An interacting neuronal network with inhibition: theoretical analysis and perfect simulation

Branda Goncalves

Workshop : New trends on Hawkes processes

June, 2022

CERGY PARIS UNIVERSITÉ

LPTM
Laboratoire de Physique Théorique et Modélisation

Introduction

What is a neuron?

A neuron is an excitable cell that propagates information as an electrical signal.

Introduction

What is a neuron?

A neuron is an excitable cell that propagates information as an electrical signal.

How does a neural network ?

The functioning of a neural network is based on the fact that neurons send discharges to each other, more precisely, presynaptic neurons send discharges to their postsynaptic partners. These discharges are called action potentials or spikes.

Introduction

What is a neuron?

A neuron is an excitable cell that propagates information as an electrical signal.

How does a neural network ?

The functioning of a neural network is based on the fact that neurons send discharges to each other, more precisely, presynaptic neurons send discharges to their postsynaptic partners. These discharges are called action potentials or spikes.

When a neuron spikes, this spike changes the rate at which other neurons spike. In other words, neurons excite and/or inhibit each other.

Description of the model

For all $1 \leq i \leq N$, $X_t^{i,N}$ describes the state of inhibition of neuron i at time t. Let us consider the case where we have N neurons that are related to each other.

Description of the model

For all $1 \leq i \leq N$, $X_t^{i,N}$ describes the state of inhibition of neuron i at time t. Let us consider the case where we have N neurons that are related to each other.

When the neuron i spikes,

- The current state of inhibition of neuron i is replaced by a new value Y^i independently of anything else with distribution F^i.

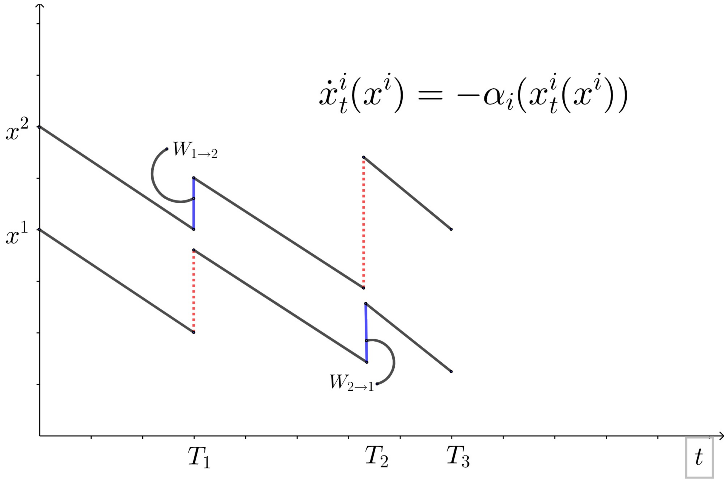
Description of the model

For all $1 \leq i \leq N$, $X_t^{i,N}$ describes the state of inhibition of neuron i at time t. Let us consider the case where we have N neurons that are related to each other.

When the neuron i spikes,

- The current state of inhibition of neuron i is replaced by a new value Y^i independently of anything else with distribution F^i.
- The state of inhibition of any neuron $j \neq i$ is increased by a positive value $W_{i \to j}$ at time t.

Description of the model



Description of the model

We consider the piecewise deterministic Markov process (PDMP) $X_t^N = (X_t^{1,N}, \cdots, X_t^{N,N}) \in \mathbb{R}_+^N$. For $i \in \{1, \cdots, N\}$, the dynamic of $X_t^{i,N}$ is given by:

$$dX_t^{i,N} = -\alpha_i \left(X_{t-}^{i,N}\right) dt + \int_0^\infty \int_0^\infty (y^i - X_{t-}^{i,N}) \mathbf{1}_{\left\{r \leq \beta_i\left(X_{t-}^{i,N}\right)\right\}} M^i\left(dt, dr, dy^i\right) + \sum_{j \neq i} W_{j \to i} \int_0^\infty \int_0^\infty \mathbf{1}_{\left\{r \leq \beta_j\left(X_{t-}^{j,N}\right)\right\}} M^j\left(dt, dr, dy^j\right), \quad (1)$$

where M^i is a random Poisson measure with intensity $dtdrF^i(dy)$ and for all i, the M^i are all independent.

