Learning hetero-synaptic delays for motion detection in a single layer of spiking neurons

Antoine Grimaldi, Camille Besnainou, Hugo Ladret, Laurent U Perrinet

Institut de Neurosciences de la Timone (UMR 7289); Aix Marseille Univ, CNRS;

ABSTRACT

The response of a biological neuron depends on the precise timing of afferent spikes. This temporal aspect of the neuronal code is essential in understanding information processing in neurobiology and applies particularly well to the output of neuromorphic hardware such as event-based cameras. Though, most artificial neuronal models do not take advantage of this minute temporal dimension and here, we develop a model for the efficient detection of temporal spiking motifs based on a layer of neurons with hetero-synaptic delays. Indeed, the variety of synaptic delays on the dendritic tree allows to synchronize synaptic inputs as they reach the basal dendritic tree. We show this can be formalized as time-invariant logistic regression which can be trained using labelled data. We apply this model for solving the specific problem of detecting motion and demonstrate its use by applying it on synthetic naturalistic videos transformed to event-based streams similar to the output of a silicon retina. In particular, we show how its accuracy may evolve as a function of the total computational load. This end-to-end event-based computational brick could help improve the performance of future Spiking Neural Network solutions currently used in neuromorphic chips.

***Index Terms*—** time code, event-based computations, spiking neural networks, motion detection, efficient coding, logistic regression

1 INTRODUCTION

In 1982, Abeles asked if the role of cortical neurons is whether to integrate synaptic inputs or rather to detect coincidences in temporal spiking patterns [1]. While the first role favors the rate coding theory, the second possibility highlights the need for temporal precision in the neural code. Since, numerous studies demonstrated the emergence of synchronicity in neuron population activity [2, 3], efficient encoding thanks to the use of spike latencies [4, 5] or precise timing in the auditory system [6, 7]. All these findings, and more [8], highlight the importance of the temporal aspect of the neural code and suggest the existence of repeated spatio-temporal patterns in the spike train.

In neuronal models, an efficient use or detection of these spatio-temporal patterns embedded in the spike train comes with the integration of heterogeneous delays [9, 10, 11]. Notably, Izhikevich [12] introduced the notion of polychronous groups (PGs) as a repetitive motif of spikes defined by a subset of neurons with different, yet precise, spiking delays. This representation has a much greater information capacity in comparison to other neural coding approaches through their connectivity and the possible coexistence of numerous superposed PGs.

In this work, we study the emergence of such spatio-temporal spiking motifs when training a single layer of spiking neurons on a supervised classification task (see Fig. 1). We develop a SNN-like method able to learn hetero-synaptic delays to perform motion detection on a synthetic event-based dataset. Because neuromorphic devices are, by design, good candidates to integrate computations with time, we highlight the fact that this event-driven algorithm is transferable to neuromorphic hardware.

2 METHODS

In a recent study, we have introduced a classification model applied to event streams based on Multinomial Logistic Regression (MLR) [13]. To represent temporal relationships between events, this model first builds “time surfaces”, an image computed using the time difference to the last recorded events [14]. By transforming each event in the stream as a vectorial input, this MLR classifier is able to make a decision for every single event. We have demonstrated on several datasets that it provides with online computations resulting in ultra-fast classification. Additionally, we made a formal bridge between the event-based MLR and a Spiking Neural Network (SNN), demonstrating the bio-plausibility of this method and its possible integration to neuromorphic hardware. Here, we propose to extend such a model to a layer of spiking neurons which include hetero-synaptic delays. In particular, one afferent may be connected to different delays and, crucially, we will explicitly use the delay as a computational process. The objective in this model, by including the

AG and LP received funding from the European Union ERA-NET CHIST-ERA 2018 research and innovation program under grant agreement N° ANR-19-CHR3-0008-03 (“APROVIS3D”). HL and LP received funding from the ANR project N° ANR-20-CE23-0021 (”AgileNeuroBot”). HL received funding from a University of Montreal Artificial Intelligence scholarship.

temporal delays' axis, is to increase the representational capacities of the classifier. We will also titrate quantitatively the best trade-off between robustness and computation time when increasing the number of these hetero-synaptic delays.

