ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0. Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

Using event-based metric for event-based neural network weight adjustment

Bruno Cessac[2] and Rodrigo Salas[3] and Thierry Viéville[1] *

(1) Inria Cortex, France `http://cortex.loria.fr`
(2) Inria Neuromathcomp, France
(3) Universidad de Valparaiso. Departamento de Ingenieria Biomedica. Chile

Abstract. The problem of adjusting the parameters of an event-based network model is addressed here at the programmatic level. Considering temporal processing, the goal is to adjust the network units weights so that the outcoming events correspond to what is desired. The present work proposes, in the deterministic and discrete case, a way to adapt usual alignment metrics in order to derive suitable adjustment rules. At the numerical level, the stability and unbiasness of the method is verified.

1 Introduction

Studying the computational power of neural networks with event-based activity (e.g.: [1, 2]) is a well-addressed topic, see [3, 4] for a recent review about spiking network computation, while [5] provides a detailed discussion on temporal aspects of such computations. See [6] for further details on the related modeling choices. In order to contribute to this general topic, we develop here a framework allowing us to effectively adjust the network parameters in order to tune the outcoming events.

Position of the problem We consider an input/output dynamical system with N units, governed by a recurrent function, $\mathbf{V} = \{\cdots V_n[t] \cdots\}$ being the *output* state variable value of the units of output index $n \in \{0, N\{$ (i.e., $0 \leq n < N$) at time $t \in \{0, T\{$. Some output units may be "hidden", i.e. not observed. Here $\mathbf{W}$ stands for the network parameters or "weights", to be tuned. The exact form of $\mathbf{V}$ is not relevant at this stage, but the gradient $\nabla_\mathbf{W} \mathbf{V}_\mathbf{W}$ must be well-defined in order to adjust $\mathbf{W}$. One track is to consider regular forms of $\mathbf{V}$. For a spiking neuron network this means that we have to consider either Hodgkin-Huxley equations, or some suitable reduction like the FitzHugh-Nagumo model or the SRM model [2]. Another track, is to "mollify" $\mathbf{V}$, i.e., defines it as the limit of a series of regular functions, as experimented in [7].

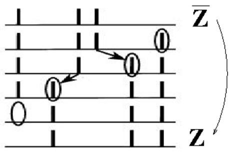
We define an event $Z_n[t] \stackrel{\text{def}}{=} H(V_n[t] - \theta) \in \{0, 1\}$, where H is the Heaviside function, as the fact that the output value is higher than a threshold θ. The goal is thus to adjust the output events $\mathbf{Z}$ of the deterministic discrete-time

*Partially supported by the ANR KEOpS project and CORTINA associated team.

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0. Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

dynamical system $\mathbf{V}$, with respect to a reference output events $\bar{\mathbf{Z}}$. The key point is to deal with the fact that the notion of event is intrinsically "discontinuous".

Considering alignment metric We define the distance between two finite event-trains $\bar{\mathbf{Z}}$, $\mathbf{Z}$ as *the minimum cost of transforming one event-train into another*. See [8, 9] for a general introduction. Following [5], we consider a generalized alignment metric: non-stationary cost (e.g., recent events may count more than older ones) and non-linear shift (e.g., neglecting tiny delays), as described in Fig. 1. Two kinds of operations are defined for an alignment metric.
(i) Event insertion/deletion, the cost of each operation being set to $\gamma^{\pm}_{\bar{t}}$ at time $\bar{t}$, e.g., $\gamma^{\pm}_{\bar{t}} = 1$, while non-stationary different insertion/deletion costs may be defined.
(ii) Event shift, the cost to shift from one event in $\bar{\mathbf{Z}}$ at time $\bar{t}$ to one event $\mathbf{Z}$ at time t, being an increasing positive function of the non-stationary normalized shift delay $\phi_{\bar{t}}((\bar{t}-t)/\tau)$, for a given time-constant τ (e.g. $\phi_{\bar{t}}((\bar{t}-t)/\tau) = |\bar{t}-t|/\tau$), while non-stationary non-linear different forward/backward shift-cost may be defined, since $\phi()$ is parameterized by $\bar{t}$.



From the upper to the lower event train is shown, using from top to bottom an insertion, a rightward shift, a leftward shift and a deletion respectively

Fig. 1: An example of minimal alignment (borrowed from [8]).

