Provable local learning rule by expert aggregation for a Hawkes network

Sophie Jaffard[1] **Samuel Vaiter**[1]
[1]Université Côte d'Azur, CNRS, LJAD

Alexandre Muzy[2] **Patricia Reynaud-Bouret**[1]
[2]Université Côte d'Azur, CNRS, i3S

Abstract

We propose a simple network of Hawkes processes as a cognitive model capable of learning to classify objects. Our learning algorithm, named HAN for Hawkes Aggregation of Neurons, is based on a local synaptic learning rule based on spiking probabilities at each output node. We were able to use local regret bounds to prove mathematically that the network is able to learn on average and even asymptotically under more restrictive assumptions.

1 INTRODUCTION

Recordings of human brain suggest that concepts are represented through sparse set of neurons that fire when the concept is activated (Legenstein et al., 2016). In this sense, Spiking Neural Networks (Tavanaei et al., 2019) are more biologically plausible than Artificial Neural Networks if one wants to understand how the brain encodes information at neuronal level, and stochastic modelling is particularly relevant (Buesing et al., 2011).

Neuroscientists have identified local learning rules to adjust synaptic weights, regrouped in the concept of Spike-Timing-Dependent Plasticity (STDP) process (Tavanaei et al., 2019; Caporale and Dan, 2008; Nessler et al., 2009). This is a form of Hebbian learning (Hebb, 1949), where connections between neurons are strengthened or weakened depending on their relative spike times in a short time-window. Other biological rules have been used, for instance to model the olfactory system (Kepple et al., 2019). However, to our knowledge there is no mathematical proof that such local rules enable to learn. If there was, it would help in understanding how local transformations can lead to global learning.

Hawkes processes (Hawkes, 1971) are point processes that are frequently used as models in a variety of settings: network analysis, financial transactions, seismic or health data (Hall and Willett, 2016; Zuo et al., 2020). In particular, a classic application consists in modeling interactions between neurons (Reynaud-Bouret et al., 2013; Lambert et al., 2018). Many works deal with estimation in these models (Yang et al., 2017; Wang et al., 2020), sometimes using recurrent neural networks (Sharma et al., 2019; Zhang et al., 2020). Simulation of large networks of these processes is also widely studied in the literature (Bacry et al., 2017; Phi et al., 2020; Mascart et al., 2022, 2023). Generalizations of these interaction models have also been studied for estimation purposes using deep networks (Mei and Eisner, 2017; Zuo et al., 2020).

Our purpose in the present work is totally different from estimation or simulation. As a first step towards proving mathematically that bio-inspired networks using local learning rules can learn, we use Hawkes networks as a model for a cognitive network that can provably learn to classify objects into one of several categories by updating synaptic weights with a local learning rule. See an illustrative example in Figure 1. In this network, the output nodes are post-synaptic neurons that produce spikes as a discrete-time Hawkes process (Ost and Reynaud-Bouret, 2020; Bremaud and Massoulie, 1996), whose spiking probability is a function of the weighted sum of the activity of the pre-synaptic neurons at the previous time step. In the case of a linear Hawkes process, Kalikow decomposition (Ost and Reynaud-Bouret, 2020) allows us to interpret these synaptic weights in the previous sum as a probability distribution. In particular, it is possible to randomly choose the presynaptic neuron of interest instead of doing the whole sum over all presynaptic neurons.

This interpretation of the synaptic weights leads to the following local vision: for an output neuron, its presynaptic neurons can be seen as so many experts and the distribution, given by the weights, can be related to an expert aggregation problem. This is why we use

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

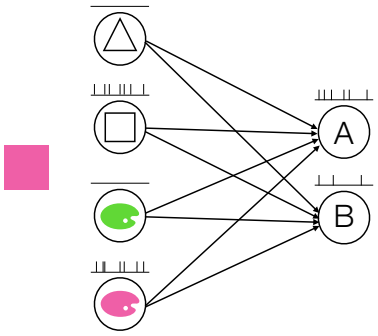

Figure 1: Illustrative example of the network. The presented object excites the neurons encoding its features. Then it is classified in the class coded by the output neuron which spiked the most, here class A.

at this stage an expert aggregation algorithm (Cesa-Bianchi and Lugosi, 2006), or also known as mixture of experts (MoE), to update the weights. However, the key ingredient is that gains of presynaptic neurons are not arbitrary, as usual in expert aggregation, but are selected depending on their spiking probability.

The resulting algorithm is called HAN (Hawkes Aggregation of Neurons), and is general enough for any expert aggregation algorithm.

Contributions. We propose a Hawkes network that learns to classify objects with a local learning rule. More precisely, our contributions are the following:
• We interpret the optimization of synaptic weights as small **expert aggregation** problems that are solved locally by each output neuron, see Algorithm 1 (HAN).
• In the case of a linear Hawkes process, we prove that the network learns to correctly classify objects on average for any expert aggregation algorithm verifying a certain regret bound (Theorem 3.3). More precisely, we have an **oracle inequality** (with constant 1): the obtained network has the same **network discrepancy** in spiking probability as the best possible network up to an additive error in $O(M^{-1/2})$, where M is the number of objects presented to the network during its learning phase.
• In the case of a general Hawkes process, we explicitly compute the limit of the weights when using the specific expert aggregation algorithm EWA (Exponentially Weighted Average) (Cesa-Bianchi and Lugosi, 1999, 2006; Stoltz, 2010).

Related Work. The proposed network is inspired by the Component-Cue model (Gluck and Bower, 1988). In this cognitive model, the objects classified by the network have several features, and the network learns to classify them in the right category by learning combinations of features which predict correctly the category. The features of the objects to classify are represented by input nodes and their categories by output nodes. Mezzadri et al. (2022); Mezzadri (2020) compared this model to the ALCOVE model (Kruschke, 2020), where objects are classified according to their similarities with previously learned objects, and showed that the Component-Cue model is most of the time a better fit to human learning than the ALCOVE model. At the difference with the present work, the original Component-Cue model does not incorporate firing patterns of neurons nor local learning rule.

Kalikow decomposition has been mainly used to prove existence of stationary processes (Galves and Löcherbach, 2013; Galves et al., 2013; Hodara and Löcherbach, 2017). Recently it has been used (Phi et al., 2020) to simulate neurons in interaction with a potentially infinite neural network.

Online learning in a context of a Hawkes network has been used by Hall and Willett (2016), where a dynamic mirror descent is performed to track how events influence future events, and by Yang et al. (2017), to estimate the triggering functions of the processes. However, in these works, online learning has been used to estimate the parameters of the Hawkes processes, whereas in the present work, online learning is used to update synaptic weights to make the Hawkes network learn how to classify objects by itself.

In neuroscience, two main local synaptic rules have been proposed to link a behavior to a corresponding synaptic mechanism and STDP is not one of them. The three-factors rule (Gerstner et al., 2018) assumes that a synaptic weight update depends on (i) the presynaptic activation, (ii) the post-synaptic activation, and (iii) the eventual outcome of the overall behavior. The rate-based learning rule (Kempter et al., 1999) assumes that the weight update depends on the firing rate of pre- and post-synaptic neurons. Our local rule is closer to this approach than to STDP. To the best of our knowledge, the present work mathematically proves for the first time that such local learning rules make a very simple network learn.

2 FRAMEWORK

2.1 First notations and set-up

All the notations are listed in Appendix 6. The objects to be classified have different natures $o \in \mathcal{O}$, each o having different features, each feature being the version of a general characteristic. For instance, a blue square can have the feature "blue" which characteristic is color.

We work in discrete time. A number M of objects are successively presented to the network, each for N time steps. We denote by $o(m) \in \mathcal{O}$ the nature of the object presented to the network during the m^{th} round. We want the network to learn to classify the objects in classes. For class j, M^j is the number of objects belonging to class j in the first M presented objects.

The present network is made of two layers. We denote by I the set of input neurons, and J the set of output neurons. Each output neuron j corresponds to a class, also noted j, in which one wants to classify the objects that are shown. Then the network activity is a sequence of random variables $(X^i_{m,t})_{i\in I\cup J, 1\leq m\leq M, 1\leq t\leq N}$ where

$$X^i_{m,t} = \begin{cases} 1 \text{ if neuron } i \text{ spiked at time } t \text{ for object } m \\ 0 \text{ otherwise.} \end{cases}$$

2.2 The input layer

Each neuron of the input layer emits a discrete process that is a sequence of independent random variables following a Bernoulli distribution (note that we assume temporal independence for one neuron but not independence between neurons). The parameter of the Bernoulli distribution is denoted

$$p^i_m := \mathbb{P}(X^i_{m,1} = 1),$$

hence p^i_m is the spiking probability of neuron i at any time step when presented with the m^{th} object (see Step 4 of Algorithm 1). We assume that p^i_m does not depend on m per see but only on eventually the nature $o(m) \in \mathcal{O}$. For instance, if p^i_m corresponds to an input neuron i detecting the feature "blue" then p^i_m will be the same for all objects with the same feature blue. It could potentially be different for each o, but we are interested in practise by the case where input neurons describe objects through their features, not their nature, as in the Component-Cue model, which describes well human learning (Mezzadri et al., 2022).

2.3 The output layer

The output layer is made of neurons coding for the classes in which the objects are classified, so J is used for both the set of output neurons and classes.

The *rule for classifying the objects* is as follows: the object is classified in the class coded by the output neuron which spiked the most during its presentation, *i.e.,* in $\arg\max_{j\in J} \widehat{p^j_m}$ where $\widehat{p^j_m} := \sum_{t=1}^N X^j_{m,t}/N$.

Each input neuron can impact output neuron j via an inhibitory or excitatory connection. To make the distinction, we consider now the set of signed input neurons $I^{+/-} := I \times \{+,-\}$. From now we denote $i^+ = (i,+) \in I^{+/-}$ an excitatory connection and $i^- = (i,-) \in I^{+/-}$ an inhibitory connection. It is possible that a given I has two connections with j via its i^+ and i^-. These two presynaptic connections will be in this case considered as two experts for j. Let $I^j_+ \subset I^{+/-}$ be all the excitatory presynaptic neurons of j and $I^j_- \subset I^{+/-}$ be all the inhibitory presynaptic neurons of j such that $I^j := I^j_+ \cup I^j_-$ is the set of connections and experts of j.

We assume that input neurons start spiking K time steps before output neurons. Then the conditional spiking probability of neuron j at time t of object m knowing the network past activity is given by

$$p^{j,\text{cond}}_{m,t}(w^j_m) := \varphi\Big(\alpha^j + \sum_{i\in I^j_+} w^{i^+\to j}_m \sum_{k=1}^K g_+(k) X^i_{m,t-k} - \sum_{i\in I^j_-} w^{i^-\to j}_m \sum_{k=1}^K g_-(k) X^i_{m,t-k}\Big) \quad (1)$$

where $\varphi : \mathbb{R} \mapsto \mathbb{R}$ is a Lipschitz function, α^j is the spontaneous activity of neuron j, $w^{i^+\to j}_m$ (resp. $w^{i^-\to j}_m$) is the weight of the excitatory (resp. inhibitory) connection from neuron i to j, $w^j_m := (w^{i^\bullet\to j}_m)_{i^\bullet\in I^j}$ is the weight family, and g_+ and g_- are functions representing the dependency on the past. We denote by $w^j_{1:M} := (w^j_m)_{1\leq m\leq M}$ the total family of synaptic weights of neuron j.

Synaptic weights are updated after every time period during which an object $o(m)$ is presented, so they depend on m. Moreover they represent a probability distribution, that is: for all $j \in J$, $i^\bullet \in I^j$, $1 \leq m \leq M$, $w^{i^\bullet\to j}_m > 0$, and $\sum_{l\in I^j} w^{i^\bullet\to j}_m = 1$. Besides, for all $\bullet \in \{+,-\}$, $g_\bullet$ is such that $g_\bullet(k) \geq 0$ and $\sum_{k=1}^K g_\bullet(k) = 1$. The parameters φ and α^j are chosen such that $p^{j,\text{cond}}_{m,t}(w^j_m) \in [0,1]$ a.s. for any weights w^j_m.

In section 3, we consider the linear case $\varphi = \text{Id}$, $\alpha^j = 0$ and $I^j_- = \emptyset$ (the last two criterion ensuring that $p^{j,\text{cond}}_{m,t}(w^j_m) \in [0,1]$ a.s.). In this framework, one can simulate $X^j_{m,t}$ thanks to the Solo steps of Algorithm 1: the simulation of the activity of only one input neuron is needed. Hence in this case the algorithm is called HAN Solo. The fact that this method indeed gives a process satisfying (1) comes from the Kalikow decomposition of the Hawkes process. For more details about the legitimacy of this operation and the particular case of discrete Hawkes processes, we refer the reader to Section 4.2.3 of Ost and Reynaud-Bouret (2020) and Appendix 11.

2.4 Learning rule

We use an expert aggregation algorithm to update the weights. One interpretation of the expert aggregation

problem (Cesa-Bianchi and Lugosi, 2006) is as follows: a forecaster can choose between several experts, each with an unknown gain, during M rounds. In each round, the forecaster defines a strategy, *i.e.,* a probability distribution over the set of experts, and receives the corresponding aggregate sum of the gains. An expert aggregation algorithm is a function used to update the probability distribution in order to maximize gains. This update depends only on past gains.

