Reproducing Fear Conditioning of Rats with Unmanned Ground Vehicles and Neuromorphic Systems

Noah Zins, Hongyu An
Department of Electrical and Computer Engineering
Michigan Technological University, Michigan, USA
nwzins@mtu.edu, hongyua@mtu.edu

2023 24th International Symposium on Quality Electronic Design (ISQED) | 979-8-3503-3475-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISQED57927.2023.10129317

Abstract

Deep learning accomplishes remarkable success through training with massively labeled datasets. However, the high demands on the datasets impede the feasibility of deep learning in edge computing scenarios and suffer the data scarcity issue. Rather than relying on labeled data, animals learn by interacting with their surroundings and memorizing the relationship between concurrent events. This learning paradigm is referred to as associative memory. The successful implementation of associative memory potentially achieves self-learning schemes analogous to animals to resolve the challenges of deep learning. The state-of-the-art implementations of associative memory are limited to small-scale and offline paradigms. Thus, in this work, we implement associative memory learning with an Unmanned Ground Vehicle (UGV) and neuromorphic chips (Intel Loihi) for an online learning scenario. Our system reproduces the classic associative memory in rats. In specific, our system successfully reproduces the fear conditioning with no pretraining procedure and labeled datasets. In our experiments, the UGV serves as a substitute for the rats. Our UGV autonomously memorizes the cause-and-effect of the light stimulus and vibration stimulus, then exhibits a movement response. During associative memory learning, the synaptic weights are updated by Hebbian learning. The Intel Loihi chip is integrated with our online learning system for processing visual signals. Its average power usages for computing logic and memory are 30 mW and 29 mW, respectively.

Keywords

Associative Memory; Hebbian Learning; Neuromorphic Computing; Unmanned Ground Vehicles.

1 Introduction

Deep learning demonstrates remarkable capabilities in multiple cognitive tasks, such as image recognition, self-driving vehicles, and natural language processing [1]. These capabilities stem from the training of large-scale Deep Neural Networks (DNNs) with a large amount of data [1]. However, the huge datasets and large-scale DNNs significantly prolong training time and increase energy demands. The excessive demand for computational resources makes the application of deep learning highly reliant on supercomputers. Unfortunately, these bulky supercomputers cannot meet applications with strict requirements of Size, Weight, and Power (SWaP), such as deep-ocean and planetary exploration [1, 2]. In addition, it is costly and laborious to build massive and labeled data sets, and sometimes the data is not practical to collect. For instance, the terrain data on the Moon and Mars are both extremely difficult to collect [2].

A modern and novel approach for implementing Artificial Intelligence (AI) is proposed to resolve these challenges: Neuromorphic Computing (NC). NC is a software and hardware co-design approach to achieving a power-efficient AI system by emulating nervous systems using hardware, algorithms, and computational models. In an NC system, the neurons are biologically plausible neuron models, such as Leaky Integrate and Fire neurons (LIF) [3]. The signal communication among neurons is low-frequency spikes in a neuromorphic system forming Spiking Neural Networks (SNN) [4]. Several training methods for SNNs have been proposed, including converting traditional ANNs into an SNN [4], biologically plausible algorithms [4], the approximation methods [4], etc. These training methods still utilize labeled datasets rather than mimic the learning process of animals. In real-world scenarios, animals learn from interacting with their surroundings and memorizing concurrent events. This self-learning scheme is referred to as associative memory [5-7]. Several studies attempted to implement associative memory using neuromorphic computing [8], electronic neurons, and synapse [6, 9-11]. However, these studies merely complete a small-scale association with a few neurons in simulation platforms rather than the experiments in real-world scenarios. Furthermore, the process of pretraining with labeled datasets is still required for these studies [12-16]. Thus, in this work, we design a large-scale neuromorphic system with associative memory and deploy it to an Unmanned Ground Vehicle (UGV) for online learning through interacting with surroundings in real-world scenarios. To our best knowledge, we for the first time implement associative memory into a UGV to complement real-time online learning by directly constant surroundings with no pre-trained procedure. The Loihi chip is integrated with UGVs to further enhance energy efficiency [17]. In specific, we utilize our UGV to reproduce the fear conditioning experiment with rats. In our experiment, the brightness of a light emulates the visual stimulus, and the vibration measured with the accelerometer mimics the electric shock to rats. The fear response, escaping, will be emulated by a motion of the UGV away from the vibration source. The perception of the light and the vibration are separately processed within two different neural assemblies. In our experimental design, vibration is the unpleasurable stimulus and light is the neutral stimulus. The signal pathway from vibration unconditionally evokes the movement of the UGV, so it is referred to as the unconditional signal pathway. Meanwhile, the light stimulus does not stimulate a movement response until the associative memory learning accomplished, making it is a conditional signal pathway.

