In the quest of efficient hardware (& soft.) implementations of dynamic neural fields: an experimental study on the influence of the kernel shape

Benoît Chappet de Vangel bdevangel@gmail.com
Jérémy Fix Jeremy.Fix@centralesupelec.fr

Tuesday, 26th July 2016
Session TP-1: Attractor and Oscillatory Neural Networks

Introduced by [Beurle(1956), Wilson Cowan(1972), Amari(1977)]

"**Tissue level models** that **describe the spatiotemporal evolution of coarse grained variables** such as synaptic or firing rate activity in populations of neurons."

S. Coombes (2006)

Equation

$$\tau \frac{\partial u}{dt}(x,t) = -u(x,t) + \int_y f(u(y,t))w(x,y) + I(x,t)$$

- $u(x,t)$: membrane potential at time t, position x
- $f(.)$: transfer function, membrane potential $\mapsto$ firing rate
- $w(.,.)$: weight function
- $I(x,t)$: input at time tt, position x
- τ : time constant

Visual attention

A dynamic neural field approach to the covert and overt deployment of spatial attention, Fix et al.(2010)

Dynamic self-organization

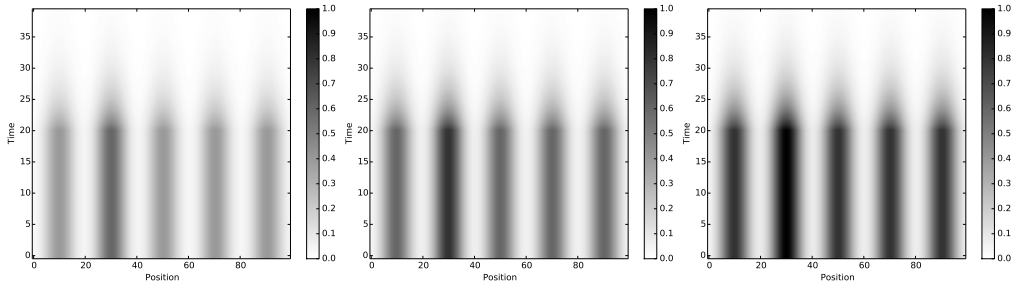
Dynamic formation of self-organizing maps, Fix (2014)

Robot control [Bicho et al, 2000], Object tracking [Spencer et al., 2012], Saccade generation [Taouali et al., 2015], . . .

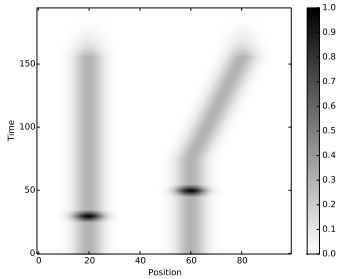
Which are the building blocks we will focus on ?

$$\tau \frac{\partial u}{dt}(x,t) = -u(x,t) + \int_y f(u(y,t))w(x,y) + I(x,t)$$

Competition



Working Memory



What is computationally expensive ?

Space and time discrete 1D DNF

Take a 1D, time/space discrete, homogeneous and isotropic DNF

$$u(x, t+1) = u(x, t) + \Delta u(x, t)$$
$$\frac{\tau}{\Delta t}\Delta u(x, t) = -u(x, t) + \sum_y f(u(y, t))w(x-y) + I(x, t)$$

What is computationally expensive ?

Space and time discrete 1D DNF

Take a 1D, time/space discrete, homogeneous and isotropic DNF

$$u(x, t+1) = u(x, t) + \Delta u(x, t)$$
$$\frac{\tau}{\Delta t}\Delta u(x, t) = -u(x, t) + \sum_y f(u(y, t))w(x-y) + I(x, t)$$

Effort in computational resources

In the worst case (fully connected), the convolution is :

- in software, at best in $O(N \log(N))$
- in hardware, very demanding in silicium surface

