Written by Yarden Nativ, mail: nativyardenac@gmail.com

1^{st} Recitation 16.3.23

Signal sampling, firing rate, convolution and stochastic processes

The recitation notes are based on:

- Drongelen, Signal processing for neuroscience, 2^{nd} edition (2018)
- Lecture notes by dr. Naomi Feldheim, BIU math course: Probability and Stochastic Processes

Data sampling:

The signals recorded from the brain can be represented in two discrete ways:

- Values of physiological measurements in a given temporal resolution such as LFP or membrane potential in milliseconds or bold signal in seconds. Example:

Time (ms)	0	1	2	3	4	5
LFP (mV)	-1	-2.5	0	1.7	2	1.3

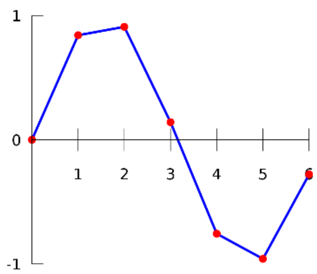
  Correspondent vector: [-1,-2.5,0,1.7,2,1.3]

- Vector in which only if a value is crossing a given threshold they are presented, mainly useful for spike train vectors. Example:

Time (ms)	0	1	2	3	4	5
Spike				V	V	

  Correspondent vector: [0,0,0,1,1,0]

As mentioned before, every recording of continuous signal makes it discrete by using, either averaging or a delta function. For example, a recording of LFP with a resolution of 10 ms can stand for 10 samplings made every millisecond and averaged to a single value. By so, we understand that sampling a continuous signal will give only a single value at a specific moment, and in between we interpolate the measurements. In neuroscience, we assume that 1 ms resolution recording is enough for even the fastest processes, but still we have to keep in the back of our mind that our measurements are discrete and the connection between them is interpolated. Illustration:



Written by Yarden Nativ, mail: nativyardenac@gmail.com

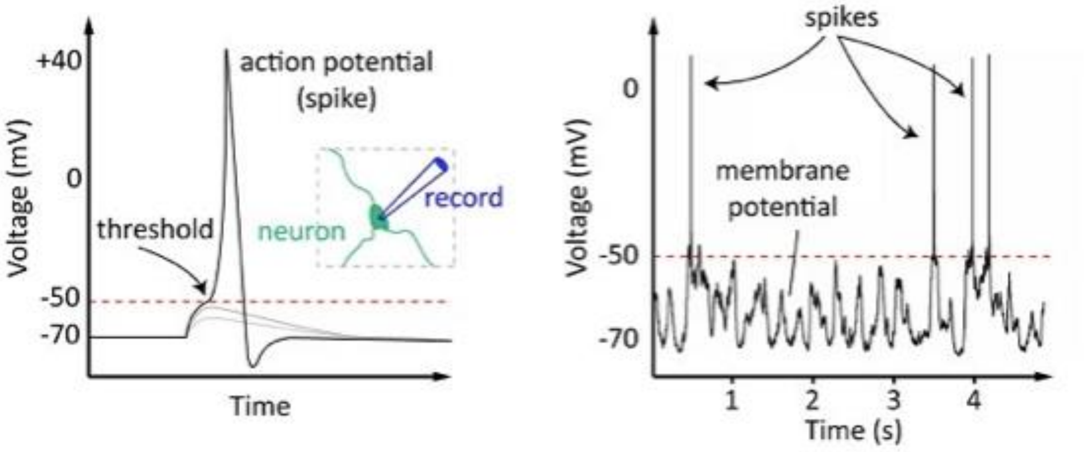
Delta functions:

Each recording method of the brain is using a delta function, measuring only one measure at a given moment. The two delta functions are given by the following definitions:

$$\textbf{Dirac: } \delta(x-a) = \begin{cases} 0 & x \neq a \\ 1 & x = a \end{cases} \qquad \textbf{Kronecker: } \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & else \end{cases}$$

Class Exercise:

Given the following patch clamp recording, write a spike train vector (0 and 1) with resolution of 1 second.