The contributions of this paper are summarized as follows:

1) Unlike other off-line associative memory [12-16], to our best knowledge, we first implemented associative memory with a UGV in an online learning scenario using an Intel Loihi chip, achieving an average power usage of 59 mW.
2) The work reproduces the classic fear conditioning of rats with solid biological rationales from a cellular level (Hebbian learning) to the behavior level (fear conditioning).
3) No pre-training process and labeled dataset are required.

2 Background of Associative Memory

The underlying mechanism behind associative memory is the synaptic plasticity among neurons. The studies on the sea slug: Aplysia [5] reveal the causality between synaptic plasticity and associative memory. Two signal pathways of Aplysia are from the siphon to the gill and from the tail to the gill. Normally, the gill neuron does not respond to siphon stimulation, which is analogous to the sound of the bell in fear conditioning experiments on dogs. The underlying reason for this nonresponse comes from the blocked signal pathway from the siphon to the gill. The signals received from the stimulus of the siphon cannot be delivered to the gill of Aplysia due to the attenuation of the synaptic connection strength between them. However, this blockage caused by the synapse can be changed if the stimulus from the siphon and tail are applied at the same time or with a small lag [5]. If the stimulus from the siphon and tail are provided at the same time, the signals overlap with each other. The paired signals will enhance the synaptic connection strength between the siphon and gill. As a result, the signals crossing them are not significantly attenuated. The larger signal from the siphon of Aplysia will be transferred to the response neuron of its gill, leading to the gill contracting.

In more advanced animals, the perception neurons are not simply one neuron such as in Aplysia. The captured signals in these animals are processed with a group of neurons known as neural assemblies [18]. For instance, the electrical shock and sound signals in the rats are processed at the auditory and somatosensory thalamus, respectively. The output signals from these neural assemblies merge at the lateral nucleus. This is analogous to what happens in Aplysia. The main difference is that the individual neurons in Aplysia are replaced by the assemblies of neurons in rats. Associative learning can be conducted with the similar training procedures. Initially, the rats ignore a neutral tone as the signal pathway from the Lateral Nucleus is blocked. When the tone is presented immediately before a foot shock, the animal learns to associate the tone with the shock. After multiple repetitions, the tone alone will provoke a fear response as well. As the neural tone stimulus will not stimulate a fear response without an associative memory process, it is referred to as a conditional stimulus (CS). Meanwhile, the shock stimulus is defined as an unconditional stimulus (US) because it unconditionally stimulates the fear response [5].

3 Reproducing Fear Conditioning with UGV

With a comprehensive understanding of the fear conditioning of rats and the associative memory mechanism in Aplysia, we design a neuromorphic system to reproduce the fear conditioning of rats and validate it with experiments. In our experiment, a UGV is placed on a vibration platform and unconditionally responds to the vibration signals. These vibration signals emulate the unpleasurable/fear stimulus in the fear conditioning experiment in rats. In addition, the light is provided as a conditional stimulus (CS). To render the UGV capable of reproducing a similar response to fear conditioning, we design several special types of neurons to convert the brightness of light and acceleration of the vibration platform into spiking signals. Moreover, the movement of UGV is controlled by the motion neurons we designed. All these neurons are customized from classic Leaky Integrate and Fire (LIF) neurons. The LIF neurons are expressed with the equations [19]:

$$C_m \frac{dV_m}{dt} = G_L(E_L - V_m) + A * I_{app}; \quad if\ V_m > V_{th}\ then\ V_m = V_{reset}, \tag{1}$$

$$\tau_{RC} = C_m / G_L \tag{2}$$

where C_m defines the membrane capacitance, V_m is the membrane potential, G_L is the leak conductance, E_L is the leak potential, I_{app} is the applied input current, A is the input signal gain, and τ_{RC} is the membrane RC time constant. For all the neurons in Table 1, the membrane potential is fixed at 1 V and input gain is modified instead. They all use the Nengo LIF model's default τ_{RC} of 0.02 seconds because it was sufficient for our desired functionality. The other two parameters are calculated and optimized based on our experimental setups so that they can produce the desired responses for their respective uses. Specifically, the vibration detection neuron's gain (A) and bias (V_{reset}) were empirically derived so that it fires with vibration stimulus input but not small sudden movements. The motion neuron is a typical LIF configured to spike whenever it receives any sustained input spikes, either from vibration neurons or brightness neurons. The brightness neuron is the Layer 3 neuron in Table 3 and is a LIF neuron with empirically derived gain and bias so that it fires only when the light feature neurons have a high enough collective output.