2.1 Task definition: fast detection of motion

To test our model, we will quantify its ability to categorize different motions. In that order, we use a set of synthetic stimuli, Motion Clouds [15] which are natural-like random textures for which we can control for velocity, among other parameters (see Fig. 1-(b)). In particular, we will set the spatial size to $(N_\mathrm{x}, N_\mathrm{y})$. Also, we will in the following consider a discretization of time with a time step of 1 such that $t \in \mathbb{N}$. Movies' duration are set to $N_\mathrm{t} = 400$. This procedure defines a set of textures with different spatial properties and different motions $\vec{\mathbf{v}}_k$ with $1 \leq k \leq N_v$ and $N_v = 8$ defined by a constant speed and linearly separated directions (see Fig. 1-(a) for an illustration):

$$\vec{\mathbf{v}}_k = \begin{pmatrix} \cos\left(2\pi \cdot \frac{k}{N_v}\right) \\ \sin\left(2\pi \cdot \frac{k}{N_v}\right) \end{pmatrix}$$

This method provides a rich dataset of textured movies for which we know the ground truth. To transform each movie into events, we compute a gradient image (initialized at zero) by adding the gradient of the pixels' intensity over two successive frames. If, on a specific pixel at that specific timestamp, the absolute value of this gradient exceeds a threshold, an event is generated. The event has either an OFF or ON polarity, whether the gradient is positive or negative. This signed threshold value is then subtracted from the residual gradient image. When applied to the whole movie, the event stream is then similar to the output of a neuromorphic camera [16], that is, a list of events defined by x and y (their position on the pixel grid), their polarity p_r (ON or OFF) and time t_r (see Fig. 1-(c)). The goal here is to predict the correct motion by observing the events.

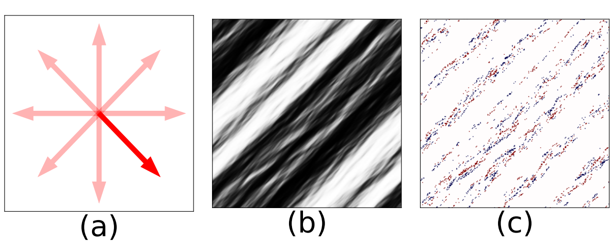

Fig. 1. Task definition. (a) The motion direction represented as the plain red vector, other possible motion directions are represented in light red. (b) A screenshot of the generated motion cloud stimulus at a specific time t_r. (c) The corresponding ON (in red) and OFF (in blue) event stream generated from the stimuli on the left.