Here, each output neuron j is a forecaster, and the experts are its connections to input neurons I^j. A round corresponds to the presentation of an object, and the synaptic weights, that we reinterpret as a probability distribution thanks to the intuition given by Kalikow decomposition in the linear case, correspond to the probability distribution chosen by neuron j in the expert aggregation. The gain of connection $i^\bullet$ (the expert) w.r.t. the output neuron j (the forecaster) at round m is denoted by $g_m^{i^\bullet\to j}$ and is defined precisely in the next section. We denote by $G_m^j := \sum_{m'=1}^m \sum_{i^\bullet\in I_j} w_{m'}^{i^\bullet\to j} g_{m'}^{i^\bullet\to j}$ the cumulated gain of output neuron j until round m, and $G_m^{i^\bullet\to j} := \sum_{m'=1}^m g_{m'}^{i^\bullet\to j}$ the cumulated gain of connection $i^\bullet$.

The weights of neuron j for the next round are then updated thanks to previously acquired knowledge about the gains, that is, G_m^j and $(G_m^{i^\bullet\to j})_{i^\bullet\in I_j}$:

$$w_{m+1}^j = f(G_m^j, (G_m^{i^\bullet\to j})_{i^\bullet\in I_j}) \tag{2}$$

where the function f is the expert aggregation algorithm. Let us give two examples of such algorithms (see details and other algorithms in Cesa-Bianchi and Lugosi (2006)) that we will use in sections 3.2 and 4:

• EWA (Exponentially Weighted Average)

$$w_{m+1}^{i^\bullet\to j} = \frac{\exp\left(\eta^j G_m^{i^\bullet\to j}\right)}{\sum_{k\in I^j}\exp\left(\eta^j G_m^{k\to j}\right)}. \tag{3}$$

The parameter η^j is called the learning rate.

• PWA (Polynomially Weighted Average)

$$w_{m+1}^{i^\bullet\to j} = \frac{(G_m^{i^\bullet\to j} - G_m^j)_+^{\beta^j-1}}{\sum_{k\in I_j}(G_m^{k\to j} - G_m^j)_+^{\beta^j-1}} \tag{4}$$

where $\beta^j \geq 2$ is a parameter to choose.

Note that EWA and PWA implement two different strategies: EWA only takes into account the cumulated gains of the experts (*i.e.,* connections i^+ and i^-) and assigns strictly positive weights, whereas PWA compares them with those of the forecaster (*i.e.,* neuron j), and as soon as an expert is outclassed by the forecaster, it is assigned a weight equal to zero.

This defines Algorithm 1 (HAN, for Hawkes Aggregation of neurons), which is given for any expert aggregation algorithm f. The algorithm's complexity is determined by the number of calls made to the pseudorandom generator for obtaining Bernoulli variables (and potentially the cost of expert update). With D the output nodes degree, we need to simulate $\mathcal{O}(NM(|J| + |I|)$ Bernoulli r.v., and perform $\mathcal{O}(NM|J|KD)$ (resp. $\mathcal{O}(M(|J|D+N|I|))$) elementary operations (scalar addition, exponential) for HAN (resp. HAN-Solo).

Algorithm 1: HAN

Initialization: $G_0^{i^\bullet\to j} := 0$, $w_1^{i^\bullet\to j} := 1/|I^j|$

1 **for** $m = 1$ ***to*** M **do**
2 **for** $t = 1$ ***to*** K **do**
3 **for** $i \in I$ **do**
4 $X_{m,t}^i \sim \mathcal{B}(p_m^i)$.
5 **for** $t = K+1$ ***to*** N **do**
6 **for** $i \in I$ **do**
7 $X_{m,t}^i \sim \mathcal{B}(p_m^i)$.
8 **for** $j \in J$ **do**
9 $X_{m,t}^j \sim \mathcal{B}(p_{m,t}^{j,\text{cond}})$ { **Solo steps** $\hat{\imath}^+ \sim w_m^j$; $\hat{k} \sim g_+$; $X_{m,t}^j \leftarrow X_{m,t-\hat{k}}^{\hat{\imath}}$
10 **for** $j \in J$ **do**
11 **for** $i^\bullet \in I^j$ **do**
12 Compute $g_m^{i^\bullet\to j}$ according to (5).
13 $w_{m+1}^j \leftarrow f(G_m^j, (G_m^{i^\bullet\to j})_{i^\bullet\in I^j})$
 `// aggregate experts using` (2)

Output: $(\arg\max_j \widehat{p_m^j})_m$
 `// classifications`

2.5 Gain formula

To make the network realistic, neurons can learn only thanks to the knowledge of the spikes emitted by the network in the past, and do not have access to spiking probabilities. We use the following gain, computed in Step 12 of Algorithm 1:

$$g_m^{i^+\to j} = \begin{cases} \widehat{p_m^i} \times \frac{M}{M^j} & \text{if } o(m) \in j \\ -\widehat{p_m^i} \times \frac{M}{M^{j'}} \times \frac{1}{|J|-1} & \text{if } o(m) \in j' \end{cases} \tag{5}$$

where $j' \neq j$, $\widehat{p_m^i} := \sum_{t=1}^N X_{m,t}^i/N$ and $g_m^{i^-\to j} = -g_m^{i^+\to j}$. Indeed, if the object belongs to class j, then the network classifies correctly the object if neuron j spikes more than the others, so excitatory (resp. inhibitory) connections get positive (resp. negative) gains to force j to spike more. Otherwise, j should spike less than the neuron coding for the correct class, so the excitatory (resp. inhibitory) connections get negative (resp. positive) gains (*i.e.,* penalties or losses) to get j to spike less. In the present work, the gain $g_m^{i^\bullet\to j}$

depends on the correct class of the presented object, but not on the network own classification, *i.e.,* the category in which the network classified the previous objects. Whether the network correctly classifies the object or not does not influence the gain.

Using this gain supposes that we know in advance the value of M^j, that is the number of presented objects in class j among the first M objects, and this, for every $j \in J$. If it is not the case, we can replace $\frac{M}{M^j}$ by $\frac{m}{m^j}$ at object m, which can be seen as an estimator of the proportion of objects belonging to class j.

2.6 Feasible weight family

In order to study HAN theoretically, we need to define families of feasible weights, which will be ideal weights that do not vary in time and whose performance we want to match. As stated in section 2.2, the activity of input neurons depends only on the nature of the presented object, and not on time. Hence, the spiking probability of neuron j with constant synaptic weights $q^j = (q^{i^\bullet \to j})_{i^\bullet \in I^j}$ when object with nature $o \in \mathcal{O}$ is presented to the network does not depend on time either and is $p_o^j(q^j) := \mathbb{E}[p_{m,t}^{j,\text{cond}}(q^j)]$.

Definition 2.1 (Feasible weight family)**.** *A* feasible weight family *is a constant weight family $q = (q^j)_{j\in J}$ independent of m such that for all $j \in J$, $o \in j$, $j' \neq j$,*

$$p_o^j(q^j) > p_o^{j'}(q^{j'}).$$

The constant

$$Disc_{safe}(q) := \min_{j\in J, o\in j, j'\neq j} \left\{ p_o^j(q^j) - p_o^{j'}(q^{j'}) \right\}$$

is called the safety discrepancy *of the family q. We denote $\mathcal{Q}$ the set of feasible weight families.*

Hence, a feasible weight family is a weights family which enables the network to correctly classify the objects in general; the larger $\text{Disc}_{\text{safe}}(q)$, the lesser it will be mistaken. In Appendix 8, we give examples of such a feasible weight family in a particular case.

In section 3.1, we give theoretical results about HAN Solo when f meets certain conditions; in section 3.2, we study the limit behavior of HAN with the EWA algorithm, and in section 4 we study a specific case of network and we compare numerically HAN with EWA, HAN with PWA and the Component-Cue model from which HAN is inspired.

3 THEORETICAL RESULTS

3.1 Average learning

In this section, we give theoretical guarantees that our network learns to classify objects as well as any feasible weight family on average under certain conditions. We consider the HAN Solo case $\varphi = \text{Id}$, $\alpha^j = 0$ and $I_-^j = \emptyset$. In this simplified framework the set I^j is identified with the set I_+^j so the experts are the input neurons linked to neuron j and we use the notation i instead of i^+.

Expert aggregation algorithms are designed to achieve low regret bounds. More precisely, when applied to our setting, the regret of the forecaster/neuron j is

$$R_M^j := \max_{q^j \in \mathcal{X}^j} \sum_{i\in I^j} q^{i\to j} G_M^{i\to j} - G_m^j$$

where $\mathcal{X}^j$ is the set of probability distributions over I^j. Note that the maximum is achieved for any combination of diracs on the experts with maximum cumulated gain (there can be ties). This regret can be translated in spiking probabilities of neuron j thanks to our particular choice of gain (see Appendix 12). However, if we want to understand how the network learns, we need a more global notion involving the activity of all output neurons. Let

$$\widehat{p_m^j(q_m^j)} := \sum_{i\in I^j} q_m^{i\to j} \widehat{p_m^i}$$

where $q_{1:M}^j := (q_m^j)_{1\leq m\leq M}$ is a weight family. This an estimator of the spiking probability of neuron j if the synaptic weights were given by q_m^j during the presentation of the m^{th} object. For any weights $q_{1:M}^j := (q_m^j)_{1\leq m\leq M}$, we then interpret

$$\widehat{P_M^{j,j'}(q_{1:M}^j)} := \frac{1}{M^{j'}} \sum_{m,\ o(m)\in j'} \widehat{p_m^j(q_m^j)}$$

as an estimator of the average spiking probability of neuron j with weights $q_{1:M}^j$ during the presentation of objects in class j'. In the notation $P_M^{j,j'}$, index j refers to a neuron, whereas index j' refers to a class.

Then the *class discrepancy* of class j is for a network governed by weights $q_{1:M} := (q_{1:M}^j)_{j\in J}$ is defined by

$$\text{Disc}_M^j(q_{1:M}) = \widehat{P_M^{j,j}(q_{1:M}^j)} - \frac{1}{|J|-1} \sum_{j'\neq j} \widehat{P_M^{j',j}(q_{1:M}^{j'})}.$$

It measures how much neuron j fires more than the other neurons when an object of class j is shown, and it is therefore a global information at the network level. We give another choice of gain (with some drawbacks) in Appendix 12, where the class discrepancy can be expressed in terms of empirical spiking probabilities.

Finally, the average class discrepancy of output neurons is called the *network discrepancy* and is defined by

$$\text{Disc}_M(q_{1:M}) = \frac{1}{|J|} \sum_{j\in J} \text{Disc}_M^j(q_{1:M}).$$

Like safety discrepancy for a feasible weight family, the network discrepancy measures how much output neurons fire more than the others when an object of their class is shown. However, unlike safety discrepancy, it gives average information rather than quantifying the worst possible deviation.

Assumption 3.1 (Regret bound)**.** *The expert aggregation algorithm f used in HAN Solo is such that for any deterministic sequence of gains $g^j_{1:M} \in [a,b]$,*

$$R^j_M \leq K(|I|, b-a)\sqrt{M}$$

where $K(|I|, b-a)$ is a constant depending on $|I|$ and $b-a$.

Both EWA (3) and PWA (4) satisfy Assumption 3.1. See Appendix 9.1 for details on the bounds.

Assumption 3.2. *There exists a constant $\xi > 0$ independent of M such that for every $j \in J$, $M_j/M \geq \xi$.*

Assumption 3.2 means that every class of objects is well represented during the learning phase.

Theorem 3.3 (Oracle inequality)**.** *Suppose Assumptions 3.1 and 3.2 hold. Let $\alpha \in (0,1]$. Suppose $\mathcal{Q}$ is non-empty. Then with probability greater than $1-\alpha$,*

$$Disc_M(w_{1:M}) \geq \max_{q\in\mathcal{Q}} Disc_{safe}(q) - E_{tot}(N,M,\alpha)$$

where $E_{tot}(N,M,\alpha) = E_{reg}(M) + E(N,M,\alpha)$ with

$$E_{reg}(M) := K\Big(|I|, \frac{|J|}{\xi(|J|-1)}\Big)\frac{1}{\sqrt{M}}$$

and

$$E(N,M,\alpha) := \sqrt{\ln\Big(\frac{2|I||J|}{\alpha}\Big)\frac{2}{\xi NM}}.$$

In a nutshell, assuming the error $E_{\text{tot}}(N,M,\alpha)$ is negligible, this result on network discrepancy means that in average, a class neuron spikes more than the other neurons when presented with an object in its class with high probability. Thus, in average, the network correctly classifies the objects under the hypothesis that a feasible weight family exists. The error is twofold: one part, $E_{\text{reg}}(M)$, comes from the regret bound and is in $O(M^{-1/2})$, whatever N. The other part, $E(N,M,\alpha)$, comes from the inherent randomness of our system and is in is in $O((NM)^{-1/2})$. Hence if M is large enough, the total error $E_{\text{tot}}(N,M,\alpha)$ is negligible compared to the constant $\max_{q\in\mathcal{Q}} \text{Disc}_{\text{safe}}(q)$. In this sense, HAN performs as well as an oracle that would know the best feasible weight family in advance: its network discrepancy is larger than the best safety discrepancy, with asymptotic multiplicative constant 1. However if M is not large enough we pay a price in $O(M^{-1/2})$ for having seen only that many objects and being initialized with a weight family that is not feasible. Finally if M and N are not large enough, the randomness in the system increases, and the approximation of the activity of input neurons given by the gains may be insufficient to find the best experts among them, or the time of the presentation of an object could be too short to see which output neuron significantly spikes the most.

