Learning hetero-synaptic delays of spiking neurons for motion detection

Antoine Grimaldi
antoine.grimaldi@univ-amu.fr
Aix Marseille Univ, CNRS

Laurent U Perrinet
laurent.perrinet@univ-amu.fr
Aix Marseille Univ, CNRS

Abstract

The response of a biological neuron depends on the precise timing of afferent spikes. This temporal aspect of the neural code is essential in understanding information processing in neurobiology and applies particularly well to the output of neuromorphic hardware such as event-based cameras. However, most artificial neural models do not take advantage of this minute temporal dimension. Inspired by this neuroscientific observation, we develop a model for the efficient detection of temporal spiking motifs based on a layer of neurons with hetero-synaptic delays. Indeed, the connectivity of the dendritic tree allows to discriminate between different temporal sequences, and we show that this can be formalized as a time-invariant logistic regression which can be trained using labelled data. We apply this model to solve one specific computer vision problem, motion detection, and demonstrate its application to synthetic naturalistic videos transformed into event streams similar to the output of event-based cameras. In particular, we quantify how its accuracy can vary with the total computational load. This end-to-end event-driven computational brick could help improve the performance of future spiking neural network (SNN) algorithms currently used in neuromorphic chips.

1. Introduction

In 1982, Abeles queried 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 possibility favors the rate coding theory, the second highlights the function of temporal precision in the neural code. Since, numerous studies demonstrated the emergence of synchronicity in neuron population activity [4, 14], efficient encoding thanks to the use of spike latencies [7, 12] or precise timing in the auditory system [3, 5]. All these findings, and more [2], highlight the importance of the minute temporal structure of the neural code and suggest the existence of repeating spatio-temporal patterns in natural spike trains.

However, most current neuroscience-inspired computer vision algorithms (for instance convolution neural networks) do not make use of this dynamic aspect. A novel emerging solution is that proposed by the representation used in event-based cameras, in which each pixel independently processes its input and emits an event for positive or negative increments of the log-luminance. The shift from the classical, dense representations to this sparse encoding of visual information offers a better analogy to neurobiology, but also offers more energy-efficient computations. Yet, flexible and robust algorithms for classical computer vision tasks, such as motion detection, are still not able to compete with the state-of-the-art of dense computer vision solutions.

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. 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 such hardware.

2. Methods

In a recent study, we have introduced a classification model applied to event streams based on Multinomial Logistic Regression (MLR) [8]. This model first builds *time surfaces*, an image representing the time difference to the last recorded events [9]. 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 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, in addition to weights, include hetero-synaptic delays. In particular, each afferent may be connected with multiple delays and, crucially, we will explicitly utilize the delay as a computational process. The objective in this model, by including the dimension of temporal delays, is to increase the representational capacities of the classifier. In the perspective of building energy-efficient algorithms, 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: motion detection

To test our model, we will quantify its ability to categorize different motions. In that order, we will first define a set of synthetic stimuli, *Motion Clouds* [10], which are natural-like random textures for which we can control for velocity, among other parameters (see figure 1-(b)). In particular, we will set the spatial size to (N_X, N_Y) and we will consider a discretization of time with a time step of 1 such that $t \in \mathbb{N}$. Movies' duration are set to $N_\text{t} = 400$. This procedure defines a set of textures with different spatial properties and different motions v_k with $1 \leq k \leq N_v$ and $N_v = 8$ defined by linearly spaced directions with a constant speed v (see figure 1-(a)): $v_k = (v \cdot \cos(2\pi \cdot \frac{k}{N_v}), v \cdot \sin(2\pi \cdot \frac{k}{N_v}))$. For any given velocity, we also varied the parameters of the textures, such as the mean and variance of the orientation or spatial frequency content. 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, respectively, whether the gradient is negative or positive. 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 [13], that is, a list of events defined by x_r and y_r (their position on the pixel grid), their polarity p_r (ON or OFF) and time t_r (see figure 1-(c)). The goal here is to infer the correct motion solely by observing these events.