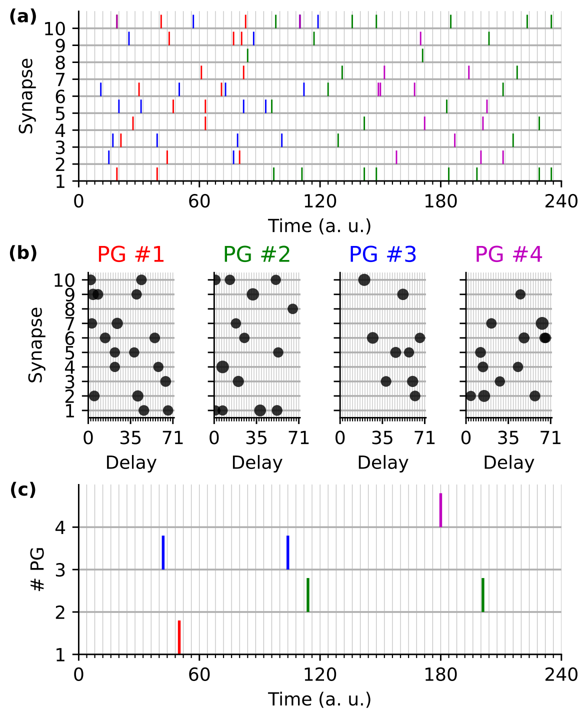

Fig. 2. The hetero-synaptic delays model applied on synthetic data. **(a)** The afferent information consists of the repeated occurrence of groups of precise motifs of spikes that we call "polychronous groups" (PGs). We highlight them by different colors, an information hidden to a detection model. **(b)** The model is defined as an assembly of neurons (here for 4 PGs) each defined by a set of different synapses defined by weights (increasing with the radius of the black dots) at each different delay. Note that each afferent may be connected with multiple weights at different delays. The propagation of the afferent information through these delay may generate at each time step a synchronous pattern on a subset of synapses. **(c)** The output of the model provides with the predicted probability of occurrence of each PG pattern at any time, which may be used to generate a spike as a Bernoulli trial, providing in this particular case with an exact identification of PGs occurrences.

2.2 Hetero-synaptic delays model

The sensory signal representing the output of an event-based camera forms a discrete stream of events (see Fig. 2-(a)). This can be formalized as an ordered set of addresses: $\{a_r\}_{r\in[0,N_{ev})}$ where $N_{ev} \in \mathbb{N}$ is the total number of events in the data stream and the rank r is the index of the event. Each address is typically in the form $a_r = (x_r, y_r, \mathbf{p}_r)$. This defines an address space $\mathcal{A} = [0, N_X) \times [0, N_Y) \times [0, N_\mathrm{p}) \subset \mathbb{N}^3$ where (N_X, N_Y) is the size of the sensor in pixels (width, height) and N_p is the number of polarities ($N_\mathrm{p} = 2$ for ON and OFF polarities). Each event is associated with a time t_r. We may now define a layer of neurons $b \in \mathcal{B}$ by first describing how each neuron connects to presynaptic afferents from $\mathcal{A}$. In biology, a single cortical neuron has generally several thousands of synapses, and each synapse may be defined by the synaptic weight and its delay, that is, the time it takes for one spike to travel from the presynaptic neuron's soma to that of the postsynaptic neuron. Note that one neuron may contact an afferent neuron multiple times with different delays. Formally, we will define in the assembly of neurons $\mathcal{B}$ a set of N_s synapses as $\{\omega_s\}_{s\in[0,N_s)}$, where each synapse ω_s is defined by a weight w_s at different delays τ_s, presynaptic address $a_s \in \mathcal{A}$ and postsynaptic address $b_s \in \mathcal{B}$. As each synapse is assigned one unique address, one may also define the function which gives the index of the synapse as $s = S(a, b, \tau)$.

For each sensory input, the corresponding presynaptic spikes $\{(a_r, t_r)\}_{r\in[0,N_{ev})}$ will be transformed by this synaptic set and notably by the respective delays which will multiplex in time all possible patterns. In particular, some specific motifs may become tightly synchronized as they reach the basal dendritic tree (see Fig. 2-(b)). This may be formalized at any given time t and for each neuron b by registering how the active spikes form a subset of addresses $\mathcal{A}^b(t) \subset \mathcal{A}$:

$$\mathcal{A}^b(t) = \{a_r | t = t_r + \tau_s \text{ and } b_s = b\}_{r\in[0,N_{ev}),\, s\in[0,N_s)}$$

We will now see how these motifs may be detected by postsynaptic neurons.

2.3 Temporal Logistic Regression

By assuming that the polychronous motifs appear repeatedly (see Fig. 2-(a)), we may indeed consider that the occurrence of spikes in $\mathcal{A}^b(t)$ may be formalized as binary latent variables on $\mathcal{A}$. Following our recent work [13], we will assign an output neuron b to each motion direction k for both polarities. In a MLR model with N_{class} classes ($N_{\text{class}} = 2 \cdot N_v$), a probability value is predicted for each event a_r at time t_r as a softmax function of the linear combination of the count of events at time t on the basal dendrite of b, weighted by the synaptic weights $w^b = \{w_s; s \in [0, N_s) | b_s = b\}$ of that postsynaptic neuron is $Pr(k = b|t) = \frac{1}{Z}\exp\big(\mathcal{C}^b(t) + w_0^b\big)$ where $\mathcal{C}^b(t) = \langle w^b, \mathcal{A}^b(t)\rangle = \sum_{a_r\in\mathcal{A}^b(t)} w_{s(a_r)}$ counts the weights on synapses which are active, w_0^b is the bias term and Z normalizes the probability overs all classes.

