Stochastic neural networks: intensity-based model

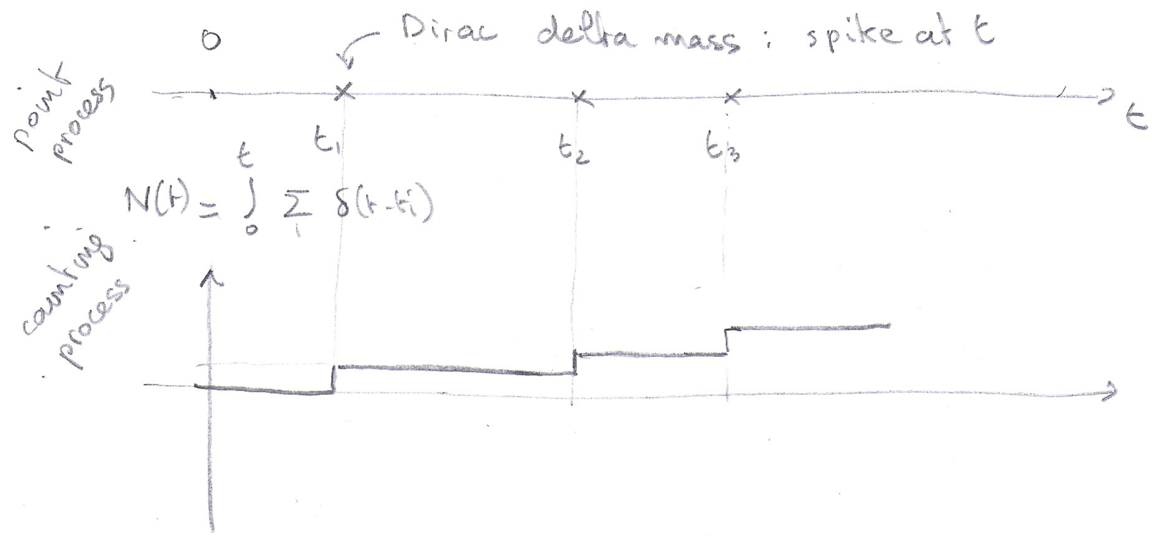
* Neural variability: responses of neurons to the same stimulus or perturbations virtually always exhibit some variability in terms of spike timing or spike counts, at least at the cortical level.

* Neural noise model neural variability
  ↳ In principle neural noise is either intrinsic, due to the fundamental stochasticity of biological systems, or extrinsic due to the influence of uncontroled/unobserved changing parameters (which may not be random). The distinction between extrinsic and intrinsic noise depends on the neural system under consideration.

Ex: For a neuron: intrinsic: stochastic ion channel opening, and closing / discrete noise due to molecular processing and gene expression (synaptic release).

extrinsic: constant background bombardment from other neurons (neural networks are spontaneously active), neuromodulation via chemical agents (dopamine...)
↑
not necessarily noise per se but a potential source of variability

* To First-approximation, network neural activity can be represented in terms of processes which only registers the times at which a neuron spikes:



<u>Experimental observation</u>: cortical neural activity ~ Poissonian variability with weak correlations between neurons. Mostly established by checking that the Fano Factor of spike counts is ~ 1 (not true for all bin size; in short bin only one spike can happen due to refractory period, in large window the Fano Factor integrates unaccounted slow modulation (attention?) and is larger than 1)

$$\text{Fano Factor} = \frac{Var(N)}{\mathbb{E}[N]} \leftarrow 1 \text{ for Poisson variable}$$

