)
Compiling Editors:	Thomas NOWOTNY, UoS (P106)
Contributors:	
SciTechCoord Review:	
Editorial Review:	

Abstract:	In this milestone report we describe our initial object recognition model network built with standard elements supported on the BrainScaleS system and show proof of concept results that demonstrate that the network model is working on a simple object recognition task.
Keywords:	BrainScales I, neuromorphic algorithm, object recognition

Table of Contents

Table of Tables

Table of Figures

Object Recognition Model Network (SGA2 Milestone 9.5.3)

1. Introduction

Object recognition is a key component of the day-to-day experience of animals, from bees detecting flowers' typical scents to humans recognizing icons on a computer screen. In this report we present a spiking neural network (SNN) which is able to cluster its inputs via unsupervised learning. The basic model presented here will then be used in further research aiming to modify the network architecture and/or plasticity parameters through an outer optimization loop to improve efficiency of the unsupervised learning algorithm.
Models for object recognition inspired by insect anatomy have proven to be useful for multiple tasks [4,2]. These models are realised as SNNs; recently, specialized hardware has been developed to simulate such networks in an efficient manner [3]. However, achieving state-of-the-art performance in complex classification problems has proven difficult, partially due to a large number of unconstrained parameters. A recent trend for challenges of this kind is learning-to-learn [8] or transfer learning approaches in which a variety of learning methods, including gradient-based and gradient-free algorithms are used to modify (learn) network (meta-) parameters [6,1].

2. Results

The basic model is based on insect anatomy where signals from the environment are transformed by sensory neurons, in the case of olfaction, olfactory receptor neurons, which, in turn, generate a relatively sparse representation of that information in primary sensory brain regions (antennal lobe (AL) in Figure 2). For our experiments we produced a proof of concept with noisy samples of 10 arbitrary sparse vectors of dimension 100, representing the activity patterns of 100 projection neurons in the AL. In these patterns, 20% of the neurons where active simultaneously. One of the samples is presented to the network every 50ms with an additional, random temporal shift in the range of $[-0.5, 0.5]$ ms (with a 0.1ms resolution) for each spike. This reflects noise in the typical input received in a sensory system but was here done mainly to enable competition through mutual inhibition at the level of KC and DN populations. The network then projects the AL representations into an even higher-dimensional space by sparsely connecting to a

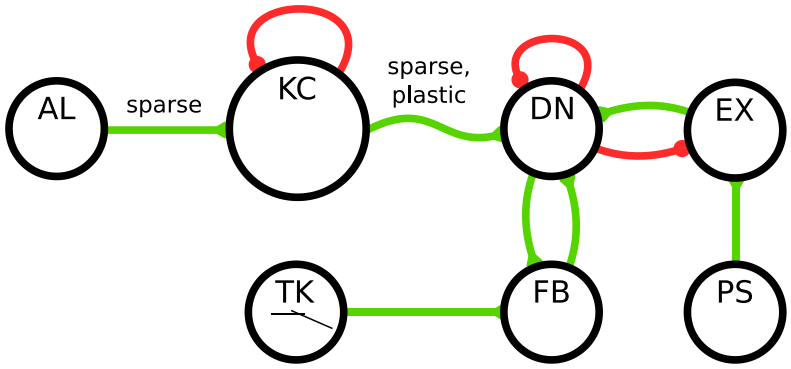

Figure 2: Simplified diagram of the network architecture after the additional populations have been introduced.

AL, KC, DN populations as before. EX population provides excitatory feedback to DN neurons except after they were active, FB provides excitatory feedback to DN to induce an additional "depression spike". EX and FB can be turned on and off by PS and TK respectively.

larger population (2500 Kenyon cells (KC) in Figure 2). Mutual inhibition between KCs reduces the number of active units. Plastic synapses connect the KC population with the output layer (detector neurons (DN) in Figure 2) which serves as a classifier for the network's input.

Because the final target for this work is the BrainScaleS platform [3] we constrained the learning rule to basic spike-timing-dependent plasticity (STDP) [7] and the neurons to leaky integrate-and-fire (LIF) neurons. These constraints required the introduction of additional neuron populations. The first additional population (FB in Figure 2) provides a feedback signal so that neurons which just spiked will do so again just before a new pattern comes into the network. Since the pre-synaptic KC population will not have been active for about 40ms prior to this additional spike, the spike leads to additional depression of synaptic weights, effectively "extending" the weight-depressing region of the STDP rule (Figure 3).