Table 1: LIF neuron parameters

Neuron Types	τ_{RC}	A	V_{reset} (V)	V_{th}(V)
Vibration neuron	0.02	1.3	0.6	1.0
Brightness neuron	0.02	0.3	-1.0	1.0
Motion neuron	0.02	1.0	0.01	1.0

3.1 Neuron Coding for Perception and Movement of UGV

The accelerometer within the onboard inertial measurement unit (IMU) is used to measure the acceleration of UGV at the vibration platform. These measured accelerations of the three axes are illustrated in Figure 1. The vibration stimuli is from the vibration platform operating at 25 Hz and 1.2 mm amplitude. As shown in Figure 1, the acceleration in the z-axis (vertical)

has the largest magnitude due to the intrinsic gravity of the earth (9.8 m/s^2). We remove this by subtracting the standard gravity of the earth from the z-axis acceleration. Eventually, the resultant acceleration, which is used for evaluating the vibration states, is calculated by the equation:

$$a_{res} = \sqrt{{a_x}^2 + {a_y}^2 + (a_z - 9.8)^2}, \quad (3)$$

where a_{res} is the resultant acceleration, a_x, a_y, a_z are the accelerations in Y-axis, Y-axis, and Z-axis, respectively. The resultant acceleration is illustrated in Figure 2. The resultant acceleration calculated is the input for the vibration detection neuron.

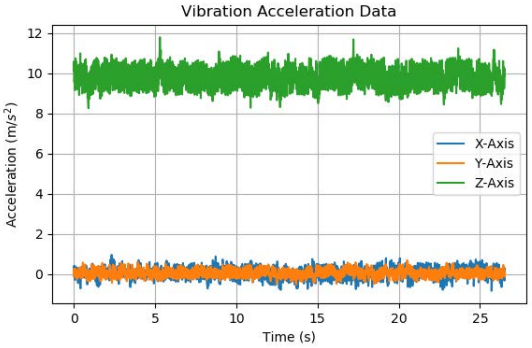

Figure 1: Acceleration data of IMU for three dimensions denoted as X-axis (blue), Y-axis (orange), and Z-axis (green).

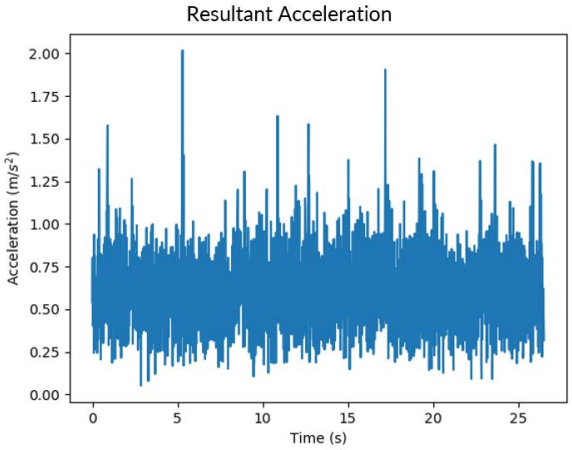

Figure 2: Resultant acceleration waveform.

In our experiments, the Nengo simulator was used to implement the system, and the vibration signal is imported to the sensory neuron.

A vibration detection neuron, implemented with LIF, is connected to the accelerometer, and it fires in response to the vibration detected as shown in Figure 3. At last, to make the robot move away from the vibration platform, the motion neurons are designed specifically to control the direction and speed of movement. The motion neuron is also implemented by a LIF neuron with parameters listed in Table 1. The active motion neuron will trigger a specific escape (fear) response that commands the UGV to move away from the vibration source with a speed of 0.3 m/s.

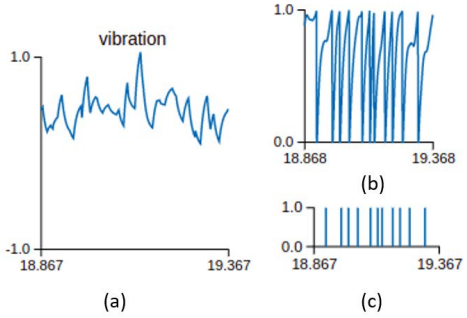

Figure 3: Vibration detection neuron response to the acceleration input: (a) input vibration signals; (b) membrane potential of the vibration detection neuron; (c) output spiking signals of the vibration detection neuron.

3.2 Processing of Visual Signals

For processing visual stimuli, we designed a neural network that activates an output neuron if it detects the light is on. Figure 4 shows the brightness of the light captured with the stereo camera equipped at our UGV. The stream of visual signals captured by the camera is sent to the computer via Robot Operating System (ROS). As the images arrive, their resolution is reduced to 24x48 pixels, and the pixel brightness is normalized to the range between -1 and 1.

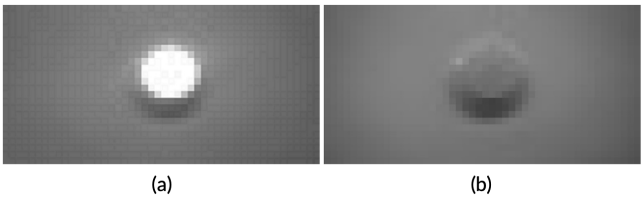

Figure 4: Images of the light on and off: (a) the light is on; (b) the light is off.