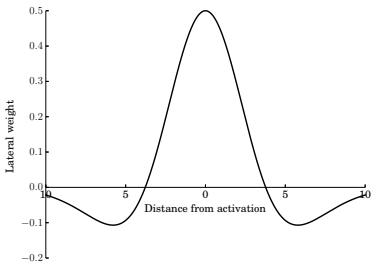
Method

Space and time discrete DNF

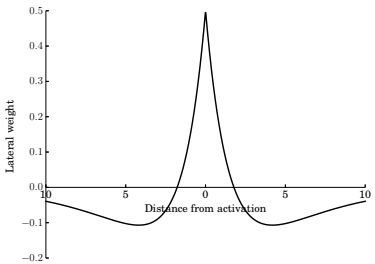
1D (N=100), time/space discrete, homogeneous and isotropic DNF

$$u(x, t+1) = u(x, t) + \Delta u(x, t)$$
$$\frac{\tau}{\Delta t}\Delta u(x, t) = -u(x, t) + \sum_y f(u(y, t))w(x-y) + I(x, t) + h$$

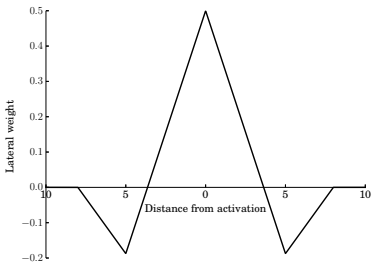
- f : Heaviside function
- w : 4 parameters $A_+, \sigma_+, A_-, \sigma_-$



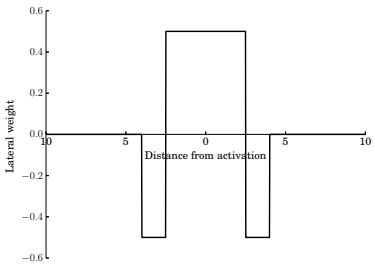
Diff. Of Gaussians



Diff. Of Exp.



Diff. Of Linear



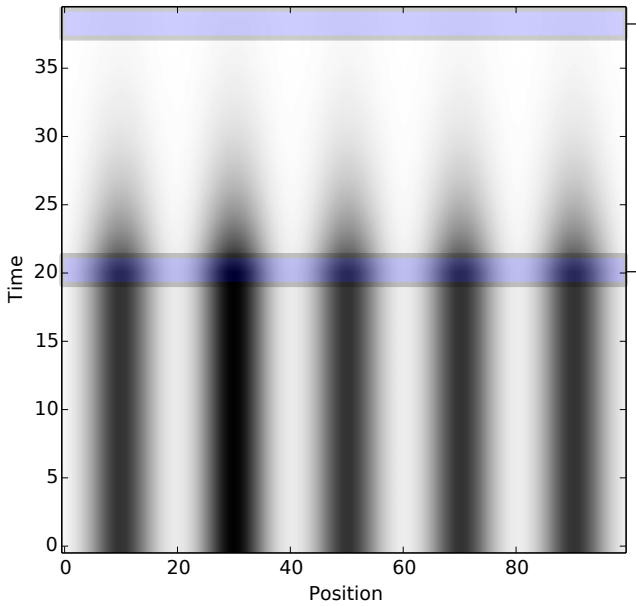
Diff. of Step

Existence and stability of pulses guaranteed for these kernels

Quantifying the quality of a Dynamic Neural Field.

A loss $J(\frac{\Delta t}{\tau}, h, A_+, \sigma_+, A_-, \sigma_-)$ is defined from the firing rates $f(u(x, t_i)) \in \{0, 1\}$ at specific times.

Competition scenario

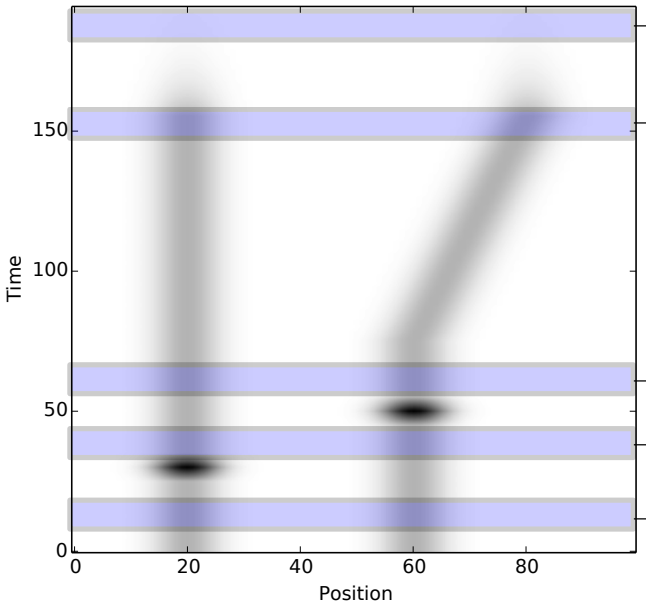


$\sum_x f(u(x,T))$

$\sum_{x,|x-x_0|\leq\sigma}(1 - f(u(x, \frac{T}{2})) + \sum_{x,|x-x_0|>\sigma} f(u(x, \frac{T}{2}))$

A loss $J(\frac{\Delta t}{\tau}, h, A_+, \sigma_+, A_-, \sigma_-)$ is defined from the firing rates $f(u(x, t_i)) \in \{0, 1\}$ at specific times.