Infinitesimal generator

The infinitesimal generator associated with this model is given by:

$$G^N V(x) = -\sum_{i=1}^{N} \alpha_i(x^i)\frac{\partial}{\partial x^i}V(x) + \sum_{i=1}^{N} \beta_i(x^i)\int_0^{\infty} F^i(dy^i)[V(x + e^i y^i - e^i x^i + \sum_{j \neq i} e^j W_{i\to j}) - V(x)] \quad (2)$$

where V is a smooth function and e^i is the $i-th$ unit vector.

First jump time

Let S_1 be the first jump time of the first neuron to jump. For all $t > 0$,

$$\mathbb{P}(S_1 > t) = \prod_{i=1}^{N} e^{-\int_0^t \beta_i(x_s^i(x^i))ds}. \tag{3}$$

First jump time

Let S_1 be the first jump time of the first neuron to jump. For all $t > 0$,

$$\mathbb{P}(S_1 > t) = \prod_{i=1}^{N} e^{-\int_0^t \beta_i(x_s^i(x^i))ds}. \tag{3}$$

Moreover, if $t < \min_i t_0(x^i)$ where

$$t_0(x^i) := \int_0^{x^i} \frac{dy}{\alpha_i(y)}$$

is the time for the neuron i hit 0 starting from x^i,

First jump time

We can write

$$\mathbb{P}(S_1 > t) = \prod_{i=1}^{N} e^{-[\Gamma_i(x^i) - \Gamma_i(x_t^i(x^i))]},$$

with $\Gamma_i(x^i) := \int^{x^i} \gamma_i(y)dy$ and $\gamma_i(x^i) = \beta_i(x^i)/\alpha_i(x^i)$.

First jump time

We can write

$$\mathbb{P}(S_1 > t) = \prod_{i=1}^{N} e^{-[\Gamma_i(x^i) - \Gamma_i(x_t^i(x^i))]},$$

with $\Gamma_i(x^i) := \int^{x^i} \gamma_i(y)dy$ and $\gamma_i(x^i) = \beta_i(x^i)/\alpha_i(x^i)$.

Then, $\mathbb{P}(S_1 = \infty) = 0$ if and only if

Assumption 1

$\Gamma_i(0) = -\infty$ *for all* $1 \leq i \leq N$

Description

- We fix a neuron $i \in \mathbb{Z}$. We are interested in finding the position of i at time 0 in the stationary regime.

Description

- We fix a neuron $i \in \mathbb{Z}$. We are interested in finding the position of i at time 0 in the stationary regime.

- We explore the past in order to determine all sets of indices and times which affect the value of neuron i at time 0. The set of all such couples (j, s) will be called the clan of ancestors of neuron i. (see Galves and Löcherbach [2], Galves et al. [3])

Description

- We fix a neuron $i \in \mathbb{Z}$. We are interested in finding the position of i at time 0 in the stationary regime.

- We explore the past in order to determine all sets of indices and times which affect the value of neuron i at time 0. The set of all such couples (j, s) will be called the clan of ancestors of neuron i. (see Galves and Löcherbach [2], Galves et al. [3])

- We consider the case where each neuron i has exactly two neighbours so that the neuron i interacts only with the neurons $i + 1$ and $i - 1$.

- T is the time vector.

Algorithm (backward procedure)

Algorithm (backward procedure)

- We simulate , $\forall\, l \in \mathbb{Z}$, $N_t^{l,s}$ and $N_t^{l,p}$ two Poisson processes with respective intensities β_* and $\beta^* - \beta_*$.

Algorithm (backward procedure)

Algorithm (backward procedure)

- We simulate , $\forall\, l \in \mathbb{Z}$, $N_t^{l,s}$ and $N_t^{l,p}$ two Poisson processes with respective intensities β_* and $\beta^* - \beta_*$.

- The jump times of $N_t^{l,s}$ and $N_t^{l,p}$ are respectively $T_n^{l,s}$ and $T_n^{l,p}$ for the neuron l after n jumps.

Algorithm (backward procedure)

Algorithm (backward procedure)

- We simulate , $\forall\, l \in \mathbb{Z}$, $N_t^{l,s}$ and $N_t^{l,p}$ two Poisson processes with respective intensities β_* and $\beta^* - \beta_*$.

