Neuromorphic System for the Identification of Maneuvers in Three-Dimensional Space

Diogo Silva
MEEC
Instituto Superior Técnico
Lisbon
diogohmsilva@hotmail.com

ABSTRACT

This work presents an approach for developing a neuromorphic system, that relies on spiking neural networks to identify human workout activities. A tridimensional characterization of maneuvers was acquired, using gyroscope and accelerometer information, collected by a small IoT device (SensorTag). Running in place, torso rotation, jumping jacks and cross toe-touch were the performed maneuvers. A pre-existing firmware was adapted to enable sensor data collection at higher frequency rates using, Bluetooth Low Energy. The samples were subsequently encoded into spike trains using a population of neurons with spatial characteristics and fed into a spiking neural network. The network incorporates a competition layer that implements a one-winner-take-all mechanism to ensure that the competitive neurons distribute themselves to learn different patterns. These neurons implement the leaky-integrate-and-fire neuron model. Also, a learning rule based on synaptic-time-dependent-plasticity is used to train the competitive neurons on the input pattern. The system was able to reach a 100% recognition accuracy for a dataset comprising of activities performed by a singular participant.

KEYWORDS

Spiking Neural Network, Leaky-Integrate-and-Fire, Synaptic-Time-Dependent-Plasticity, Winner-Take-All

1 Introduction

Motion analysis plays a crucial role in various domains such as sports science, healthcare, robotics and human-computer interaction. Drone and robot control, elderly, and patient surveillance are some specific applications. Such analysis, based on data characterizations of human movement patterns, enables computerized systems to autonomously react to predicted behaviors (e.g. monitor patient falling). Numerous machine learning based solutions have been develop to solve motion analysis problems, demonstrating their effectiveness in various applications. However, most existing solutions present high computational and energy costs. This can pose challenges when deploying such solutions in for devices with limited processing power and energy constraints, such as IoT devices and smartwatches. As a promising alternative, spiking neural networks (SNNs) demonstrate the potential to address these challenges effectively.

1.1 Objectives

The final purpose of this work is to propose an approach for the development of a neuromorphic system capable of accurately identifying maneuvres, from a pre-defined set, within a three-dimensional space. To achieve this we must complete three major tasks:

1. Create a dataset comprising of workout activities
2. Design a SNN capable of learning spatiotemporal patterns
3. Assess the network's recognition performance using the compiled dataset

The remaining content of this document is structured as follows. Section 2 provides an overview of neuronal dynamics and explores how it influenced the development of artificial neural networks. Next, Section 3 describes the methodology employed for the creation of the dataset. In Section 4, the proposed system for maneuver identification is presented. It also explains the architecture and dynamics of the implemented SNN and provides a description of the implementation process. Section 5 presents the network experimental recognition performance. Finally, some conclusions are provided in Section 6.

2 Related work

2.1 Biological Neural Networks

In animals, aggregations of neurons form large and complex webs such as the nervous system [1]. This particular system is known to regulate animal's actions and sensory inputs by carrying electrical impulses all over the body. The neuron is the elementary unit of the nervous system. It is a cell that can switch between an electrically excited or inhibited state by interacting with other neurons. These interactions consist of electrical stimuli that travel through neural networks. Neurons have three main parts: dendrites, cell body and axon. They also possess a membrane potential that reacts to incoming stimuli. While no stimuli (impulses) arrive, the neuron rests at a resting potential. As more impulses are received, the membrane potential increases. When the membrane potential reaches a threshold value, the neuron will fire, whereby an impulse will leave the cell body through its axon and reach the axon terminals, connected to the dendrites of neighbouring neurons. The cell's membrane potential will thus rapidly decrease to a resting potential. This biological event is known as action potential and is the basis for inter-neuron communication.

Donald Hebb [2], a psychologist, considered how stimuli triggered physical changes between neuron's connections. He defended that when the repeated firing of a neuron A caused a neuron B to also repeatedly fire, the connection between these two neurons would be strengthened and they will be more likely to fire together again in the future. On the other hand, if two neurons fire in an uncorrelated manner, their connections will become weaker. This idea can summarized by the motto "Neurons that fire together, wire together. Neurons that fire apart, wire apart". Here, firing is a term used to describe the case when a neuron membrane potential saturates and causes the release of an action potential, or 'spike', from its cell body to its axon terminals. This hypothesis, that latter became known as Hebbian learning, predicted a biological event that occurs between neurons. This learning mechanism is known as spike-timing-dependent-plasticity (STDP) [3]. In this process, the strength of a connection between a pre- and a post-neuron (i.e., the sending and receiving neuron), will increase if the post-neuron repeatedly fires right after the pre-neuron. The amount of variation in the post-voltage translates into the connection strength (stronger connections lead to a greater variation in the voltage). As already mentioned, for STDP the times at which pre- and post-neurons fire relative to each other determines if their connection will be strengthened or weakened. In [4], the authors have identified and characterized a critical time window during which synaptic modifications can occur. If the pre-synaptic neuron fires in the preceding 20ms window before the post-neuron fires, long term potentiation (LTP) occurs and the connection between these neurons is strengthened. If the pre-synaptic neuron fires within a window of 20ms after the post-synaptic neuron, long term depression (LTD) will lead to the weakening of their connection. Outside these windows, connections strength will remain unchanged.