Working memory scenario



$\sum_x f(u(x))$

$\sum_{x \in |x-x_1| \leq \sigma \cup |x-x_2| \leq \sigma} (1 - f(u(x))) + \sum_{x \notin |x-x_1| \leq \sigma \cup |x-x_2| \leq \sigma} f(u(x))$

$\sum_{x \in |x-x_1| \leq \sigma \cup |x-x_2| \leq \sigma} (1 - f(u(x))) + \sum_{x \notin |x-x_1| \leq \sigma \cup |x-x_2| \leq \sigma} f(u(x))$

$\sum_{x, |x-x_1| \leq \sigma} (1 - f(u(x)) + \sum_{x, |x-x_1| > \sigma} f(u(x))$

$\sum_x f(u(x))$

We solve $\min J(\frac{\Delta t}{\tau}, h, A_+, \sigma_+, A_-, \sigma_-)$ with Standard PSO 2006 [Clerc(2012)]. See paper for the bounds on the parameters.

$$\begin{aligned} p_i(t+1) &= p_i(t) + v_i(t+1) \\ v_i(t+1) &= wv_i(t) + c_1 r_1 (p_i^l(t) - p_i(t)) + c_2 r_2 (p_i^g(t) - p_i(t)) \end{aligned}$$

$w = \frac{1}{2\log 2}$ and $c_1 = c_2 = \frac{1}{2} + \log 2$. Random informants, $K = 3$

Setting for competition

Parameter	value
Number of particles	20
Number of epochs	100
Number of FE	2000
Number of trials	1000
Optimisation time	1 min.

Setting for Working memory

Parameter	value
Number of particles	200
Number of epochs	1000
Number of FE	200000
Number of trials	1000
Optimisation time	1 h.

Results

Competition scenario : Fitnesses and success rates

Trials are considered successfull if $J(\dots) = 0$.

Kernel	Success rate
Diff. Of Gauss.	98.9%
Diff. Of Expon.	84.1 %
Diff. of Linears	95.2 %
Diff. of Steps	95.2 %

- $\frac{\Delta t}{\tau}$ lower bounded
- $h \in [-.5, 0]$ monomodal, $\bar{h} = -.15$
- $A_+ \simeq A_-$
- $\sigma_+ \simeq \frac{\sigma_-}{2}$, σ_- lower bounded

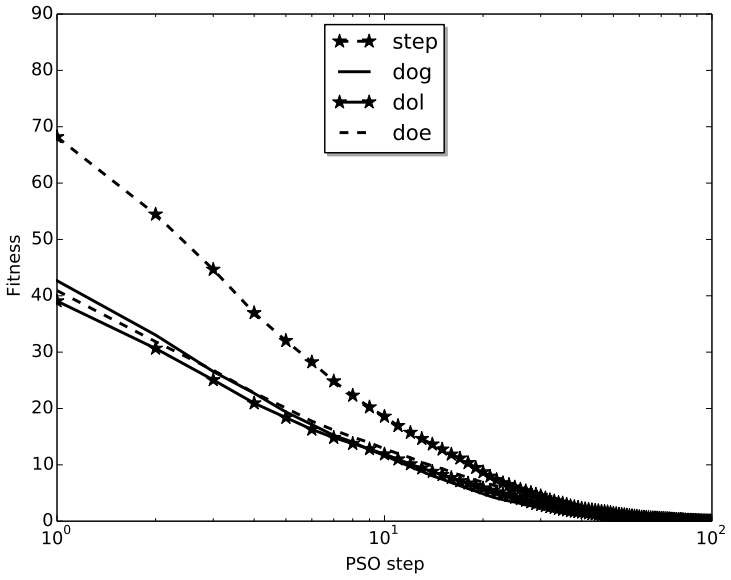
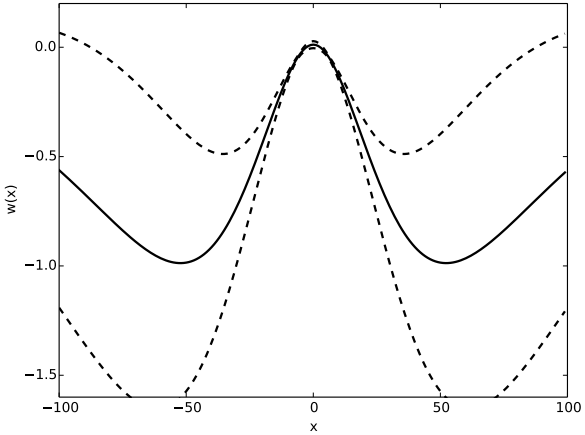