- The jump times of $N_t^{l,s}$ and $N_t^{l,p}$ are respectively $T_n^{l,s}$ and $T_n^{l,p}$ for the neuron l after n jumps.

- The jump times $T_n^{l,s}$ will be considered as times of sure jumps (counted by the process $N_t^{l,s}$) and the jump times $T_n^{l,p}$ will be considered as times of possible jumps (counted by the process $N_t^{l,p}$)

Algorithm (backward procedure)

- Let $i \in \mathbb{Z}$ fix and

$$T_1 = \inf\{\, T_1^{i\pm 1,s},\, T_1^{i\pm 1,p},\;\; T_1^{i,s}\;\; T_1^{i,p}\}.$$

Algorithm (backward procedure)

- Let $i \in \mathbb{Z}$ fix and

$$T_1 = \inf\{ T_1^{i\pm1,s}, T_1^{i\pm1,p},\ T_1^{i,s}\ T_1^{i,p}\}.$$

- If $T_1 = T_1^{i\pm1,s}$, we set $C^i_{T_1} = \{i\}$ and we put $I_1 = i \pm 1$.

Algorithm (backward procedure)

- Let $i \in \mathbb{Z}$ fix and

$$T_1 = \inf\{T_1^{i\pm1,s}, T_1^{i\pm1,p}, \ T_1^{i,s} \ T_1^{i,p}\}.$$

- If $T_1 = T_1^{i\pm1,s}$, we set $C^i_{T_1} = \{i\}$ and we put $I_1 = i \pm 1$.

- If $T_1 = T_1^{i\pm1,p}$, we set $C^i_{T_1} = \{i, i \pm 1\}$. We put $I_1 = i \pm 1$

Algorithm (backward procedure)

- Let $i \in \mathbb{Z}$ fix and

$$T_1 = \inf\{ T_1^{i\pm1,s}, T_1^{i\pm1,p},\ T_1^{i,s}\ T_1^{i,p}\}.$$

- If $T_1 = T_1^{i\pm1,s}$, we set $C^i_{T_1} = \{i\}$ and we put $I_1 = i \pm 1$.

- If $T_1 = T_1^{i\pm1,p}$, we set $C^i_{T_1} = \{i, i \pm 1\}$. We put $I_1 = i \pm 1$

- If $T_1 = T_1^{i,p}$, we set $C^i_{T_1} = \{i\}$. We put $I_1 = i$.

Algorithm (backward procedure)

- Let $i \in \mathbb{Z}$ fix and

$$T_1 = \inf\{T_1^{i\pm1,s}, T_1^{i\pm1,p}, \ T_1^{i,s} \ T_1^{i,p}\}.$$

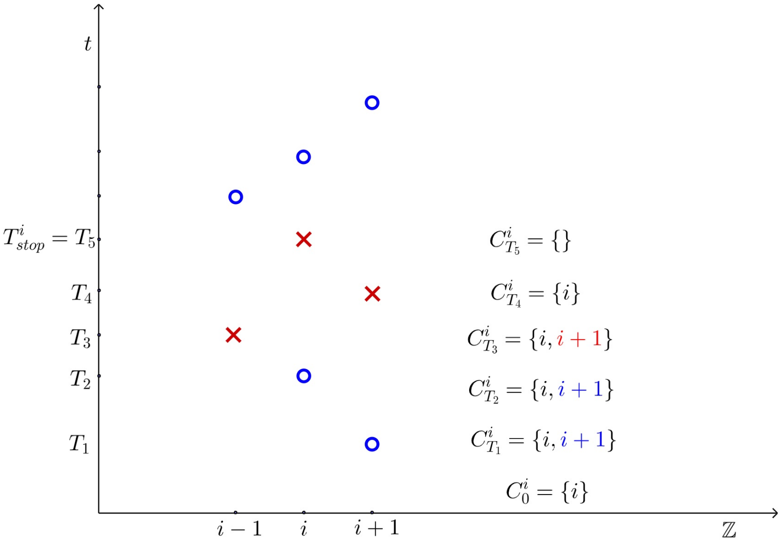
- If $T_1 = T_1^{i\pm1,s}$, we set $C^i_{T_1} = \{i\}$ and we put $I_1 = i \pm 1$.