2.2 Spiking Neural Networks

The ability to accurately model and analyze data has become crucial in today's world with many companies using it to model customer behaviors and make informed data-driven decisions. In healthcare, researchers use it to predict patient outcomes and identify disease patterns. Artificial Neural networks (ANNs) have become a prominent technology for uncovering and modelling such behaviors and patterns. With architectures inspired in biological learning systems found in animals [5], ANNs result from the aggregation of interconnected nodes, termed as neurons, that function as the computational units of the network. These systems are able to acquire knowledge (or 'learn') from multiple observations in a process designated as training, and later become able to classify new input instances based on what they have learned in the training phase.

There are three techniques that can be used to train an ANN: supervised, unsupervised and reinforcement learning. Supervised and unsupervised learning work in opposite ways. In the former method, the training data is complemented with classification data (labels) that tell the model what is being presented at its input. This way, later, when it receives a similar input instance, it knows what to output. In unsupervised learning algorithms, the training dataset is just a set of examples with no specific labels. The model is responsible for finding hidden patterns within the dataset. Lastly, reinforcement learning is an iterative process where an agent (artificially intelligent agent) attempts to improve in a specific task. If the agent actions lead him towards a pre-specified goal, it will be rewarded. The agent will try to find the best next step until a final reward is given.

In his influential paper [6], Maass classifies ANNs according to the functionality of their underlying neurons. He considers three generations of ANNs. M-P neurons, perceptrons with continuous and differentiable activation functions and spiking neurons are, in this order, the computational units that engender the different generations of ANNs. The first generation, McCulloch-Pitt neurons, are characterized by taking binary inputs and producing a binary output. This output will equal one if the sum of the inputs exceeds a pre-specified threshold value, and zero otherwise. The second generation of ANNs, as classified by Maass, is known as multilayer perceptron and result from having multiple perceptron units, with a continuous and differentiable activation function, grouped by layers. Adopting this neuron model enables the use of the gradient descent backpropagation algorithm. This is the most widely used method for training a multilayer perceptron given a set of labelled inputs (supervised learning). The third generation of ANNs are the Spiking Neural Networks (SNNs). These are ANNs based on spiking neurons. Some of the most used spiking neuron models include the leaky-integrate-and-fire (LIF) [7] [8] (see Figure 1) and the Hodgin-Huxley [9] models. This generation provides the most accurate representation of the dynamics of a biological neuron. In SNNs neurons share synapses through which they exchange sequences of spikes.

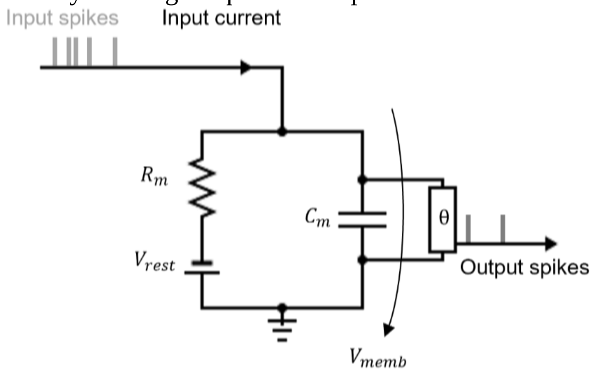

Figure 1. Physical circuit representation of a LIF neuron.

Each neuron has an internal membrane potential, V_{memb}, that changes according to a pattern of incoming spikes. Every spike entering the neuron will cause this potential to increase. When it reaches a threshold membrane potential, V_{thresh}, it decreases to a reference potential, V_{reset}, and the neuron emits a spike. Also, the neuron enters the, so called, refractory period, during which the input is disregarded. SNNs are more energy efficient than the earlier generations (and thus better for low energy platforms) as each cell of the network is only active when a spike is received or emitted, in contrast with multilayer perceptrons that always have their units active. Because this is a spike-based model, the input data must first be encoded into spikes that can be used as input stimulus. Spikes can be generated using different encoding algorithms, meaning that the same input stream of data can take different sequential representations before it is sent to the SNN. Although it is still unknown how the brain encodes information, some encoding schemes have been proposed to explain this mechanism [10]. Rate coding, time-to-first spike (TTFS) and population encoding are the most used methods to encode spatial data into spikes. The first, which is the most widely used, encodes real values using Poisson impulse trains, with firing rates that are proportional to the magnitude of the input value. The hypothesizes that the brain uses such method to encode information appears to be inaccurate and lacks supporting evidence. The human brain's exceptional ability to perform a wide range of complex tasks while operating at relatively low frequencies make it super energy efficient [11]. The rate coding scheme mentioned relies on a time window, which needs to be big enough to ensure a reliable representation of a value. Also, it leads to a system that is not energy-efficient, since the number of produced spikes increases with the magnitude of the values. On the other hand, TTFS uses time to represent real values with a single spike. The spike time, within a time window, is inversely proportional to the value itself, leading to earlier impulses for higher values. Figure 2 exemplifies how these encoding methods can be applied to encode grayscale pixel values (ranging from 0 to 255).

