

# A consistent estimator of non trivial stationary solutions of dynamic neural fields

Eddy Kwessi\*

## Abstract

Dynamics of Neural Fields are tools used in neurosciences to understand the activities generated by large ensembles of neurons. They are also used in networks analysis and neuroinformatics in particular to model a continuum of neural networks. They are mathematical models that describe the average behavior of these congregations of neurons, which are often in large amounts, even in small cortexes of the brain. Therefore, change of average activity (potential, connectivity, firing rate, etc) are described using systems of partial different equations. In their continuous or discrete forms, these systems have a rich array of properties, among which the existence of nontrivial stationary solutions. In this paper, we propose an estimator for nontrivial solutions of dynamical neural fields with a single layer. The estimator is shown to be consistent and a computational algorithm is proposed to help carry out implementation. An illustrations of this consistency is given based on different inputs functions, different kernels, and different pulse emission rate functions.

**Keywords:** Dynamic neural fields, nontrivial, stationary, estimator, consistent.

## 1 Introduction

It is known that any small piece of human or animal cortex contains a vast amount of neurons. Therefore, a continuum approach in modeling these large ensembles makes sense and was pioneered by the work of Beurle (1956). This work was designed to accommodate only excitable networks of neurons and was subsequently generalized by Wilson and Cowan (1972) to include inhibitory neurons as well. Amari (1977) considered ensembles of neurons when studying pattern formation. Since then, there have been applications and extensions of his work in several directions and the birth of the field of dynamic field theory as bi-product. These extensions have for instance enabled analyses of Electroencephalograms (Nunez and Srinivasan (2006)), short term memory (Camperi and Wang (1998)), visual hallucinations (Ermentrout and Cowan (1979), Tass (1995)), and most recently robotics using dynamics neural fields. Applications to robotics has proven very effective, see for instance the works of Bicho et al. (2000), Erlhagen and Bicho (2006), Erlhagen and Schöner (2001), and Bicho et al. (2010). The authors of the latter provided studies in which robots to humans interactions were implemented based on information from Dynamic Neural Fields (DNF). The theoretical aspects started with the work of Wilson and Cowan (1972) and Amari (1977) and will be summarized below.

Let  $\Omega \subseteq \mathbb{R}^d$  be a manifold. In the presence of neurons located a position  $\xi \in \Omega$  at time  $t$  arranged on  $L$  layers, the average potential function  $V_k(\xi, t)$  is often used to understand the continuous field on the  $k$ th layer.  $V_k(\xi, t)$  is the average membrane potential of the neurons

---

\*Corresponding author, Department of Mathematics, Trinity University, San Antonio, TX78212, USA, ekwessi@trinity.edu

located at position  $\xi$  at time  $t$  of the  $k$ th layer. When  $L = 1$ ,  $V(\xi, t)$  can also be understood as the synaptic input or activation at time  $t$  of a neuron at position or direction  $\xi$ . It satisfies the Amari equation (see Amari (1977)) which is given as

$$\frac{\partial V_k(\xi, t)}{\partial t} = -V_k(\xi, t) + \sum_{l=1}^L \int_{\Omega} K_{kl}(\xi, y) G(V_l(y, t)) dy + S_k(\xi, t), \quad (1.1)$$

where  $K_{kl}(\xi, y)$  is the intensity of the connection between a neuron located at position  $\xi$  on the  $k$ th layer with a neuron a position  $y$  on the  $l$ th layer,  $G(V_k(\xi, y))$  is the pulse emission rate (or activity) at time  $t$  of the neuron located at position  $\xi$  on the  $k$ th layer.  $G$  is often chosen as a monotone increasing function.  $S_k(\xi, t)$  represents the intensity of the external stimulus at time  $t$  arriving on the neuron at position  $\xi$  on the  $k$ th layer.

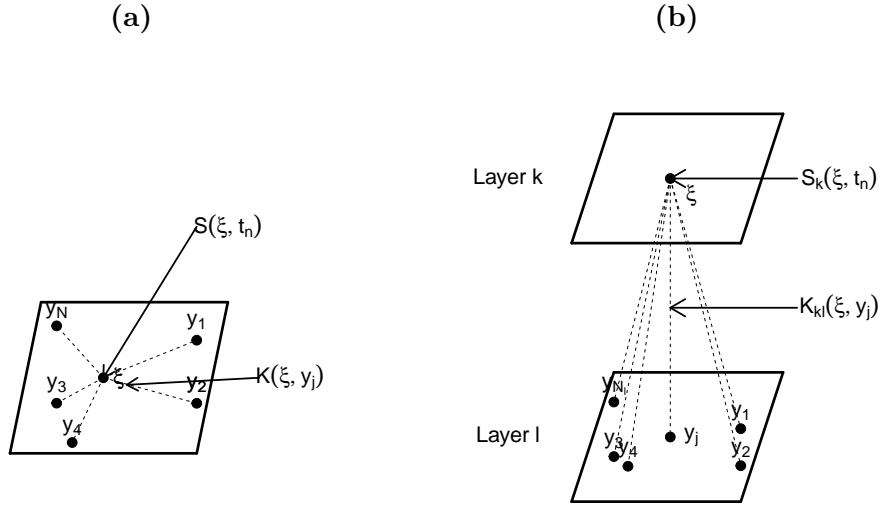


Figure 1: An illustration of the DNF for single layer (a) and multiple layers (b).

DNFs have also branched out to dynamical systems, for instance, Beim and Hutt (2014) studied an heterogeneous aspect of DNF and found existence of attractors and saddle nodes for solutions of (1.1). Existence of solutions of DNF is based on fixed point theory using Hammerstein integral equation (see Hammerstein (1930)) such as in Djitte and Sene (2013). Now based on recent developments in recurrent neural networks (RNN), equation (1.1) can be discretized using nearly exact discretization schemes (see Kwessi et al. (2018)) to give rise to discrete dynamical neural fields as

$$V_{k,n+1}^{(i)} = F(V_{k,n}^{(i)}) := \alpha_n V_{k,n}^{(i)} + \beta_n \sum_{l=1}^L \sum_{j=1}^N K_l^{(ij)} V_{k,n}^{(j)} + \eta_{k,n}^{(i)}, \quad (1.2)$$

where  $V_{k,n}^{(i)}$  represents the state of the membrane potential on the neuron at position  $\xi_i$  at time  $t_n$  on the  $l$ th layer,  $\alpha_n = 1 - \exp(-\delta_n)$  (where  $\delta_n = t_{n+1} - t_n$ ) is a time scale parameter,  $\beta_n = (1 - \alpha_n) \frac{|\Omega|}{N}$  is a parameter depending on the time scale and the size  $|\Omega|$  of the manifold

$\Omega$ ,  $K_k^{ij}$  are heterogeneous weights representing the connectivity between a neuron at position  $\xi_i$  on the  $k$ th layer with a neuron at position  $y_j$  on the  $l$ th layer, and  $\eta_{k,n}^{(i)}$  is the intensity of the external stimulus arriving at the neuron at position  $\xi_i$  at time  $t_n$  on the  $k$ th layer. We observe that (1.2) represents a discrete dynamical system. To study the stability analysis of the discrete dynamical system (1.2), one needs to first find the stationary solutions given as  $V_{i,n+1}^{(k)} = V_{i,n}^{(k)} := V_{i,*}^{(k)}$  and evaluate the derivative of the map  $F$  at these stationary solutions. This is a difficult if not impossible task if we do not know how to estimate the stationary solutions for the DNF. This is one of the main motivating factors behind these current paper.

Moreover, from Elman (1990), Williams and Zipser (1990), and most recently, Durstewitz (2017), this equation is also a RNN. Therefore, the tools of discrete dynamical systems can be applied non only to single-layer DNFs, but to multiple-layers DNFs, where conditions for stability are well-known.