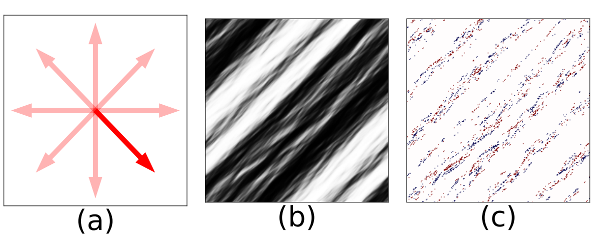

Figure 1. **Motion detection task.** *(a)* The motion direction represented as the plain red vector, other possible motion directions are represented in light red. *(b)* A screenshot of one generated naturalistic textured stimulus at a specific time. *(c)* The corresponding ON (in red) and OFF (in blue) event stream generated from the stimuli on (b) and constituting the input to the spiking neural network.

2.2. Hetero-synaptic delays model

The sensory signal representing the output of an event-based camera forms a discrete stream of events, which can be formalized as an ordered set of addresses and timestamps: $\epsilon = \{(a_r, t_r)\}_{r \in [1, N_{ev}]}$ where $N_{ev} \in \mathbb{N}$ is the total number of events in the data stream and the rank r is the index of each event ϵ_r. This event has a time of occurrence t_r and an associated address, which is typically in the form $a_r = (x_r, y_r, p_r)$. This defines a presynaptic address space $\mathcal{A} = [1, N_\text{X}] \times [1, N_\text{Y}] \times [1, N_\text{p}] \subset \mathbb{N}^3$ where (N_X, N_Y) is the size of the sensor in pixels and N_p is the number of polarities ($N_\text{p} = 2$ for ON and OFF polarities).

We may now define a layer of neurons $n \in \mathcal{B}$ by first describing how each neuron connects to presynaptic afferent from $\mathcal{A}$. In biology, a single cortical neuron has generally several thousands of synapses, and each synapse may be defined by its 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. A postsynaptic neuron $n \in \mathcal{B}$ is then not only described by synaptic weights connecting to a presynaptic afferent from $\mathcal{A}$ but also by the set of possible delays. For each neuron n, we define a set of N_s^n synapses, as $\sigma^n = \{(a_s^n, w_s^n, \delta_s^n)\}_{s \in [1, N_s^n]}$, where each synapse σ_s^n is associated to a weight w_s^n, a delay δ_s^n and a presynaptic address a_s^n. Note that a neuron may contact an afferent neuron with multiple, different delays.

The corresponding input presynaptic spikes ϵ will be integrated by this synaptic set and notably by the respective delays, which will multiplex in time all possible patterns. For each time t the integration of ϵ is defined by a list of weights $\mathcal{W}^n$ linked to the synapses that match a precise spatio-temporal motif as input: $\mathcal{W}^n(t) = \{w_s^n | a_r = a_s^n \text{ and } t = t_r + \delta_s^n\}_{r \in [1, N_{ev}], s \in [1, N_s^n]}$. The activation function of our spiking neuron is a softmax function implementing a form of Multinomial Logistic Regression (MLR) [8], in analogy to a spiking Winner-Take-All network [11]. It transforms this list of weights into a probability with the following formula: $Pr(k = n \mid t) = \frac{1}{Z} \exp(\mathcal{C}^n(t) + b^n)$ where $\mathcal{C}^n(t) = \sum \mathcal{W}^n(t)$ is the sum of the synaptic weights and b^n is the bias linked to neuron n. In particular, some specific motifs may become tightly synchronized as they reach the basal dendritic tree, leading to a high postsynaptic activity which makes it progressively more likely to generate an output spike.