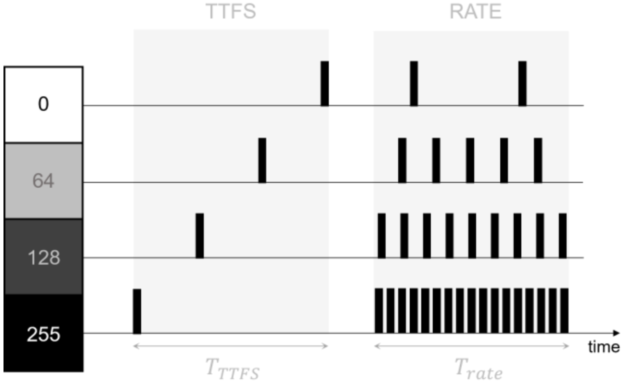

Figure 2. Pixel values encoded with TTFS and rate encoding. For both methods, a time window is chosen during which the real-valued pattern will be presented (T_{TTFS} and T_{Rate}).

STDP, the biological mechanism behind synaptic plasticity has given rise to an unsupervised learning algorithm [12]. It delivers an unsupervised approach for updating the weights of a SNN, based on the difference between the firing times of the pre- and post-synaptic neurons.

Besides excitatory synapses, neurons can also share inhibitory connections. Inhibitory connections allow neurons to suppress each other's activity, thus providing a way for incorporating competition in a SNN.

Neural simulators enable researchers to simulate and test models of biological neural networks. This is not only a key element for consolidating new knowledge obtained from in-vivo observations, but also to realize more powerful algorithms that leverage from these findings. Brian 2 [13], NEURON [14] and NEST [15] are some of the most used spiking neural networks simulators. When deciding on which simulator to use, it is important to consider the requirements of the target application, as each simulator offers different capabilities and features. NEURON focuses on the simulation of biologically accurate neural models. NEST leverages from parallel computations to efficiently simulate large-scale networks, with up to millions of neurons. Brian2 offers higher flexibility in model definition when compared to other simulators, since models are defined by simply writing down their equations.

3 Setup for data acquisition

The development of a dataset involved the use of the SensorTag, an IoT development kit designed by Texas Instruments (TI). The SensorTag runs the Texas Instruments Real Time Operating System (TI-RTOS), which is a real-time operating system developed and supported by TI. Real time operating systems are known for their deterministic behavior, ensuring precise scheduling of computational tasks to meet specific timing requirements. To facilitate code development, debugging and testing for TI's microcontrollers, TI provides an integrated development environment (IDE) called Code Composer Studio (CCS), which was used in this work.

The dataset consists of samples of human workout activities, illustrated in Figure 3. These activities were characterized by data acquired with an accelerometer and a gyroscope, both integrated into the SensorTag. The data was transmitted to a computer through Bluetooth Low energy (BLE) for further processing. The SensorTag stock firmware enables data acquisiton with a 100 Hz sampling frequency (for the used sensors). At the start of this work, the ideal sampling frequency was unknown. This is because it was dependent on the chosen spike encoding method and network topology. A pre-existing SensorTag firmware [16] was modified to enable an increase of this frequency and thus overcome any potential limitations resulting from a low sampling frequency. Such modification led to the maximization of both the BLE transmission frequency as well as the size of the data packets being transmitted. The latter the transmission of a custom message consisting of multiple sensor readings (instead of just one), allowing sensor data to be acquired with a frequency that is not limited to the BLE throughput. A BLE service was implemented to transmit such custom message. Services structure the content of BLE messages. This configuration transmits packets of 251 bytes (approximately 20 sensor readings) at 100 Hz. This improvement turned out to be unnecessary as the sampling frequency offered by the original SensorTag firmware showed sufficient.

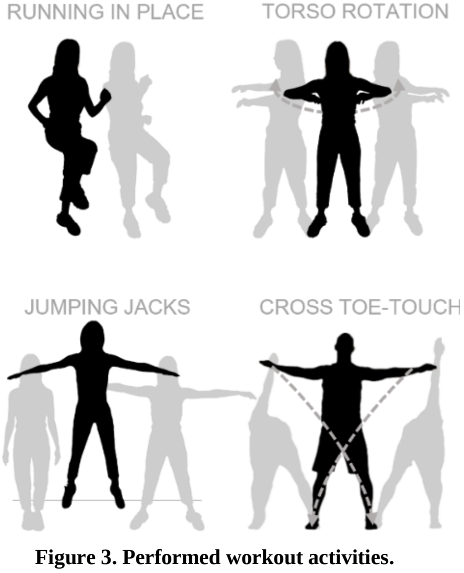

Figure 3. Performed workout activities.

During data acquisition, the SensorTag was worn on the wrist to simulate a smartwatch-like setup. Since the performed activities exhibited a regular periodicity, a straightforward but effective segmentation technique was employed to divide the sequential repetitions of activities into individual samples. The sensors axis exhibiting the most sinusoidal shape was chosen and subsequently filtered to enhance such waveform. This facilitated the identification of the filtered data extremes, which were used as reference to divide the data streams into single movement samples.