Another interesting aspect of DNFs is that if we restrict  $\Omega$  to the unit circle  $\mathbb{T}$  where  $\mathbb{T} = \{z \in \mathbb{C} : |z| = 1\}$ , then solutions may exist in the complex unit disk  $\mathbb{D} = \{z \in \mathbb{C} : |z| < 1\}$  with the absence of external stimulus, such solutions would also be solutions of a Dirichlet problem associated with equation (1.1), see Green and Krantz (1997). Indeed, suppose that  $V = V(z, t)$  and let  $F(z, t) = G(V(z, t))$  for  $z \in \mathbb{T}$  and  $t \geq 0$ , for some complex-valued function  $G$ . Consider the Poisson kernel on  $\mathbb{D}$  is defined as  $K(z, \omega) = \frac{1}{2\pi} \frac{1 - |z|^2}{|1 - z\bar{\omega}|^2}$ , where  $z, \omega \in \mathbb{D}$ . From the theory of complex analysis (see Rudin (1987)), consider then the complex single-layer Amari equation

$$\frac{\partial V(z, t)}{\partial t} = -V(z, t) + \int_{\mathbb{T}} K(z, \omega) G(V(\omega, t)) d\omega + S(z, t). \quad (1.3)$$

Suppose  $F$  is a smooth function on  $\mathbb{T}$  or  $F$  is a distribution (in a functional sense) on  $\mathbb{T}$ . If a nontrivial stationary solution  $V(z)$  for this equation exists, then it satisfies the equation

$$V(z) = G(V(z)) + S(z). \quad (1.4)$$

An obvious corollary is that if the complex function  $G(z)$  has a fixed point, then a non trivial solution for the complex Amari equation (1.3) without stimulus ( $S(z) = 0$ ) when it exists is a harmonic function in  $\mathbb{D}$ , in that the Laplacian operator applied to  $V$  is identically zero. Therefore, as a harmonic function, such a solution may be written as a power series  $V(z) = \sum_{n \in \mathbb{Z}} a_n z^n$ , where the coefficient  $a_n$  are to be determined. This would be an interesting aspect of this non trivial solutions of DNF worth investigating akin to the Lotka-Volterra expansion proposed by Beim and Hutt (2014).

Most analyses of DNFs focus on their applications and theoretical properties. However, given that kernels often used in practice are either Gaussian, Laplacian, or tangent hyperbolic kernels, and the function  $G$  is monotone increasing, there are avenues to also study statistical properties of the DNFs, albeit in specific situations. Indeed, the aforementioned kernels can be thought of as density functions of a random variable  $Y$  so that the integrand in equation (1.1) can be viewed as the average of the random variable  $G(V(Y))$  over the manifold  $\Omega$ . With that understanding at hand, our goal is to use this new statistical paradigm to propose a consistent estimator for non trivial solutions of DNF. The remainder of the paper is organized as follows. In section 2, we state the necessary definitions and the main result, in section 3, we propose a computation algorithm for the implementation of the estimator. In Section 4, we state the technical considerations to be used in the implementation, by proposing other functions  $G$  beyond the usual sigmoid function. In Section 5, we perform the Monte Carlo simulations based on different kernel functions and make concluding remarks Section 6.

## 2 Main Results

We observe that non-stationary solutions of (1.1) are given as

$$V_k(\xi) = \sum_{l=1}^L \int_{\Omega} K_{kl}(\xi, y) G(V_l(y)) dy + S_k(\xi) .$$

Henceforth for simplicity sake, given that at up to a kernel, the stationary solutions would have the same form, we consider a dynamic neural field with a single layer, so that (1.1) becomes

$$\frac{\partial V(\xi, t)}{\partial t} = -V(\xi, t) + \int_{\Omega} K(\xi, y) G(V(y, t)) dy + S(\xi, t) . \quad (2.1)$$

Let  $V$  be a stationary solution of the integro-differential equation (2.1). According to Hammerstein (1930), such a nontrivial solution exists if  $K(x, y)$  is symmetric positive definite and  $G$  satisfies  $G(V) \leq \mu_1 V + \mu_2$ , where  $\mu_1, \mu_2$  are positive constants. We know that  $V$  is defined over the domain  $\Omega$  as

$$V(\xi) = \int_{\Omega} K(\xi, y) G(V(y)) dy + S(\xi) \quad \text{for } \xi \in \Omega . \quad (2.2)$$

**Definition 1.** The indicator function  $I_A$  of the set  $A$  is defined as

$$I_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases} .$$

We recall the distribution of a consistent estimator.

**Definition 2.** Let random sample  $X_1, X_2, \dots, X_N$  from distribution with parameter  $\theta$ . Let  $\hat{\theta}_N = h(X_1, X_2, \dots, X_N)$  be an estimator of  $\theta$ . Then  $\hat{\theta}_N$  is said to a consistent estimator for  $\theta$  if for any real number  $\epsilon > 0$

$$\lim_{N \rightarrow \infty} Pr(|\hat{\theta}_N - \theta| \geq \epsilon) = 0 .$$

**Theorem 3.** For a given  $\xi \in \Omega$ , suppose  $Y$  is a random variable supported on  $\Omega$  with probability density function distribution  $f(y|\xi) = K(\xi, y)$ . Suppose  $G$  is the cumulative distribution function of some random variable  $U$  supported on  $V(\Omega)$ .

Then given positive integers  $n$  and  $m$ , define

$$V_{n,m}(\xi) := \frac{1}{nm} \sum_{i=1}^m \sum_{j=1}^n I_{[u_i, \infty)}(V(y_j)) + S(\xi) , \quad (2.3)$$

where  $u_i, 1 \leq i \leq m$  and  $y_j, 1 \leq j \leq n$  are random points from  $U$  and  $Y$  respectively.

Then for  $\xi \in \Omega$ , we have that

$$V_{n,m}(\xi) \text{ is a consistent estimator of } V(\xi) .$$

*Proof.* From the Markov Inequality, we know that given  $\nu > 0$   $Pr(|V_{n,m} - V| \geq \nu) \leq \frac{E[V_{n,m} - V]}{\nu}$ , therefore, it will be enough to prove that given  $\xi \in \Omega$ ,

$$\lim_{n,m \rightarrow \infty} E_{Y|\xi}[V_{n,m} - V] = 0 .$$

We have that

$$\begin{aligned}
|E_{Y|\xi}[V_{n,m} - V]| &= \left| \int_{\Omega} (V_{n,m}(\xi) - V(\xi)) f(y|\xi) dy \right| \\
&\leq \left| \int_{\Omega} \frac{1}{n} \left( \frac{1}{m} \sum_{i=1}^m I_{[u_i, \infty)}(V(y_j|\xi)) - Pr(U \leq V(y_j|\xi)) \right) f(y|\xi) dy \right| + \\
&+ \left| \int_{\Omega} \frac{1}{n} \sum_{j=1}^n (Pr(U \leq V(y_j|\xi)) - G(V(y|\xi))) f(y|\xi) dy \right| \\
&+ \left| \int_{\Omega} \left( \frac{1}{n} \sum_{j=1}^n G(V(y|\xi)) - V(\xi) \right) f(y|\xi) dy \right|
\end{aligned}$$

Put

$$\begin{aligned}
I_1 &= \left| \int_{\Omega} \frac{1}{n} \sum_{j=1}^n \left( \frac{1}{m} \sum_{i=1}^m I_{[u_i, \infty)}(V(y_j|\xi)) - Pr(U \leq V(y_j|\xi)) \right) f(y|\xi) dy \right| \\
I_2 &= \left| \int_{\Omega} \frac{1}{n} \sum_{j=1}^n (Pr(U \leq V(y_j|\xi)) - G(V(y|\xi))) f(y|\xi) dy \right| \\
I_3 &= \left| \int_{\Omega} \left( \frac{1}{n} \sum_{j=1}^n G(V(y|\xi)) - V(\xi) \right) f(y|\xi) dy \right|
\end{aligned}$$

Since  $F_m(x) = \frac{1}{m} \sum_{i=1}^m I_{[u_i, \infty)}(x)$  is the empirical distribution of the random variable  $U$ , we have that

$$\lim_{m \rightarrow \infty} F_m(V(y_j|\xi)) = Pr(U \leq V(y_j|\xi))$$

and since  $f(y|\xi)$  is density function, we have

$$I_1 \leq \frac{1}{n} \sum_{j=1}^n |F_m(V(y_j|\xi)) - Pr(U \leq V(y_j|\xi))| \rightarrow 0 \quad \text{as } n, m \rightarrow \infty.$$

We observe that by definition  $Pr(U \leq V(y_j|\xi)) = G(V(y_j|\xi))$ , therefore

$$I_2 \leq \int_{\Omega} \left| \frac{1}{n} \sum_{j=1}^n [G(V(y_j|\xi)) - G(V(y|\xi))] \right| f(y|\xi) dy.$$

Given  $\epsilon$ , by continuity of  $G$ , there exists  $\delta > 0$  such that

$$|V(y_j|\xi) - V(y|\xi)| < \delta \implies |G(V(y_j|\xi)) - G(V(y|\xi))| < \epsilon.$$

It follows that for any  $\epsilon > 0$

$$I_2 \leq \frac{\epsilon}{n} \rightarrow 0 \quad \text{as } n, m \rightarrow \infty.$$

To finish, we note that

$$V(\xi) = \int_{\Omega} K(\xi, y) G(V(y)) dy = \int_{\Omega} G(V(y|\xi)) f(y|\xi) dy = E_{Y|\xi}[G(V(Y|\xi))].$$

Therefore

$$\begin{aligned}
I_3 &= \left| \int_{\Omega} \left( \frac{1}{n} \sum_{j=1}^n G(V(y|\xi)) - V(\xi) \right) f(y|\xi) dy \right| \\
&= \left| \int_{\Omega} (G(V(y|\xi)) - V(\xi)) f(y|\xi) dy \right| \\
&= \left| \int_{\Omega} G(V(y|\xi)) f(y|\xi) dy - V(\xi) \int_{\Omega} f(y|\xi) dy \right| \\
&= \left| \int_{\Omega} G(V(y|\xi)) f(y|\xi) dy - V(\xi) \right| = 0
\end{aligned}$$

This concludes the proof that  $V_{n,m}$  is a consistent estimator for  $V$ .  $\square$

**Remark 4.**

1. We observe that  $V_{n,m}(\xi)$  depends on the knowledge of  $V(y_i)$ , which is not known in general. However, we observe that  $V_{n,m}(y_j)$  given in equation (2.3) has minimum  $V_{n,m}^{min}(y_j) = S(y_j)$  and a maximum of  $V_{n,m}^{max}(y_j) = \frac{1}{n} + S(y_j)$  because  $\frac{1}{m} \sum_{i=1}^m I_{[u_i, \infty)}(V(y_j))$  has minimum 0 and maximum 1.