Probabilistic setting

parameters : receptive field / network structure

$$P(y \mid s, k) \quad \leftarrow \text{probabilistic model}$$

input : spike trains / stimulus

output : spike trains / decision

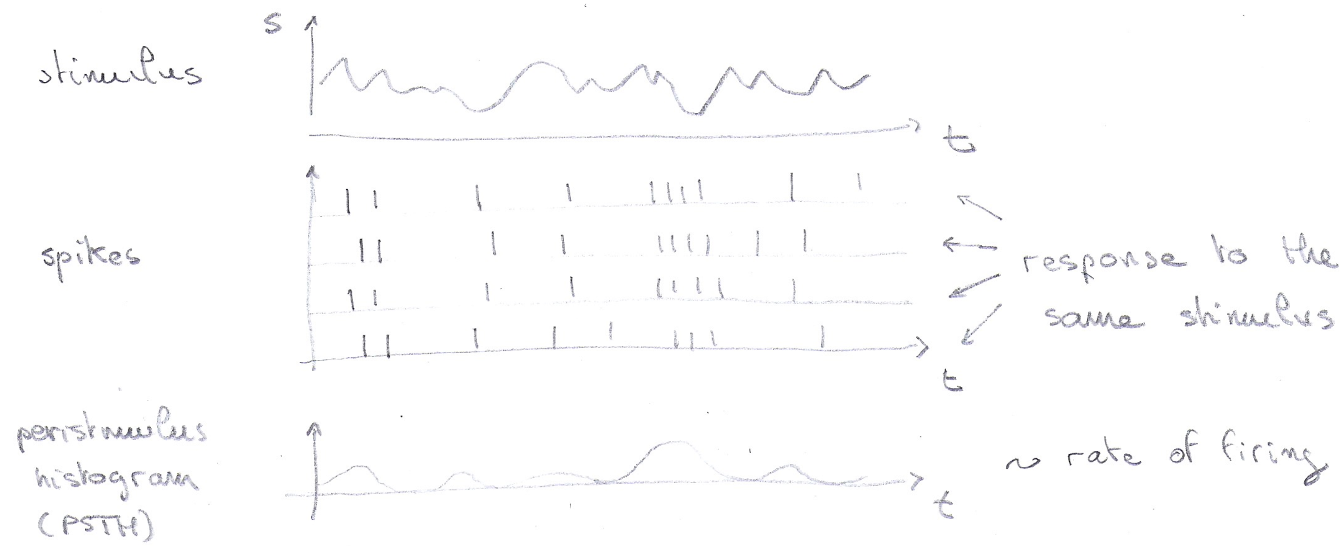
1 - P, as a function of y, is the encoding distribution

2 - P, as a function of k, is the likelihood function useful for parameter estimation given the data

3 - P, as a function of x, is the input likelihood function useful for input reconstruction given the data (decoding)

Probabilistic models are used to infer parameters (fitting + validation) or to study simplified dynamics of large neural networks (ex: balanced network hypothesis / supralinearly - stabilized networks ...)

Simplest rate model - Poisson process



stimulus $s(t)$ $\xrightarrow[\text{receptive field}]{\text{non-linear filtering}}$ rate $r(t)$ $\xrightarrow[\text{Poisson variability}]{\text{stochastic realization}}$ spike trains $\sum_i \delta(t-t_i)$ ← Dirac mass

Poisson process:

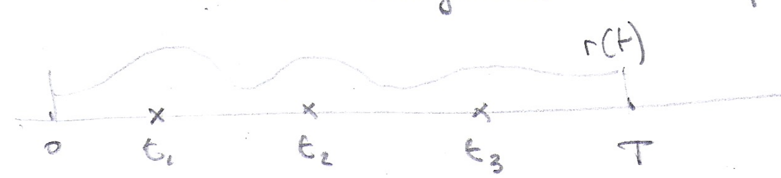
Properties on $\mathbb{R}$ follow intuitively from the construction.

↑ independent binomial variable with probability p_i, $1 \leq i \leq N$

$$\boxed{N \to +\infty,\ \tfrac{i}{N} \to t : \frac{p_i N}{T} \longrightarrow r(t)}$$

↓

limit : inhomogeneous Poisson process



As a counting process:

$N(t) = \int_0^t \sum_i \delta(t_i - s)\, ds$:

* $N(0) = 0$
* $\mathbb{P}\{N(t+h) - N(t) = 1\} = \lambda(t)h + o(h)$
* $\mathbb{P}\{N(t+h) - N(t) \geqslant 2\} = o(h)$
* $\mathbb{P}\{N(t) = k\} = \dfrac{e^{-\int_0^t r(s)ds}\left(\int_0^t r(s)ds\right)^k}{k!}$ ← Poisson law
* $p(t_1, \ldots, t_k \mid N(t) = k) = r(t_1)\, r(t_2) \cdots r(t_k) \Big/ \left(\int_0^t r(s)ds\right)^k$ ← independent with density r

General linear / Generalized linear model

$r(t) = \mathcal{F}(\{s(t')\}_{t'<t})$ ← instantaneous rate as a function of the stimulus

<u>In principle</u>: $r(t) = r_0 + \int_{-\infty}^{t} k(t_1)\, s(t_1)\, dt_1$

Volterra expansion →

$$+ \int_{-\infty}^{t}\int_{-\infty}^{t} k(t_1,t_2)\, s(t_1)\, s(t_2)\, dt_1\, dt_2$$

$$+ \int_{-\infty}^{t}\int_{-\infty}^{t}\int_{-\infty}^{t} k(t_1,t_2,t_3)\, s(t_1)\, s(t_2)\, s(t_3)\, dt_1\, dt_2\, dt_3$$