- If $T_1 = T_1^{i\pm1,p}$, we set $C^i_{T_1} = \{i, i \pm 1\}$. We put $I_1 = i \pm 1$

- If $T_1 = T_1^{i,p}$, we set $C^i_{T_1} = \{i\}$. We put $I_1 = i$.

- If $T_1 = T_1^{i,s}$, we set $C^i_{T_1} = \emptyset$ and we stop the algorithm. We put $I_1 = i$.

Algorithm (backward procedure)



Algorithm (backward procedure)

Suppose T_n is the $n-$th jump time of $C^i_{T_n}$. We have:

$$T_{n+1} = \inf\{\, T^{j,s}_m, T^{j,p}_m > T_n : |j - C^i_{T_n}| \leq 1,\ T^{k,s}_m > T_n,\ k \in C^i_{T_n}, \}.$$

Algorithm (backward procedure)

Suppose T_n is the $n-$th jump time of $C^i_{T_n}$. We have:

$$T_{n+1} = \inf\{\, T^{j,s}_m, T^{j,p}_m > T_n : |j - C^i_{T_n}| \leq 1,\ T^{k,s}_m > T_n,\ k \in C^i_{T_n}, \}.$$

- If $T_{n+1} = T^{j,p}_m$ we set:

$$\begin{cases} \text{If } j \in C^i_{T_n}, & C^i_{T_{n+1}} = C^i_{T_n} \\ \text{If } j \notin C^i_{T_n}, & C^i_{T_{n+1}} = C^i_{T_n} \cup \{j\} \end{cases}$$

Algorithm (backward procedure)

Suppose T_n is the $n-$th jump time of $C^i_{T_n}$. We have:

$$T_{n+1} = \inf\{ T^{j,s}_m, T^{j,p}_m > T_n : |j - C^i_{T_n}| \leq 1, \; T^{k,s}_m > T_n, \; k \in C^i_{T_n}, \}.$$

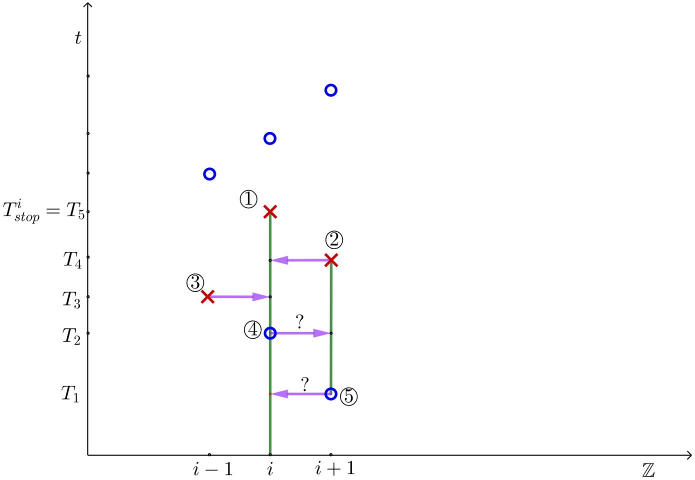
- If $T_{n+1} = T^{j,p}_m$ we set:

$$\begin{cases} \text{If } j \in C^i_{T_n}, & C^i_{T_{n+1}} = C^i_{T_n} \\ \text{If } j \notin C^i_{T_n}, & C^i_{T_{n+1}} = C^i_{T_n} \cup \{j\} \end{cases}$$

- If $T_{n+1} = T^{k,s}_m$ we set:

$$\begin{cases} \text{If } k \in C^i_{T_n}, & C^i_{T_{n+1}} = C^i_{T_n} \setminus \{k\} \\ \text{If } k \notin C^i_{T_n}, & C^i_{T_{n+1}} = C^i_{T_n} \end{cases}$$

Forward procedure



Extinction time T^i_{stop}

The procedure will stop at time T^i_{stop} where $T^i_{stop} = \inf\{t : C^i_t = \emptyset\}$.

Extinction time T^i_{stop}

The procedure will stop at time T^i_{stop} where $T^i_{stop} = \inf\{t : C^i_t = \emptyset\}$.