3.2 Limit behavior

In this section, independent of the previous one, we are going to study the limit behavior of the network. We are no longer interested in average results: the linearity of φ is not needed anymore and *we are in the general case (1), where inhibition is allowed.* However, we want to conduct a more precise analysis of the network's limit behavior and this analysis can only be carried out on a case-by-case basis, depending on the expert aggregation algorithm chosen. Here, we have decided to use the EWA algorithm, for its simplicity and universality.

Instead of assuming that a feasible weight family exists as in the previous section, we want to build directly the limit of the weights, hoping that this limit makes sense from a learning point of view. But if the input neurons encoding the features have nothing to do with the output class (e.g. two classes "blue" and "red" and all neurons having the same firing rates whatever the color) the problem cannot be solved. This is why we introduce the notion of *feature discrepancy* of connection i^+ (resp. i^-) with respect to class j, defined by:

$$d^{i^+\to j} := \frac{1}{n^j}\sum_{o\in j} p^i_o - \frac{1}{|J|-1}\sum_{j'\neq j}\frac{1}{n^{j'}}\sum_{o\in j'} p^i_o$$

(resp. $d^{i^-\to j} = -d^{i^+\to j}$) where n^j is the number of natures of objects belonging to class j. For excitatory connections (resp. inhibitory), this feature discrepancy is the difference between the average firing rate of neuron i when presented objects belonging to class j, and the average firing rate of neuron i when presented objects belonging to other classes (resp. the opposite). It indicates the extent to which neuron i has higher-than-usual firing rate when presented with objects in category j. Thus one can define the set of connections that are the most sensible to class j: $\tilde{I}^j = \arg\max_{i^\bullet\in I^j} d^{i^\bullet\to j}$, as well as the gap in discrepancy if $\tilde{I}^j \neq I^j$:

$$\gamma^j = \max_{i^\bullet\in I^j} d^{i^\bullet\to j} - \max_{i^\bullet\in I^j\setminus\tilde{I}^j} d^{i^\bullet\to j},$$

which measures how good the most sensible connections are with respect to the others. Note in particular that if all the $d^{i^\bullet\to j}$'s are null, $\tilde{I}^j = I^j$, there is no gap and nothing can be learned from the network because the

input neurons are in fact not sensible to important features for the classification.

Theorem 3.4. *Suppose each nature of object is presented the same amount of times: for all $o \in \mathcal{O}$, $|\{m, o(m) = o\}| = \frac{M}{|\mathcal{O}|}$. Let $\eta^j = \frac{1}{|\mathcal{O}|}\sqrt{2\frac{\ln(|I^j|)}{M}}$ and $w_\infty := (w_\infty^{i^\bullet \to j})_{j\in J, i^\bullet \in I^j}$ where*

$$w_\infty^{i^\bullet \to j} = \begin{cases} |\tilde{I}^j|^{-1} & \text{if } i^\bullet \in \tilde{I}^j \\ 0 & \text{otherwise.} \end{cases}$$

Then with probability $1-\alpha$, for all $j \in J$:
• if $\tilde{I}^j \neq I^j$, then the weights $w_{M+1}^{i^\bullet \to j}$ at the end of the learning phase satisfy

$$|w_{M+1}^{i^\bullet \to j} - w_\infty^{i^\bullet \to j}| \leq E^j(N,\alpha) + E^j_{EWA}(M)$$

where

$$E^j_{EWA}(M) := \max\left\{1, \frac{|I^j|}{|\tilde{I}^j|} - 1\right\} \frac{1}{|\tilde{I}^j|} e^{-\frac{\gamma^j}{|\mathcal{O}|}\sqrt{2\ln(|I^j|)M}}$$

and

$$E^j(N,\alpha) := |I^j|\sqrt{\ln\left(\frac{2|I||J|}{\alpha}\right)\frac{\ln(|I^j|)}{|\mathcal{O}|N}}.$$

• if $\tilde{I}^j = I^j$, then the weights $w_{M+1}^{i^\bullet \to j}$ at the end of the learning phase mostly did not evolve, i.e.

$$|w_{M+1}^{i^\bullet \to j} - |I^j|^{-1}| \leq E^j(N,\alpha).$$

This choice of η^j has good theoretical guarantees (if the time horizon is unknown, we can use a similar time-dependent learning rate providing similar results, see details in Appendix 9.1). Note that this result applies for weights w^j_{M+1} and not w^j_m: it is true only at the end of the learning phase. In the first case, the synaptic weights converge to the constant family w_∞, which is uniform on input neurons with maximal feature discrepancy. The error is twofold. One part, $E^j(N,\alpha)$, coming from the randomness of our system is in $O(N^{-1/2})$ and is the equivalent of the error term $E(N,M,\alpha)$ in the oracle inequality (Theorem 3.3). The other part, $E^j_{EWA}(M)$, coming from the fact that we have only seen M objects and very specific to the EWA algorithm is in $\exp(-O(\gamma^j\sqrt{M}))$ and is the equivalent of the error term $E_{\text{reg}}(M)$ which was specific to the expert aggregation algorithm used in HAN Solo. This points out that the larger the gap in discrepancy the quicker the learning. The second case is the non interesting one, where there is nothing to learn (for large N weights are back to their initial value) either because HAN with EWA is good since initialization or because the features are so badly encoded by the input neurons that nothing can be learned.

Going from the local notion of feature discrepancy to the global notion of network discrepancy is not straightforward, even if it seems intuitive to hope that the weight family with the largest feature discrepancy also achieves the largest network discrepancy. This is why, to complete the circle, we need to assume that w_∞ is a feasible weight family in the next corollary.

Corollary 3.5. *Suppose the assumptions of Theorem 3.4 hold and w_∞ is a feasible weight family. Let L be the Lipschitz constant of φ. Then at the end of the learning phase, with probability $1-\alpha$, for all $j \in J$, $j' \neq j$, $t \in \{1,\dots,N\}$, supposing that we present an object $o(M+1) \in j$ to the network we have*

$$\begin{aligned} &p^{j,cond}_{M+1,t}(w^j_{M+1}) - p^{j',cond}_{M+1,t}(w^{j'}_{M+1}) \\ &\geq \left(p^{j,cond}_{M+1,t}(w^j_\infty) - p^{j',cond}_{M+1,t}(w^{j'}_\infty)\right) - E^{j,j'}_{tot}(N,M,\alpha) \end{aligned}$$

where

$$E^{j,j'}_{tot}(N,M,\alpha) := L \sum_{h\in\{j,j'\}} |I^h|(E^h(N,\alpha) + E^h_{EWA}(M)).$$

Note that $\mathbb{E}[p^{j,\text{cond}}_{M+1,t}(w^j_\infty) - p^{j',\text{cond}}_{M+1,t}(w^{j'}_\infty)] \geq \text{Disc}_{\text{safe}}(q)$. This means that with high probability, at the end of the learning phase the network classifies objects as well as it would with the feasible weight family w_∞, with a decreasing error term of the same order as in Theorem 3.4. This corollary can be seen as a non-average version of the oracle inequality of Theorem 3.3.

4 A CONCRETE EXAMPLE

In this section, we give a specific case for which the limit can be guessed beforehand and is a feasible weight family under certain conditions in both HAN and HAN Solo framework, and we compare numerically HAN with EWA, with PWA and the Component-Cue model (Gluck and Bower, 1988).

4.1 Framework

In this section, we assume that the processes emitted by input neurons are mutually independent. There are c characteristics, each declined in n features. Therefore, a nature of object is identified with c given features, one for each characteristic, and there are n choices for each feature. We consider two classes, class B, containing one nature of object, and class A, containing every possible other nature of object. Class B represents an exception. The features are denoted by $f_{k,l}$, where $k \in \{1,\dots,c\}$ and $l \in \{1,\dots,n\}$, and for each feature $f_{k,l}$, there is one input neuron, also denoted $f_{k,l}$, which spikes with probability p when presented with an object having the feature $f_{k,l}$. Each nature of

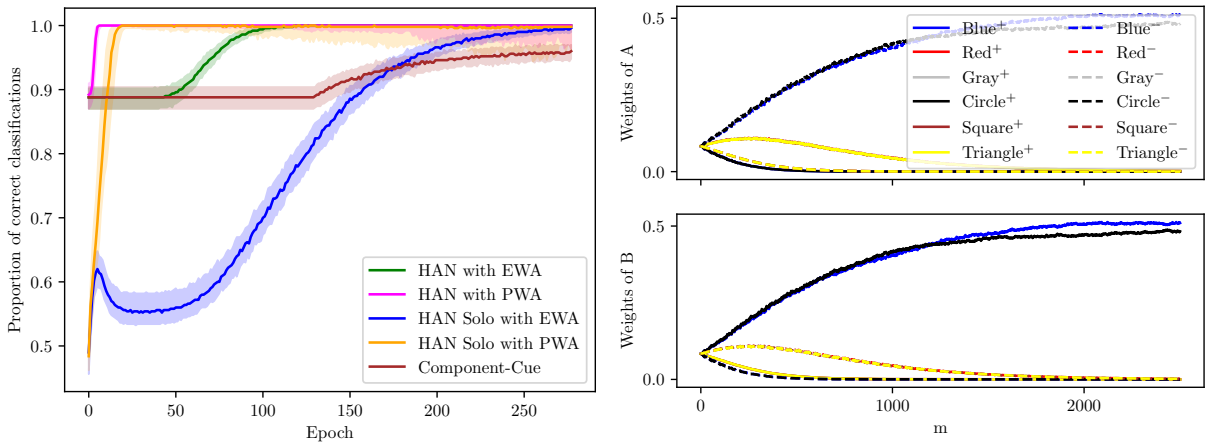

Figure 2: Numerical results with $M = 2502$, $K = 1$, $N = 1000$, $p = 0.2$, $q = 0.3$, $\alpha^A = 0.2$, $\alpha^B = 0$, $\beta^j = 2$ and $\eta^j = \frac{1}{|\mathcal{O}|}(2\frac{\ln(|I^j|)}{M})^{-1/2}$. Parameters of Component-Cue: $\lambda_w = 0.005$ and $\phi = 10$ (see details in Appendix 10). On the left, evolution of the proportion of correct classifications for HAN and HAN Solo with EWA and PWA and Component-Cue with time. A number 100 of realizations were made; for each realization, a testing set of 500 objects drawn randomly was generated. Then the network was trained for 278 epochs, an epoch being a random sequence of the 9 nature of objects. After each epoch the weights were frozen and the network performance was evaluated on the testing set. On the x-axis, number of epochs. On the y-axis, proportion of correctly classified objects of the testing set with confidence interval of level 0.9. On the right, evolution of the weights of neurons A and B with time for one realization of HAN with EWA.

object is presented the same amount of times to the network.

• **HAN**: we study the case $I^j_+ = I^j_- = I$, $\varphi = (\cdot)_+ \wedge 1$, $\alpha^B = 0$ and $K = 1$. Then under some assumptions the limit of the weights, computed in Appendix 8.1, is a feasible weight family and the network correctly classify the objects asymptotically.

• **HAN Solo**: we study the case $I^j_+ = I$, $I^j_- = \emptyset$, $\varphi = \mathrm{Id}$, $\alpha^j = 0$. To replace inhibition, we add input neurons to the network: for each feature $f_{k,l}$, we add a neuron $\tilde{f}_{k,l}$ which spikes with probability q when presented with an object which does not have feature $f_{k,l}$. Hence, neuron $f_{k,l}$ detects the presence of feature $f_{k,l}$, and neuron $\tilde{f}_{k,l}$ detects its absence. Then under some assumptions the limit of the weights, computed in Appendix 8.2, is a feasible weight family and the network correctly classify the objects on average and asymptotically.

4.2 Numerical results

To illustrate this specific case, we use $c = 2$ characteristics with $n = 3$ features for each: the shape, corresponding to the features circle, square and triangle, and the color, corresponding to the features blue, gray and red. The classes are $A = \{\square, \triangle, \bigcirc, \square, \triangle, \bigcirc, \square, \triangle\}$ and $B = \{\bigcirc\}$.

The evolution of the proportion of correct classifications of HAN and HAN Solo with the setting of section 4.1 for both EWA and PWA and Component-Cue are visible on the left of Figure 2 (for a description of Component-Cue and its parameters see Appendix 10). For both PWA and EWA, HAN reaches perfect performance faster than HAN Solo: the use of non-linear φ and inhibition was more effective than the addition of neurons coding for the absence of features. Besides, for both HAN and HAN Solo, PWA learns faster than EWA but its variance is higher and seems to grow in time, whereas the one of EWA decays. Component-Cue is the only one which does not achieve perfect performance at the end of the learning phase, and its variance does not seem to evolve.