→ General linear: $r(t) = r_0 + \int_{-\infty}^{t} k(t_1)\, s(t_1)\, dt_1$

↳ receptive field ~ kernel

→ Generalized linear: $r(t) = f\left(r_0 + \int_{-\infty}^{t} k(t_1)\, s(t_1)\, dt_1\right)$

↳ static non linearity

Estimate of kernel k

* <u>reverse correlation method</u>: (LMS method)

- Error: $E = \frac{1}{T}\int_0^T dt \left(r_0 + \int_0^{+\infty} d\tau\, k(\tau)\, s(t-\tau) - r(t)\right)^2 = E[k]$ ← functional

r_0: true mean response; $k(\tau)$: unknown; $s(t-\tau)$: stimulus; $r(t)$: true response

$k^* = \text{argmin } E$ solves $\frac{\delta E}{\delta k} = 0$, which evaluates to:

$$\int_0^T dt \left(r_0 - r(t) + \int_0^{+\infty} d\tau'\, k(\tau')\, s(t-\tau')\right) \int_0^{+\infty} d\tau\, \delta k(\tau)\, s(t-\tau) = 0$$

Interverting the integrals leads to:

$$\int_0^{+\infty} d\tau' k(\tau') \underbrace{\left(\frac{1}{T} \int_0^T dt\, s(t-\tau)\, s(t-\tau') \right)}_{A(\tau-\tau') :\ \text{stimulus autocorrelation}} = \underbrace{\frac{1}{T} \int_0^T dt\, (r(t) - r_0)\, s(t-\tau)}_{C(-\tau) :\ \text{stimulus response cross correlation}}$$

K is determined by solving the convolution equation

$$\int_0^{+\infty} k(\tau') A(\tau-\tau') d\tau' = C(-\tau) \quad \leftarrow \text{reverse correlation analysis}$$

* maximum likelihood method. suppose we have

$$r(t) = \int_0^{+\infty} k(\tau) s(t-\tau) + \eta_t \leftarrow \text{Gaussian white noise}$$

$$\langle \eta_t \eta_s \rangle = \delta(t-s) : \text{autocorrelation}$$

the goal is to estimate the most likely form of k given the observation r, s and the noise model.

In practice such estimates are carried out on discretized data: $r_i = \sum_{j \geq 0} k_j s_{i-j} + \eta_j \leftarrow$ Gaussian i.i.d. $N(0,\sigma)$

likelihood function: $p(r \mid k, s) = \prod_{i=1}^{T} \left(\frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(- \frac{\left(r_i - \sum_{j\geq 0} k_j s_{i-j}\right)^2}{2\sigma^2} \right) \right)$

maximum likelihood: $k^* = \arg\max\, p(r \mid k, s)$

$$= \arg\max \underbrace{\log p(r \mid k, s)}_{L[k]\ \text{log likelihood}}$$

$$L[k] = -\sum_i \left(r_i - \sum_{j \geq 0} k_j s_{i-j} \right)^2 + ct = -E[k] + cte$$

$$\boxed{\text{reverse correlation kernel} \Longleftrightarrow \text{Gaussian linear model}}$$

There are maximum likelihood methods for generalized linear model (Jonathan Pillow / Liam Paninski).

For a Poisson noise model:

$$p(n_i | K, s) = \frac{(\Delta t r_i)^{n_i}}{n_i!} e^{-\Delta t r_i}, \quad \Delta t: \text{ bin size}$$

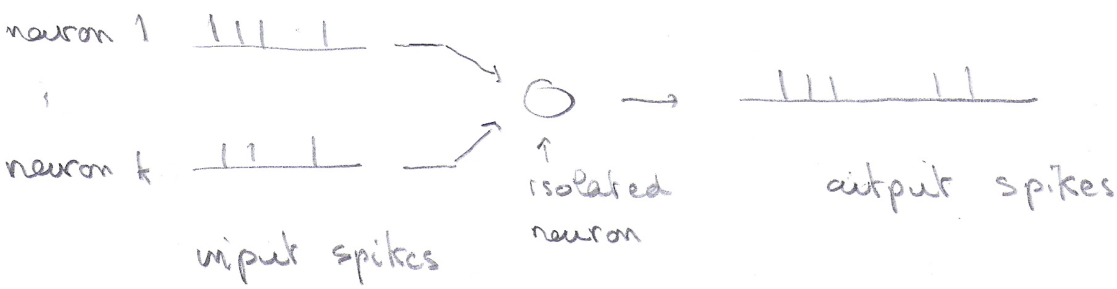
spike count in bin i

$$r_i = f\left(\sum_{j \geq 0} K_j s_{i-j}\right)$$

static non linearity (assumed to belong to a family of curves)