A deep neural network model based on 2D sparse coding is used for recognizing the brightness of light. The goal of sparse coding is to represent an input vector with a linear combination of features from a dictionary. This can be modeled by the LASSO equivalent optimization function:

$$E(a) = \frac{1}{2}\|x - \Phi \cdot a\|_2^2 + \lambda \cdot \|a\|_1 \quad (4)$$

$$a^* = \underset{a}{\operatorname{argmin}}\, E(a) \quad (5)$$

Where E(a) is the cost, a is the "sparse code" vector consisting of the feature coefficients a_i, x is the input signal, and Φ is the dictionary matrix, the columns of which are the features Φ_i.

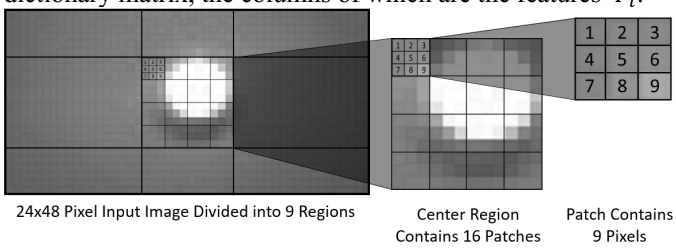

Figure 5: Image region layout and patch structure

The cost is determined by how close input matches the reconstruction $\Phi \cdot a$, and the size of the sparse code vector $\|a\|_1$, where λ is an additional sparsity penalty parameter to

control the tradeoff between sparsity and accuracy. Eq. (5) shows a^* is the optimal sparse code for minimizing the cost in Eq. (4).

Typically, the sparse code a^* is used to reconstruct x, or as the input to a classifier. In 2D image sparse coding, the image is divided into patches the size of the features Φ_j. Thus, each patch is a sparse coding optimization problem. The image is first divided into 9 sub-regions which are further partitioned into patches as shown in Figure 5. The Locally Competitive Algorithm (LCA) [20] is used for solving Eq. (4) and Eq. (5) because it is a biologically plausible method inspired by lateral inhibition observed in neuroscience and uses it to emulate the V1 region of the primary visual cortex. The algorithm uses only local competition between neighboring neuron elements, and it can be implemented with SNNs, unlike other solutions to the sparse coding problem. The essential idea is each dictionary feature is represented by a neuron, with its firing rate or activation indicating its features contribution to the input. The activation is achieved by the weights of the connections from the input to these "feature neurons". The higher a feature neuron's activity level is, the more it inhibits, or reduces, its neighboring neurons' activity levels. The following equations describe the Spiking LCA (S-LCA) model [21].

$$\dot{u} = \frac{1}{\tau}(\Phi^T x - u - (\Phi^T \phi - I) \cdot a), a = T_\lambda(u) \quad (6)$$

$$T_\lambda(u) = 0 \; if \; u \leq \lambda, else \; T_\lambda(u) = u - \lambda \quad (7)$$

Where τ is the discrete timestep, a_i is the average firing rate of neuron i, u_i is the average soma current for the neuron, and T_λ is the thresholding function that determines if neuron i is going to fire. This is achieved by adjusting the firing threshold of the neurons to λ. It has been shown that the S-LCA system dynamics converge to the set of average firing rates a_i corresponding to the optimum solution a^*.

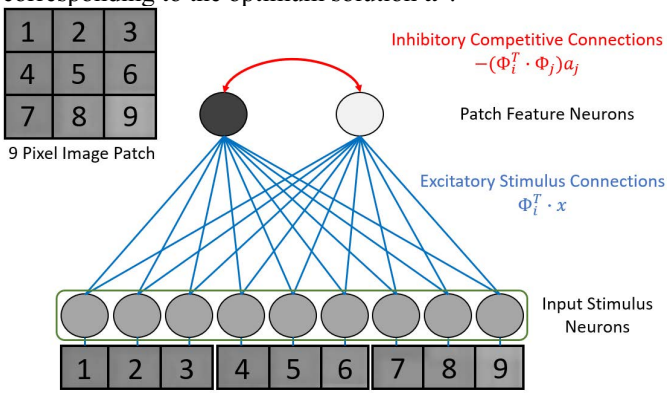

Figure 6: LCA network for one patch including input neurons (Layer 1) and feature neurons (Layer 2)

Figure 6 illustrates a spiking LCA network for solving one image patch, which is depicted in Figure 5. The model consists of one layer for the input x and another layer for the sparse code a^*. The firing rates of each neuron are the coefficients a_i. The neurons in the second layer are referred to as feature neurons because each of them is associated with one feature Φ_i. Typically, an overcomplete dictionary is used in sparse coding, resulting in a having a larger size than x. However, our model represents each patch with only two features, light and dark. Thus, the dictionary in Figure 6 only contains two features. The system could be made overcomplete by using more variations of exclusively light and dark features to create more feature vectors than the input (9 pixels in the patch).