Obviously the distance is zero (no editing operation) if and only if both trains are equal, is always bounded by the number of events in both event-trains (i.e. the cost of deleting/inserting all events), thus also by twice the number of samples in the discretized case. For small τ, the distance approaches the number of non-coincident events, since instead of shifting events it is cheaper to insert/delete non-coincident events, while when $\tau \to 0, \gamma^{\pm}_{t} = 1$ we obtain the coincidence (or Hamming) distance equal to the number of non-coincident events. Given two time sequences with the same number of events, there is always a τ high enough for the distance to correspond to the weighted sum of time differences between both train events, as used in, e.g., [4]. More generally, for high τ, the distance basically equals the difference in event number (rate distance) [8].
When considering event-trains with more than one unit, our approach consists to sum the distances for each unit alignment, i.e., consider each unit independently, avoiding the related estimation to suffer from NP-completeness [9].

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0. Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

2 Defining indexed alignment divergence

Since we want to tune the $\mathbf{Z}$ events in order to approximate the $\bar{\mathbf{Z}}$ events, let us introduce an alignment indexation as follow: $\delta(t) = \bar{t} - t$ if the two events $Z[t] = 1$ and $\bar{Z}[\bar{t}] = 1$ are aligned by a shift, $\delta(t) = \pm 0$ to code for an insertion/deletion, while $\delta(t) = 0$ otherwise. In words, we not only compute the distance but make explicit the alignment operations (*shift*, *deletion*, *insertion*) allowing to "edit" $\mathbf{Z}$ in order to obtain $\bar{\mathbf{Z}}$. The δ code function is used to explicitly match both trains.

The distance $d_{\bar{k},k}$ between the first $\bar{k}$ events in $\bar{\mathbf{Z}}$ and the first k events $\mathbf{Z}$ and the related δ indexing are iteratively defined by induction, after [8] but now generalized (see [7] for a detailed derivation). We write $t_k, k > 0$ the k-th value such that $Z[t_k] = 1$, with a similar notation for $\bar{t}_{\bar{k}}, \bar{k} > 0$. On one hand, $d_{\bar{k},0} = \sum_{\bar{l}<\bar{k}} \gamma^-_{\bar{t}_{\bar{l}}}$, since the distance between any event-train and the empty event-train corresponds to the cost of deleting all events, while $\delta(t_{\bar{k}}) = -0$ in this case. Similarly, $d_{0,k} = \sum_{l<k} \gamma^+_{t_l}$ corresponds to inserting all events, with $\delta(t_{\bar{k}}) = +0$. On the other hand:

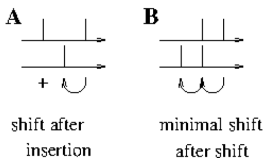
$$
d_{\bar{k}+1,k+1} = \min \begin{cases} d_{\bar{k},k+1} + \gamma^-_{\bar{t}_{\bar{k}}} & (\textit{deletion}), \quad \delta(t_{k+1}) = -0 \\ d_{\bar{k}+1,k} + \gamma^+_{t_k} & (\textit{insertion}), \quad \delta(t_k) = +0 \\ d_{\bar{k},k} + \phi_{\bar{t}_{\bar{k}}}\left(\frac{\bar{t}_{\bar{k}} - t_k}{\tau}\right) & (\textit{shift}), \quad \delta(t_k) = \bar{t}_k - t_k. \end{cases} \tag{1}
$$

Obviously, several alignment operation sequences may lead to the same minimal alignment cost. In order to make a choice, from the last time to the previous time, we consider that shift is preferable to insertion/deletion, since it is a reasonable assumption to heuristic that it is going to have a less important influence on the dynamics than the apparition/cancellation of an unexpected event. This defines algorithmically a unique well-defined indexing function for a given distance, as illustrated in Fig. 2, thus solves the ambiguities. On the reverse, solving these ambiguities allows us to define algorithmically a unique indexing function.

Although computing such a distance and indexes seems subject to a combinatorial complexity, this is a quadratic algorithm (i.e. with a complexity equal to the product of the numbers of events), and its derivation, done by induction, is similar to usual alignment distance calculations [9, 5]. Regarding indexing, this means that we do not have to explore all possible alignment operation sequences, in order to define a globally well-defined process.

This indexing definition also allows us to enrich the original alignment distance by not only considering a composite number describing the distance in terms of shift and insertion/deletion, but allowing to make explicit a numerical approximation of the number of shifts *versus* insertion/deletion. This is the same feature as in message-passing alignment mechanisms [10], but defined here in a much simpler context.

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0. Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.



Edition is realized from the most recent event to the oldest event in the past, i.e., from right to left. The previous specification solves the ambiguity. **A** From right to left, shift is preferred to insertion, thus shift precedes insertion. **B** From right to left, two minimal shifts are preferred to a higher shift and a event coincidence.