For simple family of static non linearity maximum likelihood estimation is a convex optimization problem

Network models: neural networks modeled as interacting point processes:



Hawkes-type models:

$$r_i(t) = F_i\left(\int_0^{+\infty} \sum_{j \neq i} h_{ij}(s)\, dN_j(t-s) \right)$$

F_i: static nonlinearity

$h_{ij}(s)$: "synaptic" kernel

$dN_j(t-s)$: $\sum_{t_{k_j} \geq 0} \delta(t - t_{k_j})$ ← spike trains of neuron $j \neq i$

Regular Hawkes (self exciting process)

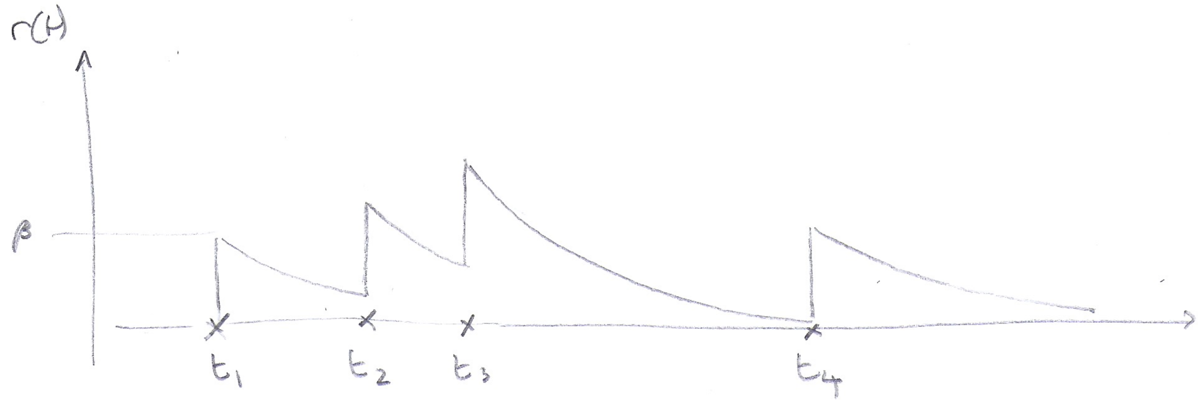
$$r(t) = r_0 + \int_0^{+\infty} \beta e^{-\alpha s}\, dN(t-s)$$

$dN(t-s)$: "Poisson point process with rate $r(s)$"

The instantaneous rate is a function of the past spiking history of the process. In particular it is a random variable since the past spiking history is itself stochastic. This idea of past-dependent stochastic instantaneous rate is formalized via the notion of stochastic intensity.

$$r(t) = r(t, \mathcal{F}_t)$$

$\mathcal{F}_t$: past spiking history registering all the previous spikes



⚠ <u>Existence of stationary regime</u>? At stationarity, the mean stochastic intensity is independent of the time:

$$\langle r(t) \rangle = \langle r_0 + \int_0^{+\infty} \beta e^{-\alpha s} \, dN(t-s) \rangle$$

$$= r_0 + \int_0^{+\infty} \beta e^{-\alpha s} \langle dN(t-s) \rangle$$

$$= r_0 + \int_0^{+\infty} \beta e^{-\alpha s} \, r(t-s)$$

$$r(t) = r = \text{cte} \Rightarrow \quad r = r_0 + \frac{\beta}{\alpha} r \quad \Leftrightarrow \quad r = \frac{r}{1 - \beta/\alpha}$$

explosion if $\beta/\alpha \to 1^-$, no stationary regime for $\beta/\alpha > 1$

<u>One must be careful</u>: see (Bremaud Massoulie 1996)

However, there are maximum likelihood methods to infer h_{ij} and f_i from neural activity recording. These have been shown to work quite well to fit and predict the activity of the retina. (Pillow et al 2008)

Key mathematical insights for the analysis of intensity-based network:

Ex: "counting" neuron with reset.