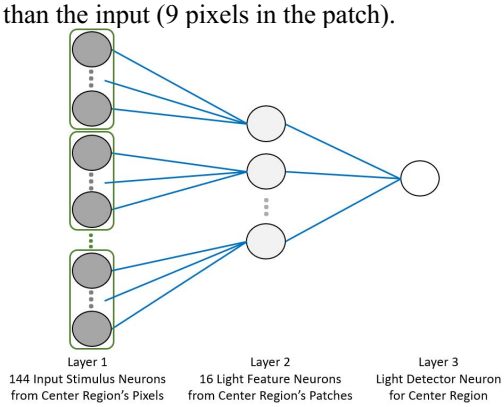

Figure 7: Neural network for light detection in the center region. Note: The dark feature neurons in Layer 2 are not shown.

The convolutional stride between the image patches is equal to the width of one patch (3 pixels) resulting in no overlap between them. This simplifies the LCA model by removing connections between feature neurons in overlapping patches. The neural network contains a third layer with one neuron for each of the 9 regions in the image. The portion of the network for the center region is shown in Figure 7. The third layer has only one neuron as an output neuron that integrates the light feature neurons of every patch. In Layer 1 each neuron is a spike generator and has a firing rate proportional to the pixel intensity it represents. The Layer 2 neurons are Integrate and Fire LCA neurons, or "patch feature neurons". The sparsity penalty λ is implemented via the V_{reset} parameter in the LCA neurons, which was empirically adjusted until the desired response was achieved from the feature neurons. The other parameters are kept constant according to the LCA model used in [21]. The Layer 3 neurons are the same LIF neurons with the parameters listed in Table 2.

Table 2: LCA neuron parameters in Layer 2 and 3

LCA Neuron Layer	τ_{RC} (s)	A	V_{reset} (V)	V_{th} (V)
Layer 2	∞	1.0	-λ = -0.85	1.0
Layer 3	0.02	0.3	-1.0	1.0

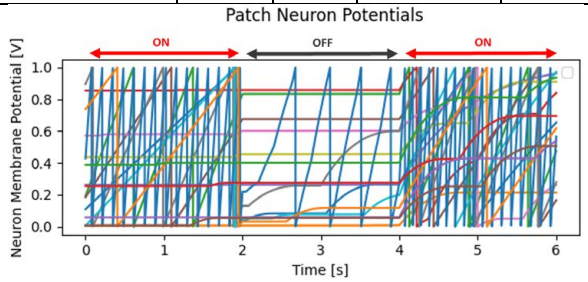

Figure 8: Membrane potential of Layer 2 light feature neurons in center region image patches.

When the light-on image (Figure 4 (a)) is the input to the network, the light feature neurons in the center patches start to fire. The light-off image (Figure 4 (b)) subsequently reduces the

activity in the neurons. These images are alternately presented as inputs, shown in Figure 8. This causes the neurons in layer 3 to fire for the center region from 0s-2s and 4s-6s as shown in Figure 9. It is the period the light is off in our experiment.

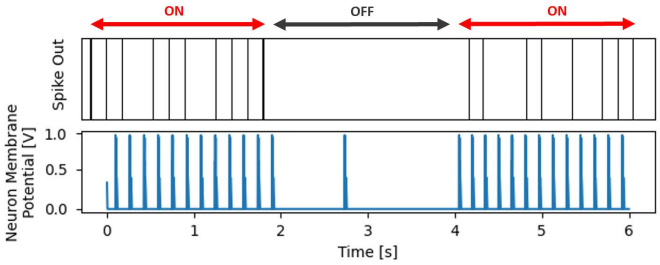

Figure 9: Spike output of Layer 3 neuron for the center region.

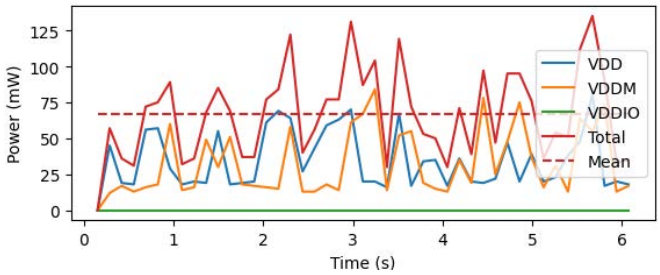

Figure 10: Power consumption for sparse coding network with Loihi chip. The VDD represents the compute logic, VDDM is the SRAMs, and VDDIO is the IO interface.