The evolution of the weights for one realization of HAN with EWA is visible on the left of Figure 2, which illustrates Theorem 3.4: the weights of A converge to uniform distribution on connections Blue$^-$ and Circle$^-$, and the weights of B converge to uniform distribution on connections Blue$^+$ and Circle$^+$, which is a feasible weight family. Hence A is inhibited by neurons coding for features of $\bigcirc$, which is the object of class B, whereas B is excited by these same neurons.

In Table 1, we provide an ablation study to see how HAN and HAN Solo perform with missing input neurons. The Table gives the percentage of correct classifications for each model at the end of the learning phase depending on the number of ablated features, with the same parameters as in Figure 2. One ablated feature

Table 1: Ablation study

Ablated features	HAN EWA	HAN PWA	HAN Solo EWA	HAN Solo PWA
0/6	99.9	99.5	99.4	98.6
1/6	93.0	92.1	92.8	90.1
2/6	88.6	87.1	82.2	81.6
3/6	83.4	85.2	74.8	68.6
4/6	84.5	84.2	58.7	51.9
5/6	84.9	85.1	55.8	52.5

corresponds to one ablated neuron (resp. two ablated neurons) for HAN (resp. HAN Solo): if feature $f_{k,l}$ is ablated, then neuron $f_{k,l}$ is ablated for HAN and neurons $f_{k,l}$ and $\tilde{f}_{k,l}$ are ablated for HAN Solo. We can see that the network performance is comparable when using either EWA or PWA for both HAN and HAN Solo. However, HAN seems to perform better than HAN Solo with ablated features for both EWA and PWA.

Figure 3 shows a comparison with the well-known perceptron learning algorithm. The performance of the perceptron is comparable to that of HAN with PWA. It should be noted that the comparison is not fair since the perceptron does not involve spikes and is designed for performance, whereas HAN is designed to be cognitively relevant.

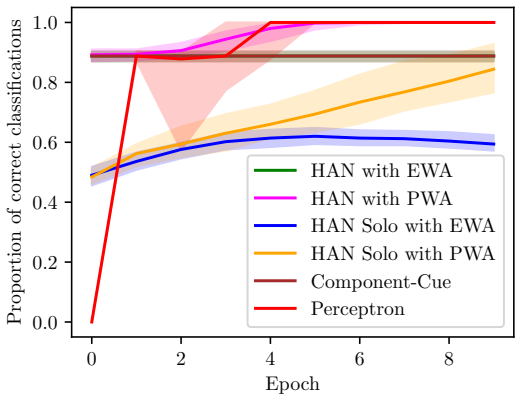

Figure 3: Comparison with the perceptron learning algorithm. Same parameters as in Figure 2, learning rate 1 for the perceptron. Zoom on the 10 first epochs.

5 CONCLUSION

In this paper, we introduced a Hawkes network that provably learns to classify objects thanks to a local learning rule using an expert aggregation method. The main point of our paper is to rigorously prove why our Hawkes network learns. Indeed, our learning rule led to an algorithm (HAN) allowing us to prove an oracle inequality on the network discrepancy in the case of linear Hawkes process, and even limits and rates of convergence in the general case for a specific expert aggregation algorithm. A promising – but ambitious – line of research is to understand if such local rules can be generalized for Hawkes network with one, or more, hidden layers. A first step in this direction could be to add an intermediate layer with neurons detecting correlations in the activity of feature neurons. Another line of research could be to try to prove similar regret bounds for STDP, three-factors or rate-based learning rules.

Acknowledgment

This research was supported by the French government, through CNRS (eXplAIn team), the UCA^{Jedi} and 3iA Côte d'Azur Investissements d'Avenir managed by the National Research Agency (ANR-15 IDEX-01 and ANR-19-P3IA-0002), directly by the ANR project ChaMaNe (ANR-19-CE40-0024-02) and GraVa (ANR-18-CE40-0005), and finally by the interdisciplinary Institute for Modeling in Neuroscience and Cognition (NeuroMod).

References

Emmanuel Bacry, Martin Bompaire, Philip Deegan, Stéphane Gaïffas, and Søren V Poulsen. tick: A python library for statistical learning, with an emphasis on hawkes processes and time-dependent models. *JMLR*, 18(1):7937–7941, 2017.

Pierre Bremaud and Laurent Massoulie. Stability of nonlinear hawkes processes. *The Annals of Probability*, 24(3):1563–1588, 1996. ISSN 00911798.

Lars Buesing, Johannes Bill, Bernhard Nessler, and Wolfgang Maass. Neural dynamics as sampling: A model for stochastic computation in recurrent networks of spiking neurons. *PLOS Computational Biology*, 7(11):1–22, 11 2011.

Natalia Caporale and Yang Dan. Spike timing–dependent plasticity: A hebbian learning rule. *Annual Review of Neuroscience*, 31(1):25–46,

2008. doi: 10.1146/annurev.neuro.31.060407.125639. PMID: 18275283.

Nicolo Cesa-Bianchi and Gábor Lugosi. On prediction of individual sequences. *Ann. Stat.*, 27(6):1865–1895, 1999.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Antonio Galves and Eva Löcherbach. Infinite systems of interacting chains with memory of variable length—a stochastic model for biological neural nets. *J. Stat. Phys.*, 151(5):896–921, 2013.

Antonio Galves, Nancy Lopes Garcia, Eva Löcherbach, and Enza Orlandi. Kalikow-type decomposition for multicolor infinite range particle systems. *Ann. Appl. Probab.*, 23(4):1629–1659, 2013.

Wulfram Gerstner, Marco Lehmann, Vasiliki Liakoni, Dane Corneil, and Johanni Brea. Eligibility traces and plasticity on behavioral time scales: experimental support of neohebbian three-factor learning rules. *Front. Neural Circuits*, 12:53, 2018.

Mark A Gluck and Gordon H Bower. From conditioning to category learning: an adaptive network model. *J. Exp. Psychol.*, 117(3):227, 1988.

Eric C Hall and Rebecca M Willett. Tracking dynamic point processes on networks. *IEEE Trans. Inf. Theory*, 62(7):4327–4346, 2016.

Alan G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58 (1):83–90, 1971.

D.O. Hebb. *The organization of behavior*. Psychology Press, 1949.

Pierre Hodara and Eva Löcherbach. Hawkes processes with variable length memory and an infinite number of components. *Adv. Appl. Probab.*, 49(1):84–107, 2017.

Richard Kempter, Wulfram Gerstner, and J Leo Van Hemmen. Hebbian learning and spiking neurons. *Phys. Rev. E*, 59(4):4498, 1999.

Daniel R Kepple, Hamza Giaffar, Dmitry Rinberg, and Alexei A Koulakov. Deconstructing odorant identity via primacy in dual networks. *Neural Computation*, 31(4):710–737, 2019.

John K Kruschke. Alcove: An exemplar-based connectionist model of category learning. In *Connectionist psychology: a text with readings*, pages 107–138. Psychology Press, 2020.

Regis C Lambert, Christine Tuleau-Malot, Thomas Bessaih, Vincent Rivoirard, Yann Bouret, Nathalie Leresche, and Patricia Reynaud-Bouret. Reconstructing the functional connectivity of multiple spike trains using hawkes models. *J. Neurosci. Methods*, 297:9–21, 2018.

Robert Legenstein, Christos H Papadimitriou, Santosh S Vempala, and Wolfgang Maass. Variable binding through assemblies in spiking neural networks. In *CoCo@ NIPS*, 2016.

C. Mascart, D. Hill, A. Muzy, and P. Reynaud-Bouret. Scalability of large neural network simulations via activity tracking with time asynchrony and procedural connectivity. *Neural Comput.*, 34(9):1915–1943, 2022.

Cyrille Mascart, David Hill, Alexandre Muzy, and Patricia Reynaud-Bouret. Efficient simulation of sparse graphs of point processes. *ACM Trans. Model. Comput. Simul.*, 33(1–2), feb 2023. ISSN 1049-3301. doi: 10.1145/3565809.

Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. In *NeurIPS*, 2017.

Giulia Mezzadri. *Statistical inference for categorization models and presentation order*. Theses, Université Côte d'Azur, December 2020.

Giulia Mezzadri, Thomas Laloë, Fabien Mathy, and Patricia Reynaud-Bouret. Hold-out strategy for selecting learning models: Application to categorization subjected to presentation orders. *J. Math. Psychol.*, 109:102691, 2022.

Bernhard Nessler, Michael Pfeiffer, and Wolfgang Maass. Stdp enables spiking neurons to detect hidden causes of their inputs. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009.

Guilherme Ost and Patricia Reynaud-Bouret. Sparse space–time models: Concentration inequalities and lasso. *Ann I. H. Poincare B.*, 56(4):2377–2405, 2020.

Tien Cuong Phi, Alexandre Muzy, and Patricia Reynaud-Bouret. Event-scheduling algorithms with kalikow decomposition for simulating potentially infinite neuronal networks. *SN Comput. Sci.*, 1(1):1–10, 2020.

Patricia Reynaud-Bouret, Vincent Rivoirard, and Christine Tuleau-Malot. Inference of functional connectivity in neurosciences via hawkes processes. In *GlobalSIP*, pages 317–320, 2013.

Abhishek Sharma, Aritra Ghosh, and Madalina Fiterau. Generative sequential stochastic model for marked point processes. In *ICML Time Series Workshop*, 2019.

Gilles Stoltz. Sequential aggregation of predictors: General methodology and application to air-quality fore-

casting and to the prediction of electricity consumption. *J. Soc. Fr. Stat.*, 151(2):66–106, 2010.

Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothée Masquelier, and Anthony Maida. Deep learning in spiking neural networks. *Neural Networks*, 111:47–63, 2019. ISSN 0893-6080.

Haoyun Wang, Liyan Xie, Alex Cuozzo, Simon Mak, and Yao Xie. Uncertainty quantification for inferring hawkes networks. In *NeurIPS*, 2020.

Yingxiang Yang, Jalal Etesami, Niao He, and Negar Kiyavash. Online learning for multivariate hawkes processes. In *NeurIPS*, 2017.

Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive hawkes process. In *ICML*, 2020.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *ICML*, 2020.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes.**
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **Yes.** HAN/HAN Solo's complexity is directly linked to the number of rv simulation calls.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. **Yes.**

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. **Yes.**
   (b) Complete proofs of all theoretical results. **Yes.**
   (c) Clear explanations of any assumptions. **Yes.**

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **Yes.**
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). **Yes.**
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **Yes.**
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **Yes.** Personal computer without GPU (no timing)

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. **N/A.**
   (b) The license information of the assets, if applicable. **N/A.**
   (c) New assets either in the supplemental material or as a URL, if applicable. **N/A.**
   (d) Information about consent from data providers/curators. **N/A.**
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. **N/A.**

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. **N/A.**
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. **N/A.**
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. **N/A.**

Appendix for "Provable local learning rule by expert aggregation for a Hawkes network"

Appendix 6 provides a table of notations, Appendix 7 − 12 provide additional theoretical and numerical results, and Appendix 13 provides the proofs of our results.

6 TABLE OF NOTATIONS

All the notations are listed in Table 2.

7 ADDITIONAL EXPERIMENTS

On Figure 4, the evolution of the empirical spiking probabilities (*i.e.,* $\frac{N_m^A}{N}$ and $\frac{N_m^B}{N}$) with time for each nature of object is visible, for the realization of HAN Solo with EWA illustrated on the left of Figure 2.

We can see that neuron B does not spike at all when presented with objects having no feature in common with the blue circle (*i.e.,* the red square, red triangle, gray square and gray triangle), and spikes less than A when presented with objects having one feature in common (*i.e.,* the blue square, blue triangle, red circle and gray circle). At the beginning, B spikes less than A when presented with the blue circle and after some time it starts spiking more. This explains why on the right of Figure 2, the curve of HAN with EWA is constant at around 8/9 (8 of the 9 natures of object are well classified) and after some time rises to 1.

On Figure 5, we can see the evolution of the proportion of correct classifications for HAN and HAN Solo with EWA and PWA and Component-Cue with the same parameters as in Figure 2, but here the objects are randomly selected with replacement so the network is not guaranteed to see all natures of objects in a given epoch. All the variances are increased compared to Figure 2; however, the one of HAN with EWA decreases after some time and HAN with EWA is the only algorithm reaching almost perfect performance. Although HAN Solo with EWA has a high variance, its performance is improving and could reach higher values with more time, but the curves of the other algorithms do not seem to be converging towards 1.

8 DETAILS ABOUT THE LIMIT WEIGHTS OF SECTION 4.1

8.1 Study of HAN with EWA

Here $I_+^j = I_-^j = I$, $\varphi = (\cdot)_+ \wedge 1$, $\alpha^B = 0$ and $K = 1$.