4 Proposed system

The final purpose of this work is to propose an approach for the development of a system for accurately identifying maneuvres within a three-dimensional space. The block diagram of the architecture of the proposed system for solving such problem is depicted in Figure 4. The data acquisition and data processing modules correspond, respectively, to the processes of data collection and segmentation described in section 3. The segmented samples resulting from these modules are then converted into sequences of spikes, after an encoding method is applied, in the encoding module. Finally, the identification module predicts the different maneuvres belonging to the dataset when their corresponding sequence of spikes has been provided at its input. To achieve this, a SNN undergoes a training process, to learn a subset of encoded samples from the dataset. Afterwards, another subset of samples is used to assess the performance of the network at identifying maneuvres.

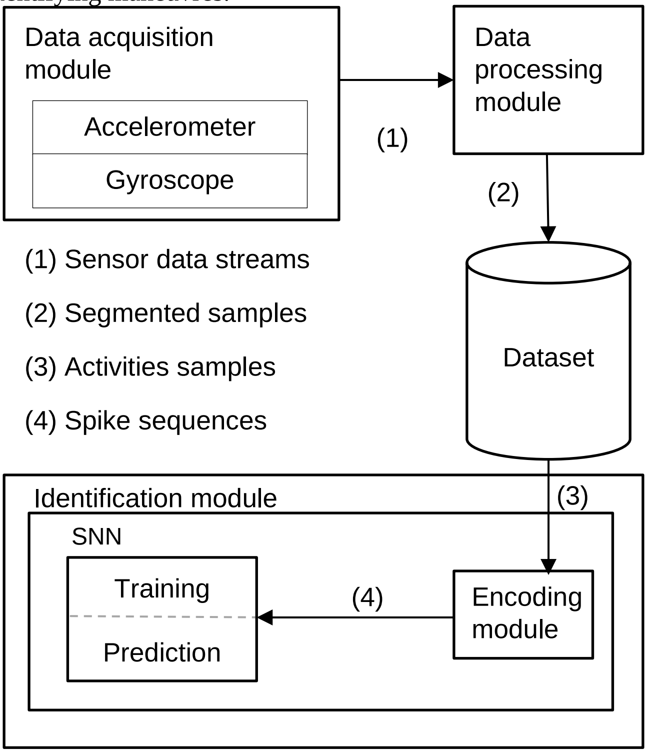

Figure 4. Block diagram of the proposed system for maneuvres identification.

4.1 Network architecture

The adopted SNN is composed of three layers (see Figure 5). The first layer encodes the movement samples into spike-sequences. Each input neuron is connected to all the second layer neurons. These synaptic connections exhibit synaptic plasticity, with weights that are learned throughout the experiment, using a STDP-like trace-based learning rule. The second layer is composed of LIF neurons, with dynamics described by equation (1). The neurons membrane potential, v, rests at a potential V_{rest} while no spikes are received. Otherwise, pre-spikes increase the excitatory synaptic current, I_e, and, subsequently, v. If v crosses the threshold potential, the neuron emits a spike, resets its potential and enters a refractory period where it disregards any input. I_i is the inhibitory synaptic current and its increment leads to the decrease of v. These currents increase with the arrival of spikes, and decrease exponentially otherwise, as expressed by equation (2). R_e and R_i are, respectively, the excitatory and inhibitory membrane resistances. These values influence the synaptic currents amplitude, and, consequently, the neurons response speed to incoming excitatory and inhibitory spikes. Second layer neurons, referred to as competitive or WTA neurons, share lateral inhibitory connections. The connections are established between each neuron and the remaining ones. Every time a neuron fires, it inhibits the remaining from firing in the following time window of duration τ_{inh} milliseconds. This lateral inhibition scheme enforces a WTA mechanism that drives the neurons to compete and distribute themselves to cover the different patterns presented in the input. Besides the neural mechanisms discussed so far, an additional homeostatic process was introduced in the LIF model. Each neuron is also attributed with an adaptive threshold which increases each time a neuron fires, after which it decays exponentially to a reference value otherwise. Such mechanism effectively limits the neurons firing rate and prevents single neurons from continuously firing, thus inhibiting other neurons, therefore dominating the WTA layer. Although the biological STDP mechanism uses the firing time difference between pre- and post-neurons to perform the weights update, there are more computationally efficient approaches to implement a learning rule which provides a similar outcome. Instead of saving every firing time instant, each synapse is attributed with two additional variables. These are the pre- and post-synaptic traces, a_{pre} and a_{post} (see equations (3) and (4)), respectively. These variables keep track of pre- and post-synaptic activity, respectively. Whenever a pre- or post-neuron fires, the correspondent trace variable is updated as described by equations (5) and (6), respectively, after which it decays exponentially. Also, when a pre- or post-neuron fires, their weights are updated according to equations (7) and (8), respectively. The condition described in equation (9) ensures that the decrease in weight for LTD is greater than the increase in weight for LTP. The third, and last layer, is the recognition layer. It has as many neurons as the number of classes in the dataset. These neurons follow the leaky-integrate model, since they don't have a firing condition. The activity of each of the recognition neurons is used to assess the network performance. The recognition layer is only active during the testing phase (i.e., after the learning has been carried out). Each of the output neurons share static and delayed synapses with the competitive neurons (i.e., neurons that share lateral inhibitory connections). The neurons in the third layer are of the leaky-integrate type since they don't have a firing condition (dynamics also described by equation (1) without the inhibition terms).