Fig. 2: Solving ambiguous equal distance alignments, in the case where a shift cost equals the insertion/deletion cost.

3 Mollified version of the alignment distance

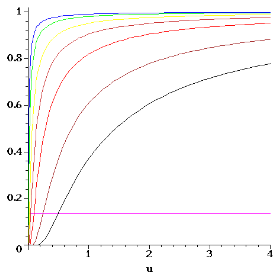
From the previous construction we now introduce the key idea of the paper, i.e., propose a variational expression of the alignment distance. To this aim, we "mollify" the event generation mechanism, i.e. replace the Heaviside function by a suitable convolution $H_\upsilon = \upsilon * H = H(u+\sqrt{\upsilon}\,\nu)\,\exp(-\upsilon/(u+\sqrt{\upsilon}\,\nu))$ where ν is a *margin* maintaining the state at a non-infinitesimal distance to the threshold, and $\upsilon \rightarrow 0$ is the mollification factor (see [6] for details)), as show in Fig. 3. We obtain after some algebra given in [6]: $d\left(\bar{\mathbf{Z}}, \mathbf{Z}\right) = \lim_{\upsilon \rightarrow 0} d_\upsilon\left(\bar{\mathbf{Z}}, \mathbf{V}\right)$, with:

$$\begin{array}{rcl} d_\upsilon\left(\bar{\mathbf{Z}}, \mathbf{V}\right) & = & \sum_{nt, \delta[t]=0} \gamma_t^{\pm}\, H_\upsilon\left((1 - 2\,\bar{Z}_n[t])\,(V_n[t] - \theta)\right) \\ & + & \sum_{nt, \delta[t]\neq 0} \phi_t\left(\frac{\delta[t]}{\tau}\right) \left(\begin{array}{l} Z_n[t] + \\ (1 - Z_n[t])\, H_\upsilon\left(-\frac{\delta[t]}{\tau}\, H_\upsilon(\theta - V_n[t])_{\nu=0}\right) \end{array}\right) \end{array} \quad (2)$$

The key point is that now the criterion is not defined with respect to $\mathbf{Z}$ but $\mathbf{V}$. Qualitatively an increase of $V_n[t]$ tends to shift event in the past, avoid deletion but induce insertion of event, whereas a decrease of $V_n[t]$ tends to shift event in the future, induce deletion but avoid insertion of event. Changes are now differentiable, thanks to the mollification and the gradient $\nabla_{\mathbf{W}} d_\upsilon\left(\bar{\mathbf{Z}}, \mathbf{V}\right) = \nabla_{\mathbf{V}} d_\upsilon\left(\bar{\mathbf{Z}}, \mathbf{V}\right)\, \nabla_{\mathbf{W}}\mathbf{V}$ is obvious to derive. We thus can *tune* $\mathbf{V}$, thus $\mathbf{W}$ to optimize the alignment metric as desired, with a straight-forward implementation for a feed-forward system and the need of specific method for a recurrent structure, as developed elsewhere [7]. The metric allows us to calculate the appropriate network weights by a simple numerical minimization.

Since H_υ is convex for suitable υ [6], for a fixed value of δ the criterion is convex as the sum of positively weighted functions. However the criterion is also optimized with respect to δ and as soon as an event occurrence is modified by a variation of the weights, the indexing is to be recalculated, while, up to our best knowledge, there is no chance to guarantee a global minimum, so that we now turn to numerical verification.

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0. Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.



It is drawn here for $\nu = 0$ and in *black*, *brown*, *red*, *orange*, *yellow*, *green*, *blue*, for $\upsilon = [1, 0.5, 0.2, 0.1, 0.05, 0.02, 0.01]$, respectively. The curves are convex below the magenta horizontal line.

Fig. 3: Defining the mollification of the Heaviside function $H()$.

4 Numerical experiment

Criterion calibration. In order to estimate the performance of the estimation, we must first quantify to which extends we obtain numbers "better than by chance", i.e., to which extends the minimized alignment distance yields a better result than if the output would have been random (a small distance may simply mean that events are sparse !). In order to obtain a correct order of magnitude, we considered the *normalized alignment distance* with respect to a random event train of the same rate. More precisely, if two event trains of T samples are drawn from a Bernoulli distribution of rates r (i.e., samples are random and independent), it is straightforward to obtain the average coincidence distance, i.e. $E[d(\bar{\mathbf{Z}}, \mathbf{Z})] = 2\,T\,r\,(1-r)$. However, the same derivation is not obvious for an alignment distance parameterized with τ and we have numerically interpolated the value as a power of τ, for the standard alignment distance, obtaining for $\gamma_t^{\pm} = 1, \phi_t(s) = |s|$, with a residual standard deviation better than 1.5%:

$$E[d(\bar{\mathbf{Z}}, \mathbf{Z})] = 2\,T\,r\,(1-r)\,\frac{1.183 + 0.183\,r\,(1-r)}{(\tau+1)^{0.265+1.444\,r\,(1-r)}}.$$