Figure: Mean value of the cost function for every kernel

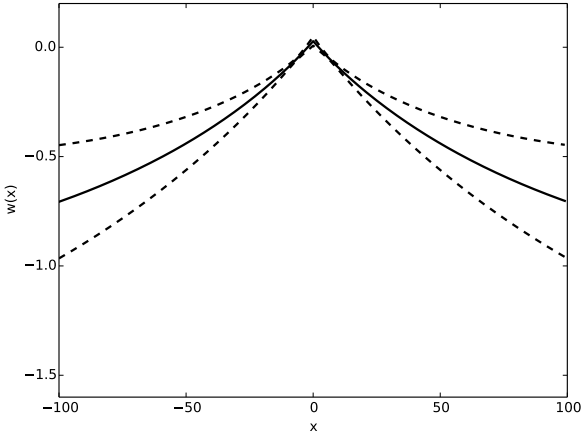
Parameters and their histograms :

`http://jeremy.fix.free.fr/Simulations/ijcnn_2016.html`

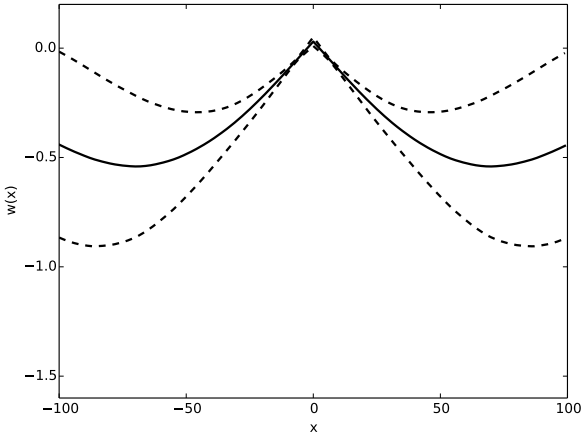
Results
Competition scenario : Weights shape



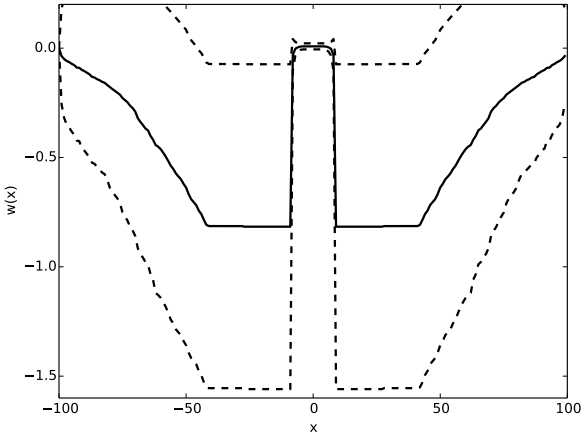
Diff. of Gauss.



Diff. of Exp.



Diff. of Linears



Diff. of Steps

Parameters and their histograms :

`http://jeremy.fix.free.fr/Simulations/ijcnn_2016.html`

Working memory scenario : Fitnesses and success rates

Trials are considered successfull if $J(\dots) \leq 8$ (set from visual inspection of the trials ordered by increasing fitness)

Kernel	Success rate
Diff. Of Gauss.	100%
Diff. Of Exp.	46 %
Diff. of Linears	100 %
Diff. of Steps	100 %

- $\frac{\Delta t}{\tau} \in [0.15, 0.3]$, larger than comp.
- $h \simeq -0.5$
- small A_+, local exc/inh
- $A_- \simeq 0.7A_+$ or $0.9A_+$