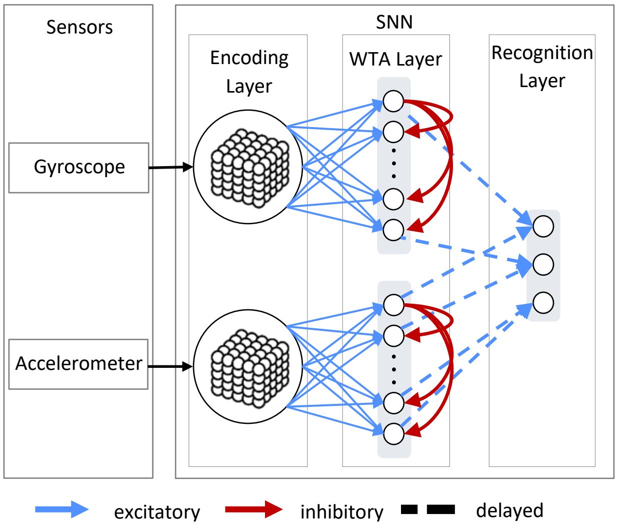

Figure 5. Implemented network architecture (a set of lateral inhibitory connections relative to a single competitive neuron is shown).

Their behaviour is analysed by extracting the maximum value of their potential over a pattern presentation time window.

$$\tau_m \frac{dv}{dt} = (V_{rest} - v) + R_e * I_e - R_i * I_i \tag{1}$$

$$\tau_I \frac{dI}{dt} = -I \tag{2}$$

$$\tau_{pre} \frac{dapre}{dt} = -apre \tag{3}$$

$$\tau_{post} \frac{dapost}{dt} = -apost \tag{4}$$

$$\Delta apre = Apre \tag{5}$$

$$\Delta apost = Apost \tag{6}$$

$$\Delta w = apre \qquad \begin{array}{l} if\ w + apre \leq w_{max} \\ else\ w = w_{max} \end{array} \tag{7}$$

$$\Delta w = apost \qquad \begin{matrix} if\ w + apost \geq 0 \\ else\ w = 0 \end{matrix} \tag{8}$$

$$\tau_{post} * Apost > \ -\tau_{pre} * Apre \tag{9}$$

4.2 Spike encoding

The proposed encoding scheme represents the input signal as a sequence of spatial patterns, treating each as an independent snapshot of data and overlooking any temporal correlations between them. Each dimension of the input data is discretized into a reduced set of possible values that are linearly spaced. The total number of input neurons is equivalent to the number of possible combinations resulting from discretizing all the values in all dimensions of the input data. The input patterns are represented using the spatial coordinates of the input neurons, providing a mechanism for encoding and processing spatiotemporal data in a distributed manner. The resolution at which the input patterns are represented increases with the chosen number for possible discrete values per dimension. Given the three-dimensional nature of the dataset samples (for each sensor), their full range is constrained to a cubic space. Taking this aspect into consideration, a population of input neurons is uniformly distributed in a cubical arrangement. Input neurons dynamics are defined by equation (10). Here, $\boldsymbol{Poisson}$ stands for a rated Poisson process. These neurons are modelled to exhibit a Poisson process firing behaviour, with a fixed firing rate, dependent on their spatial proximity to the input data. Each neuron is attributed with a three-dimensional coordinate (represented as $\boldsymbol{c}$ in equation (11)). The readings from a single sensor (represented as p(t)) are also three-dimensional coordinates. Every simulation timestep, each neuron is said to be $\boldsymbol{inzone}$ if its Euclidean distance to the current sensor reading is smaller than a threshold distance, $\boldsymbol{dth}$ (as expressed by equation (11)). Note that $\boldsymbol{inzone}$ is a Boolean variable. If so, the conditioned Poisson activity takes effect. The value of $\boldsymbol{p(t)}$ is updated according to the sensor sampling period, $\boldsymbol{Ts}$, and is constant within the interval $[\boldsymbol{t_k}, \boldsymbol{t_k} + \boldsymbol{Ts}]$ (for a reading at $\boldsymbol{t} = \boldsymbol{t_k}$). Note that $\boldsymbol{f_{zone}} >> \boldsymbol{f_{min}}$ and the latter is used to activate neurons that otherwise wouldn't fire (since the input data is never spatially close to them). This is because neurons need to fire to update their weights. Figure 6 illustrates the input response to a set of three-dimensional coordinates using this encoding scheme.

$$\begin{aligned} input\_dyn(\ in_{zone}, f_{zone}, f_{min}) \\ = \ in_{zone} * Poisson(f_{zone}) \\ + Poisson(f_{min}) \end{aligned} \tag{10}$$