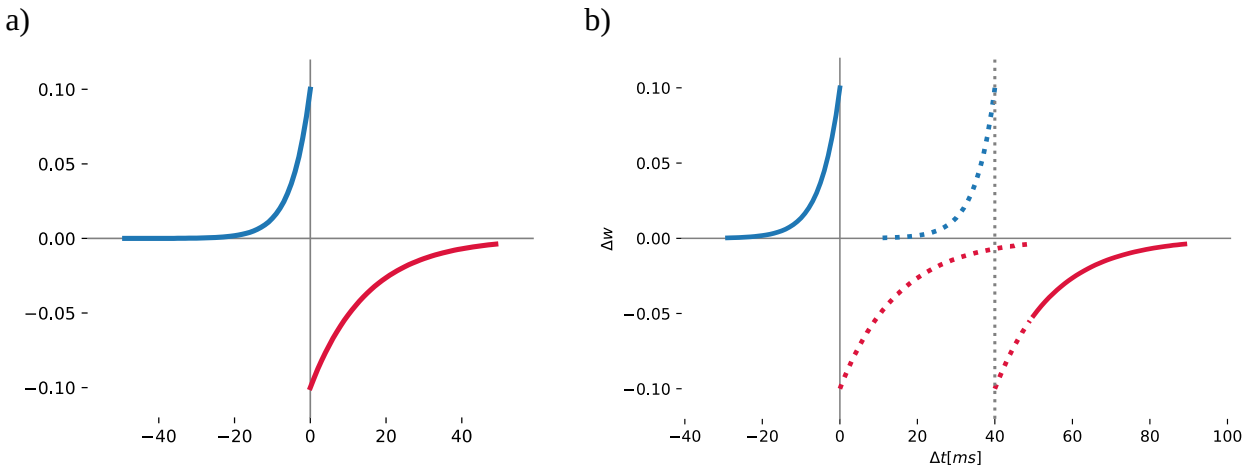

Figure 3: STDP curve and extension effect.

a) The generated curves for parameters $A_+ = 0.1, A_- = 0.1, \tau_+ = 5$ ms and $\tau_- = 15$ ms. b) The extension effect of the feedback excitation to output neurons; in the gray region (dotted curves) no pre- and post-synaptic spike pairs are present thus so no weights change will occur.

The second additional population (EX in Figure 2b) provides high frequency spikes through low-efficacy synapses to output neurons. Whenever an output neuron fires, it provides strong inhibition to a particular EX neuron, thus blocking excitation to itself for around 200 ms (Figure 4). We can think of this interaction as increasing the relative likelihood of spiking for neurons which have not yet spiked.

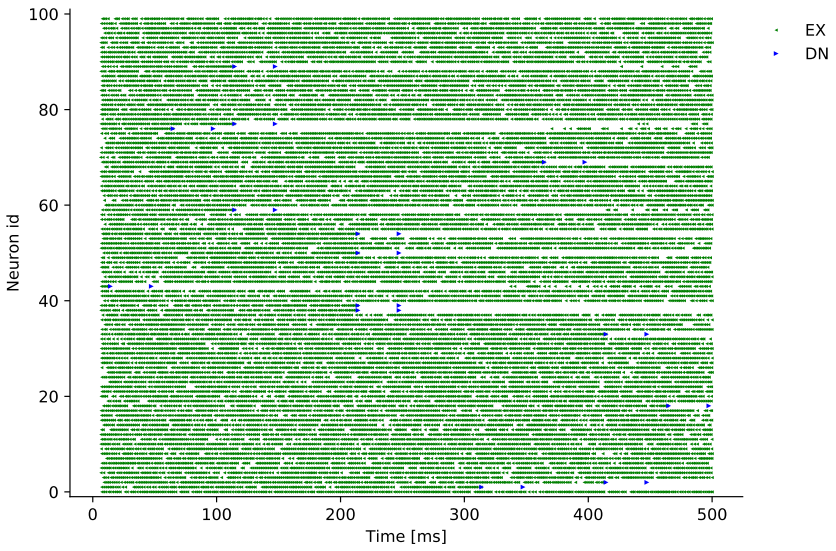

Figure 4: Spiking activity from DN and EX populations.