From the perspective of simulating these event-based computations on standard chips, it is useful to represent them using dense representations. As such, we may first write any event-based input as the boolean matrix $A \in \{0,1\}^{\mathcal{A}}$. In this simplified model, we will consider that hetero-synaptic delays are limited in range such that the synaptic set can be represented by the dense matrix K^b giving for each neuron b the weights as a function of presynaptic address and delay: $\forall s, K^b(a, \tau_s) = w_s$ (see Fig. 2-(b) for an example on synthetic data). Using this dense representation, the counting defined above becomes:

$$\mathcal{C}^b(t) = \sum_{a,\tau_s} K^b(a, \tau_s) \cdot A(a, t - \tau_s)$$

This shows that $\mathcal{C}^b$ is a temporal convolution of the dense representation of the event stream with the dense kernels formed by the synapse set: $\mathcal{C}^b = K^b * A$. This well-known computation defines a differentiable measure which is very efficiently implemented for GPUs and which we will use for learning the classification of different patterns in the event stream.

The metric $\mathcal{C}^b$ was defined in an assembly of neurons (see Fig. 2) and we may further adapt it for event-based cameras. In particular, we may extend the convolution to a 3D convolution such that the representation would also benefit from spatial invariance. For that, we design 3D kernels of shape $(K_{\text{x}}, K_{\text{y}}, K_{\text{t}}) = (15, 15, 8)$, respectively the two sizes in the spatial domain and the range of delays. Computations are performed on spatio-temporal windows, defined by the kernels, sliding around the events, that is, the center of the spatio-temporal window around the current event a_r. Finally, the output of the MLR model results in an event with the highest probability class, keeping the same timing as the event as input. The loss function of the MLR model is the binary cross entropy on the output of the classification layer. Simulations are performed thanks to the PyTorch library using gradient descent with Adam (2^{12} epochs and a learning rate of 10^{-5}).

3 RESULTS

3.1 3D kernels for motion detection

After training our model, we first observe the weights learned for the different neurons (see Fig. 3). Through the positive weights, a strong selectivity is observed along specific axes for the different kernels. These directions can be easily associated to the direction of motion controlled in the motion clouds. For instance, the third and the seventh kernels show a horizontal selectivity to motion directions. With the negative weights, one can observe an anti-selectivity for directions that do not correspond to the motion to which the kernel is selective to. This qualitative look at the 3D kernels allows the reader to infer for the 8 different motion directions used to generate our synthetic event streams.

If one focuses on the interpretation of these kernels in terms of spatio-temporal patterns embedded in the event stream, it can lead to interesting outcomes. In [13], a link between event-based MLR training and Hebbian learning is drawn, allowing to say that the present model will learn its weights according to a presynaptic activity associated to the different motion directions. Each neuron becomes selective to a specific motion direction through the learning of an associated prototypical spatio-temporal spike pattern. For the kernels not linked to the class of a training sample, an anti-Hebbian learning is also demonstrated formally in this past study and observed in the kernels of Figure 3 through the negative weights. Each voxel in the 3D kernels defines a specific timestamp and a specific address. Then, our model is able to detect precise spatio-temporal patterns embedded in the spike train and associated to the different motion direction.

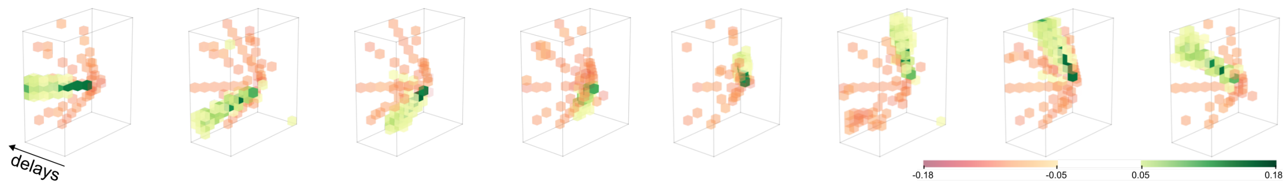

Fig. 3. Representation of the weights for the 8 learned kernels of the model corresponding to the OFF events. Because of the symmetry observed between the ON and OFF event streams, kernels are similar for the ON polarities. For the sake of clarity, the values in range $[-0.05, 0.05]$ are not shown. One sees positive (excitatory) coefficients for the specific direction of motion and negative (inhibitory) coefficients for all other directions.