The sparse coding network was ported to the Intel Loihi neuromorphic chip for power and energy profiling using Nengo's support for using Intel's NxSDK software as the backend to recreate the network with the same neuron models. NxSDK contains a built-in LCA network implementation, which can be connected to the rest of the SNN. The parameters from Table 3 were used to create the same LCA network from Nengo for deployment on Loihi. In Loihi chip, the synaptic weights only have a 4-bit resolution, instead of the 24-bit resolution of the CPU network. The length of one simulation time-step, how often the network parameters (neuron spikes, membrane potentials, etc.) are increased from 1 ms to 20 ms for the Loihi. The network was given the same input stimulus and measured power, as shown in Figure 10. The average VDD power is 30 mW and the average VDDM power is 29 mW. VDD represents the compute logic, VDDM is the SRAM, and VDDIO is the IO interface. One can see VDDIO's contribution is relatively negligible to the total power, while VDD and VDDM have effectively equal contributions. The power is measured and reported with the average consumption across every 8 timesteps during the experiment.

4 Associative Memory Simulation and Experiment with UGV

The neuromorphic system implementing associative memory is illustrated in Figure 11. In our system, the signal from light is a conditional stimulus and the acceleration signal from vibration is the unconditional stimulus. The movement away from the vibration platform emulates the fear response of rats. The experimental setup with our UGV is shown in Figure 12.

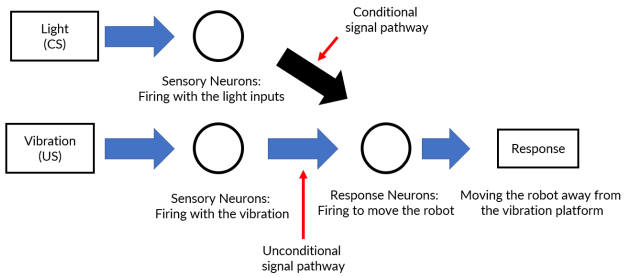

Figure 11: Neuromorphic system for associative memory learning implementation.

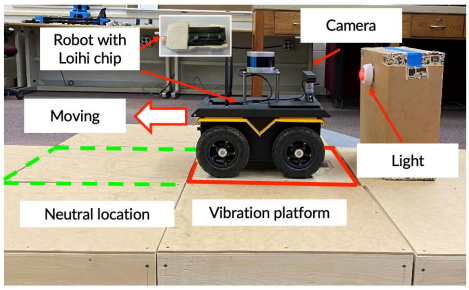

Figure 12: Experimental setup with UGV.

The synaptic weights are modified based on Hebbian learning [19, 22]. Hebbian learning states that when the pre-and postsynaptic neurons are both active at the same time, the synaptic weights between them will be modified by the equations [19, 22]:

$$w = \eta r_i r_j, \tag{8}$$

where the r_i and r_j are firing rates of pre- and postsynaptic neurons, respectively, and the η is the learning rate, determining the changing rate of synaptic weight. In our experiment, the learning rate η is 2×10^{-4}. Our simulation results are shown in Figure 14. The initial synaptic weights between the brightness detection neuron (CS) and the movement neuron are small. Consequently, the brightness stimulus of the light cannot be delivered to the movement neuron to stimulate it to fire. As a result, the synaptic weights between the brightness detection neuron and movement neuron stay constant. When the vibration (US) is applied to the vibration detection neuron, the movement neuron starts to fire. When the vibration and light are applied to the system, both the brightness detection neuron and the motion neuron are active, resulting in their synaptic weight increases.

Figure 14 illustrates that the synaptic weights increase when the vibration and light stimulus are both applied. However, the first overlapping time frame is not long enough to establish a significant synaptic weight modification. Thus, a sequence of weak spiking signals is observed from the response neuron after the vibration is removed, which is marked in Figure 14. In contrast, the second overlapping period is longer than the first one which leads to a larger increase in synaptic weights. Thereby, after training, the response neuron (motion neuron) will fire with a visual stimulus (light) even with no vibration stimulus. This demonstrates an accomplishment of associative memory.