(adopted from the website: The Brain Bank North West)

Class solution:

Firing Rate:

As we will see along the course, we have a good justification to believe neurons pass information not by single spikes, but by changes of the spiking rates, also known as firing rate (FR). Similarly to Hz which is defined by number of samples per second, firing rate is commonly used as spikes/second even if it changes every 100 milliseconds.

There are few definitions to firing rate, and it can mean some of the following:

- r- firing rate along the time duration Δt, also known as spike count rate: $\frac{number\ of\ spikes}{\Delta t}$
- $\langle r \rangle$- mean firing rate averaged across trials
- $r(t)$- firing rate during a very short time period $\Delta t \rightarrow 0$
  As you'll experience in project 1, this method of firing rate is very sensitive to the size of the chosen parameter Δt and to computational changes of the data such as convolution for smoothing.

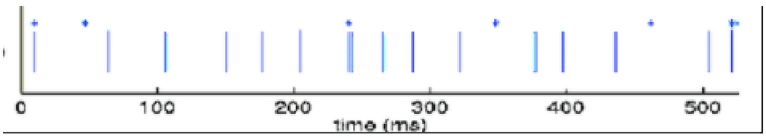
Using bins for computing firing rate

How do we transform a spike train vector containing 0,1 into a firing rate function? One of the methods is using non-overlapping time bins in which we count the number of spikes in each bin and divide by the bin duration.

Class Exercise:

Given the following spike train, write the firing rate function using:

- Bins with equal sizes of 100 ms.
- Bins with equal sizes of 50 ms.
- Bins with alternating sizes of 100 and 50 ms.

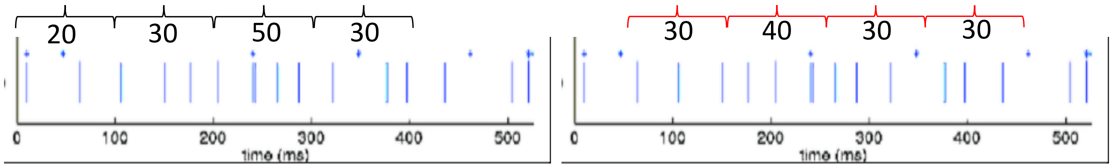


Class solution:

Written by Yarden Nativ, mail: nativyardenac@gmail.com

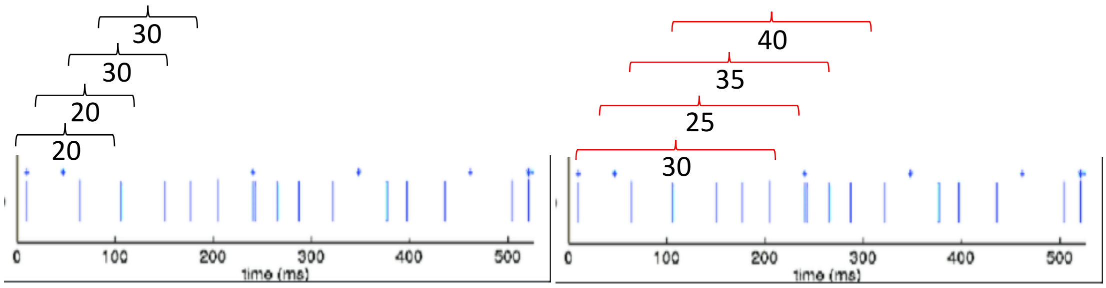
Convolution:

One of the main problems of using bins is that it requires some arbitrary decisions regarding the length and the timing of each bin. Here is an example how these two parameters can affect the resulting firing rate function:



For this reason, instead of binning the spike train, one could expect using a one bin and moving it across the spike train, each time by a single step. This method is practically using a **moving average**, one of the basic meanings of convolution.

The main advantage for this method is that any change in the firing rate can be detected immediately. Nevertheless, the size of the moving window still affects the result, and this is why each time we use a moving average, we note the size of the window.



Class questions for open discussion:

- What is the difference between sizes of the window? (level of averaging, meaning level of smoothing)
- How to handle the edges?
- How do we decide about the time to write the result?