Proposition 8.1. *Suppose each nature of object is presented the same amount of times,* $n > 2$, $(c-1)p < (1-p)^{c-1}$. *Then the conditions of Theorem 3.4 are verified and there exists* $\alpha^A > 0$ *such that the limit weights* $w_\infty := (w_\infty^A, w_\infty^B)$ *are a feasible weight family such that* w_∞^A *puts the weight* c^{-1} *on every connection* $f_{k,1}^-$ *and* w_∞^B *puts the weight* c^{-1} *on every connection* $f_{k,1}^+$. *Besides,*

$$Disc_{safe}(w_\infty) = \min\left\{\alpha^A(1-p)^{c-1} - \frac{c-1}{c}p, p - \alpha^A(1-p)^c\right\}.$$

Note that w_∞^A uniformly distributes weight on inhibitory connections to neurons active when presented with o_B, while w_∞^B uniformly distributes weight on excitatory connections to neurons active when presented with o^B, so it is easy to see why it is a feasible weight family. Hence, under the assumptions of Proposition 8.2, the conclusion of Corollary 3.5 holds: the network correctly classifies the objects asymptotically.

We can see an illustration of this proposition on Figure 2.

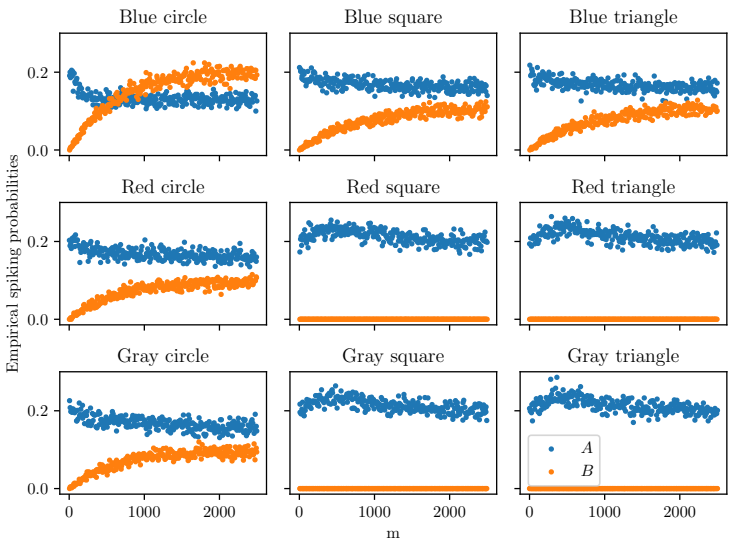

Figure 4: Evolution of the empirical spiking probabilities of neurons A and B with time by nature of object for the same realization of HAN with EWA as in Figure 2 (left). (Same parameters as in Figure 2.)

8.2 Study of HAN Solo with EWA

Here $\varphi = \mathrm{Id}$, $\alpha^j = 0$, $I^j_- = \emptyset$ and $I^j = I$.

Proposition 8.2. *Suppose each nature of object is presented the same amount of times, $n \geq 2$, $p(n-1)^{-1} < q < p(n-1)$ and $q > (c-1)p$. Then the conditions of Theorem 3.4 are verified and the limit weights $w_\infty := (w^A_\infty, w^B_\infty)$ are a feasible weight family such that w^A_∞ puts the weight c^{-1} on every neurons $\tilde{f}_{k,1}$ and w^B_∞ puts the weight c^{-1} on every neurons $f_{k,1}$. Besides,*

$$Disc_{safe}(w_\infty) = \min\left\{\frac{q-p(c-1)}{c}, p\right\}.$$

Note that w_∞ is very close to the one of section 8.1: w^B_∞ is the same and w^A_∞ selects neurons detecting absence of features instead of inhibitory connections. Hence, under the assumptions of Proposition 8.2, the conclusions of Theorem 3.3 and Corollary 3.5 hold: the network correctly classifies the objects in average and asymptotically.

9 REGRET

9.1 Details about the regret bounds of EWA and PWA

1. EWA: the regret bound given in Cesa-Bianchi and Lugosi (2006) holds for losses (*i.e.,* negative gains) taking value in $[0,1]$, but a more general demonstration for only assumed to be bounded is given in Stoltz (2010), and provides the following bound for losses taking value in the interval $[a,b]$ for any $a < b \in \mathbb{R}$:

$$R^j_M \leq \frac{\ln(|I^j|)}{\eta^j} + \eta^j \frac{(b-a)^2}{8} M.$$

With $\eta^j = \frac{1}{b-a}\sqrt{8\ln(|I^j|)/M}$, we obtain

$$R^j_M \leq (b-a)\sqrt{\frac{M}{2}\ln(|I^j|)}.$$

This choice of η^j supposes that we know the time horizon M in advance. If it is not the case, we can use a time-dependent learning rate $\eta^j_m = \frac{1}{b-a}\sqrt{8\ln(|I^j|)/m}$, which gives the bound

$$R^j_M \leq \sqrt{2M\ln(|I^j|)} + \sqrt{\frac{\ln(|I^j|)}{8}}$$

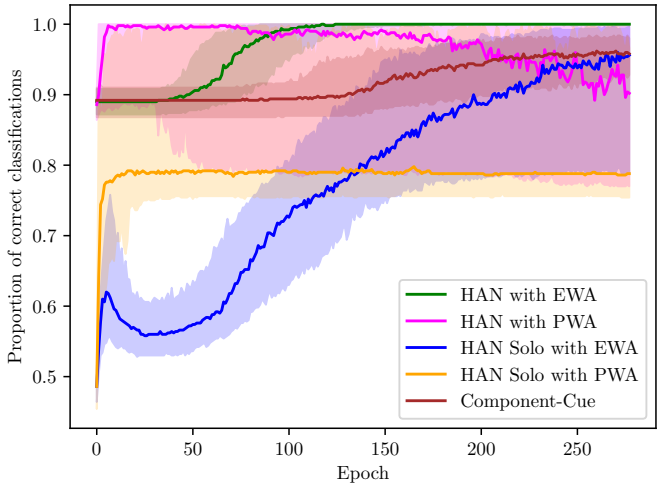

Figure 5: Evolution of the proportion of correct classifications for HAN and HAN Solo with EWA and PWA and Component-Cue with time, with the same parameters as in Figure 2. What changes here is that an epoch is a sequence of 9 objects drawn randomly with replacement: all the natures of object are not necessarily presented during one epoch.

which has the same order of magnitude. Theorem 3.4 can also be adapted with this choice of η^j. Note that under the assumptions of Theorem 3.4, for all j, $\frac{M}{M^j} = \frac{|\mathcal{O}|}{n^j}$ is bounded by $|\mathcal{O}|$ so the gains take value in $[-|\mathcal{O}|, |\mathcal{O}|]$. Hence a good choice of η^j is

$$\eta^j = \frac{1}{2|\mathcal{O}|}\sqrt{8\ln(|I^j|)/M}.$$

2. PWA: the regret bound given in Cesa-Bianchi and Lugosi (2006) holds for losses (*i.e.,* negative gains) taking value in $[0, 1]$. It gives the following inequality:

$$R_M^j \leq \sqrt{(\beta^j - 1)|I^j|^{2/\beta^j} M}.$$

For losses that are only assumed to be taking value in the interval $[a, b]$ for any $a < b \in \mathbb{R}$, we can translate them thanks to the function $x \mapsto \frac{x-a}{b-a}$. We obtain the following bound:

$$R_M^j \leq (b-a)\sqrt{(\beta^j - 1)|I^j|^{2/\beta^j} M}$$

The bound is optimal for $\beta^j = 2\ln(|I^j|)$, which gives

$$R_M^j \leq (b-a)\sqrt{e(2\ln(|I^j|) - 1)M}.$$

Then both bounds can be written in the form of Assumption 3.1 by bounding $|I^j|$ by $|I|$.

9.2 Interpretation of the regret

In the HAN Solo framework, the regret can be interpreted in terms of the neurons activity. The *neuronal discrepancy* of neuron j in a network governed by weights $q^j_{1:M}$ is defined by

$$\mathrm{disc}^j_M(q^j_{1:M}) := \widehat{P^{j,j}_M(q^j_{1:M})} - \frac{1}{|J|-1}\sum_{j' \neq j} \widehat{P^{j,j'}_M(q^j_{1:M})}. \tag{6}$$

It is the difference between the average estimated spiking probability of neuron j over the objects belonging to class j and the average estimated spiking probability of neuron j over objects belonging to other classes,

normalised by the number of objects of each class. It gives information about how much neuron j spikes more than usual when presented objects in category j and when the weights $q^j_{1:M}$ are used. It is a local information (because at neuron j).

Since $\sum_{m=1}^{M} \sum_{i \in I^j} q_m^{i\to j} g_m^{i\to j} = M \mathrm{disc}_M^j(q^j_{1:M})$, we have the following *interpretation of regret in terms of discrepancy*:

$$\frac{R_M^j}{M} = \max_{q^j \in \mathcal{X}^j} \mathrm{disc}_M^j(q^j) - \mathrm{disc}_M^j(w^j_{1:M})$$

where q^j is identified with the constant family of weights $(q^j)_{1\le m\le M}$. Therefore, the regret gives information about the proximity of the neuronal discrepancy of neuron j under HAN Solo with the maximum possible neuronal discrepancy of neuron j with constant weights. Note that this interpretation is made possible thanks to the choice $\varphi = \mathrm{Id}$. This is local information because it is only about the activity of neuron j. To understand the global behavior of the network, we need the class and network dicrepancies.

10 DETAILS ABOUT COMPONENT-CUE

The original Component-Cue algorithm (Gluck and Bower, 1988) is an artificial neural network, with three layers: an input layer receiving the stimuli, an intermediate layer decomposing the stimuli into several features, and an output layer made of category nodes. The intermediate and output layers are linked by weights, that the network updates to learn to classify objects.

Let us detail the Component-Cue algorithm. Let I be the set of features (which is also the set of intermediate neurons), J the set of classes (which is also the set of output neurons), $w_m^{i\to j}$ the synaptic weight between neurons $i \in I$ and $j \in J$ when presented with the m^{th} object. Let

$$a_m^i := \begin{cases} 1 \text{ if object } o(m) \text{ has feature } i \\ 0 \text{ otherwise.} \end{cases}$$

When presented with the m^{th} object, the output neuron j is activated by the quantity

$$O_m^j := \sum_{i \in I} a_m^i w_m^{i\to j}$$

and object $o(m)$ is classified in class j with probability

$$\frac{e^{\phi O_m^j}}{\sum_{l \in J} e^{\phi O_m^l}}$$

where ϕ is a parameter to choose. Then the weights are updated according to the formula

$$w_{m+1}^{i\to j} = w_m^{i\to j} + \lambda_w a_m^i (\tau_m^j - O_m^j)$$

where $\tau_m^j := \begin{cases} 1 \text{ if } o(m) \in j \\ -1 \text{ otherwise} \end{cases}$ and λ_w is the learning rate. It is a gradient descent step.

Note that the choice of the parameters in Component-Cue is tricky, and that the behavior of the algorithm (learning or not) highly depends on this choice (see details in Mezzadri (2020)).

Comparison with HAN: The structure of Component-Cue is very similar to the one of our network; however, Component-Cue does not have a spiking neuronal network interpretation. Indeed, in HAN, spike trains replace the quantities a_m^i, which are real numbers. Besides, unlike HAN, Component-Cue has no theoretical guarantee to correctly classify the objects.

11 KALIKOW DECOMPOSITION

Let us detail Kalikow decomposition in the case of a general discrete-time linear Hawkes process without inhibition (Ost and Reynaud-Bouret, 2020). Let I the set of neurons, $j \in I$ a neuron. Then the spiking probability of neuron i at time 0 knowing the past is

$$p_i(X) = \nu_i + \sum_{s\in\mathbb{Z}^*_-} h_{j\to i}(-s)X_{j,s}$$

where $X = (X_{j,s})_{j\in I, s\in\mathbb{Z}^*_-}$ is the network past activity, $\nu_i \geq 0$ is the spontaneous activity of neuron i, and the functions $h_{j\to i}$ are such that

- $h_{j\to i}(s) \geq 0$ for $s \in \mathbb{N}^*$
- $\sum_{s\in\mathbb{N}} h_{j\to j}(s) + \nu_i \leq 1$.

Then the Kalikow decomposition of $p_i(X)$ is

$$p_i(X) = \lambda_i(\emptyset)p_i^{\emptyset} + \sum_{v\in\mathcal{V}, v\neq\emptyset} \lambda_i(v)p_i^v(X)$$

where $\mathcal{V} = \{\{(j,s)\}, (j,s) \in I\times\mathbb{Z}^*_-\} \cup \{\emptyset\}$ is a family of neighborhoods, $\lambda_i(v)$ is non negative, $\lambda_i(\emptyset) = 1 - \sum_{s\in\mathbb{N}} h_{j\to j}(s)$, $p_i^{\emptyset} = \frac{\nu_i}{\lambda_i(\emptyset)}$, $\lambda_i(\{(j,s)\}) = h_{j\to i}(-s)$ and $p_i^{\{j,s\}}(X) = X_{j,s}$.

Then the Kalikow decomposition of $p_i(X)$ is such that $\lambda_i(v) \geq 0$ for all $v \in \mathcal{V}$, $\sum_{v\in\mathcal{V}} \lambda_i(v) = 1$, $p_i^{\emptyset} \geq 0$ and $p_i^v(X) \geq 0$ for all v. Hence, thanks to Kalikow decomposition, the activity of neuron i can be simulated the following way.