This also means that although we may not know exactly the value of  $V_{n,m}(y_j)$ , we can estimate it to be between  $S(y_j)$  and  $\frac{1}{n} + S(y_j)$ .

We can therefore select  $V(y_i)$  between  $S(y_j)$  and  $\frac{1}{n} + S(y_j)$  for  $j = 1, 2, \dots, n$  and the estimate of the nontrivial solution will exist in a small interval of length (bandwidth)  $\frac{1}{n}$ . Hence the if the domain  $\Omega$  is very dense in points  $y_i$ 's, then the nontrivial solution will be a small perturbation of the initial external input, and if the domain is sparse, then perturbation will be greater. Henceforth for simplicity sake, we will assume that  $V(y_j) \sim Unif(S(y_j), \frac{1}{n} + S(y_j))$ .

2. Another observation is that  $S$  depends on the position of the neuron  $\xi$  in a single-layer system, however, in a multiple-layer system, it would be reasonable to think of  $S$  as depending on both the layer  $k$  and the position of the neuron on the layer, that is,  $S(V_k(\xi, t))$ .

### 3 Computational Algorithm

In this section, we have the setting of Theorem 3 above. From Remark 4, above we can use the following algorithm to estimate  $V(\xi)$ :

1. Step 1: **Select positive integers  $n$  and  $m$ .**

Here, the experimenter should choose the values of  $m, n$  relative to how much computational capabilities ones has, knowing that very large values can lead to a significant slowdown of convergence

2. Step 2: **Select  $y_1, y_2, \dots, y_n$  from the distribution  $f(y|\xi)$ .**

Knowing that  $f(y|\xi) = K(\xi, y)$  is a known probability distribution (Gaussian, Laplace, or tangent hyperbolic, see the section below), this should be achievable with relative ease from any software.

3. Step 3: **Select  $u_1, u_2, \dots, u_m$  from the distribution of  $g(u)$  of  $U$  associated with  $G$ .**

As in the previous step, sampling from a known probability distribution  $g(u)$  should be

achievable. However, if  $G$  is not given as bounded function between 0, and 1, we can still truncate it adequately to obtain a probability distribution, see for instance Section 4.1. below

4. Step 4: **For  $j = 1, \dots, n$ , select  $V(y_j)$  from a uniform distribution  $Unif(S(y_j), \frac{1}{n} + S(y_j))$ .**

This steps assumes that we have an external stimulus  $S$  arriving on the neuron at position  $\xi$  given as a function of  $\xi$ .

5. Step 5: **For given  $\xi \in \Omega$ , evaluate  $V_{n,n}(\xi) := \frac{1}{nm} \sum_{j=1}^n \sum_{i=1}^m I_{[u_i, \infty)}(V(y_j)) + S(\xi)$ .**

In this final step, one can choose different values of  $\xi$  to plot the estimator in the space  $\Omega$ .

We will use step 4 only to evaluate  $V_{n,m}^{\max}(\xi)$ . From the above algorithm, it is clear that the activation function  $V(\xi)$  of the neuron at position  $\xi$  is the sum of the average of activations of neurons at position  $y_i$  and the external stimulus arriving at  $\xi$ . So essentially, the function  $V(\xi)$  is a perturbation of the function  $S(\xi)$  by the quantity  $\frac{1}{nm} \sum_{j=1}^n \sum_{i=1}^m I_{[u_i, \infty)}(V(y_i))$  that depends on  $\xi$  and possibly of parameters of the distribution of random variables  $U$  and  $Y$ . These parameters play the role of smoothing to compensate from the noise created by small values of  $n, m$ , see Section 5.

## 4 Technical considerations

In this section, we discuss the choices for the pulse emission rate function  $G$  and the connection intensity  $K(x, y)$ .

### 4.1 Pulse emission rate function

We note that Amari consider the dynamic of neural fields with pulse emission function  $G$  defined as the sigmoid function. However, equation still has non-trivial stationary solution even if  $G$  is not the sigmoid. In fact, there are is large class of nonlinear functions  $G$  for which this is true, see Figure 2 below. For example the following functions, often used in for training algorithm in artificial neural networks, that have been adequately truncated for our purposes. Here,  $\theta$  is a positive real number.

Denomination	Formulation	Conditions
Sigmoid : $G_1(v) =$	0	if $v < v_0$
	$\frac{1}{1 + e^{-\theta(v-v_0)}}$	if $v \geq v_0$
Weighted Sigmoid : $G_2(v) =$	0	if $v < v_0$
	$(v - v_0)G_1(v)$	if $v \geq v_0$
Hyperbolic Tangent : $G_3(v) =$	0	if $v < v_0$
	$1 - 2G_1(-2v)$	if $v \geq v_0$
Tangent inverse : $G_4(v) =$	$\arctan(v - v_0)$	if $v \geq v_0$
	0	if $v < v_0$
Heaviside : $G_5(v) =$	0	if $v < v_0$
	1	if $v \geq v_0$
Ramp : $G_6(v) =$	0	if $v < v_0$
	$\theta(v - v_0)$	$v_0 \leq v < v_0 + \theta^{-1}$
	1	if $v \geq v_0 + \theta^{-1}$

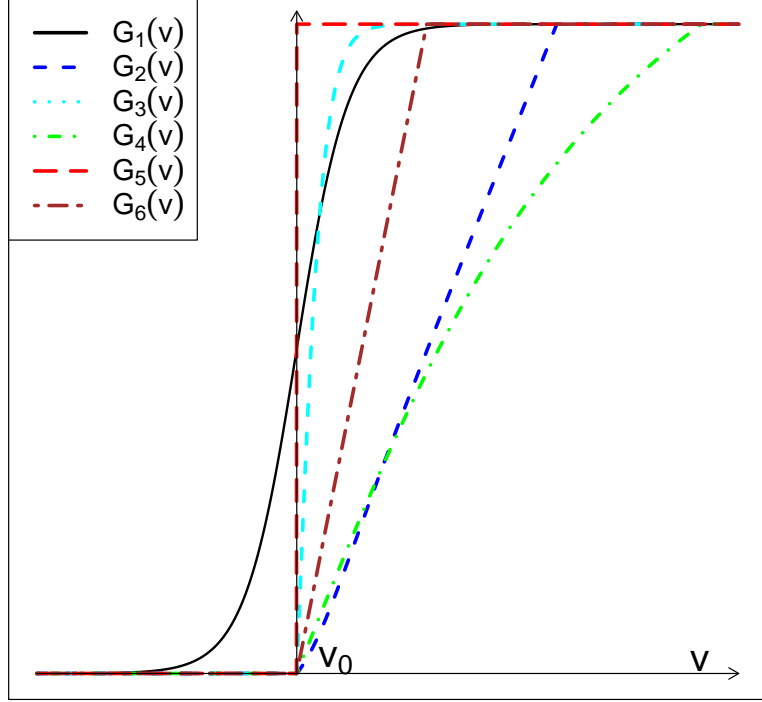


Figure 2: Five different activation functions.

**Remark 5.**

1.) We observe that the choice of the sigmoid activation function  $G_1(v) = (1 + e^{-\theta(v-v_0)})^{-1}$  is widely preferred in the literature for its bounded nature, without condition.

2.) Another reason is the fact that it is also suitable when the  $V_n^{(i)}$ 's are binary, that is, they may take values 0,1, where 0 represents a non active neuron at time  $n$  and 1 represents an

active neuron at time  $n$ . In this case,  $G_1\left(W_{i0} + \sum_{j=1}^N W_{ij}V_n^{(j)} + \eta_n^{(i)}\right) = Pr\left(V_{n+1}^{(i)} = 1\right)$  would represent the probability that there is an activity on neuron at position  $\xi_i$  at time  $n + 1$ .