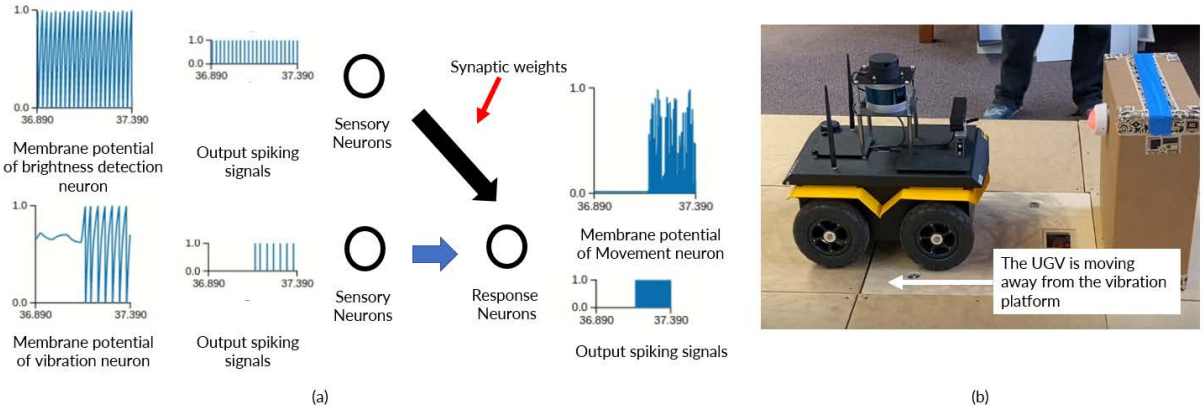

Figure 13: Real-time experiment data of associative memory with UGV: (a) The membrane potentials and spiking outputs of the brightness detection neuron, vibration detection neuron, and movement neuron. (b) The UGV is moving away from the vibration platform

Next, our neuromorphic system with associative memory is validated using real-time experiments. Our experimental setup is illustrated in Figure 12. We replace the rats in fear learning with the UGV. The UGV is placed on a testing platform, which is constructed with 9 wooden boxes. The dimension of each box is 23 in (L) × 23 in (W) × 8 in (H).

A vibration plate is installed underneath the center platform, which is marked in red square in Figure 12 (a). The vibration plate can provide vibration signals (15-40 Hz) emulating an unpleasurable stimulus (the electrical shock) in fear learning on the rats. The other eight platforms with no installed vibration plate are marked in green-dashed squares in Figure 12.

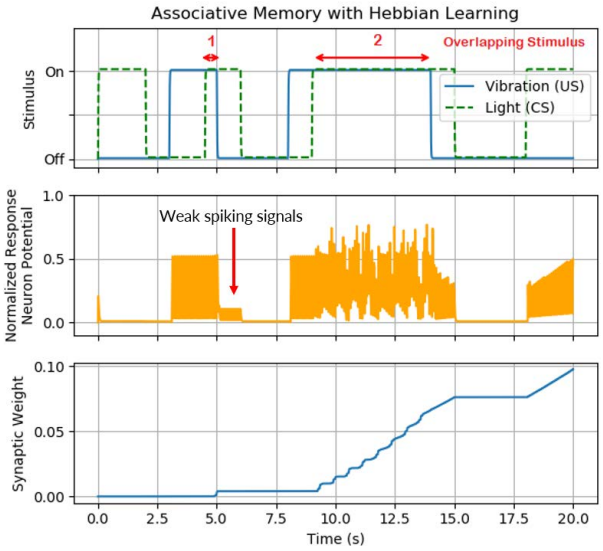

Figure 14: Change of synaptic weight during associative learning.

With the unpleasurable stimulus applied, the UGV will move away from the vibration platform to the neutral location. Figure 13 illustrates our experiments of associative memory learning on UGV in real-time. In the experiments, the synaptic weights between the brightness detection neuron and the motion neurons are modified during the training process. As a result, after associative memory learning, the UGV will move away from the vibration platform under the stimulus of light even with no vibration signal presented, demonstrating successful online learning in real time. Several state-of-the-art relevant works are summarized in Table 3. Our work outperforms these works in several aspects. First, our associative memory learning model is constructed with more than thousands of neurons and synapses, which outperforms a few neurons in [12-15, 23, 24]. The large scale of neural networks enables our model to have the capability of processing more complicated signals, such as visual and acceleration signals. More importantly, other associative memory learning processes in [8, 12-15, 23, 24] are accomplished in simulation scenarios rather than in experiments in the real world. In the real world, the system requires more robustness in noisy and unpredictable environments. At last, our UGV provides a platform with sensors that can accomplish associative memory learning by directly collecting signals from surroundings rather than manually-labeled the data.

Table 3: Comparison of scale and association capability with other state-of-the-art works

	Neuron	Synapse	Dataset	Learning Scheme
[12]	6	3	N/A	Simulation
[13]	3	1	N/A	Simulation
[14]	5	6	N/A	Simulation
[15]	3	1	N/A	Simulation
[23]	3	1	N/A	Simulation
[24]	3	2	N/A	Simulation
[8]	20	100	Pre-trained with datasets	Simulation
This work	1419	1420	No dataset for pretraining	Experiment

5 Conclusion

In this paper, we implement associative memory with a UGV and Intel Loihi for an online learning scenario. Our system successfully reproduces the fear conditioning of rats with no pretraining procedure and labeled datasets. Our UGV autonomously memorizes the cause-and-effect of the light

stimulus and vibration stimulus. The Intel Loihi chip is integrated with our online learning system for processing visual signals. The average power usage is 59 mW.