- A neighborhood $v \in \mathcal{V}$ is drawn thanks to the probability distribution λ.
- Then neuron i spikes with probability $p_i^v(X)$ (resp. $p_i^{\emptyset}$ if $v = \emptyset$).

In the HAN Solo case, there is no spontaneous activity. The neighborhoods are the tuples $\{(i,s)\}$ where $i \in I^j$ and $s \in \{t-1,\dots,t-K\}$.

12 OTHER POSSIBLE GAIN

In the HAN Solo framework, another choice of gain is possible:

$$g_m^{i\to j} = \begin{cases} \frac{N_m^{i\to j}}{Nw_m^{i\to j}} \times \frac{M}{M^j} & \text{if } o(m) \in j \\ -\frac{N_m^{i\to j}}{Nw_m^{i\to j}} \times \frac{M}{M^{j'}} \times \frac{1}{|J|-1} & \text{if } o(m) \in j' \neq j \end{cases}$$

if $w_m^{i\to j} > 0$ and

$$g_m^{i\to j} = 0$$

otherwise, where $N_m^{i\to j}$ is the amount of times neuron j spiked after choosing input neuron i in Kalikow decomposition when presented with the m^{th} object. Note that $\frac{N_m^{i\to j}}{Nw_m^{i\to j}}$ is also an estimator of p_m^i: indeed, knowing the weights w_m^j, the variable $N_m^{i\to j}$ follows a binomial distribution with parameter N and $p_m^i w_m^{i\to j}$. Using this gain, the estimator that we consider for p_m^i is

$$\widehat{p_m^i} = \frac{N_m^{i\to j}}{Nw_m^{i\to j}}.$$

The neuronal discrepancy (6) becomes

$$\mathrm{disc}_M^j(w_{1:M}^j) = \frac{1}{M^j} \sum_{m,\ o(m)\in j} \frac{N_m^j}{N} - \frac{1}{|J|-1} \sum_{j'\neq j} \frac{1}{M^{j'}} \sum_{m,\ o(m)\in j'} \frac{N_m^j}{N}$$

where N_m^k is the number of spikes emitted by neuron k during the presentation of the m^{th} object, and the class discrepancy becomes

$$\mathrm{Disc}_M^j(w_{1:M}) = \frac{1}{M^j} \sum_{m,\ o(m)\in j} \frac{N_m^j}{N} - \frac{1}{|J|-1} \sum_{j'\neq j} \frac{1}{M^j} \sum_{m,\ o(m)\in j} \frac{N_m^{j'}}{N}.$$

Therefore, with this choice of gain, the class discrepancy directly compares the number of spikes emitted by output neurons, which gives a better indicator about the network's classifications because the rule to classify objects is precisely about the number of spikes, so its interpretation is easier.

However, this gain causes difficulties because of the division by $w_m^{i\to j}$: when $w_m^{i\to j}$ is close to zero, the network behaviour is difficult to study theoretically. First of all, the gains are not bounded anymore so the regret bounds will depend on the supremum of the gains, which can be large. Besides, an error term in $O\Big(\frac{\ln(M)}{Nw_m^{i\to j}}+\sqrt{\frac{\ln(M)}{Nw_m^{i\to j}}}\Big)$ appears in the regret bound instead of the error term $E(N,M,\alpha)$, which was in $O(\frac{1}{NM})$. This new error is much worse, and converges to zero only if $N >> \frac{\ln(M)}{w_{i\to j}^m}$. However, the weights of connections corresponding to experts which does not have optimal gains tend to converge to zero, as stated in Theorem 3.4. Hence, we cannot be assured to be in this favorable regime. Besides, this gain cannot be generalized in the HAN framework.

13 PROOFS

13.1 Proof of Theorem 3.3

We need a preliminary proposition.

Proposition 13.1 (Regret bound)**.** *Suppose Assumptions 3.1 and 3.2 hold. Then for all $j \in J$,*

$$disc_M^j(w_{1:M}^j) \geq \max_{q^j\in\mathcal{X}^j} disc_M^j(q^j) - E_{reg}(M) \quad a.s.$$

where q^j is identified with the constant family of weights $(q^j)_{1\leq m\leq M}$ and

$$E_{reg}(M) := K(|I|,(1+(1+|J|)^{-1})\xi^{-1})M^{-1/2}.$$

Proof. According to Assumption 3.2, for all $j' \in J$, $M^{j'}/M \geq \xi$. Hence the gains $g_m^{i\to j}$ take value in $[-\frac{1}{\xi(|J|-1)},\frac{1}{\xi}]$ a.s. so according to Assumption 3.1, we have

$$R_M^j \leq K(|I|,(1+(|J|-1)^{-1})\xi^{-1})\sqrt{M} \quad \text{a.s.}$$

We get the result by dividing the previous inequality by M. □

Let $q \in \mathcal{Q}$. We want to bound from below $\mathrm{Disc}_M(w_{1:M})$. We have almost surely

$$\begin{aligned}\mathrm{Disc}_M(w_{1:M}) &= \frac{1}{|J|}\sum_{j\in J}\mathrm{Disc}_M^j(w_{1:M})\\ &= \frac{1}{|J|}\sum_{j\in J}\widehat{P_M^{j,j}(w_{1:M}^j)} - \frac{1}{|J|}\sum_{j\in J}\frac{1}{|J|-1}\sum_{j'\neq j}\widehat{P_M^{j',j}(w_{1:M}^{j'})}\end{aligned}$$

Let us exchange the name of the indexes j and j' in the second term.

$$\mathrm{Disc}_M(w_{1:M}) = \frac{1}{|J|}\sum_{j\in J}\widehat{P_M^{j,j}(w_{1:M}^j)} - \frac{1}{|J|}\sum_{j'\in J}\frac{1}{|J|-1}\sum_{j\neq j'}\widehat{P_M^{j,j'}(w_{1:M}^j)}$$

Let us exchange the sums in the second term.

$$\begin{aligned}\mathrm{Disc}_M(w_{1:M}) =& \frac{1}{|J|}\sum_{j\in J}\widehat{P_M^{j,j}(w^j_{1:M})} - \frac{1}{|J|}\sum_{j\in J}\frac{1}{|J|-1}\sum_{j'\neq j}\widehat{P_M^{j,j'}(w^j_{1:M})}\\ =& \frac{1}{|J|}\sum_{j\in J}\Big(\widehat{P_M^{j,j}(w^j_{1:M})} - \frac{1}{|J|-1}\sum_{j'\neq j}\widehat{P_M^{j,j'}(w^j_{1:M})}\Big)\\ =& \frac{1}{|J|}\sum_{j\in J}\mathrm{disc}_M^j(w^j_{1:M})\\ \geq& \frac{1}{|J|}\sum_{j\in J}\Big(\mathrm{disc}_M^j(q^j) - E_{\mathrm{reg}}(M)\Big)\end{aligned}$$

thanks to Proposition 13.1.

$$\begin{aligned}\mathrm{Disc}_M(w_{1:M}) \geq& \frac{1}{|J|}\sum_{j\in J}\Big(\widehat{P_M^{j,j}(q^j)} - \frac{1}{|J|-1}\sum_{j'\neq j}\widehat{P_M^{j,j'}(q^j)}\Big) - E_{\mathrm{reg}}(M)\\ =& \frac{1}{|J|}\sum_{j\in J}\widehat{P_M^{j,j}(q^j)} - \frac{1}{|J|}\sum_{j\in J}\frac{1}{|J|-1}\sum_{j'\neq j}\widehat{P_M^{j,j'}(q^j)} - E_{\mathrm{reg}}(M)\end{aligned}$$

Let us exchange the sums in the second term.

$$\mathrm{Disc}_M(w_{1:M}) \geq \frac{1}{|J|}\sum_{j\in J}\widehat{P_M^{j,j}(q^j)} - \frac{1}{|J|}\sum_{j'\in J}\frac{1}{|J|-1}\sum_{j\neq j'}\widehat{P_M^{j,j'}(q^j)} - E_{\mathrm{reg}}(M)$$

Let us exchange the name of the indexes j and j' in the second term.

$$\begin{aligned}\mathrm{Disc}_M(w_{1:M}) \geq& \frac{1}{|J|}\sum_{j\in J}\widehat{P_M^{j,j}(q^j)} - \frac{1}{|J|}\sum_{j\in J}\frac{1}{|J|-1}\sum_{j'\neq j}\widehat{P_M^{j',j}(q^{j'})} - E_{\mathrm{reg}}(M)\\ =& \frac{1}{|J|}\sum_{j\in J}\Big(\widehat{P_M^{j,j}(q^j)} - \frac{1}{|J|-1}\sum_{j'\neq j}\widehat{P_M^{j',j}(q^{j'})}\Big) - E_{\mathrm{reg}}(M)\\ =& \mathrm{Disc}_M(q) - E_{\mathrm{reg}}(M)\end{aligned}$$

Now we want to compare $\mathrm{Disc}_M(q)$ and $\mathrm{Disc}_{\mathrm{safe}}(q)$. We need the following result.

Proposition 13.2. *Let $\alpha > 0$. Suppose Assumption 3.2 holds. Then*

$$\mathbb{P}\Big(\forall j\in J, \forall i\in I, \Big|\frac{1}{M^j}\sum_{m,o(m)\in j}(\widehat{p^i_m} - p^i_m)\Big| \leq \sqrt{\ln\Big(\frac{2|I||J|}{\alpha}\Big)\frac{1}{2\xi NM}}\Big) \geq 1-\alpha.$$

Proof. Let $j\in J$. The variables $(X^i_{m,t})_{1\leq t\leq N, m \text{ s.t. } o(m)\in j}$ are independent bounded by 1, of mean p^i_m. Hence, according to Hoeffding's inequality, for all $\beta > 0$

$$\mathbb{P}\Big(\Big|\frac{1}{NM^j}\sum_{m,o(m)\in j}\sum_{t=1}^{N}(X^i_{m,t} - p^i_m)\Big| \geq \beta\Big) \leq 2e^{-2\beta^2 NM^j}.$$

According to Assumption 3.2, $M^j \geq \xi M$ so

$$\mathbb{P}\Big(\Big|\frac{1}{NM^j}\sum_{m,o(m)\in j}\sum_{t=1}^{N}(X^i_{m,t} - p^i_m)\Big| \geq \beta\Big) \leq 2e^{-2\beta^2 NM\xi}.$$

Let D^c be the event $\{\exists j \in J, \exists i \in I, |\frac{1}{NM^j}\sum_{m,o(m)\in j}\sum_{t=1}^{N}(X^i_{m,t} - p^i_m)| \geq \beta\}$. Then

$$\begin{aligned}\mathbb{P}(D^c) &\leq \sum_{j\in J, i\in I}\mathbb{P}\Big(\frac{1}{NM^j}\sum_{m,o(m)\in j}\sum_{t=1}^{N}(X^i_{m,t} - p^i_m) \geq \beta\Big)\\ &\leq 2|I||J|e^{-2\beta^2 NM\xi}.\end{aligned}$$

Let us choose β such that $2|I||J|e^{-2\beta^2 NM\xi} = \alpha$, *i.e.*, $\beta = \sqrt{\ln\Big(\frac{2|I||J|}{\alpha}\Big)\frac{1}{2NM\xi}}$.

On D, for all $j \in J, i \in I$

$$\Big|\frac{1}{M^j}\sum_{m,o(m)\in j}(\widehat{p^i_m} - p^i_m)\Big| \leq \sqrt{\ln\Big(\frac{2|I||J|}{\alpha}\Big)\frac{1}{2NM\xi}}$$

so we can conclude. □

Let $\alpha > 0$. Let us work on the event D defined in the proof of Proposition 13.2. Let

$$p^j_m(q^j) := \sum_{i\in I^j} q^{i\to j}p^i_m$$

the spiking probability of neuron j with weights q^j when presented with the m^{th} object,

$$P^{j',j}_M(q^{j'}) := \frac{1}{M^j}\sum_{m,\ o(m)\in j} p^{j'}_m(q^{j'}_m)$$

the average spiking probability of neuron j' when presented with objects in class j. For all $j, j' \in J$,

$$\begin{aligned}|\widehat{P^{j',j}_M(q^{j'})} - P^{j',j}_M(q^{j'})| &\leq \sum_{i\in I^{j'}} q^{i\to j'}|\frac{1}{M^j}\sum_{m,o(m)\in j}(\widehat{p^i_m} - p^i_m)|\\ &\leq \sqrt{\ln\Big(\frac{2|I||J|}{\alpha}\Big)\frac{1}{2\xi NM}}\end{aligned}$$

Thanks to Proposition 13.2. Let

$$\overline{\mathrm{Disc}}^j_M(q^j) = P^{j,j}_M(q^j) - \frac{1}{|J|-1}\sum_{j'\neq j}P^{j',j}_M(q^{j'})$$

and

$$\overline{\mathrm{Disc}}_M(q) = \frac{1}{|J|}\sum_{j\in J}\overline{\mathrm{Disc}}^j_M(q^j).$$

For all $j \in J$ we have

$$|\overline{\mathrm{Disc}}^j_M(q^j) - \mathrm{Disc}^j_M(q^j)| \leq 2\sqrt{\ln\Big(\frac{2|I||J|}{\alpha}\Big)\frac{1}{2\xi NM}}$$

so

$$|\overline{\mathrm{Disc}}_M(q) - \mathrm{Disc}_M(q)| \leq 2\sqrt{\ln\Big(\frac{2|I||J|}{\alpha}\Big)\frac{1}{2\xi NM}}.$$

Besides, since q is a feasible weight family,

$$\overline{\mathrm{Disc}}_M(q) \geq \mathrm{Disc}_{\mathrm{safe}}(q).$$

Finally,

$$\mathrm{Disc}_M(w_{1:M}) \geq \mathrm{Disc}_{\mathrm{safe}}(q) - E_{\mathrm{reg}}(M) - E(N, M, \alpha).$$

This is true for any feasible weight family q and $\mathbb{P}(D) \geq 1 - \alpha$ so we can conclude.