Written by Yarden Nativ, mail: nativyardenac@gmail.com

Class Exercise: Moving average as scalar multiplication

We have the following spike train with sampling rate of 1000 Hz:

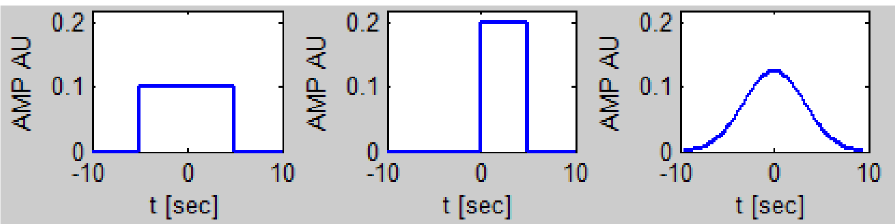
[0,0,1,1,0,1]

- Without the edges, calculate the convolved firing rate using a moving window of 3 samples.
- Scalarly multiply each triplet sequence by the vector [1/3,1/3,1/3].
- Show computationally why the two results are the same. Clue: multiply vector [a,b,c] by [1/3,1/3,1/3] and show it is the same as simple average.
- Using the second method for convolution, how can zero padding help us?
- What is the averaging meaning of using a vector with a triangle? And with gaussian distribution? Conclusion: Weighted arithmetic mean, each convolution window requires span (width) and shape (relevant function)
- Using the second method for convolution, what will be the difference between using the vector [0.5 2 0.5] and [1 4 1]? Does both represent the same moving weighted average? Conclusion: convolution function has to normalize the window to the sum of the elements
- What will be the result of using a vector with length m moving across vector n convolutionally? Conclusion: n+m-1

Convolution implementations for smoothing and for filtering:

Commonly we use convolution operation over our data for smoothing and filtering purposes, due to biological noise. The convolution as filter will narrow a biological noise with specific frequency (for example heart beats, respiratory etc.). The reason for that will be understood later in the course using the frequency domain transformations. The smoothing is doing the same just without picking a specific frequency. The most common convolution windows are:

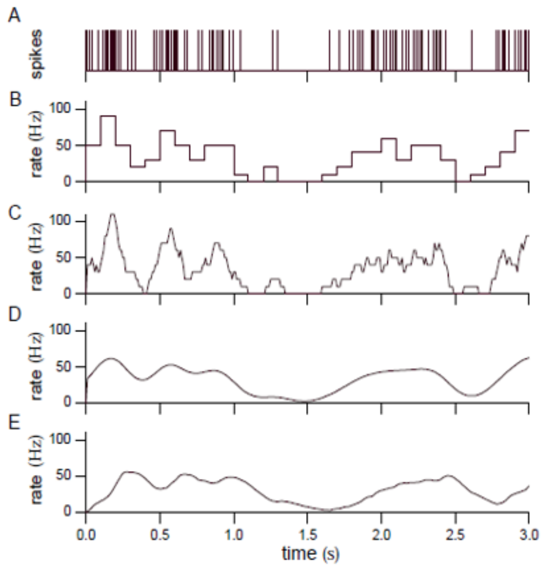
Rectangual window Causal window Gaussian window



Written by Yarden Nativ, mail: nativyardenac@gmail.com

Two examples:

a. From spike trains to rate functions:

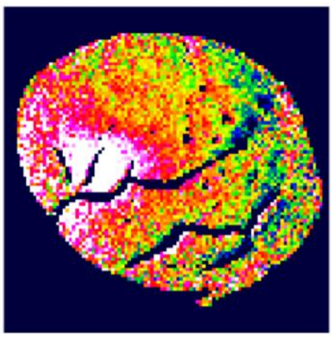


- A: Spikes
- B: Binned count
- C: Sliding window
- D: Sliding Gaussian kernel
- E: Sliding causal kernel

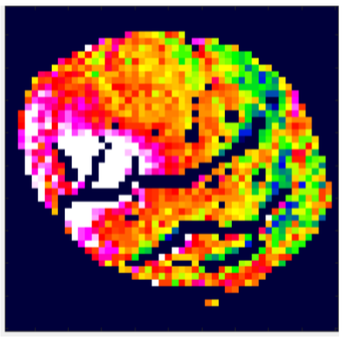
Last causal kernel is half a gaussian.