$$inzone = (||c - p(t)|| \leq \ dth) \tag{11}$$

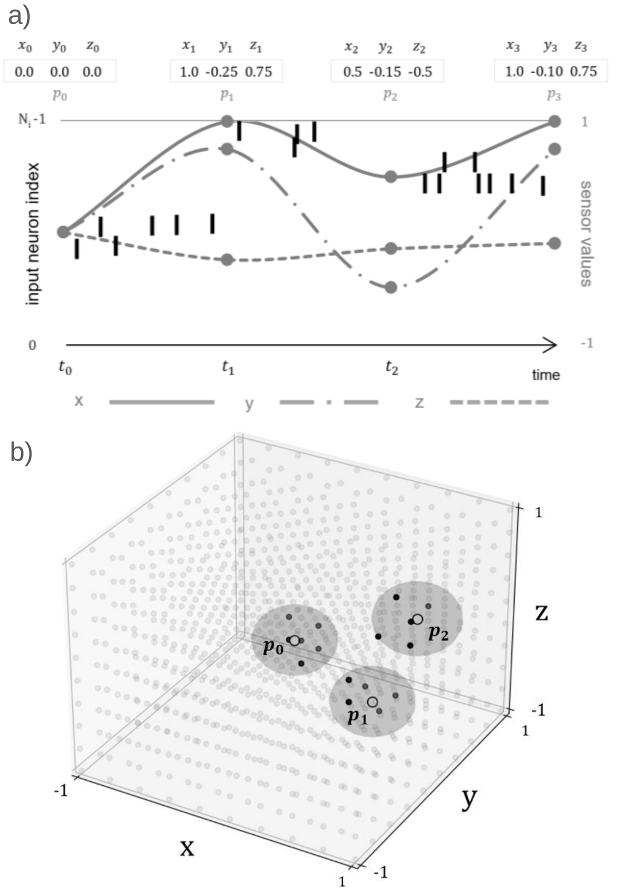

Figure 6. Example of the proposed encoding spatial-dependent spike encoding scheme (a) Three-dimensional values over time and resulting spike pattern (b) The first three 3D coordinates are represented as white points. The shades of grey indicate the areas where input neurons are permitted to fire. Black dots represent firing input neurons.

4.3 Implementation

The modelling code was developed in Python and the SNN was simulated with Brian2. While other simulators focus on biologically plausible neural representations, Brian2 stands out as a more versatile tool, since it enables the user to define equation-based neural and synaptic models. This feature expands the spectrum for SNN simulations, as it enables the research of both biologically and non-biologically plausible models. The complete workflow was performed using Jupyter Notebooks. A set of Jupyter Notebooks is available at [17] for reference and reproducibility purposes. The adapted SensorTag firmware custom firmware is also available in this repository. The experiment comprises of three stages: training, supervised parameter estimation (via analysis of firing behaviour) and recognition. The dataset was partitioned into training and testing sets with a ratio of 70/30, respectively.

Initially, the network was trained to learn the patterns in the training dataset. The weights between the input and WTA layers were randomly initialized within a small range relative to the maximum weight value. During each simulation iteration, the network is presented with a sample from each class. After each sample presentation, there is a resting stage, without input spikes, where the variables reset to their resting values. The order of class presentation is maintained consistently across iterations to ensure an equitable stimulation between classes and prevent biases. The weights are recorded at fixed time intervals while the training set is iterated multiples times.

A similar network is simulated for testing, the difference being that the weights between the input and the WTA layer are now static and equal to the weights deduced in the learning phase. The network is then simulated with the testing dataset. The purpose of this first trial with the test set is to analyse the neurons behaviour when equipped with the learned weights. After learning, certain WTA neurons exhibit a resembling behaviour when presented with patterns from the same class. More specifically, the neurons firing time, relative to the beginning of the pattern, is proximate. The neurons behaviour analysis is restricted to only the first emitted spike inside each time window corresponding to a movement pattern presentation. The neurons whose behaviours are more consistent are selected and assigned to a class. To be eligible for a class, neurons must fire at least 90% of the times for that class. Additionally, their spike emission times (relative to the beginning of the pattern presentation) need to be relatively consistent across testing iterations. Each class is represented by an output neuron. The selected WTA neurons are then connected to the corresponding output neurons. The firing time differences between WTA neurons assigned to the same class are used to setup delays between the WTA and the recognition neurons. These delays were calculated to synchronize the WTA neurons activity, producing a higher response in the output neurons. WTA neurons will distribute and become responsive to the different spatial patterns that compose the spatiotemporal patterns of the movements. By synchronizing their responses through delayed synapses, it is possible to assemble a spatiotemporal pattern detector [18]. As depicted in figure 8a, for each neuron, this delay is calculated as the difference between its average firing time of neuron and the average firing time of the neuron with the highest average firing time. Figure 8b shows the responses five of competitive neurons when presented to the testing set. It is possible to observe that some neurons exhibit similar firing times when presented with patterns belonging to the same class. Finally, a network with the learned weights and asserted delays is once again presented with the testing dataset and the performance of the network is assessed. The accuracy recognition for each class is determined by calculating the percentage of the number of times each output neuron presented the highest membrane potential, when presented with its corresponding class, relative to the number of presentations per class.

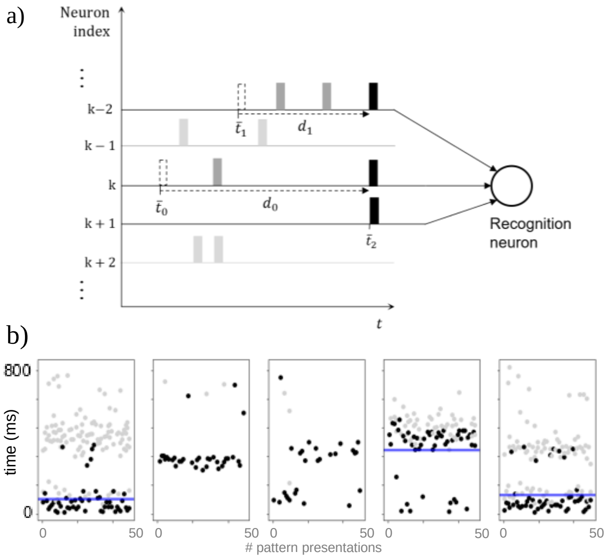

Figure 7. (a) Illustration of neural response synchronization using delayed synapses (b) Firing time (relative to the start of the pattern) of 5 WTA neurons to 50 testing samples of single a class (blue lines correspond to the average firing time). Black and grey dots refer to the first and following spikes (for each pattern presentation), respectively.