Whenever a DN neuron spikes (small blue triangle), the corresponding EX neuron stops spiking for a while (green dots).

Initial testing of the model was performed using the GeNN simulator as a backend [9]. We generated 1000 noisy samples of each input pattern (i.e. 10000 total samples for 10 classes), 85% of which were used for training and 15% for testing. We enabled spiking from the EX population for

Table 1: STDP parameters used in our proof of concept runs.

w_{min}	w_{max}	A_+	A_-	τ_+	τ_-
-0.36	0.18	0.1	0.1	5 ms	15 ms

the initial learning period which was defined as the time from the start until on average 100 samples per pattern had been seen, e.g. 50 seconds in the case of 10 classses. Feedback from from the FB population was on during the entire learning period. Both populations were silenced during testing. Plastic synapses were modified using the basic STDP rule with parameters as presented in Table 1. Note that synapses' weights can change sign (i.e. from excitatory to inhibitory and vice versa). Although this is not biologically plausible it was required for the network to effectively classify the inputs.

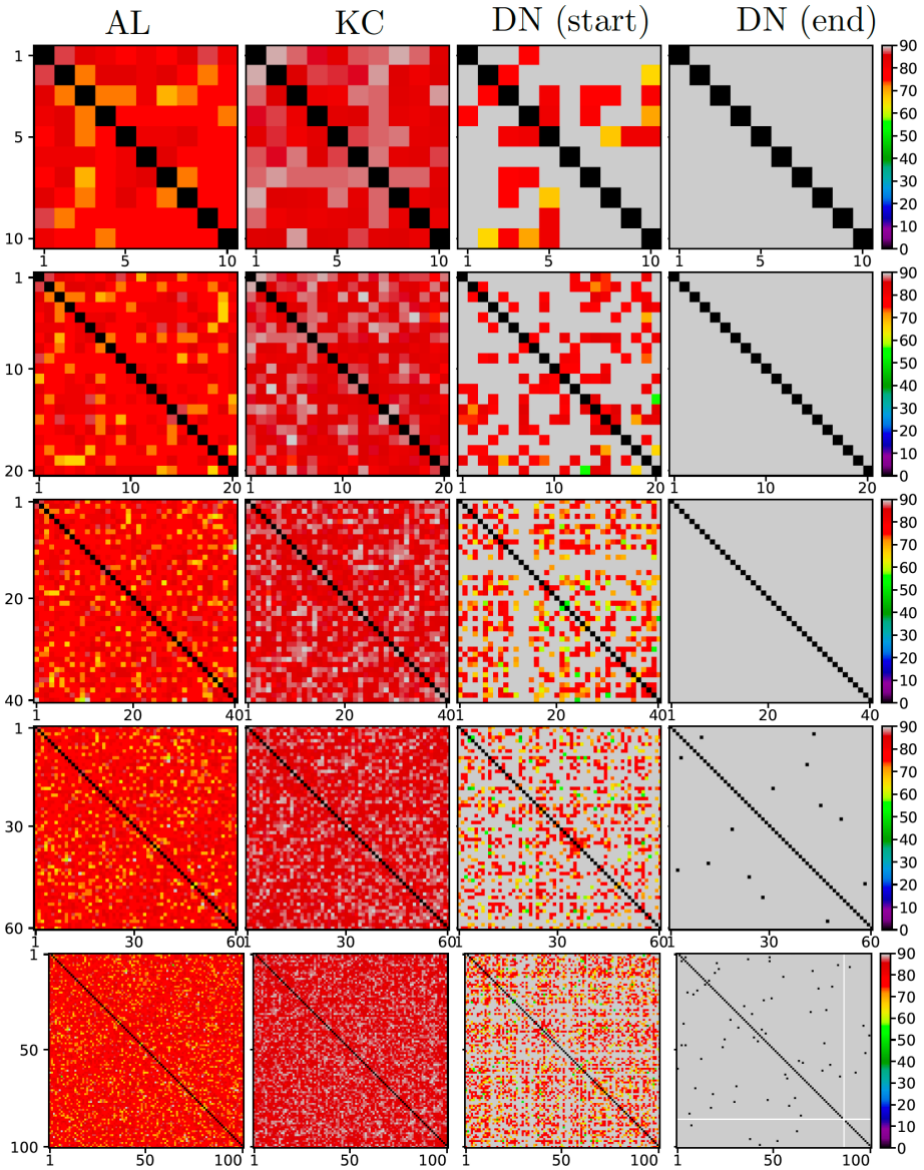

Figure 5: Object recognition results at the different layers of the model network.