Does this algorithm always stop at finite time? Or do we need to explore the increasingly distant past?

Extinction time T^i_{stop}

The procedure will stop at time T^i_{stop} where $T^i_{stop} = \inf\{t : C^i_t = \emptyset\}$.

Does this algorithm always stop at finite time? Or do we need to explore the increasingly distant past?

Theorem 1

We set $\delta = \frac{\beta_*}{\beta^* - \beta_*}$. *There exists a critical value* $0 < \delta_c < \infty$ *such that:*

- *if* $\delta > \delta_c$, *then the extinction time is finite almost surely that is,* $\mathbb{P}(\forall i,\ \ T^i_{stop} < \infty) = 1$
- *if* $\delta < \delta_c$, *then the extinction time is infinite with a positive probability that is,* $\mathbb{P}(\forall i,\ \ T^i_{stop} = \infty) > 0$.

Sketch of proof

- We first show that $T^i_{stop} < +\infty$ almost surely for sufficiently large δ.

- We can upper bound $|C^i_t|$ by Z_t almost surely for all $t \geq 0$ where $Z_0 = 1$ and $(Z_t)_{t\geq 0}$ is a branching process.

Sketch of proof

- We first show that $T^i_{stop} < +\infty$ almost surely for sufficiently large δ.

- We can upper bound $|C^i_t|$ by Z_t almost surely for all $t \geq 0$ where $Z_0 = 1$ and $(Z_t)_{t\geq 0}$ is a branching process.

- The associated infinitesimal generator of $(Z_t)_{t\geq 0}$ as follows :

$$Af(n) = n[(\beta^* - \beta_*)(f(n+1) - f(n)) + \beta_*(f(n-1) - f(n))].$$

Sketch of proof

- We first show that $T^i_{stop} < +\infty$ almost surely for sufficiently large δ.

- We can upper bound $|C^i_t|$ by Z_t almost surely for all $t \geq 0$ where $Z_0 = 1$ and $(Z_t)_{t\geq 0}$ is a branching process.

- The associated infinitesimal generator of $(Z_t)_{t\geq 0}$ as follows :

$$Af(n) = n[(\beta^* - \beta_*)(f(n+1) - f(n)) + \beta_*(f(n-1) - f(n))].$$

- If we take, $f(n) = n$, then, for $\delta > 1$, we have

$$Af(n) = -cf(n)$$

where $-c = (\beta^* - \beta_*)(1 - \delta)$.

Sketch of proof

- Assume $x_t = \mathbb{E}(f(Z_t))$ and using the Itô formula, we have:

$$x_t = x_0 + \mathbb{E}\int_0^t Af(Z_s)ds = x_0 - c\int_0^t x_s ds = x_0 e^{-ct}.$$

Sketch of proof

- Assume $x_t = \mathbb{E}(f(Z_t))$ and using the Itô formula, we have:

$$x_t = x_0 + \mathbb{E}\int_0^t Af(Z_s)ds = x_0 - c\int_0^t x_s ds = x_0 e^{-ct}.$$

- When $t \to \infty$, we have $x_t \to 0$.

- If $\delta > 1$, $\mathbb{P}(T_{stop}^i < \infty) \geq \mathbb{P}(\lim_{t\to\infty} Z_t = 0) = 1$ thus ensuring that $\delta_c \leq 1$.

Sketch of proof

We show that for all $\delta < \delta_c,\ T^i_{stop} = +\infty$ with positive probability.

Sketch of proof

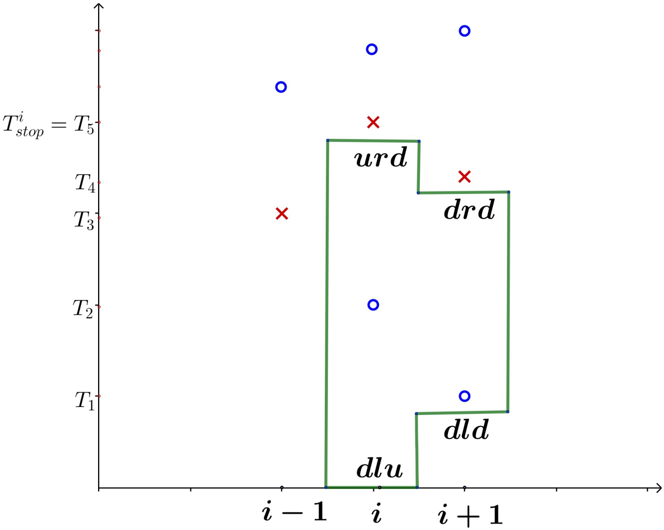
We show that for all $\delta < \delta_c,\ T^i_{stop} = +\infty$ with positive probability.