5 Results

5.1 Dataset

Table 1 provides with some insights about the acquired samples.

Activity	# samples	$\bar{T}$ (s)	σ_T (ms)
Cross-toe touch	219	1.65	73.90
Jumping jacks	229	0.92	57.74
Running in place	233	0.85	44.97
Torso rotation	259	1.79	157.19
	940	1.3	83.45
	Total	Average	

Table 1. Dataset number of samples and duration per class. $\bar{T}$ and σ_T refer to the samples average duration and the standard deviation of the duration, respectively.

Also, two visual representations of the encoded input patterns are provided. Figure 8 relates the firing times of the input neurons with their index and Figure 9 with their spatial coordinate.

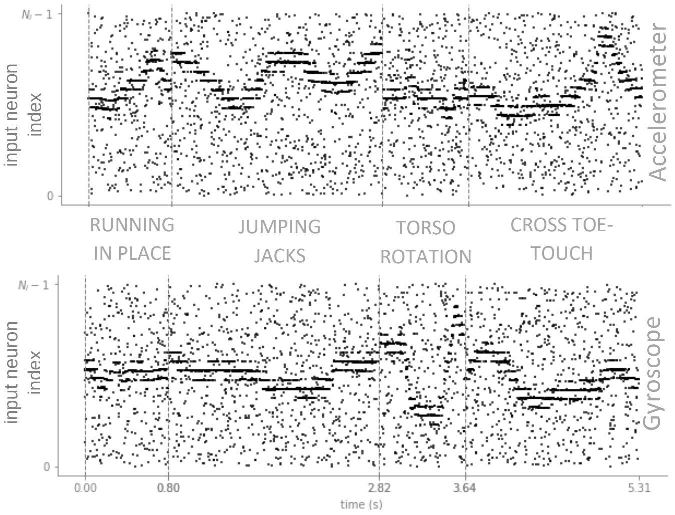

Figure 8. Dataset movements spike patterns.

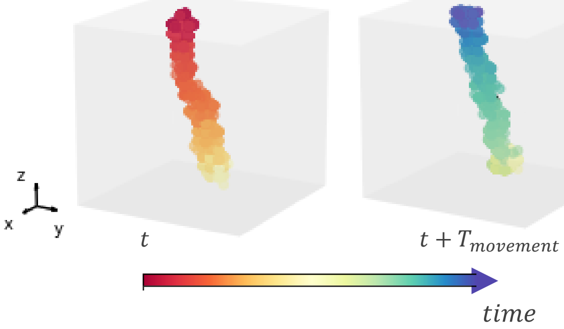

Figure 9. Spatial distribution of input spikes for torso rotation activity referred to the gyroscope.

5.2 Network performance

The implemented solution demonstrated a good performance when tested with various parameters sets (see Table 2). 150 samples were used for training, and 50 for testing. During training, the weights were recorded with a fixed periodicity. After, the network recognition performance was assessed for each weight recording, allowing to extract its progressive improvement, as shown by Figure 10. This assessment was performed using the testing samples.

f_{zone} (Hz)	Accuracy (%)	Time to reach maximum accuracy (s)
50	98.5	640
60	98.5	320
70	99.5	680
80	99.5	680
90	99.5	440
100	100.0	200

Table 2. Network accuracy.

Figure 11 shows the learned weights for one the experiments referred above. It is possible to observe that different weights stimulations occurred for each WTA neuron, indicating the successful learning of distinct input pattern by different WTA neurons.

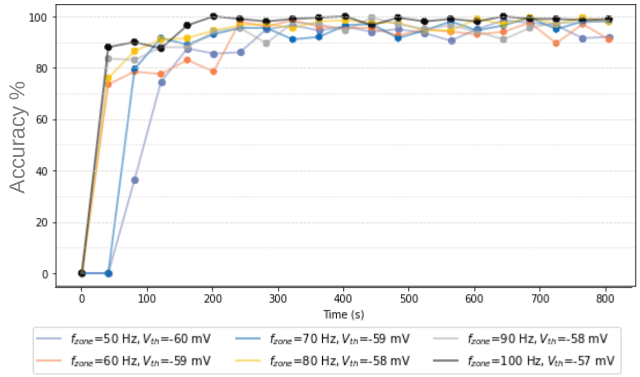

Figure 10. SNN accuracy for multiple input rates.