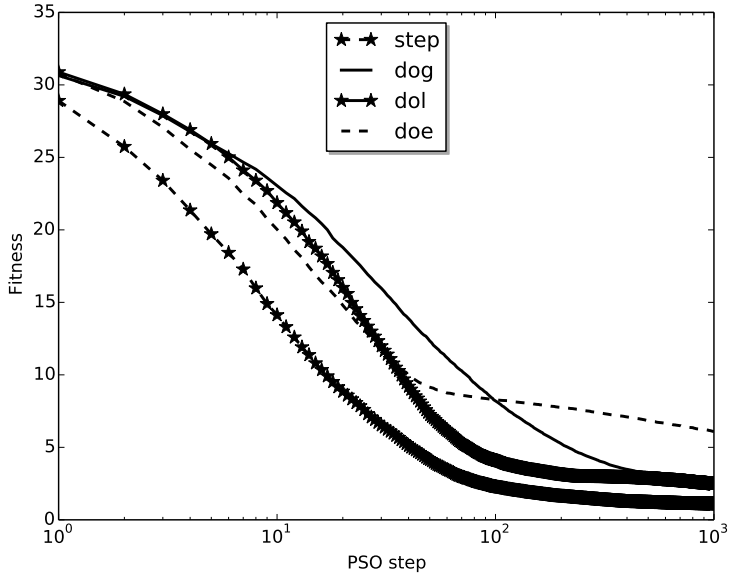
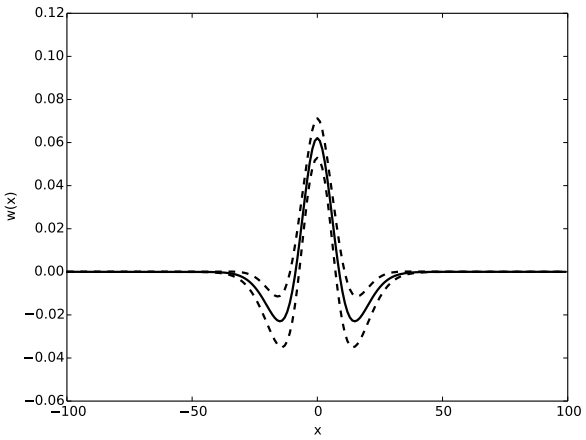


Figure: Mean value of the cost function for every kernel

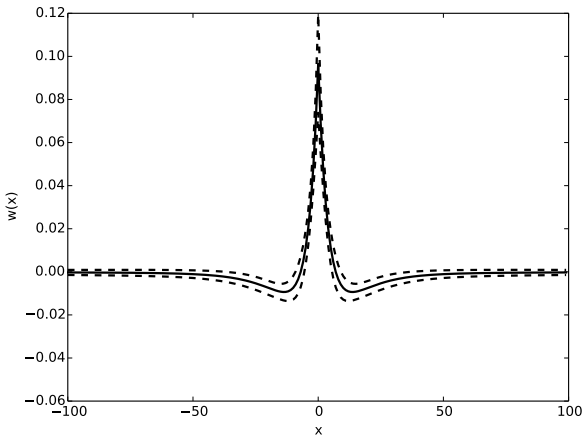
Parameters and their histograms :

`http://jeremy.fix.free.fr/Simulations/ijcnn_2016.html`

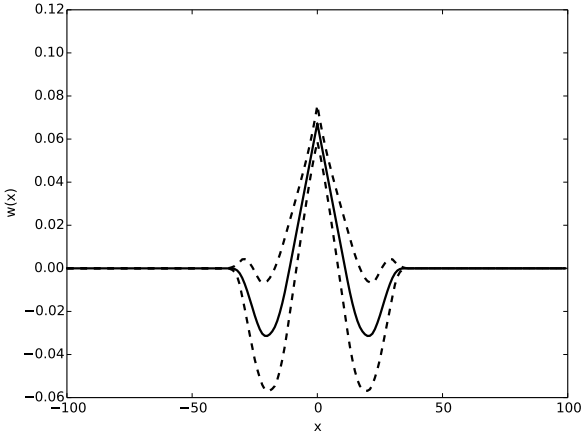
Results
Working memory scenario : Weights shape