Acknowledgment

As Intel Neuromorphic Research community members, the authors would like to thank Intel Labs for providing their neuromorphic Loihi chips for our online learning studies.

References

[1] I. Goodfellow, B. Yoshua, and C. Aaron, "Deep Learning," *Deep Learning,* p. 785, 2016, doi: 10.1016/B978-0-12-391420-0.09987-X.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805,* 2018.

[3] N. Zins, Y. Zhang, C. Yu, and H. An, "Neuromorphic Computing: A Path to Artificial Intelligence Through Emulating Human Brains," in *Frontiers of Quality Electronic Design (QED)*: Springer, 2023, pp. 259-296.

[4] F. Zenke and S. Ganguli, "Superspike: Supervised learning in multilayer spiking neural networks," *Neural computation,* vol. 30, no. 6, pp. 1514-1541, 2018.

[5] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. A. Siegelbaum, and A. Hudspeth, *Principles of neural science*. McGraw-hill New York, 2000.

[6] J. Sun, G. Han, Z. Zeng, and Y. Wang, "Memristor-based neural network circuit of full-function pavlov associative memory with time delay and variable learning rate," *IEEE transactions on cybernetics,* 2019.

[7] T. Kohonen, *Self-organization and associative memory*. Springer Science & Business Media, 2012.

[8] H. An, Q. An, and Y. Yi, "Realizing Behavior Level Associative Memory Learning Through Three-Dimensional Memristor-Based Neuromorphic Circuits," *IEEE Transactions on Emerging Topics in Computational Intelligence,* 2019.

[9] S. G. Hu *et al.*, "Associative memory realized by a reconfigurable memristive Hopfield neural network," *Nat Commun,* vol. 6, pp. 1-5, 2015, doi: 10.1038/ncomms8522.

[10] K. Moon *et al.*, "Hardware implementation of associative memory characteristics with analogue-type resistive-switching device," *Nanotechnology,* vol. 25, 2014, doi: 10.1088/0957-4484/25/49/495204.

[11] H. An, Z. Zhou, and Y. Yi, "Memristor-based 3D neuromorphic computing system and its application to associative memory learning," *2017 IEEE 17th International Conference on Nanotechnology, NANO 2017,* pp. 555-560, 2017, doi: 10.1109/NANO.2017.8117459.

[12] J. Yang, L. Wang, Y. Wang, and T. Guo, "A novel memristive Hopfield neural network with application in associative memory," *Neurocomputing,* vol. 227, pp. 142-148, 2017, doi: 10.1016/j.neucom.2016.07.065.

[13] X. Liu, Z. Zeng, and S. Wen, "Implementation of memristive neural network with full-function pavlov associative memory," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 63, no. 9, pp. 1454-1463, 2016.

[14] X. Hu, S. Duan, G. Chen, and L. Chen, "Modeling affections with memristor-based associative memory neural networks," *Neurocomputing,* vol. 223, pp. 129-137, 2017, doi: 10.1016/j.neucom.2016.10.028.

[15] K. Moon *et al.*, "Hardware implementation of associative memory characteristics with analogue-type resistive-switching device," *Nanotechnology,* vol. 25, no. 49, p. 495204, 2014.

[16] S. B. Eryilmaz *et al.*, "Brain-like associative learning using a nanoscale non-volatile phase change synaptic device array," *Front Neurosci-Switz,* vol. 8, 2014.

[17] M. Davies *et al.*, "Advancing Neuromorphic Computing With Loihi: A Survey of Results and Outlook," *P Ieee,* 2021.

[18] M. Li, J. Liu, and J. Z. Tsien, "Theory of connectivity: nature and nurture of cell assemblies and cognitive computation," *Frontiers in neural circuits,* vol. 10, p. 34, 2016.

[19] P. Miller, *An introductory course in computational neuroscience*. MIT Press, 2018.

[20] C. J. Rozell, D. H. Johnson, R. G. Baraniuk, and B. A. Olshausen, "Sparse coding via thresholding and local competition in neural circuits," *Neural computation,* vol. 20, no. 10, pp. 2526-2563, 2008.

[21] P. T. P. Tang, T.-H. Lin, and M. Davies, "Sparse coding by spiking neural networks: Convergence theory and computational results," *arXiv preprint arXiv:1705.05475,* 2017.

[22] P. Dayan and L. F. Abbott, *Theoretical neuroscience: computational and mathematical modeling of neural systems*. Computational Neuroscience Series, 2001.

[23] M. Ziegler *et al.*, "An Electronic Version of Pavlov's Dog," *Adv Funct Mater,* vol. 22, no. 13, pp. 2744-2749, 2012, doi: 10.1002/adfm.201200244.

[24] Y. V. Pershin and M. Di Ventra, "Experimental demonstration of associative memory with memristive neural networks," *Neural Networks,* vol. 23, no. 7, pp. 881-886, 2010.