3.) A third reason which will be important in our situation is that it has an inverse that can be written in close form, unlike many other activation functions sometime used in the artificial neural networks (see for instance [Kwessi and Edwards \(2020\)](#)) making it easy to generate random numbers from. The other functions would require the use of numerical inversion methods such as the bisection method, or the secant method, or the Newton-Raphson method, all which are computationally intensive, see for instance Chapter 4 in [Devroye \(2010\)](#).

**4.2 Connection intensity function**

There are varieties of connection intensities functions (or kernel) that one can choose from. This include a Gaussian kernel  $K(x, y) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2\sigma^2}\|x-y\|^2}$  introduced above. One could also consider also the Laplacian kernel defined as  $K(x, y) = \frac{1}{2\sigma}e^{-\frac{1}{\sigma}\|x-y\|}$ , the hyperbolic tangent kernel  $K(x, y) = \tanh(x - y)$ .

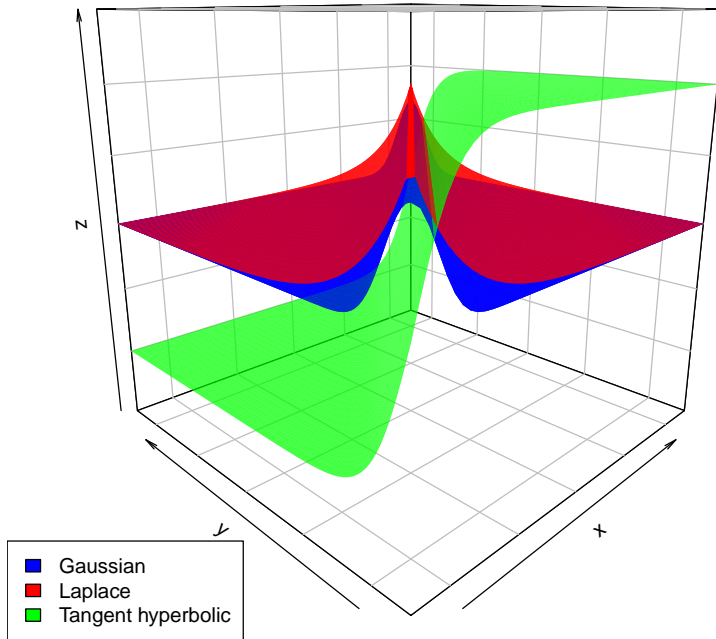


Figure 3: An illustration of Gaussian, Laplacian, and Tangent hyperbolic kernels.

## 5 Simulations

In each of the simulations below, we will use the function  $V(\xi) = \sin(\xi)$  as if it were the true solution, just to evaluate  $V(y_j)$  for  $j = 1, 2, \dots, n$ . Using the algorithm above, will then compare the estimates of  $V(\xi)$  obtained using our estimator  $V$  with  $V$  unknown (red curves) and  $V$  known (blue curves), using various kernel functions  $K(x, y)$  and various external stimulus functions  $S(x)$ . In all simulations below, we selected  $n = m = 100$ . We will use Gaussian, Laplacian, and hyperbolic tangent kernels with a sigmoid function  $G$ . The value of  $\sigma$  is set at 1 for the Gaussian and Laplacian kernels.

### 5.1 Simulation 1: constant external stimulus

In this first simulation, we illustrate the algorithm above by selection a constant intensity of external stimulus arriving at at point  $\xi$ . To check if the algorithm is correct, we will select  $S(\xi) \equiv 1$ , with true function  $V(\xi) = \sin(\xi)$ , see Figure 4–6 below:

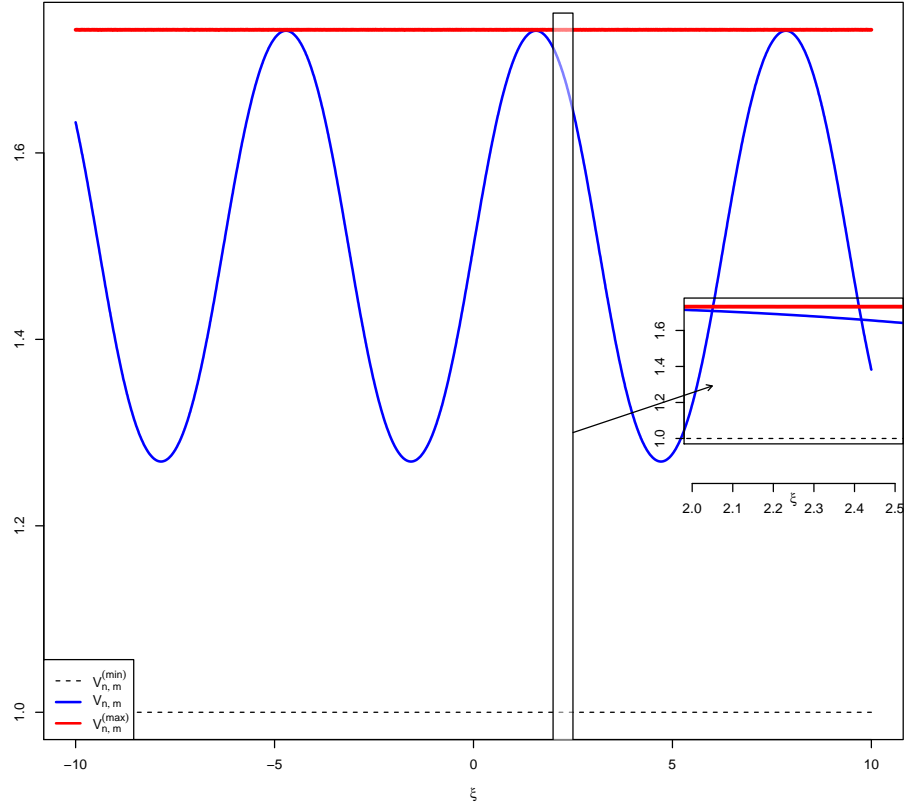


Figure 4: The dotted line represents the input  $S(\xi) = 1$ , which from above is similar to the  $V_{n,m}^{min}$ . The kernel is a Gaussian. The maximum estimator  $V_{n,m}^{min}$  (red) has the same patterns as the external stimulus and the estimator  $V_{n,m}$  (blue) takes the form of the true sine function.

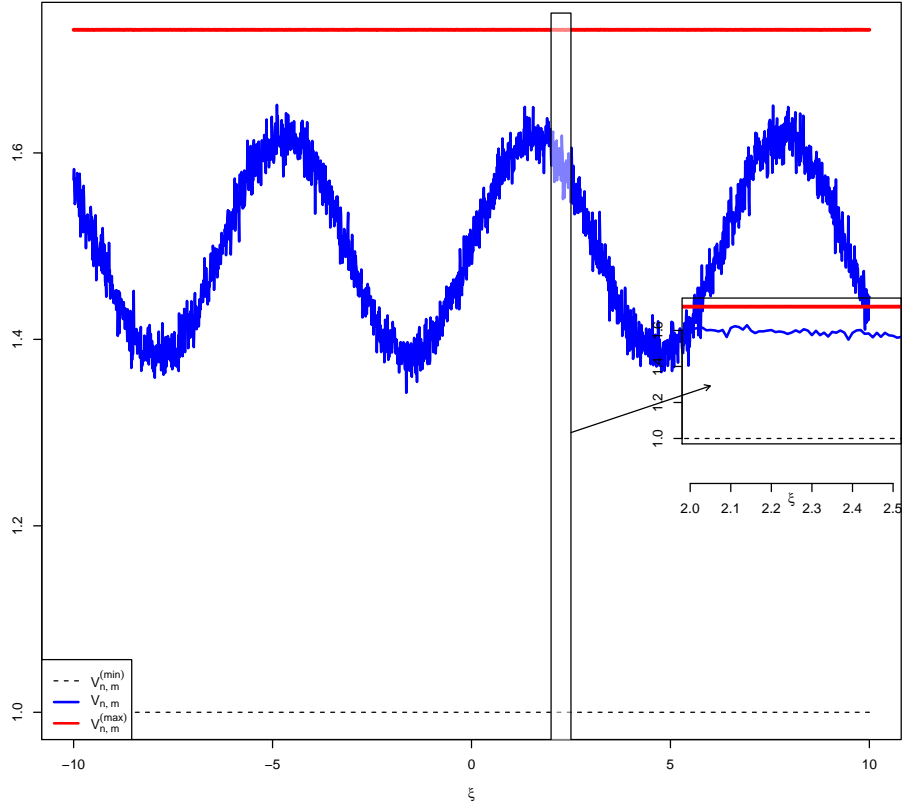


Figure 5: The dotted line still represents the input  $S(\xi) = 1$ , which from above is similar to the  $V_{n,m}^{min}$ . The kernel is Laplacian. The maximum estimator  $V_{n,m}^{min}$  (red) has the same patterns as the external stimulus and the estimator  $V_{n,m}$  (blue) takes the form of the true sine function, but noisier at low  $n, m$  and progressively smoother at high values of  $n, m$ .