2.3. Temporal Logistic Regression

In our MLR model with $N_\text{class} = N_v$ classes, a probability value is predicted for each event at address a_r and at time t_r as a softmax function of the linear combination of the list of events on the basal dendrite of a neuron n in association to a specific class. The linear combination can be defined by a set of synapses σ^n as described in the hetero-synaptic delays model. From the perspective of simulating

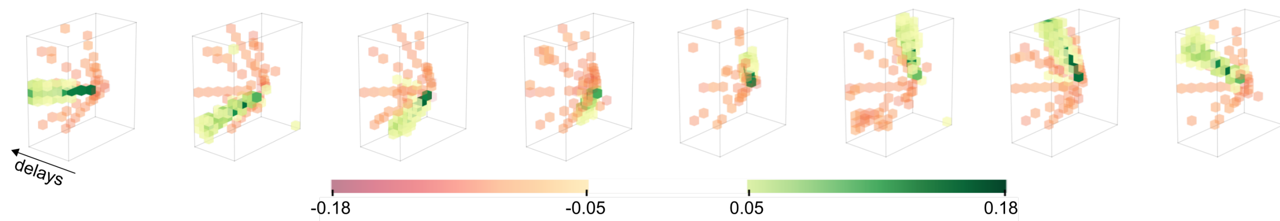

Figure 2. Representation of the weights for the 8 learned kernels of the model corresponding to the OFF polarities and selective to the different motion directions (because of the symmetry observed between the ON and OFF event streams, kernels are similar for the ON polarities). For the sake of clarity, 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.

such event-based computations on standard chips, it is useful to transform this sparse representation into a 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^n giving for each neuron n the weights as a function of presynaptic address and delay: $\forall s \in [1, N_s^n], K^n(a_s^n, \delta_s^n) = w_s^n$. Using this dense representation, the counting defined above becomes:

$$\mathcal{C}^n(a,t) = \sum_{a,\delta_s^n} K^n(a_s^n, \delta_s^n) \cdot A(a, t - \delta_s^n)$$

This shows that $\mathcal{C}^n$ is a temporal convolution of the dense representation of the event stream with the dense kernels formed by the set of synapses: $\mathcal{C}^n = K^n * A$. This well-known computation defines a time-invariant, 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. In particular, we may extend the convolution to a 3D convolution such that the representation would also benefit from spatial invariance. The use of spatiotemporal filters on a stream of events was shown to improve CNN performances for an action recognition task in [6]. For that, we design 3D kernels of shape $(K_\mathrm{x}, K_\mathrm{y}, K_\mathrm{t}) = (15, 15, 8)$, respectively representing the two spatial dimensions 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 ϵ_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. Spatio-temporal kernels for motion detection

After training our model, we first observe the weights learned for the different neurons (see figure 2). Focusing on the positive weights, a strong selectivity is observed along specific axes for the different kernels. These directions can be easily associated to that of motion which is controlled in the stimuli. 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. In [8], 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. Each voxel in the 3D kernels defines a specific timestamp and a specific address. Our model is able to detect precise spatio-temporal patterns embedded in the spike train and associate them to the different motion directions.

3.2. Accuracy for the motion detection task

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 as implemented on a neuromorphic chip would be dominated by the number of spikes multiplied by the number of synapses. As a consequence, computational load scales with the number of nonzero synaptic weights. To assess the robustness of the classification as a function of this load, we will prune the weights which are lower than a defined threshold. In figure 3, we plot classification accuracy as a function of the relative number of computations, or active weights, per decision for each neuron of the layer.

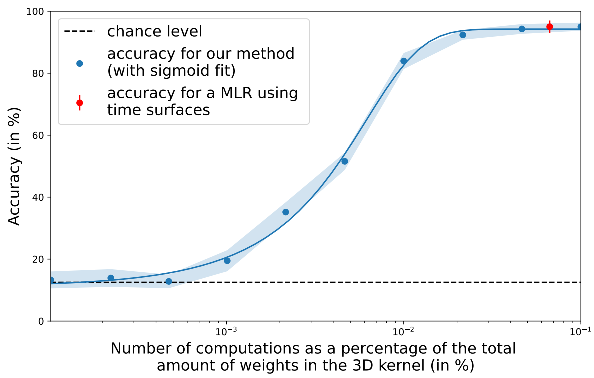

Figure 3. Accuracy as a function of the number of computation load for the hetero-synaptic delays model (in blue) and for a method using 2D time surfaces (in red). The relative computational load (on a log axis) is controlled by changing the percentage of active weights relative to the dense convolution kernel. We observe a similar accuracy than HOTS, yet that our model could achieve a similar accuracy with significantly less coefficients.