3.2 Accuracy for the motion detection task

Once our MLR is trained with PyTorch, we obtain 3D kernels corresponding to the weights associated to the hetero-synaptic delays of our layer of spiking neurons and which may be used for detection. We observed that the distribution of the kernels' weights are sparse with most values near zero. As shown in the formalization of our event-based model, the computational cost of our model if implemented on a neuromorphic chip would be dominated by the number of spikes times the number of synapses. This scales with the number of nonzero synaptic weights. To assess the robustness of the classification as a function of the computational load, we will prune the weights in $\{\omega_s\}_{s\in[0,N_s)}$ that are below a defined threshold.

In Figure 4, we plot classification accuracy as a function of the relative number of computations, or active weights, per decision for each neuron of the layer. As a comparison and to account for the necessity of hetero-synaptic delays, we provide the accuracy obtained with a MLR model using 2D time surface (in red) as in [13]. This latter method is based on delays from the last recorded events and uses fewer computations (in our case 15×15) than the dense 3D kernels without any pruning ($8 \times 15 \times 15$). The evolution of accuracy as a function of the log percentage of active weights follows a sigmoid curve. Half-saturation level is reached at $3.5\times10^{-3}\%$ of active weights, corresponding to a total amount of 6 computations per decision. While less computations is needed, the classification performance obtained for the model using time surfaces is similar to our method using all the weights of the kernels. However, accuracy of our method is maintained to its top performances when dividing the number of computations by a factor up to about 200.

4 DISCUSSION

Here, we have introduced a generic SNN using hetero-synaptic delays and shown how it compares favorably with a state-of-the-art event-based algorithm used for classification. This shows that we may use the precise timing of spike to enhance neural computations.

One advantage of our model is the generality of the possible kernels. However this can be a limit as this increases the number of free parameters in the learning algorithm. As we observe that the timing of events in natural scene may be slightly variable (in the order of the synaptic time constant of 5 ms) an extension of the model would be to include a fixed filtering stage similar to that in the tempotron model [9]. This could be efficiently added as a temporal convolution on $\mathcal{C}^b(t)$.

Note that this supervised learning scheme can be extended to a variety of tasks. It would follow from the learning the emergence of new kernels adapted to this new task. This constitutes a major advantage over other algorithms which derive event-based algorithms from specific physical rules (see for instance [17] for computing the optic flow using the luminance conservation rule). We aim at extending the application of this model on more generic datasets acquired in natural conditions for progressively more complex tasks from motion extraction, optic flow or time-to-contact maps.

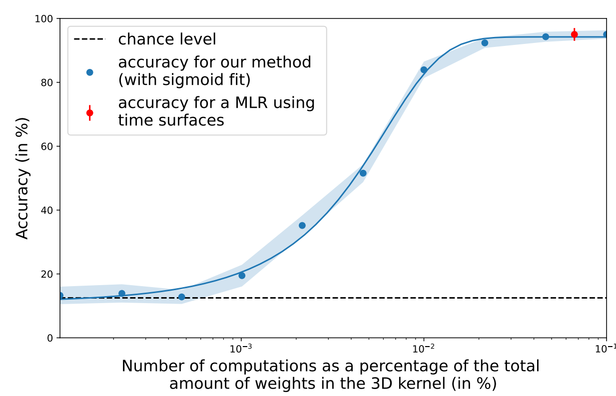

Fig. 4. Accuracy as a function of the number of computation load for the hetero-synaptic delays model (in blue) and for a method using 2D time surfaces (in red) on a log axis. The relative computational load is controlled by changing the percentage of active weights relative to the dense convolution kernel. We observe a similar accuracy than HOTS, yet that our model could achieve a similar accuracy with significantly less coefficients.