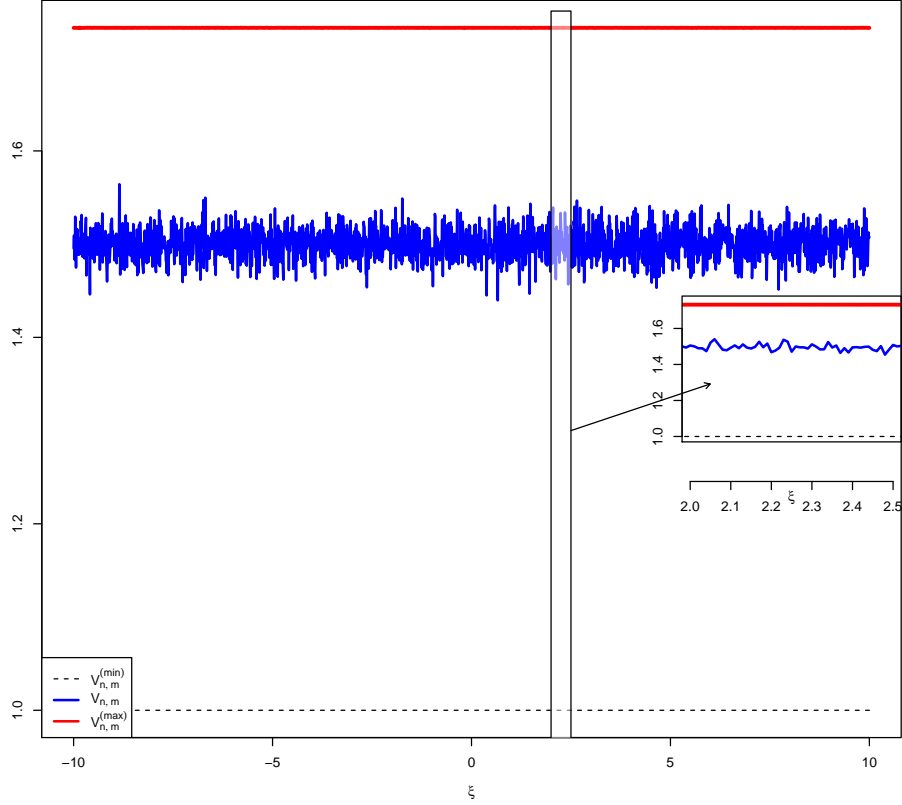


Figure 6: The dotted line represents the input  $S(\xi) = 1$ , which from above is similar to the  $V_{n,m}^{min}$ . The kernel is a Hyperbolic tangent. The maximum estimator  $V_{n,m}^{min}$  (red) has the same patterns as the external stimulus and the estimator  $V_{n,m}$  (blue) does take the form of the initial external input, unable to replicate the form of the true since function, even at high values of  $n, m$ .

## 5.2 Simulation 2: logarithm external stimulus

In this second simulation, we illustrate the algorithm above by selection a constant intensity of external stimulus arriving at at point  $\xi$ . To check if the algorithm is correct, we will select  $S(\xi) = \ln(1 + \xi^2)$ , with true function  $V(\xi) = \sin(\xi)$ , see Figure 7–9 below:

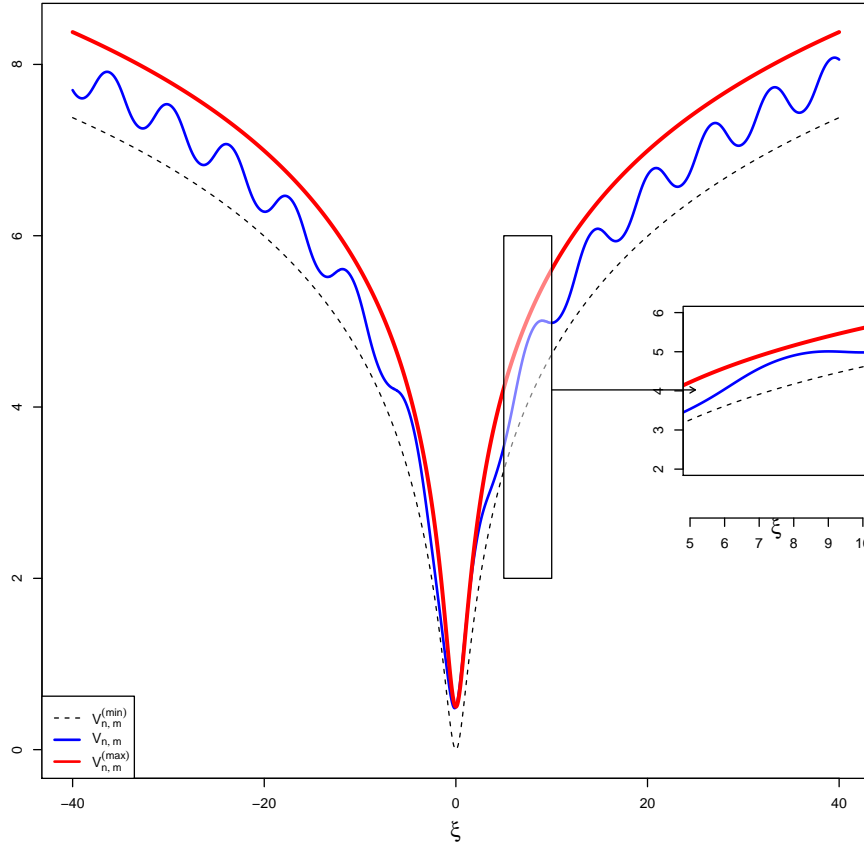


Figure 7: The dotted line represents the input  $S(\xi) = \ln(1 + x^2)$ , which from above is similar to the  $V_{n,m}^{min}$ . The kernel is Gaussian. The maximum estimator  $V_{n,m}^{min}$  (red) has the same patterns as the external stimulus and the estimator  $V_{n,m}$  (blue) a distorted version of the original since function, distortion that is increased around 0, which is caused by the presence of  $S(\xi)$  which in the neighborhood of 0, dominates the sine function which we know is close to 0 in a small neighborhood of 0. But as we get farther from zero, the influence of external stimulus wanes and the estimator starts to take the shape of the true sine function.

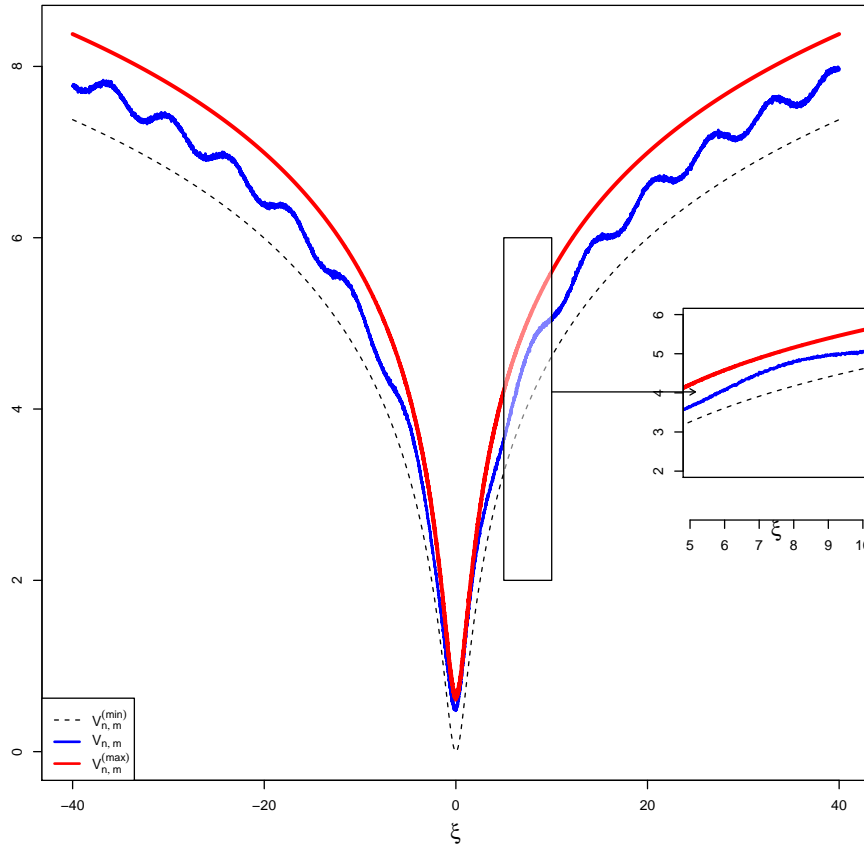


Figure 8: In this case, we use a Laplacian Kernel and we observe a similar pattern as above. However, the estimator is much noisier. There is also a noticeable phase difference between the estimations from a Gaussian kernel and a laplacian kernel.