13.2 Proof of Theorem 3.4

Let

- $\overline{g}_m^{i^+\to j} := \begin{cases} p_i^m \times \frac{M}{M^j} & \text{if } o(m) \in j \\ -p_i^m \times \frac{1}{|J|-1} \times \frac{M}{M^{j'}} & \text{if } o(m) \in j' \neq j \end{cases}$
- $\overline{g}_m^{i^-\to j} = -\overline{g}_m^{i^+\to j}$
- $\overline{G}_m^{i^\bullet\to j} = \sum_{m'=1}^{m} \overline{g}_{m'}^{i^\bullet\to j}$
- $\overline{w}_{m+1}^{i^\bullet\to j} = \frac{\exp(\eta^j \overline{C}_{i^\bullet\to j}^m)}{\sum_{k\in I_j} \exp(\eta^j \overline{C}_{k\to j}^m)}$.

We need the following proposition:

Proposition 13.3. *Suppose each nature of object is presented the same amount of times. Let $\alpha > 0$. Then with EWA with $\eta^j = \frac{1}{|\mathcal{O}|}\sqrt{2\frac{\ln(|I^j|)}{M}}$, we get*

$$\mathbb{P}\Big(\forall j \in J, i^\bullet \in I^j, |w_{M+1}^{i^\bullet\to j} - \overline{w}_{M+1}^{i^\bullet\to j}| \leq |I^j|\sqrt{\ln\Big(\frac{2|I||J|}{\alpha}\Big)\frac{\ln(|I^j|)}{|\mathcal{O}|N}}\Big) \geq 1-\alpha$$

Proof. Let $j \in J$, $i^\bullet \in I^j$, $h^l : \mathbb{R}^{|I^j|} \mapsto \mathbb{R}$ such that $h^l(x_1,\dots,x_{|I^j|}) = \frac{\exp(\eta^j x_l)}{\sum_{k=1}^{|I^j|}\exp(\eta^j x_k)}$. Then for all $(x_1,\dots,x_{|I^j|})$,

$$||\nabla h^l(x_1,\dots,x_{|I^j|})|| \leq \eta^j\sqrt{|I^j|}.$$

Besides, according to the mean value theorem, for $i^\bullet \in I^j$

$$\begin{aligned} |w_{i^\bullet\to j}^{M+1} - \overline{w}_{i^\bullet\to j}^{M+1}| &= |h^l((G_{k\to j}^M)_{k\in I^j}) - h^l((\overline{G}_{k\to j}^M)_{k\in I^j})| \\ &\leq \eta^j\sqrt{|I^j|}||(G_{k\to j}^M)_{k\in I^j} - (\overline{G}_{k\to j}^M)_{k\in I^j}||. \end{aligned}$$

Besides, each nature of object is presented the same amount of times so Assumption 3.2 holds with $\xi = \frac{1}{|\mathcal{O}|}$. Hence according to Proposition 13.2, with probability $1-\alpha$ we have that for all $j \in J$, $k \in I^j$,

$$\begin{aligned} |G_{k\to j}^M - \overline{G}_{k\to j}^M| &\leq M\sqrt{\ln\Big(\frac{2|I||J|}{\alpha}\Big)\frac{1}{2\xi NM}} \\ &= \sqrt{\ln\Big(\frac{2|I||J|}{\alpha}\Big)\frac{M|\mathcal{O}|}{2N}} \end{aligned}$$

i.e.,

$$|w_{M+1}^{i^\bullet\to j} - \overline{w}_{M+1}^{i^\bullet\to j}| \leq \eta^j|I^j|\sqrt{\ln\Big(\frac{2|I||J|}{\alpha}\Big)\frac{M|\mathcal{O}|}{2N}}.$$

Then we find the result by replacing η^j by its value $\frac{1}{|\mathcal{O}|}\sqrt{2\frac{\ln(|I^j|)}{M}}$.

□

Proposition 13.4. *Suppose each nature of object is presented the same amount of times: for all $o \in \mathcal{O}$, $|\{m, o(m) = o\}| = \frac{M}{|\mathcal{O}|}$. Then for all $j \in J$:*

- *if $\tilde{I}^j \neq I^j$, then*

$$|\overline{w}_{M+1}^{i^\bullet\to j} - w_\infty^{i^\bullet\to j}| \leq E_{EWA}^j(M)$$

where

$$E^j_{EWA}(M) = \max\Big\{1, \frac{|I^j|}{|\tilde{I}^j|} - 1\Big\}\frac{1}{|\tilde{I}^j|}e^{-\frac{\gamma^j}{|\mathcal{O}|}\sqrt{2\ln(|I^j|)M}}$$

- *if* $\tilde{I}^j = I^j$*, then*

$$\overline{w}^{i^\bullet\to j}_{M+1} = |I^j|^{-1}.$$

Proof. First, let's prove that for all $m \geq 1$, for all $j \in J$ and $i \in I^j$

$$\overline{G}^{i^\bullet\to j}_M = d^{i^\bullet\to j}M. \tag{7}$$

Indeed,

$$\begin{aligned}
\overline{G}^{i^+\to j}_M &= \sum_{m=1}^{M} \overline{g}^{i^+\to j}_m \\
&= \sum_{m,\ o(m)\in j} \frac{M}{M^j} \times p^i_m - \sum_{j'\neq j}\ \sum_{m,\ o(m)\in j'} \frac{1}{|J|-1} \times \frac{M}{M^{j'}} \times p^i_m \\
&= M\Big(\frac{1}{M^j}\sum_{m,\ o(m)\in j} p^i_m - \frac{1}{|J|-1}\sum_{j'\neq j}\frac{1}{M^{j'}}\sum_{m,\ o(m)\in j'} p^i_m\Big) \\
&= M\Big(\frac{1}{M^j}\sum_{o\in j}\ \sum_{m,\ o(m)=o} p^i_o - \frac{1}{|J|-1}\sum_{j'\neq j}\frac{1}{M^{j'}}\sum_{o\in j'}\ \sum_{m,\ o(m)=o} p^i_o\Big)
\end{aligned}$$

Besides, each kind of object is presented the same amount of times, so for all $j \in J$,

$$M^j = n^j \times \frac{M}{|O|}$$

so we have

$$\begin{aligned}
\overline{G}^{i^+\to j}_M &= M\Big(\frac{1}{n_j}\sum_{o\in j} p^o_i - \frac{1}{|J|-1}\sum_{j'\neq j}\frac{1}{n_{j'}}\sum_{o\in j'} p^o_i\Big) \\
&= d^{i^+\to j}M.
\end{aligned}$$

Similarly we have

$$\overline{G}^{i^-\to j}_M = d^{i^-\to j}M.$$

Let $d^j_{\max} := \max_{i^\bullet\in I^j} d^{i^\bullet\to j}$.

Case $\tilde{I}^j = I^j$. Then for all $i^\bullet \in I_j$,

$$\overline{w}^{i^\bullet\to j}_{M+1} = \frac{\exp(\eta^j d^j_{\max}M)}{\sum_{k\in I^j}\exp(\eta^j d^j_{\max}M)} = \frac{1}{|I^j|}.$$

Case $\tilde{I}^j \neq I^j$. Then

$$\overline{w}^{i^\bullet\to j}_{M+1} = \frac{e^{\eta^j M d^{i^\bullet\to j}}}{\sum_{k\in I^j} e^{\eta^j M d^{k\to j}}} \leq \frac{e^{\eta^j M d^{i^\bullet\to j}}}{|\tilde{I}^j|e^{\eta^j M d^j_{\max}}} = \frac{1}{|\tilde{I}^j|}e^{-\eta^j M(d^j_{\max} - d^{i^\bullet\to j})}.$$

Thus

$$0 \leq \overline{w}^{i^\bullet\to j}_{M+1} \leq \frac{1}{|\tilde{I}^j|}e^{-\eta^j M(d^j_{\max} - d^{i^\bullet\to j})}. \tag{8}$$

Let $i^\bullet \in \tilde{I}^j$, $d^j_{\max \text{ bis}} := \max_{k \in I^j \setminus \tilde{I}^j} d^{k\to j}$.

$$\begin{aligned}
\overline{w}^{i^\bullet\to j}_{M+1} &= \frac{e^{\eta^j M d^{i^\bullet\to j}}}{\sum_{k\in I_j} e^{\eta^j M d^{k\to j}}} \\
&\geq \frac{e^{\eta^j M d^j_{\max}}}{|\tilde{I}^j| e^{\eta^j M d^j_{\max}} + (|I^j| - |\tilde{I}^j|) e^{\eta^j M d^j_{\max \text{ bis}}}} \\
&= \frac{1}{|\tilde{I}^j|} \frac{1}{1 + \frac{|I^j|-|\tilde{I}^j|}{|\tilde{I}^j|} e^{-\eta^j\gamma^j M}} \\
&\geq \frac{1}{|\tilde{I}^j|}\Big(1 - \frac{|I^j| - |\tilde{I}^j|}{|\tilde{I}^j|} e^{-\eta^j\gamma^j M}\Big) \\
&= \frac{1}{|\tilde{I}^j|} - \frac{|I^j| - |\tilde{I}^j|}{|\tilde{I}^j|^2} e^{-\eta^j\gamma^j M},
\end{aligned}$$

and thanks to (8),

$$\overline{w}^{i^\bullet\to j}_{M+1} \leq \frac{1}{|\tilde{I}^j|}.$$

Thus

$$\frac{1}{|\tilde{I}^j|} - \frac{|I^j| - |\tilde{I}^j|}{|\tilde{I}^j|^2} e^{-\eta^j\gamma^j M} \leq \overline{w}^{i^\bullet\to j}_{M+1} \leq \frac{1}{|\tilde{I}^j|}.$$

Let $i \in I^j \setminus \tilde{I}^j$. Then (8) tells us that

$$0 \leq \overline{w}^{i^\bullet\to j}_{M+1} \leq \frac{1}{|\tilde{I}^j|} e^{-\eta^j M\gamma^j}.$$

In particular, with $\eta^j = \frac{1}{|\mathcal{O}|}\sqrt{2\frac{\ln(|I^j|)}{M}}$, for all $i^\bullet \in I_j$

$$|\overline{w}^{i^\bullet\to j}_{M+1} - w^{i^\bullet\to j}_\infty| \leq E^j_{\text{EWA}}(M).$$

□

We get the result by combining Proposition 13.3 and Proposition 13.4.