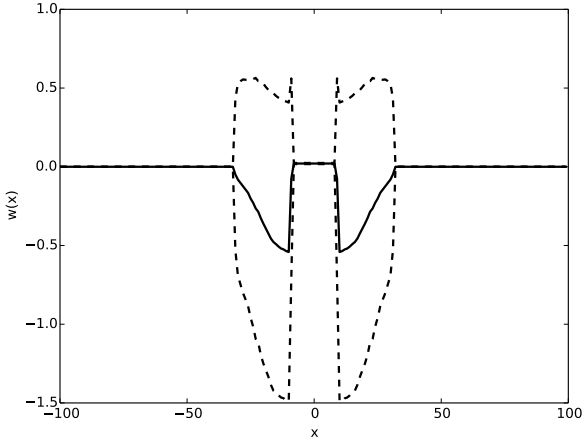
Diff. of Gauss.



Diff. of Exp.



Diff. of Linears



Diff. of Steps

Parameters and their histograms :

`http://jeremy.fix.free.fr/Simulations/ijcnn_2016.html`

Conclusion

Conclusion

$$u(x, t+1) = u(x, t) + \Delta u(x, t)$$
$$\frac{\tau}{\Delta t}\Delta u(x, t) = -u(x, t) + \sum_{y} f(u(y, t))w(x - y) + I(x, t)$$

Summary

- 1D Dynamic neural field
- with a DOG, DOE, DOL or DOS kernel
- successfully optimized with PSO
- for competition (3 scenarii) and working memory

Conclusion

$$u(x, t+1) = u(x, t) + \Delta u(x, t)$$
$$\frac{\tau}{\Delta t}\Delta u(x, t) = -u(x, t) + \sum_y f(u(y, t))w(x-y) + I(x, t)$$

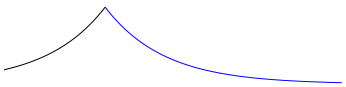
Is it more efficient to simulate/implement ?

In the worst case, i.e. fully connected (e.g. competition)

Kernel	Soft. Complexity	Hard. complexity
Diff of Gauss.	$O(N\log(N))$	$O(N^2)$
Diff of Exp.	$O(N)$	$O(N)$
Diff of Linears	$O(N)$	$O(N)$
Diff of Steps	$O(N)$	$O(N)$

$$L(x) = \sum_y f(u(y,t))w(x-y) = \sum_{y<x} \cdots + \sum_{y \geq x} \cdots = L_l(x) + L_r(x)$$

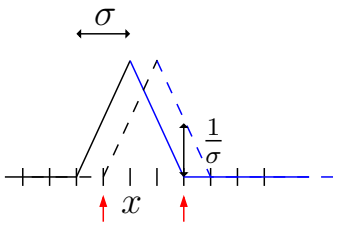
Difference of Exponentials



$$L_l(x+1) = L_l(x)e^{\frac{1}{\sigma^2}} + f(u(x+1))$$
$$L_r(x+1) = (L_r(x) - f(u(x)))e^{\frac{1}{\sigma^2}}$$

$\rightarrow \rightarrow O(N)$ operations

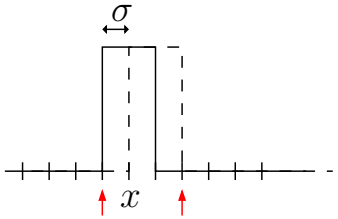
Difference of Linears



$$L_r(x+1) = (L_r(x) - f(u(x)))(1+\tfrac{1}{\sigma}) + f(u(x+\sigma))\tfrac{1}{\sigma}$$