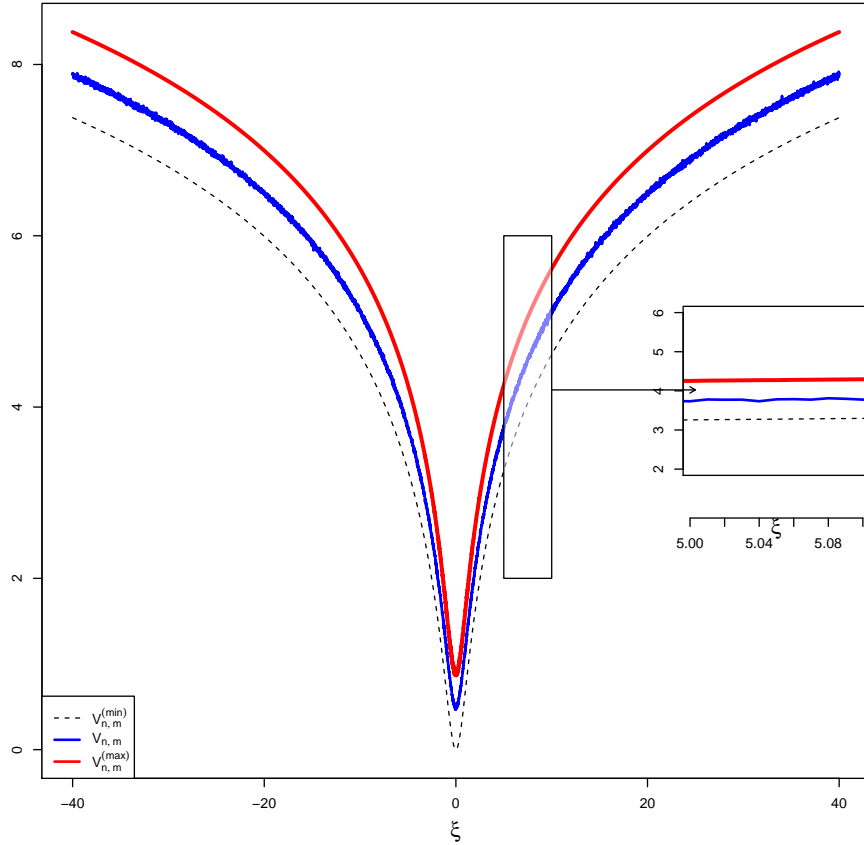


Figure 9: In this case, we used hyperbolic tangent kernel and clearly the sine pattern of the true function is never recovered. This suggests that the external input is overwhelming the noise, even after a close look within the interval  $[5.0, 5.1]$

### 5.3 Simulation 3: exponentially decaying external stimulus

In this third simulation, we illustrate the algorithm above by selection an intensity of external stimulus arriving at at point  $\xi$  as  $S(\xi) = e^{-\xi^2}$ , with true function  $V(\xi) = \sin(\xi)$ , see Figure 10–12 below:

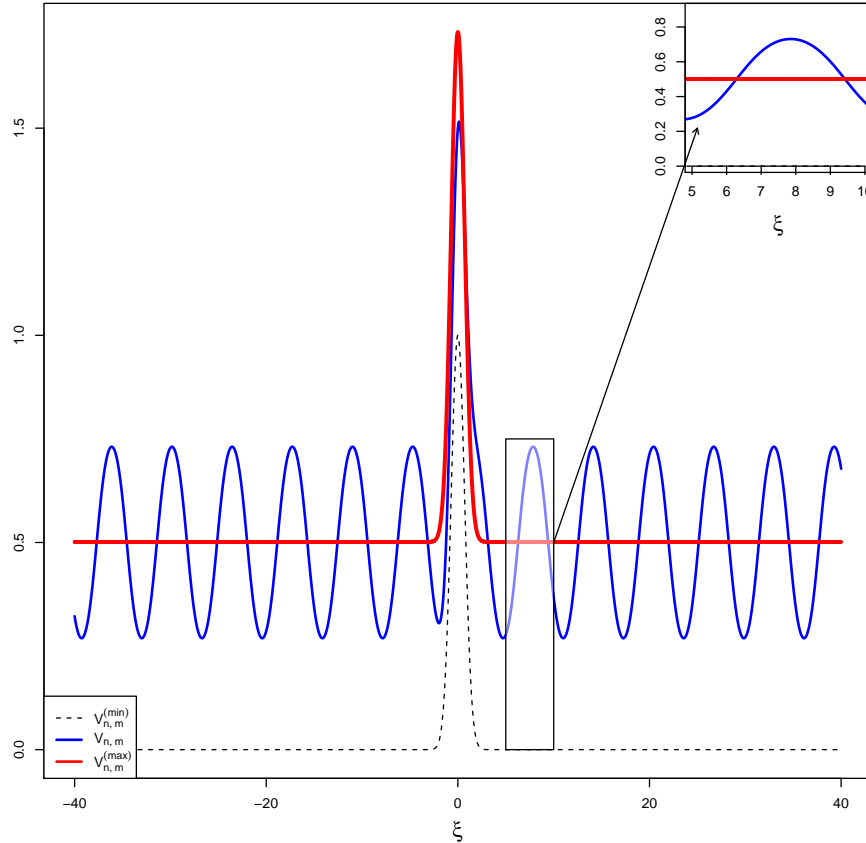


Figure 10: Clearly, with this external inputs, the situation is different from the above cases. In a small neighborhood of 0, we still have the external stimulus dominating, the noise and the estimates  $V_{n,m}$  (blue) remains between  $V_{n,m}^{max}$  and  $V_{n,m}^{min}$ . However, as we move farther away from 0,  $V_{n,m}^{max}$  takes the shape of  $S(\xi)$  but is oscillates between  $S(x)$  and  $V_{n,m}$ . This is explained but the fact that  $exp(-\xi^2)$  is trades with  $\sin(\xi)$  due to the periodic nature of the latter. The estimator on the other hand reproduces the expected pattern is this where a Gaussian kernel is used.

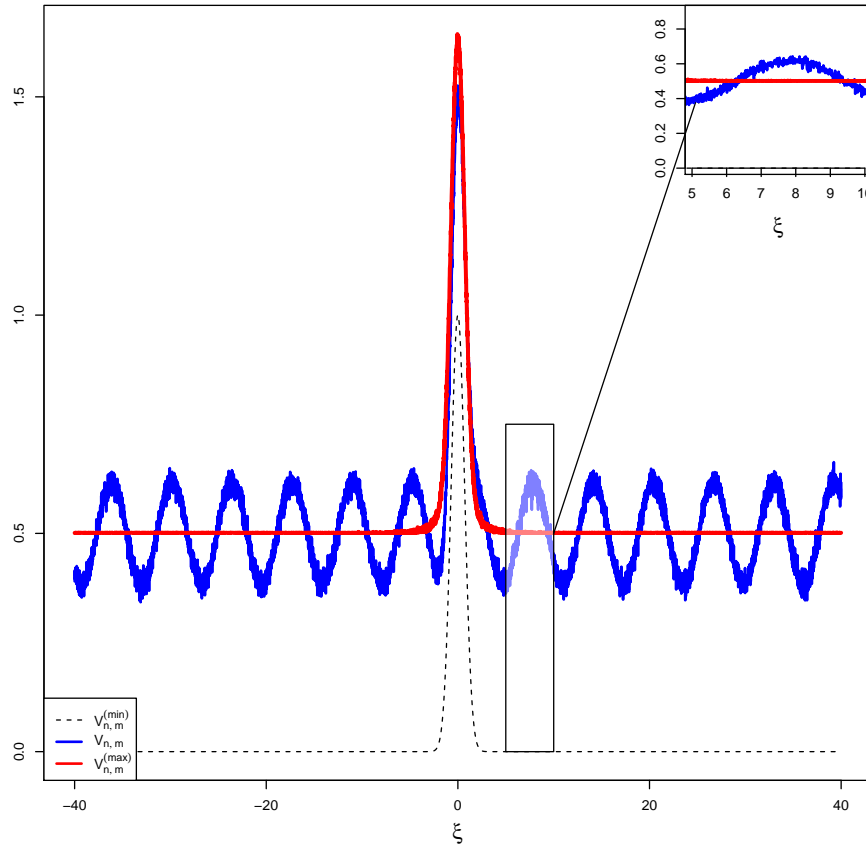


Figure 11: In this case the kernel is laplacian and the observations are the same as above. However, we observe much more noise in the estimator, together with a phase shift.