5 References

[1] Moshe Abeles, "Role of the cortical neuron: integrator or coincidence detector?," *Israel journal of medical sciences*, vol. 18, no. 1, pp. 83–92, 1982.

[2] Alexa Riehle, Sonja Grun, Markus Diesmann, and Ad Aertsen, "Spike synchronization and rate modulation differentially involved in motor cortical function," *Science*, vol. 278, no. 5345, pp. 1950–1953, 1997.

[3] Zachary W Davis, Gabriel B Benigno, Charlee Fletterman, Theo Desbordes, Christopher Steward, Terrence J Sejnowski, John H Reynolds, and Lyle Muller, "Spontaneous traveling waves naturally emerge from horizontal fiber time delays and travel through locally asynchronous-irregular states," *Nature Communications*, vol. 12, no. 1, pp. 1–16, 2021.

[4] Laurent Perrinet, Manuel Samuelides, and Simon Thorpe, "Coding static natural images using spiking event times: do neurons cooperate?," *IEEE Transactions on neural networks*, vol. 15, no. 5, pp. 1164–1175, 2004.

[5] Tim Gollisch and Markus Meister, "Rapid neural coding in the retina with relative spike latencies," *science*, vol. 319, no. 5866, pp. 1108–1111, 2008.

[6] Michael DeWeese and Anthony Zador, "Binary coding in auditory cortex," *Advances in neural information processing systems*, vol. 15, 2002.

[7] CE Carr and M Konishi, "A circuit for detection of interaural time differences in the brain stem of the barn owl," *Journal of Neuroscience*, vol. 10, no. 10, pp. 3227–3246, 1990.

[8] Sander M Bohte, "The evidence for neural information processing with precise spike-times: A survey," *Natural Computing*, vol. 3, no. 2, pp. 195–206, 2004.

[9] Robert Gütig and Haim Sompolinsky, "The tempotron: A neuron that learns spike timing–based decisions," *Nature Neuroscience*, vol. 9, no. 3, pp. 420–428, Mar. 2006.

[10] Mira Guise, Alistair Knott, and Lubica Benuskova, "A bayesian model of polychronicity," *Neural Computation*, vol. 26, no. 9, pp. 2052–2073, 2014.

[11] Malu Zhang, Jibin Wu, Ammar Belatreche, Zihan Pan, Xiurui Xie, Yansong Chua, Guoqi Li, Hong Qu, and Haizhou Li, "Supervised learning in spiking neural networks with synaptic delay-weight plasticity," *Neurocomputing*, vol. 409, pp. 103–118, 2020.

[12] Eugene M Izhikevich, "Polychronization: computation with spikes," *Neural computation*, vol. 18, no. 2, pp. 245–282, 2006.

[13] Antoine Grimaldi, Victor Boutin, Sio-Hoi Ieng, Ryad Benosman, and Laurent U Perrinet, "A robust event-driven approach to always-on object recognition," *TechRxiv preprint*, January 2022.

[14] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E. Shi, and Ryad B. Benosman, "HOTS: A Hierarchy of Event-Based Time-Surfaces for Pattern Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1346–1359, 2017.

[15] Paula Sanz Leon, Ivo Vanzetta, Guillaume S Masson, and Laurent U Perrinet, "Motion Clouds: Model-based stimulus synthesis of natural-like random textures for the study of motion perception," *Journal of Neurophysiology*, vol. 107, no. 11, pp. 3217–3226, 2012.

[16] Marco Rasetto, Qingzhou Wan, Himanshu Akolkar, Bertram Shi, Feng Xiong, and Ryad Benosman, "The Challenges Ahead for Bio-inspired Neuromorphic Event Processors: How Memristors Dynamic Properties Could Revolutionize Machine Learning," *arXiv:2201.12673 [cs]*, Jan. 2022.

[17] Ryad Benosman, "Asynchronous frameless event-based optical flow," *Neural Networks*, vol. 27, pp. 6, 2012.