b. Using fluorescent imaging with convolution:

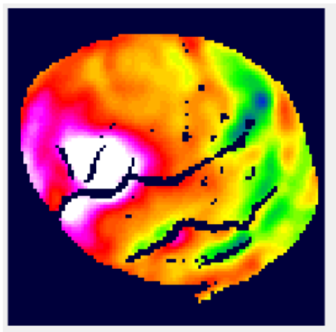
Original signal:



Bins of 4X4:



Gaussian (std=1.5 pixel)



Signal to noise ratio (SNR):

Before we use smoothing or binning to reduce the noise in the data, we aim to quantify the level of noise it has. For this the size SNR is made for, quantifying in a logarithmic scale of decibels (dB) the ratio between the signal and the noise. Higher SNR indicates less noise, for example SNR of 10 db indicates the signal is 10 times stronger than the noise, and 2 dB indicates signal is 2 times stronger than the noise.

To simplify the effects of noise, we'll assume that noise ξ can affect the recording of firing rate $r(t)$ which modeled by $f(t)$ in one of two ways:

Additively: $r(t) = f(t) + \xi$

multiplicatively: $r(t) = f(t) + g(t) \cdot \xi$ (g(t)- gaussian noise)

To decide what is the signal and what is the noise (for example motor related spiking activity is the signal and spontaneous activity from resting state is the noise), we define SNR using one of two methods:

Mean square: $SNR_{dB} = 10 \log_{10} \frac{\langle (signal)^2 \rangle}{\langle (noise)^2 \rangle}$ $\langle \underline{x}^2 \rangle \equiv \frac{1}{N} \sum_{i=1}^{N} x_i^2 \equiv \frac{1}{T} \int_0^T x^2 dx$

Root mean square: $SNR_{dB} 20 \log_{10} \frac{\sqrt{\langle (signal)^2 \rangle}}{\sqrt{\langle (noise)^2 \rangle}}$ $\sqrt{\langle \underline{x}^2 \rangle} \equiv \sqrt{\frac{1}{N} \sum_{i=1}^{N} x_i^2} \equiv \sqrt{\frac{1}{T} \int_0^T x^2 dx}$

Class Exercise: Pseudocode for SNR

One of the known phenomena in the somatosensory system is of lateral inhibition, an elevation of sensitivity threshold of neurons adjacent to stimulated one in the periphery.

- Design a 1D model of the neurons by spatially arranging them, determining their firing rate at rest and during stimulus and their threshold for changing the firing rate.
- Pick one neuron to be the stimulated one with a square pulse shifting across this neuron towards the others.
- Create a Mexican hat signal changing the threshold of adjacent neurons based on the function $\psi(x) = \frac{2}{\sqrt{3\sigma\pi^{\frac{1}{4}}}} \left(1 - \left(\frac{x}{\sigma}\right)^2\right) e^{-\frac{x^2}{2\sigma^2}}$. Note that you need to decide the length of the inhibition.
- Add a gaussian noise over this signal using the function $g(x) = Ae^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}$. Use both multiplication
- Calculate the SNR.

Convolution arithmetic definition: As we've seen using convolution is composed of the following operations:

1. (Reversing the convolution window)
2. Multiplication
3. Moving the window
4. Summation of all multiplication results

If the convolution window is given by the function $h(t)$ so a convolution over signal $x(t)$ can be written as:

Continuous convolution: $y(t) = (x * h)(t) = \int_{-\infty}^{\infty} x(\tau) \cdot h(t - \tau) d\tau$

Discrete convolution: $\underline{y}[t] = (\underline{x} * \underline{h})[t] = \sum_{\tau=-\infty}^{\infty} \underline{x}[t] \cdot \underline{h}[t - \tau]$