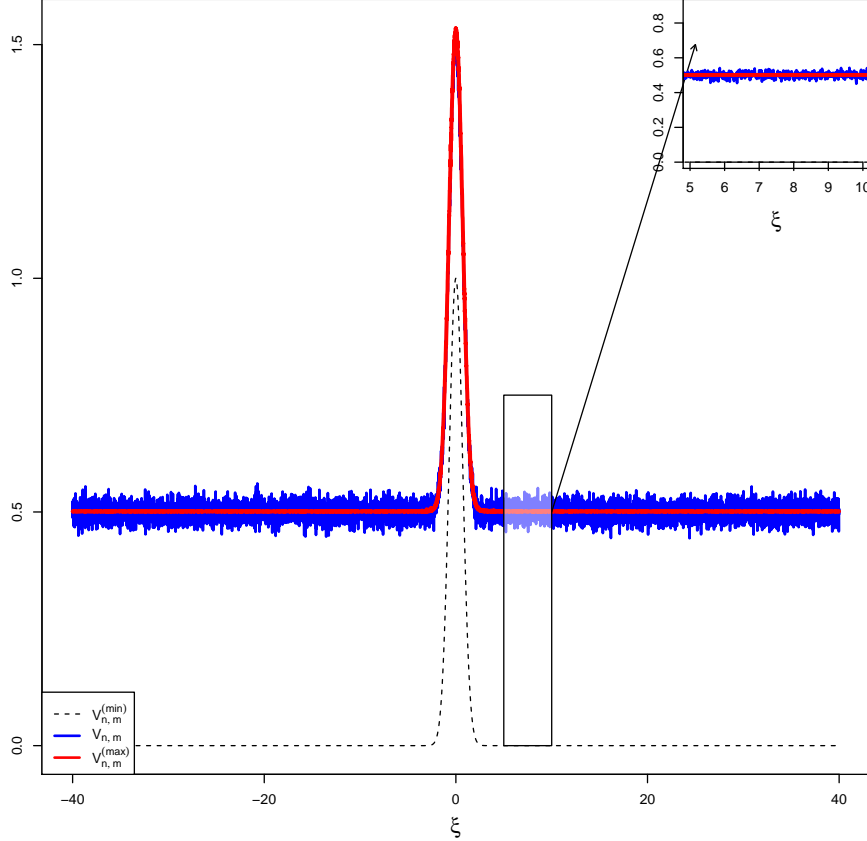


Figure 12: In this case, the expected pattern is not reproduced due to the external stimulus dominating and the estimator serving as noise.

#### 5.4 Simulation 4: Mexican hat true function

Beim and Hutt (2014) used a Gaussian Kernel to obtain an estimate of the nontrivial solution that had the form of a Mexican hat function in the space  $\Omega$ . Our method differs from theirs in two aspects: first, they assumed that  $K(x, y) = V_0(x) \cdot V_0(y)$  where  $V_0(x) = \int_{\Omega} K(x, y)G(V_0(y))dy$  which implies that  $1 = \int_{\Omega} V_0(y)G(V_0(y))dy$ . This would restrict us to only independent random variables  $X$  and  $Y$  with marginals  $V_0(x)$  and  $V_0(y)$ . We do not make such an assumption because there are many kernels (bivariate functions) that cannot be factored as the product of two marginals. Another key difference is that we do not make the assumption that  $G(V(y))$  has a power series about a certain state  $\overline{V(y)}$  so that  $G(V(y)) = \sum_{n=0}^{\infty} \frac{G^{(n)}(y)}{n!} [V(y) - \overline{V(y)}]^n$ . This assumption would obviously fail for the Heaviside, and Ramp functions. The main reason for the difference is that they were interested in second-order synaptic dynamics which is not the case here. In this section, we show that our method still yields a comparable estimate even without these assumptions. Indeed, in Beim and Hutt (2014), it was shown that the true solution obtained with a Gaussian kernel in the space has the form of a Mexican hat function. In this simulation, we will use our estimator to verify this fact, that is, we will set our external stimulus as a Gaussian distribution with mean zero and standard deviation 0.03 as in their case, and compare the estimates obtained from the use of a Gaussian, Laplacian, and Hyperbolic tangent kernels, see Figure 13 below.

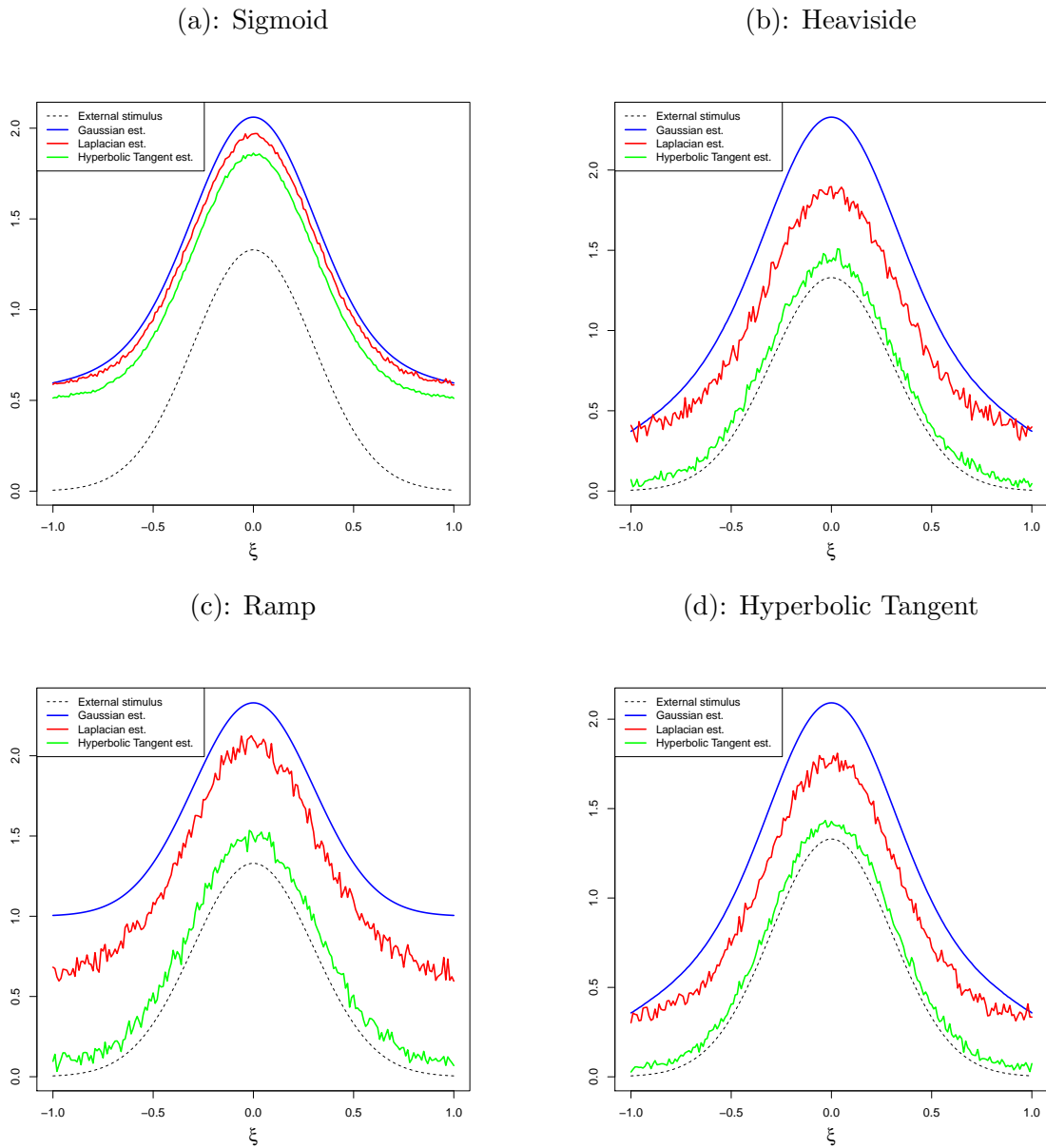


Figure 13: The curve represented are estimate of  $V(\xi)$  using the proposed estimator for different kernels, Gaussian (blue), Laplacian (red), and hyperbolic tangent (green) with a mean zero gaussian external stimulus with standard deviation 0.03. Clearly, all kernels yield the expected pattern. In (a), we used the sigmoid function  $G_1$ , in (b) we used the Heaviside function  $G_5$ , in (c), we used the Ramp function  $G_6$  with  $\theta = 3.5$ , and in (d), we used hyperbolic tangent inverse function  $G_3$ .