As a comparison, we also show as a comparison the accuracy obtained with a MLR model using the HOTS model defined in [8]. 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 ($15 \times 15 \times 8$). 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. By pruning weights, we observe that the evolution of accuracy as a function of the log percentage of active weights follows a sigmoid curve. Half-saturation is reached at $3.5 \times 10^{-3}\%$ of active weights, that is, with about 6 computations per decision. Advantageously, the accuracy for our method is maintained to its top performances with a computational load divided by a factor up to about 200.

4. Discussion

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 a spike to enhance neural computations. One advantage of our model is the generality of the approach. Indeed, this supervised learning scheme can be extended to a novel task by defining a new set of supervision pairs (for instance supervised by local orientation) which would lead to 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. We aim at extending the application of this model on more generic datasets acquired in natural conditions for progressively more complex tasks such as time-to-contact maps.

Acknowledgments

Authors received funding from the European Union ERA-NET CHIST-ERA 2018 N° ANR-19-CHR3-0008-03 ("APROVIS3D") and from ANR project N° ANR-20-CE23-0021 ("AgileNeuroBot"). Authors have applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

[1] M Abeles. Role of the cortical neuron: integrator or coincidence detector? *Israel J med sci*, 18(1):83–92, 1982. 1

[2] S Bohte. The evidence for neural information processing with precise spike-times: A survey. *Natural Computing*, 3(2):195–206, 2004. 1

[3] C Carr and M Konishi. A circuit for detection of interaural time differences in the brain stem of the barn owl. *Journal of Neuroscience*, 10(10):3227–3246, 1990. 1

[4] Z Davis, G Benigno, C Fletterman, T Desbordes, C Steward, T Sejnowski, J Reynolds, and L Muller. Spontaneous traveling waves naturally emerge from horizontal fiber time delays and travel through locally asynchronous-irregular states. *Nature Communications*, 12(1):1–16, 2021. 1

[5] M DeWeese and A Zador. Binary coding in auditory cortex. *Adv in neural information processing systems*, 15, 2002. 1

[6] R Ghosh, A Gupta, A Silva, A Soares, and N Thakor. Spatiotemporal filtering for event-based action recognition. *CoRR*, abs/1903.07067, 2019. 3

[7] T Gollisch and M Meister. Rapid neural coding in the retina with relative spike latencies. *Science*, 319(5866), 2008. 1

[8] A Grimaldi, V Boutin, S Ieng, R Benosman, and L Perrinet. A robust event-driven approach to always-on object recognition. *TechRxiv preprint*, January 2022. 1, 2, 3, 4

[9] X Lagorce, G Orchard, F Galluppi, B Shi, and R Benosman. HOTS: A Hierarchy of Event-Based Time-Surfaces for Pattern Recognition. *IEEE Trans Pattern Analysis and Machine Intelligence*, 39(7):1346–1359, 2017. 1

[10] P Leon, I Vanzetta, G Masson, and L Perrinet. Motion Clouds: Model-based stimulus synthesis of natural-like random textures for the study of motion perception. *Journal of Neurophysiology*, 107(11):3217–3226, 2012. 2

[11] B Nessler, M Pfeiffer, L Buesing, and W Maass. Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. *PLoS Comput Biol*, 9(4):e1003037, 2013. 2

[12] L Perrinet, M Samuelides, and S Thorpe. Coding static natural images using spiking event times: do neurons cooperate? *IEEE Trans Neural Networks*, 15(5):1164–1175, 2004. 1

[13] M Rasetto, Q Wan, H Akolkar, B Shi, F Xiong, and R Benosman. The Challenges Ahead for Bio-inspired Neuromorphic Event Processors: How Memristors Dynamic Properties Could Revolutionize Machine Learning. *arXiv:2201.12673 [cs]*, Jan. 2022. 2

[14] A Riehle, S Grun, M Diesmann, and A Aertsen. Spike synchronization and rate modulation differentially involved in motor cortical function. *Science*, 278(5345):1950–1953, 1997. 1