13.2.1 Proof of Corollary 3.5

We have

$$\begin{aligned}
p^{j,\text{cond}}_{M+1,t}(w^j_{M+1}) - p^{j',\text{cond}}_{M+1,t}(w^{j'}_{M+1}) =& p^{j,\text{cond}}_{M+1,t}(w^j_{M+1}) - p^{j,\text{cond}}_{M+1,t}(w^j_\infty) \\
&+ p^{j,\text{cond}}_{M+1,t}(w^j_\infty) - p^{j',\text{cond}}_{M+1,t}(w^{j'}_\infty) \\
&+ p^{j',\text{cond}}_{M+1,t}(w^{j'}_\infty) - p^{j',\text{cond}}_{M+1,t}(w^{j'}_{M+1}).
\end{aligned}$$

Let $h \in J$.

$$
\begin{aligned}
&|p^{h,\mathrm{cond}}_{M+1,t}(w^h_{M+1}) - p^{h,\mathrm{cond}}_{M+1,t}(w^h_\infty)| \\
&\qquad = \Big|\varphi\Big(\alpha^h + \sum_{i^+\in I^h_+} w^{i^+\to h}_{M+1}\sum_{k=1}^K g_+(k)X^i_{M+1,t-k} - \sum_{i^-\in I^h_-} w^{i^-\to h}_{M+1}\sum_{k=1}^K g_-(k)X^i_{M+1,t-k}\Big) \\
&\qquad\qquad - \varphi\Big(\alpha^h + \sum_{i^+\in I^h_+} w^{i^+\to h}_{\infty}\sum_{k=1}^K g_+(k)X^i_{M+1,t-k} - \sum_{i^-\in I^h_-} w^{i^-\to h}_{\infty}\sum_{k=1}^K g_-(k)X^i_{M+1,t-k}\Big)\Big| \\
&\qquad \le L\Big|\sum_{i^+\in I^h_+} (w^{i^+\to h}_{M+1} - w^{i^+\to h}_\infty)\sum_{k=1}^K g_+(k)X^i_{M+1,t-k} - \sum_{i^-\in I^h_-} (w^{i^-\to h}_{M+1} - w^{i^-\to h}_\infty)\sum_{k=1}^K g_-(k)X^i_{M+1,t-k}\Big| \\
&\qquad \le L\sum_{i^\bullet\in I^h} |w^{i^\bullet\to h}_{M+1} - w^{i^\bullet\to h}_\infty|
\end{aligned}
$$

because φ is L-Lipschitz and the variables $X^i_{M+1,t-k}$ are bounded by 1. Hence according to Theorem 3.4, with probability $1-\alpha$, for all $h \in J$, $t \in \{1,\dots,N\}$ we have

$$|p^{h,\mathrm{cond}}_{M+1,t}(w^h_{M+1}) - p^{h,\mathrm{cond}}_{M+1,t}(w^h_\infty)| \le L|I^h|(E^h(N,\alpha) + E^h_{\mathrm{EWA}}(M)).$$

Hence with probability $1-\alpha$, for all $j \in J$, $j' \neq j$, $t \in \{1,\dots,N\}$, $o(M+1) \in j$ we have

$$p^{j,\mathrm{cond}}_{M+1,t}(w^j_{M+1}) - p^{j',\mathrm{cond}}_{M+1,t}(w^{j'}_{M+1}) \ge p^{j,\mathrm{cond}}_{M+1,t}(w^j_\infty) - p^{j',\mathrm{cond}}_{M+1,t}(w^{j'}_\infty) - L\sum_{h\in\{j,j'\}} |I^h|(E^h(N,\alpha) + E^h_{\mathrm{EWA}}(M)).$$

13.3 Proofs of Proposition 8.1 and Proposition 8.2

13.3.1 Proof of Proposition 8.1

Here the assumptions of Theorem 3.4 are verified so Theorem 3.4 applies. Let us study the limit family (w^A_∞, w^B_∞). Let us compute the feature discrepancies. We have $|A| = n^c - 1$ and $|B| = 1$. Here, $|J| = 2$ so for all $k \in \{1,\dots,c\}$, $l \in \{1,\dots,n\}$, $d^{f^+_{k,l}\to A} = -d^{f^-_{k,l}\to A} = -d^{f^+_{k,l}\to B} = d^{f^-_{k,l}\to B}$.

Let $k \in \{1,\dots,c\}$ and $l \in \{2,\dots,n\}$. There are $n^{c-1}-1$ objects in A and 1 in B with the feature $f_{k,1}$, n^{c-1} in A and 0 in B with the feature $f_{k,l}$. Hence,

$$
\begin{aligned}
d^{f^+_{k,1}\to A} &= \frac{n^{c-1}-1}{n^c-1}p - p \\
d^{f^+_{k,l}\to A} &= \frac{n^{c-1}}{n^c-1}p \\
d^{f^-_{k,1}\to A} &= p - \frac{n^{c-1}-1}{n^c-1}p \\
d^{f^-_{k,l}\to A} &= -\frac{n^{c-1}}{n^c-1}p \\
d^{f^+_{k,1}\to B} &= p - \frac{n^{c-1}-1}{n^c-1}p \\
d^{f^+_{k,l}\to B} &= -\frac{n^{c-1}}{n^c-1}p \\
d^{f^-_{k,1}\to B} &= \frac{n^{c-1}-1}{n^c-1}p - p \\
d^{f^-_{k,l}\to B} &= \frac{n^{c-1}}{n^c-1}p.
\end{aligned}
$$

It is clear that $d^{f^+_{k,1}\to A} < d^{f^+_{k,l}\to A}$ and $d^{f^-_{k,l}\to A} < d^{f^-_{k,1}\to A}$. Besides,

$$d^{f^+_{k,l}\to A} < d^{f^-_{k,1}\to A} \iff n > 2.$$

Hence under the condition $n > 2$, $\tilde{I}^A = \{f^-_{k,1}, 1 \le k \le c\}$ and similarly $\tilde{I}^B = \{f^+_{k,1}, 1 \le k \le c\}$. So according to Theorem 3.4, w^A_∞ is the family such that $w^{f^-_{k,1}\to A} = 1/c$ for $k \in \{1, \dots, c\}$ and $w^{i^\bullet \to A} = 0$ for other connections, whereas w^B_∞ is the family such that $w^{f^+_{k,1}\to B} = 1/c$ for $k \in \{1, \dots, c\}$ and $w^{i^\bullet \to B} = 0$ for other connections.

Let us look at the conditions under which this is a feasible weight family. When presented with an object having h features $f_{k_1,1}$,..., $f_{k_h,1}$ in common with object o_B,

$$p^{A,\mathrm{cond}}_{m,t}(w^A_\infty) = \Big(\alpha^A - \frac{1}{c}\sum_{l=1}^{h} X^{f_{k_l,1}}_{m,t-1}\Big)_+$$

so under the condition $\alpha^A < 1/c$,

$$\begin{aligned}\mathbb{E}[p^{A,\mathrm{cond}}_{m,t}(w^A_\infty)] &= \alpha^A \mathbb{P}(X^{f_{k_1,1}}_{m,t} = 0, \dots, X^{f_{k_h,1}}_{m,t} = 0)\\ &= \alpha^A (1-p)^h.\end{aligned}$$

Besides,

$$p^{B,\mathrm{cond}}_{m,t}(w^B_\infty) = \frac{1}{c}\sum_{l=1}^{h} X^{f_{k_l,1}}_{m,t-1}$$

so

$$\mathbb{E}[p^{B,\mathrm{cond}}_{m,t}(w^B_\infty)] = \frac{h}{c}p.$$

Hence (w^A_∞, w^B_∞) is a feasible weight family if and only if we have the two following conditions:

$$\forall 1 \le h \le c-1, \quad \alpha^A(1-p)^h > \frac{h}{c}p \tag{9}$$

$$\alpha^A(1-p)^c < p \tag{10}$$

where (9) says that w_∞ correctly classifies objects in A and (10) says that that w_∞ correctly classifies the object in B.

Besides, (9) is equivalent to

$$\alpha^A(1-p)^{c-1} > \frac{c-1}{c}p$$

so w_∞ is a feasible weight family if and only if $\alpha^A \in \left(\frac{(c-1)p}{c(1-p)^{c-1}}, \frac{p}{(1-p)^c}\right)$ and $\alpha^A < 1/c$. If so, the safety discrepancy is

$$\mathrm{Disc}_{\mathrm{safe}}(w_\infty) = \min\{\alpha^A(1-p)^{c-1} - \frac{c-1}{c}p, p - \alpha^A(1-p)^c\}.$$

Besides,

$$\frac{(c-1)p}{c(1-p)^{c-1}} < \frac{p}{(1-p)^c} \iff \frac{c-1}{c}(1-p) < 1$$

which is always true so the interval is non-empty, and $\frac{1}{c} > \frac{(c-1)p}{c(1-p)^{c-1}}$ if and only if $(c-1)p < (1-p)^{c-1}$.

To conclude, if $n > 2$ and $(c-1)p < (1-p)^{c-1}$ then there exists α^A such that w_∞ is a feasible weight family.

13.3.2 Proof of Proposition 8.2

Let us compute the feature discrepancies in this new framework.

Let $k \in \{1, \dots, c\}$ and $l \in \{2, \dots, n\}$. There are $n^c - n^{c-1}$ objects in A and 0 in B without the feature $f_{k,1}$ and $n^c - n^{c-1} - 1$ in A and 1 in B without the feature $f_{k,l}$. Hence, similarly as in 13.3.1,

$$d^{f_{k,1}\to A} = \frac{n^{c-1}-1}{n^c - 1}p - p$$

$$
\begin{aligned}
d^{f_{k,l}\to A} &= \frac{n^{c-1}}{n^c-1}p \\
d^{\tilde{f}_{k,1}\to A} &= \frac{n^c-n^{c-1}}{n^c-1}q \\
d^{\tilde{f}_{k,l}\to A} &= \frac{n^c-n^{c-1}-1}{n^c-1}q - q \\
d^{f_{k,1}\to B} &= p - \frac{n^{c-1}-1}{n^c-1}p \\
d^{f_{k,l}\to B} &= -\frac{n^{c-1}}{n^c-1}p \\
d^{\tilde{f}_{k,1}\to B} &= -\frac{n^c-n^{c-1}}{n^c-1}q \\
d^{\tilde{f}_{k,l}\to B} &= q - \frac{n^c-n^{c-1}-1}{n^c-1}q.
\end{aligned}
$$

It is clear that $d^{f_{k,1}\to A} < d^{f_{k,l}\to A}$ and $d^{\tilde{f}_{k,l}\to A} < d^{\tilde{f}_{k,1}\to A}$. Besides,

$$d^{f_{k,l}\to A} < d^{\tilde{f}_{k,1}\to A} \iff \frac{p}{n-1} < q$$

and

$$d^{\tilde{f}_{k,l}\to B} < d^{f_{k,1}\to B} \iff q < (n-1)p.$$

Thus, under the hypothesis

$$\frac{p}{n-1} < q < (n-1)p,$$

the neurons having maximal feature discrepancies are neurons $\tilde{f}_{k,1}$ for A, and $f_{k,1}$ for B. So according to Theorem 3.4, w^A_∞ is the family such that $w^{\tilde{f}_{k,1}\to A} = 1/c$ for $k \in \{1,\dots,c\}$ and $w^{i\to A} = 0$ for other connections, whereas w^B_∞ is the family such that $w^{f_{k,1}\to B} = 1/c$ for $k \in \{1,\dots,c\}$ and $w^{i\to B} = 0$ for other connections. With constant weights (w^A_∞, w^B_∞), neurons A and B have the following spiking probabilities.

	$o \in A$ with a l common features with o_B	o_B
A	$q(c-l)/c$	0
B	pl/c	p

Hence with $q > (c-1)p$, the pair $w_\infty = (w^A_\infty, w^B_\infty)$ is indeed a feasible weight family.

Besides,

$$\mathrm{Disc}_{\mathrm{safe}}(w_\infty) = \min\Big\{ \min_{l\in\{0,c-1\}} \Big\{ q\frac{c-l}{c} - p\frac{l}{c} \Big\}, p \Big\}$$

The second minimum is achieved for $l = c-1$, so

$$\mathrm{Disc}_{\mathrm{safe}}(w_\infty) = \min\Big\{ \frac{q-p(c-1)}{c}, p \Big\}.$$

Table 2: Notations

NOTATION	DESCRIPTION
$\mathcal{O}$	set of objects
M	total number of objects presented to the network
N	number of time steps during which one object is presented
m	index of the current round
o	object in $\mathcal{O}$
$o(m)$	nature of the object of the m^{th} round
J	set of classes and of output neurons
I	set of input neurons
j	index of an output neuron and of a class
M^j	number of rounds with objects belonging to class j
i	index of an input neuron
I_+^j	set of excitatory input neurons of output neuron j
I_-^j	set of inhibitory input neurons of output neuron j
$i^\bullet$	connection $i^\bullet \in \{i^+, i^-\}$
$I^{+/-}$	set of signed input neurons, *i.e.,* all possible connections to output neurons
I^j	set of connections to output neuron j
φ	activation function of the Hawkes process
K	size of the support of functions g_+ and g_-
α^j	spontaneous activity of output neuron j
$X_{m,t}^i$	activity of neuron i during time step t of round m.
p_m^i	spiking probability of input neuron i during round m
$\mathcal{Q}$	set of feasible weight families
p_o^i	spiking probability of input neuron i when presented with object o
$p_{m,t}^{j,\text{cond}}$	conditional spiking probability of output neuron j knowing the past
$w_m^{i^\bullet \to j}$	synaptic weight of connection $i^\bullet$ of neuron j during round m given by HAN
w_m^j	weight family of neuron j during round m given by HAN: $(w_m^{i \to j})_{i \in I^j}$
$q^{i^\bullet \to j}$	constant synaptic weight of connection $i^\bullet$ of j
q^j	constant weight family of neuron j: $(q^{i^\bullet \to j})_{i^\bullet \in I_j}$
$g_m^{i^\bullet \to j}$	gain of connection $i^\bullet$ w.r.t. output neuron j for round m
$G_m^{i^\bullet \to j}$	cumulated gain of connection $i^\bullet$ of output neuron j until round m
G_j^m	cumulated gain of output neuron j until round m: $\sum_{m'=1}^m \sum_{i^\bullet \in I^j} w_m^{i^\bullet \to j} g_{m'}^{i^\bullet \to j}$
η^j	learning rate of EWA
β^j	parameter of PWA
A, B	classes of the specific case
c	number of characteristics in the specific case
n	number of features for each characteristic in the specific case
o^B	unique object belonging to class B
$f_{k,l}$	feature of the specific case
$f_{k,l}, \tilde{f}_{k,l}$	neurons of the specific case
p	spiking probability of $f_{k,l}$ neurons when active
q	spiking probability of $\tilde{f}_{k,l}$ neurons when active