## Discussion

- 1.) The conclusion we can draw from the first simulation is that Gaussian and Laplacian kernels both fare well when the external input is constant. The latter produces noisier outputs at low resolution values  $n, m$  and becomes smoother at high resolution values  $n, m$ .
- 2.) The major takeaway from the second and third simulation is that the external stimulus can have a significant effect on the estimator, especially near boundary points where there is

a significant frequency change between the true value and the external stimulus. In reality, in practical applications the true value is not known, therefore, a careful choice on the external input is needed if one would like to obtain accurate estimations.

3.) The fourth simulations shows that estimates obtained using a Gaussian kernel will be smoother. As for the pulse rate function  $G$ , the sigmoid function  $G_1$  fares much better for all three kernels used in comparison to other functions. 4.) Ultimately, the point of the first three simulations is the hope to extend this type of estimator beyond solutions of DNF. In fact, equation (1.1) can be considered a linear Boltzman equation with stochastic kernel if instead of thinking of  $\xi$  as the position of a neuron, we think of it as the velocity of a particle and  $V(x, t)$  as the velocity distribution overtime. In this case, nontrivial solutions are now solutions of the nonlinear Markov operator  $Pg(v(\xi)) = v(\xi)$  where  $Pg(v(\xi)) := \int_{\Omega} K(\xi, y)g(v(y))dy$ , which by the Hille-Yosida theorem exist. The estimator proposed here provides in a sense another way of thinking about this problem using successive approximations, see Lasota and Mackey (1994).

5.) A drawback of the estimator is that it depends on a great first guest of the solutions at points  $y_i \in \Omega$ . However, if enough of these points are selected, one stands a great chance of obtaining a good approximation.

6.) An advantage of this estimator is that locally, it is a great point estimator of the value of  $V(\xi)$  for a given  $\xi \in \Omega$ , and given  $y_1, y_2, \dots, y_n \in \Omega$ . As we mentioned in Section 3, though we may not know the values of  $V(y_i), i = 1, \dots, n$  needed to evaluate  $V(\xi)$ , one way to go around the issue is to select them uniformly from a small interval of length  $\frac{1}{n}$ .

7.) We observe that the computational aspect of our algorithm depends on Monte Carlo Simulations. This is not the only way of efficiently achieve this. One may also choose to achieve this using methods such as sparse grids and Bayesian Monte Carlo with appropriate priors on the parameters, see for instance Rasmussen and Ghahramani (2003), Deisenroth et al. (2009), Gerstner and Griebel (1998), Xu and Liao (2020), Movaghar and Mohammadzadeh (2020).

8.) One other possible use of this estimator that it can help initialize a RNN algorithm or also in finding the phase space diagram in a discrete dynamical system with two different layers.

## 6 Conclusion

In this paper, we have proposed an estimator for nontrivial solutions of dynamic neural fields. The estimator proposed is shown to be consistent. Moreover, the estimator proposed exists within a small interval depending on the number of points selected in the domain  $\Omega$  where these nontrivial solutions are defined. The choice of the kernel, as in previous studies, is shown to be crucial in determining the accuracy of the estimates. We also showed that Gaussian kernels provide the best balance between accuracy, smoothness, and the number of points used. In the space domain, the estimates obtained are visually similar to those obtained using for instance Lotka-Volterra series as in Beim and Hutt (2014). The estimator proposed has the advantage that it is simple to implement and may serve as initial guess or initialization when for example using a recurrent neural networks to find nontrivial solutions in the time and space domain. This is particularly important in robotics and to a certain extend in neuroinformatics because it could potentially help with accuracy of movements of robots. In this paper, we also showed how the DNF can be extended to functional and complex analysis which could further extend theoretical properties of DNFs using techniques from these areas. The estimator proposed in this paper can be used to initialize a discrete dynamical system associated with the DNF. The present work could be useful for new insights into the connection between DNF and dynamical systems and overall contribute to the literature in these areas and in computational neuroscience.

## Bibliography

- S-I. Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27(2):77–87, 1977. ISSN 0340-1200. doi: 10.1007/BF00337259.
- P. G. Beim and Axel Hutt. Attractor and saddle node dynamics in heterogeneous neural fields. *EPJ Nonlinear Biomedical Physics, EDP Sciences*, 2, 2014. pp.4.10.1140/epjnbp17.hal-00987789.
- R. L. Beurle. Properties of a mass of cells capable of regenerating pulses. *Philosophical Transactions of the Royal Society London B*, 240:55–94, 1956.
- E. Bicho, P. Mallet, and G. Schöner. Target representation on an autonomous vehicle with low-level sensors. *The International Journal of Robotics Research*,, 19:424–447, 2000.
- E. Bicho, L. Louro, and W. Erlhagen. Integrating verbal and non-verbal communication in a dynamic neural field for human-robot interaction. *Frontiers in Neurorobotics*, 4(5):1–13, 2010.
- M. Camperi and X-J. Wang. A model of visuospatial short-term memory in prefrontal cortex: recurrent network and cellular bistability. *J. Comp. Neuroscience*, 4:383–405, 1998.
- M. P. Deisenroth, M. F. Huber, and U. D. Henebeck. Analytic moment-based gaussian process filtering. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, 2009. Montreal, Canada.
- L. Devroye. Complexity questions in non-uniform random variate generation. In *Proceedings of COMPSTAT'2010*, pages 3–18. Physica-Verlag/Springer, Heidelberg, 2010.
- N. Djitte and M. Sene. An iterative algorithm for approximating solutions of hammerstein integralequations. *Numerical Functional Analysis and Optimization*, 34(12):1299–1316, 2013.
- D. Durstewitz. *Advanced data analysis in neuroscience*. Bernstein Series in Computational Neuroscience. Springer, Cham, 2017.
- J. L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.
- W. Erlhagen and E. Bicho. The dynamics neural field approach to cognitive robotics. *J. Neural Eng.*, 3:R36–R54, 2006.
- W. Erlhagen and G. Schöner. Dynamic field theory of movement preparation. *Psychological Review*, 109(3):545–572, 2001.
- G. B. Ermentrout and J. D. Cowan. A mathematical theory of visual hallucination patterns. *Biological Cybernetics*, 34:137–150, 1979.
- T. Gerstner and M. Griebel. Numerical integration using sparse grids. *Numerical algorithms*, 18(3-4):209, 1998.
- R. E. Green and S. G. Krantz. *Function theory of one complex variable*. Pure and Applied Mathematics (New York). John Wiley & Sons, Inc., New York, 1997. ISBN 0-471-80468-1. A Wiley-Interscience Publication.
- A. Hammerstein. Nichtlineare integralgleichungen nebst anwendungen. *Acta Math*, 54:117–176, 1930.

- E. Kwessi and L. Edwards. Artificial neural networks with a signed-rank objective function and applications. *Communication in Statistics–Simulation and Computation*, 2020. doi: 10.1080/03610918.2020.1714659.
- E. Kwessi, S. Elaydi, B. Dennis, and G. Livadiotis. Nearly exact discretization of single species population models. *Natural Resource Modeling*, 2018. doi: 10.1111/nrm.12167.
- A. Lasota and M. C. Mackey. *Chaos, fractals, and noise*, volume 97 of *Applied Mathematical Sciences*. Springer-Verlag, New York, second edition, 1994. ISBN 0-387-94049-9. doi: 10.1007/978-1-4612-4286-4. Stochastic aspects of dynamics.
- M. Movaghar and S. Mohammadzadeh. Bayesian monte carlo approach for developing stochastic railway track degradation model using expert-based priors. *Structure and Infrastructure Engineering*, pages 1–22, 2020.
- P. L. Nunez Nunez and R. Srinivasan. *Electric Fields of the Brain: The Neurophysics of EEG*. Oxford University Press, second edition, 2006.
- C. E. Rasmussen and Z. Ghahramani. Advances in neural information processing systems. *Bayesian monte Carlo Journal*, pages 505–512, 2003.
- W. Rudin. *Real and Complex Analysis*. McGraw-Hill, 1987.
- P. Tass. Cortical pattern formation during visual hallucinations. *Journal of Biological Physics*, 21:177–210, 1995.
- R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computations*, 1:256–263, 1990.
- H. R Wilson and J. D Cowan. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysics Journal*, 12:1–24, 1972.
- Z. Xu and Q. Liao. Gaussian process based expected information gain computation for bayesian optimal design. *Entropy*, 22(2):258, 2020.