We should note that practically our experiments never start in the far away history ($\tau \rightarrow -\infty$) and we stop our measurement sometime so for this reason we never really measure until the infinite future ($\tau \rightarrow \infty$). We tend to assume that the system was at some level of balance or resting state before we start the recording and will go back to it after we end it, and by so we can assume the "zero padding":

Continuous convolution: $y(t) = (x * h)(t) = \int_{0}^{t_{max}} x(\tau) \cdot h(t - \tau) d\tau$

Discrete convolution: $\underline{y}[t] = (\underline{x} * \underline{h})[t] = \sum_{\tau=0}^{t_{max}} \underline{x}[t] \cdot \underline{h}[t - \tau]$

For this we have the "causal window" commonly used instead of the rectangle window. To use it we must first reverse the window, the first step we commonly do.

Class Exercise: calculating convolution explicitly

- Calculate the convolution for continuous impulses $f_1(t) = t, f_2(t) = -t$
- Calculate the convolution for discrete impulses $f_1(t) = u(t) - u(t-3), f_2(t) = u(t) - u(t-1)$ with sampling rate of 1 ms.

Conclusion: convolution is graphically sliding overlapping areas of two signals.
Example 1, Example 2

Summary so far:

The advantages of convolution-

- Smoothing the data to reduce noise.
- Avoids arbitrary bins.
- Detects changes in firing rates faster than bins.
- Allows to use different kinds of weights.

The disadvantages of convolution-

- Depends on the shape and span of the window.
- Changes the time axis.
- Depends on time assignments.
- We must assume the system is in resting state before and after.
- For this reason, convolution creates “non real edges”.

Important things about creating a convolution function-

- Reverse the window.
- Normalize the window.
- Zero padding.
- Right length of outcome.

Convolution for a linear model of neuron’s computation:

One possible simplifying model for a neuron is that it is a linear time-invariant system which its output firing rate is independent in time (at any given time the same stimulus to a neuron will produce the same output) and it has superposition characteristic (its outputs are a linear sum of few other simplified outputs).

If for example a neuron responds with one spike for a given input, we can model the response $y(t)$ by the convolution:

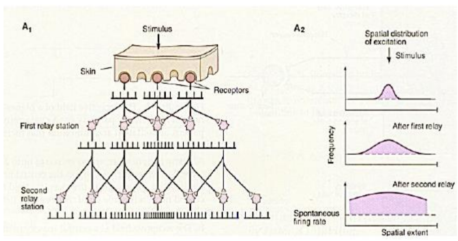
$$y(t) = \int_{-\infty}^{\infty} x(\tau)\delta(t-\tau)d\tau$$

If this neuron requires a sequence of events as input so it will respond with one spike, we can model it as a convolution with the neuron’s convolving function $h(t)$. This function can be addressed as the “weighted memory” of the neuron:

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau$$

These basic computations will allow us to develop our first simplistic model of neuron as a memoryless stochastic system (Markov property).

Example from the somatoseneory system for lateral inhibition:



Written by Yarden Nativ, mail: nativyardenac@gmail.com

Stochastic Processes:

Mathematically, stochastic matrix is a transition matrix describing the probabilities of transitions between different states of a system. This matrix must define a Markov chain which has a Markov property of memoryless process. In neurons, we refer to this memoryless property regarding the assumption that in the short term of few hours, the probability for a neuron creating new spike does not depend on how many spikes the neuron ever produced and how many stimuli it was presented with. For this reason, we use the term "stochastic process" to refer to the neuron as a random system, non-deterministic.

How do we check stochasticity of a process? If it can be written with a deterministic equation, it is not stochastic (proof by contradiction). Practically in neuroscience, we want to check if a stochastic process is one of the following:

Stationary process	Weak stationary process	Ergodic process
Same distribution all along the process	Expectation and Covariance over time are constant	Expectation and Covariance over time and over trials are constant

Class Exercise: Define the following processes

Write three examples for processes or distributions- one stationary and ergodic, another one stochastic and not stationary and another stationary but not ergodic.

3 examples:

- Rolling a dice (Stationary and ergodic)
- Birds on a tree every day (Stochastic, not stationary)
- Average time of sport for every man (Stationary, not ergodic)