The colours illustrate the angle between the activity vectors in the different populations AL, KC, DN (before learning), DN (after learning) in the network. After learning, output neurons in the DN population respond exclusively to one type of input patterns. Top row for 10 classes as described in the main text, subsequent rows for 20, 40, 60 and 100 classes, respectively.

We recorded spiking activity from the AL, KC and DN populations. In order to analyze the workings of the network model we generate "activity vectors" for each input pattern. If a neuron in a population spikes within the 50 ms analysis window, we add it to that population's representation of the input pattern. After computing the population representations we measure the angle between all pairs of vectors within each population to characterize the amount of overlap in neuron activity. The lower the angle, the more similar are the representations and the harder it is for these patterns

to be distinguished at the level of the population in question. In Figure 5, columns represent the AL (input), KC, DN (at the start of the experiment) and DN (at the end of the experiment) and the colours show the angles between pairs of patterns. The patterns are ordered according to their class, i.e. the original sparse pattern they were derived from. As can be seen, the activity patterns for different inputs are usually overlapping in AL and KC and angles between most vectors at these stages are less than 90°. The vectors for the output population at the beginning of the experiment are, in most cases also less than 90°. But by the end of the experiment the network has learned which input has been assigned to a certain output. Accordingly, the rightmost column in Figure 5 shows that neurons in the output population which spike for a particular class of inputs will not spike for others.

3. Conclusions

We have designed and tested a network architecture for object recognition (classification); although based on insect anatomy some adaptations where needed to train the network within the framework available on the BrainScaleS I system. The tests we have performed on the model have shown great performance for the current set of "toy" inputs though we are planning to increase the complexity of the recognition problem soon. Our next steps are to port the network and evaluate it on the actual BrainScaleS I platform. To do this we will need to work on different aspects of mapping the network to the hardware (e.g. population-to-chip mapping, weight discretization, STDP mechanism).
All software which has so far been developed is available in a Github repository [5] under an open source licence.

4. References

[1] Guillaume Bellec et al. "Long short-term memory and learning-to-learn in networks of spiking neurons". In: Advances in Neural Information Processing Systems. 2018, pp. 795–805.
[2] Ramon Huerta and Thomas Nowotny. "Fast and robust learning by reinforcement signals: explorations in the insect brain". Neural Computation 21.8 (2009), pp. 2123–2151.
[3] Karlheinz Meier. "A mixed-signal universal neuromorphic computing system". In: 2015 IEEE International Electron Devices Meeting (IEDM) IEEE. 2015, pp. 4–6.
[4] Thomas Nowotny et al. "Self-organization in the olfactory system: one shot odor recognition in insects". Biological cybernetics 93.6 (2005), pp. 436–446.
[5] Garibaldi Pineda García. Software repository. 2019. url: https://github.com/chanokin/brainscales-recognition/tree/master/codebase.
[6] Sebastian Schmitt et al. "Neuromorphic hardware in the loop: Training a deep spiking network on the BrainScaleS wafer-scale system". In: 2017 International Joint Conference on Neural Networks (IJCNN) IEEE. 2017, pp. 2227–2234.
[7] J. Sjöström and W. Gerstner. "Spike-timing dependent plasticity". Scholarpedia 5.2 (2010). revision #184913, p. 1362. doi: 10.4249/scholarpedia.1362.
[8] Sebastian Thrun and L. Pratt. "Learning to Learn: Introduction and Overview". In: Learning To Learn. Ed. by S. Thrun and L. Pratt. Kluwer Academic Publishers, Jan. 1998.
[9] Esin Yavuz, James Turner, and Thomas Nowotny. "GeNN: a code generation framework for accelerated brain simulations". Scientific reports 6 (2016), p. 18854.