$T^i_{stop} < \infty$ if and only if $\bar{\mathcal{C}}^i = \cup_{t \geq 0} C^i_t$ is a finite set.

- We show that $\mathbb{P}(T^i_{stop} < \infty) = \mathbb{P}(|\bar{\mathcal{C}}^i| < \infty) < 1$ for sufficiently small values of δ using classical contour techniques. (see Ferrari et al.[1])

- For this, on $|\bar{\mathcal{C}}^i| < \infty,$ we draw the contour of $\bar{\mathcal{C}}^i$ as follow.

Sketch of proof/Contour



Sketch of proof

- Set $N(dlu), N(urd), \cdots$ the number of appearances of the different direction vectors.

$$N(dru) = N(urd) - 1 \leq n/2, \;\; N(drd) + N(dru) + N(uru) + N(urd) = n.$$

Sketch of proof

- Set $N(dlu), N(urd), \cdots$ the number of appearances of the different direction vectors.

$$N(dru) = N(urd) - 1 \leq n/2, \; N(drd) + N(dru) + N(uru) + N(urd) = n.$$

- For $n = 1$, the probability of appearance of a contour of length 4 is equal to $\mathbb{P}(D_1 = urd) \leq \delta$.

Sketch of proof

- For $n = 2$, the probability of appearance of a contour of length 8 is equal to

$$\mathbb{P}(D_1 = ulu,\ D_2 = urd,\ D_3 = drd) + \mathbb{P}(D_1 = ulu,\ D_2 = uru,\ D_3 = urd) + \mathbb{P}(D_1 = uru,\ D_2 = urd,\ D_3 = dld) + \mathbb{P}(D_1 = urd,\ D_2 = drd,\ D_3 = dld) \leq 4\delta^2.$$

Sketch of proof

- For $n = 2$, the probability of appearance of a contour of length 8 is equal to

$$\mathbb{P}(D_1 = ulu,\ D_2 = urd,\ D_3 = drd) + \mathbb{P}(D_1 = ulu,\ D_2 = uru,\ D_3 = urd) + \mathbb{P}(D_1 = uru,\ D_2 = urd,\ D_3 = dld) + \mathbb{P}(D_1 = urd,\ D_2 = drd,\ D_3 = dld) \leq 4\delta^2.$$

- A very approximate upper bound on the total number of possible triplets is given by $4^{2n} = 16^n$. We get for all $\delta < \frac{1}{(16)^2}$,

$$\mathbb{P}(T^i_{stop} < \infty) \leq \delta + 4\delta^2 + \frac{(16\sqrt{\delta})^3}{1 - 16\sqrt{\delta}} =: \phi(\delta) \to 0 \text{ when } \delta \to 0$$

Bibliography

[1] Pablo Augusto Ferrari et al. "Phase transition for infinite systems of spiking neurons". In: *Journal of Statistical Physics* 172.6 (2018), pp. 1564–1575.

[2] Antonio Galves and Eva Löcherbach. "Infinite systems of interacting chains with memory of variable length- a stochastic model for biological neural nets". In: *Journal of Statistical Physics* 151.5 (2013), pp. 896–921.

[3] Antonio Galves, Eva Löcherbach, and Enza Orlandi. "Perfect simulation of infinite range Gibbs measures and coupling with their finite range approximations". In: *Journal of Statistical Physics* 138.1 (2010), pp. 476–495.

[4] Branda Goncalves. "An interacting neuronal network with inhibition: theoretical analysis and perfect simulation". working paper or preprint. Oct. 2021. URL: https://hal.archives-ouvertes.fr/hal-03403351.

Thank you for your attention!

Questions?