Numerical robustness. Let us now illustrate the previous developments considering a leaky integrate and fire (LIF) network, as in [11]. As a test, we have generated hundred of input/output data sets using a "master" network and have verified that the learning algorithm applied on another model of the same dimensions is able to find weights that reproduce the input/output function. Weights values are randomly drawn from a Gaussian distribution of zero mean and standard-deviation $\sigma \in [0.1, 10]$. The LIF resetting mechanism is mollified as for the event thresholding.

This is a basic verification of both the correctness of the code and the numerical stability of the estimation. Hundred of runs have provided correct results, as expected. For long length $T > 10^5$ and complex dynamics the method may fail finding the exact solution with the standard parameters. For small length epoch, as expected, there is always an exact solution, in fact there is one, even if the

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0. Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

raster is not generated from the same model [11].
These tests have been done for various values of $\tau \geq 0$ and using several values of margin $\nu \in [0.01, 0.1]$. The key point is that we can obtain good numerical results "even if" profiles are finally very sharp, using the proposed continuation method, consisting of numerically drive $\upsilon \to 0$. We have experimented using the conjugate gradient algorithm of the GSL (`http://www.gnu.org/s/gsl`) library, but have also checked that this is not a critical choice. Robustness has been checked for $\gamma_t^{\pm} = 1, \phi_t(s) = |s|$ and several for generalized metric also. Further numerical results are provided as supplementary material of this submission, while the code is available in the open-source EnaS library (`http://enas.gforge.inria.fr`).

5 Conclusion

The key point, here, is the non-learnability of even-based networks [12], since it is proved that this problem is NP-complete, when considering the estimation of both weights in the general case, except for exact simulation [11]. We show that we can "elude" this caveat and propose an alternate efficient estimation mechanism, inspired by alignment metrics used in spike train analysis [9], thus providing a complement of other estimation approaches [4], beyond usual convolution metric [9, 5]. At last, the proposed mollification *is* a series of convolution metric, but that converges towards the expected alignment metric.

References

[1] F. Rieke, D. Warland, Rob de Ruyter van Steveninck, and William Bialek. *Spikes, Exploring the Neural Code.* The M.I.T. Press, 1996.

[2] W. Gerstner and W. Kistler. *Spiking Neuron Models.* Cambridge University Press, 2002.

[3] H. Paugam-Moisy and S.M. Bohte. *Handbook of Natural Computing*, chapter Computing with Spiking Neuron Networks. Springer Verlag, 2009.

[4] Benjamin Schrauwen. *Towards Applicable Spiking Neural Networks.* PhD thesis, Universiteit Gent, Belgium, 2007.

[5] Bruno Cessac, H. Paugam-Moisy, and Thierry Viéville. Overview of facts and issues about neural coding by spikes. *J. Physiol. Paris*, 104:5–18, 2010.

[6] T. Viéville, B. Cessac, and R. Salas. Using event-based metric for network weight adjustment (extended version). Supplementary material for ESANN'12 http://www.loria.fr/ vthierry/results/vieville-etal-2012b.pdf, INRIA, 2011.

[7] T. Viéville and L. Bougrain. A general algorithm to estimate recurrent networks weights (extended version). Supplementary material for CAP'12 http://www.loria.fr/ vthierry/results/vieville-etal-2012a.pdf, INRIA, 2011.

[8] J.D. Victor. Spike train metrics. *Current Opinion in Neurobiology*, 15(5):585–592, 2005.

[9] C. Houghton and Victor. *Measuring representational distances - the spike-train metrics approach.* MIT Press,, 2011.

[10] J. Dauwels, F. Vialatte, T. Rutkowski, and A. Cichocki. Measuring neural synchrony by message passing. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[11] H. Rostro-Gonzalez, Bruno Cessac, Juan-Carlos Vasquez, and Thierry Viéville. Back-engineering of spiking neural networks parameters. Research report, INRIA, 2010. in preparation.

[12] Jiří Šíma and Jiří Sgall. On the nonlearnability of a single spiking neuron. *Neural Computation*, 17(12):2635–2647, 2005.