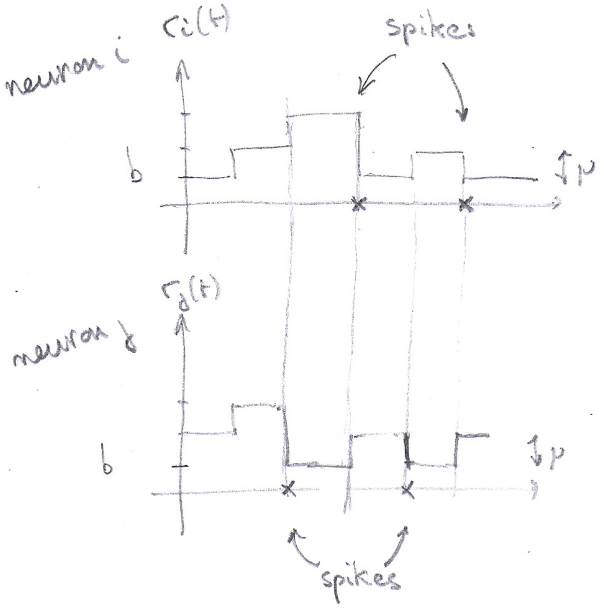
Fully connected network of K neurons

synaptic strength μ.

stochastic intensity of neuron i: $$r_i(t) = \int_{T_i(t)}^{t} \mu \sum_{j \neq i} dN_j(s) + b$$

$T_i(t)$: last-time neuron i spiked before t

N_j: point process counting the spikes of j



neurons are "counting" the number of spike they have received since last time they spiked and reset after spiking

Two insights: A/ dynamics best formulated in term of stochastic differential equations

B/ dynamics can be exactly simulated via discrete event method (Gillespie algorithm)

A] Stochastic differential equations can capture the history dependence of stochastic intensities: (generalizing the defining relation of Hawkes process)

$$r_i(t) = r_i(0) + \underbrace{\sum_{j \neq i} \int_0^t \mu \, dN_j(s)}_{\substack{\text{interaction term} \\ \text{registering received spike} \\ \text{between } 0 \text{ and } t}} + \underbrace{\int_0^t \left(r_i(t) - b\right) dN_i(s)}_{\substack{\text{reset term} \\ \text{registering spiking of } i \\ \text{between } 0 \text{ and } t}}$$

This equation are conservation rules: the change in intensity $r_i(t) - r_i(0)$ integrates Fixed size jumps $(\mu > 0)$ due to interaction and negative variable size jumps resetting the intensity to b after a spiking event.

B] The network dynamic is Markovian and can be efficiently simulated by discrete-event method, especially when the stochastic intensity is constant between interaction event. (Gillespie algorithm also used to simulate chemical reactions)

Idea: suppose the state of the network is given at $t=0$ by $(r_1, \dots, r_K) \leftarrow$ vector of stochastic intensities

In the absence of interactions, each neuron would spike at some time $\tau_1, \dots, \tau_K$, $\tau_i \sim \mathrm{Exp}(r_i)$! where $\mathrm{Exp}(r_i)$ denote the exponential distribution of parameter r_i.

Then $t = \min(\tau_1, \dots, \tau_k)$, the first time a neuron, say i, spike spontaneously is actually an interaction time of the system with interaction.

Thus at t: $r_i \leftarrow b$ reset

$r_{j\neq i} \leftarrow r_{j\neq i} + \mu$ interaction

One could proceed iteratively from there by drawing anew exponential variables: $\mathcal{E}xp(b_i)$, $\mathcal{E}xp(r_{j\neq i}+\mu)$. However, one can do better by only drawing $\tau_i \sim \mathcal{E}xp(b_i)$ for the spiking neuron and recycling the time $\tau_{j\neq i}$ via the update:

$$\tau_{j\neq i} \longleftarrow \underbrace{(\tau_{j\neq i} - t)}_{1} \frac{r_{j\neq i}}{r_{j\neq i}+\mu} \begin{array}{l} \leftarrow \text{rate before spiking} \\ \leftarrow \text{rate after spiking} \end{array}$$

Because of the memoryless property of exponential variables $(\tau_{j\neq i} - t) \sim \mathcal{E}xp(r_{j\neq i})$. Thus the update produce a new value distributed as

$$\frac{r_{j\neq i}}{r_{j\neq i}+\mu} \mathcal{E}xp(r_{j\neq i}) \sim \mathcal{E}xp(r_{j\neq i}+\mu).$$