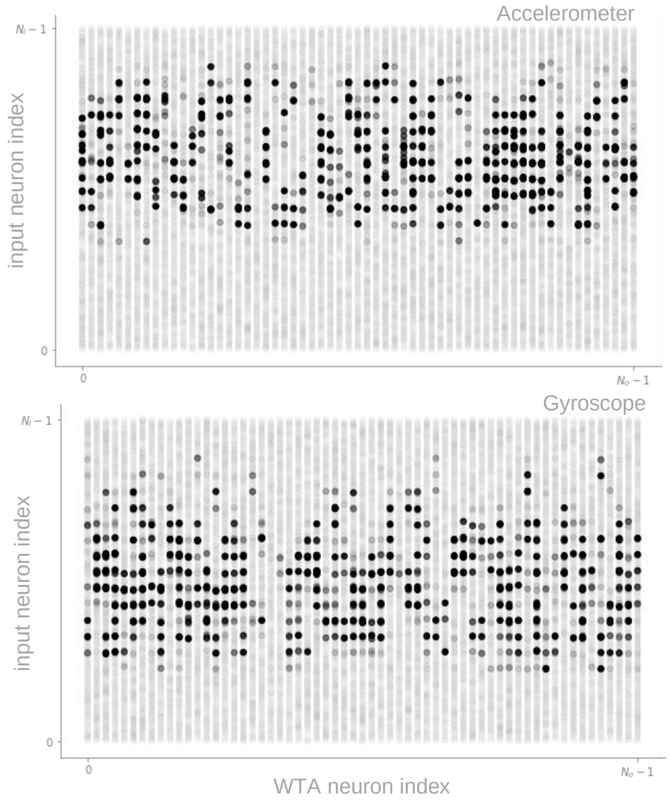

Figure 11. WTA neurons final weights. Blacker dots indicate higher weight values.

6 Conclusion

The presented results demonstrate that the competitive neurons effectively learned spatial patterns consisting of distinct sets of three-dimensional coordinates. The proposed encoding method produces multiple synapses between the input and the competition layers. This proved beneficial as it enabled the learning to be distributed across a greater number of synapses. By synchronizing the WTA neurons response (i.e., synchronize multiple three-dimensional coordinate detections), using delayed synapses, the recognition layer neurons response became maximized, and differentiated, resulting in an accurate identification of the different dataset classes. Although the network shows good performance, the results would be more meaningful if the same system had been applied to a dataset comprising several different participants. Nevertheless, the results indicate the promising potential of the proposed approach.

REFERENCES

[1] E. R. e. a. e. Kandel, Principles of neural science, New York: McGraw-hill, 2000.

[2] D. O. Hebb, "The organization of behavior: A neuropsychological theory" *The American Journal of Psychology* , vol. 63, no. 4, pp. 633-642, 1949.

[3] B. a. Poo, "Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type" *Journal of Neuroscience,* vol. 18, no. 24, pp. 10464-10472, 1998.

[4] P. M. Bi GQ, "Synaptic modification by correlated activity : Hebb's postulate revisited" *Annu. Rev. Neurosci.* , vol. 24, pp. 139-166, 2001.

[5] S. N. a. M. S. DC Somers, "An emergent model of orientation selectivity in cat visual cortical simple cells" *Journal of Neuroscience,* vol. 15, no. 8, pp. 5448-5465, 1995.

[6] W. Maass, "Networks of spiking neurons: The third generation of neural network models" *Neural Networks,* vol. 10, no. 9, pp. 1659-1671, 1997.

[7] A. M, "Role of the cortical neuron: integrator or coincidence detector?" *Isr J Med Sci,* no. 18, pp. 83-92, 1982.

[8] E. A. S. W. Konig P, "Integrator or coincidence detector? The role of the cortical neuron revisited" *Trends Neurosci,* no. 19, pp. 130-137, 1996.

[9] A. F. H. A. L. Hodgkin, "A quantitative description of membrane current and its application to conduction and excitation in nerve" *J Physiol,* vol. 4, no. 117, pp. 500-544, 1952.

[10] S. D. A. &. V. R. R. Thorpe, "Spike-based strategies for rapid processing" *Neural Networks,* vol. 14, no. 6-7, pp. 715-725, 2001.

[11] D. &. L. S. B. Attwell, "An energy budget for signaling in the grey matter of the brain" *Journal of Cerebral Blood Flow & Metabolism,* vol. 21, no. 10, pp. 1133-1145, 2011.

[12] T. Masquelier, "STDP Allows Close-to-Optimal Spatiotemporal Spike Pattern Detection by Single Coincidence Detector Neurons" *Neuroscience,* vol. 389, pp. 133-140, 289.

[13] M. S. R. B. D. F. Goodman, "Brian 2, an intuitive and efficient neural simulator" *eLife,* vol. 8, no. e47314, 2019.

[14] N. T. a. H. M. L. Carnevale, The NEURON Book, Cambridge, UK: Cambridge University Press, (2006)..

[15] M.-O. G. a. M. Diesmann, "NEST (NEural Simulation Tool)" *Scholarpedia,* vol. 2, no. 4, p. 1430, 2007.

[16] "BLE Project Zero" Texas Instruments, [Online]. Available: https://software-dl.ti.com/lprf/simplelink_academy/modules/projects/ble_projectzero/information.html. [Accessed 2023].

[17] "Identification of 3D spatiotemporal patterns using SNNs" [Online]. Available: https://github.com/diogohmsilva/HAR. [Accessed 2023].

[18] i. M. a. G. Deco, "Learning and Coding in Neural Networks." 2013.