$\rightarrow \rightarrow O(N)$ operations

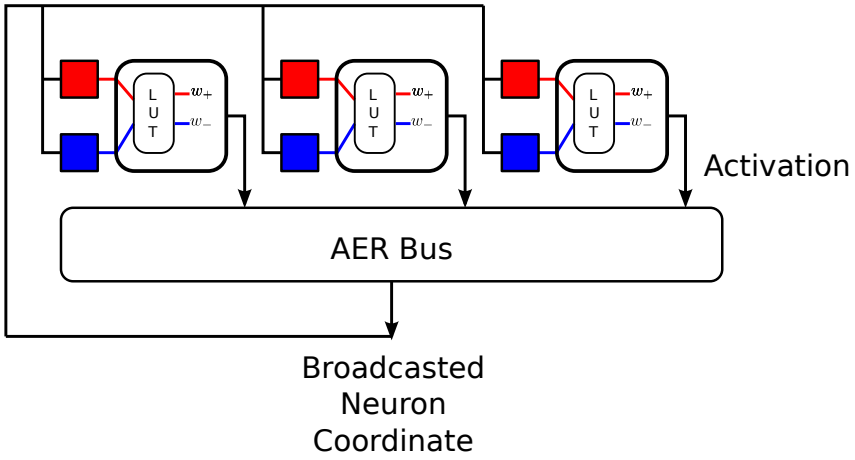
Difference of steps



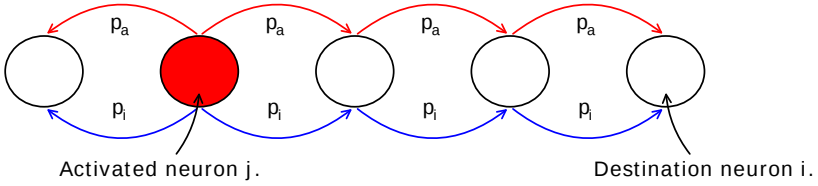
$$L(x+1) = L(x) + f(u(x+1+\sigma)) - f(u(x-\sigma))$$

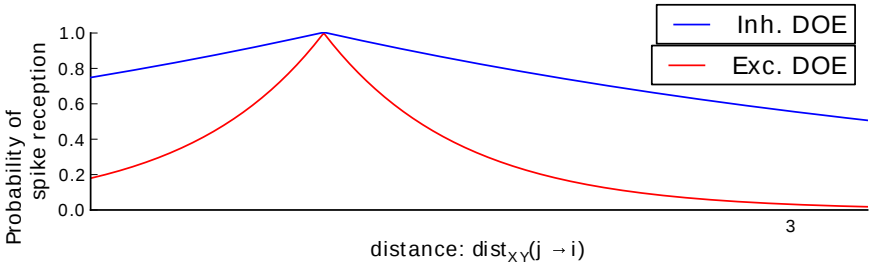
$\rightarrow \rightarrow O(N)$ operations

Difference of gaussians : Lot of space for the Look Up Tables



Difference of exponentials [De Vangel, 2015]





Difference of steps : Just 4 comparators required

Thank you for your attention

Benoît Chappet de Vangel bdevangel@gmail.com
Jérémy Fix Jeremy.Fix@centralesupelec.fr

Additional resources at :
http://jeremy.fix.free.fr/Simulations/ijcnn_2016.html

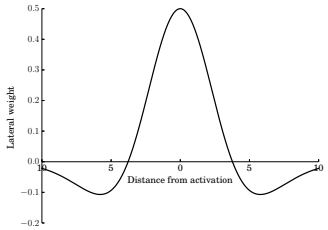
Additional references:

Benoit de Vangel et al. (2015) **Randomly spiking dynamic neural fields.**

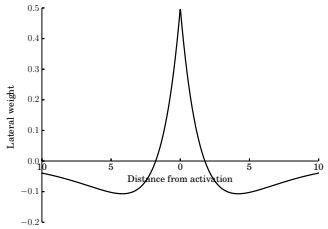
Jeremy Fix (2013) **Template based black box optimization of dynamic neural fields.**

Dynamic neural field definition

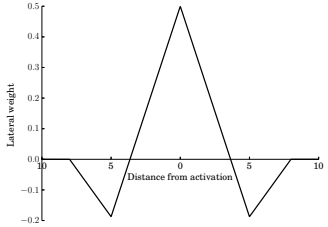
Parametrization



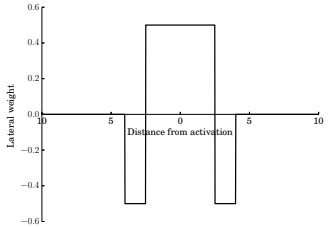
$$w(x) = A_e \exp^{\frac{-x^2}{2\sigma_e^2}} - A_i \exp^{\frac{-x^2}{2\sigma_i^2}}$$



$$w(x) = A_e \exp^{\frac{-4|x|}{\sigma_e^2}} - A_i \exp^{\frac{-4|x|}{\sigma_i^2}}$$



$$w(x) = A_e \left[1 - \frac{|x|}{2\sigma_e}\right]^+ - A_i \left[1 - \frac{|x|}{2\sigma_i}\right]^+$$



$$w(x) = A_e \mathcal{H}(|x| - \sigma_e) - A_i \mathcal{H}(|x| - \sigma_i)$$