ESANN'2008 proceedings, European Symposium on
Artificial Neural Networks - Advances in Computational Intelligence and Learning
Bruges (Belgium), 23-25 April 2008, d-side publi., ISBN 2-930307-08-0.

Computationally Efficient Neural Field Dynamics

Alexander Gepperth, Jannik Fritsch and Christian Goerick

Honda Research Institute Europe GmbH
Carl-Legien-Str.30, 63073 Offenbach am Main - Germany

Abstract. We propose a modification of the dynamic neural field model of Amari [1], aiming at reducing the simulation effort by employing space- and frequency representations of the dynamic state in parallel. Additionally, we show how the correct treatment of boundary conditions (wrap-around, zero-padding) can be ensured, which is of particular importance for, e.g., vision processing. We present theoretical predictions as well as measurements of the performance differences between original and modified dynamics. In addition, we show analytically that key properties of the original model are retained by the modified version. This allows us to deduce simple conditions for the applicability and the computational advantage of the proposed model in any given application scenario.

1 Introduction

Dynamic neural fields [2, 1, 3] are a standard tool applied in cognitive modeling [2, 4, 5], biomimetic vision [6, 7] and "intelligent" real-world systems [8]. However, the computational effort required for this technique was always a factor preventing its wide-spread use. This contribution presents a method for reducing the computational effort considerably if certain conditions can be met. The equation for neural field dynamics as proposed by Amari [1] (referred to simply as *Amari dynamics* - AD) reads

$$\tau\dot{a}(\vec{x},t) = -a(\vec{x},t) + i(\vec{x},t) + F(\vec{x}) * f[a(\vec{x},t)] + h \tag{1}$$

where $a\left(\vec{x,t}\right)$ is the state of the neural field, $i\left(\vec{x},\mathrm{t}\right)$ is a function representing the input to the neural field, $f\left(.\right)$ is a bounded monotonic, usually nonlinear function with values between 0.0 and 1.0 called "transfer function", $F\left(\vec{x}\right)$ stands for a function called the "interaction kernel", τ specifies the time scale, and h is a constant specifying the global excitation or inhibition of the field. The operator "*" indicates a spatial *convolution.* In order to solve the equation numerically, the variables $\vec{x}$,t are discretized using step sizes $\Delta\vec{x}$, Δt. By doing this, the convolution is transformed into a discrete operation. Usually, the function $F\left(\vec{x}\right)$ is concentrated in a small region around the origin and can thus be expressed as a discretized, finite *convolution filter.* Discrete convolutions are computationally costly in the space representation (SR), but it is known [9] that discrete convolutions can be computed efficiently in the Fourier representation (FR).

As a consequence, it would be desirable to perform *all* computations of one AD iteration in the FR. In order for this to work, all other quantities appearing on the right-hand side of eqn. (1) must be transformed to or maintained in the FR. The problems with this approach in the context of simulating AD are twofold: first of all, a convolution in the FR implicitly enforces cyclic boundary conditions. Secondly, the application of the transfer function is, except for linear functions, unfeasible in the FR. The issue of boundary conditions is important in many real-world applications such as, e.g., image processing [7, 6]. In the following section, we will propose a modification to AD which allows to reduce the computational cost strongly by exploiting the advantages of the FR while correctly applying arbitrary transfer functions and boundary conditions.

2 Methods

We propose to perform the numerical solution of eqn. (1) entirely in the FR which is feasible since almost all simulation steps are convolutions, scalar multiplications or sums. An exception is the application of the transfer function and possibly boundary conditions. The key idea is to apply these to a size-reduced SR of the neural field instead to the field itself.[1] For this purpose, eqn. (1) must be adapted; the proposed *modified Amari dynamics* (MAD) now reads as follows, where $S(\vec{x})$ is a smoothing operator designed to remove high spatial frequencies from the field:

$$\tau\dot{a}(\vec{x},t) = -a(\vec{x},t) + i(\vec{x},t) + F(\vec{x}) * f[S(\vec{x}) * a(\vec{x},t)] + h. \tag{2}$$

The smoothing operation makes it possible to downsample the neural field to a smaller size in the FR without introducing artefacts. Since the field is represented in the FR, smoothing is simply effected by cutting out a central part of the neural field. From the reduced-size neural field in the FR, a reduced-size field in the SR can be obtained by inverse Fourier transform which can be performed very efficiently due to the size reduction. After applying the transfer function to the field, the result is transformed back into a size-reduced FR, and the convolution with the interaction kernel can be performed. In order for this to work, the parameters in the interaction kernel

$$f(\vec{x}) = G_{\sigma_E}(\vec{x}) - G_{\sigma_I}(\vec{x}), \quad G_{\sigma}(\vec{x}) = \frac{1}{2\pi\sigma} e^{-\{\frac{\vec{x}^2}{2\sigma^2}\}}, \quad \sigma_E < \sigma_I. \tag{3}$$

must be chosen such that the kernel is band-limited (see also section 3.3).

In the case of zero-padding boundary conditions, the convolution results need to be adapted, as shall be explained later in this section. Subsequently, the neural field can be upsampled to the original size. The upsampling procedure in the FR is a simple copying operation and thus very efficient. The upsampled result can now be used to perform the whole iteration step entirely in the FR. The updated state of the neural map remains in the FR to be used in the next iteration step. An iteration is schematically shown in Fig. 1 (left).

Zero-padding boundary conditions are frequently used when performing discrete convolutions: they imply that, whenever a convolution filter element is

[1] Otherwise, the proposed method would not be more efficient than the alternative: to solve in the SR and only to transform to the FR for computing the convolution term.

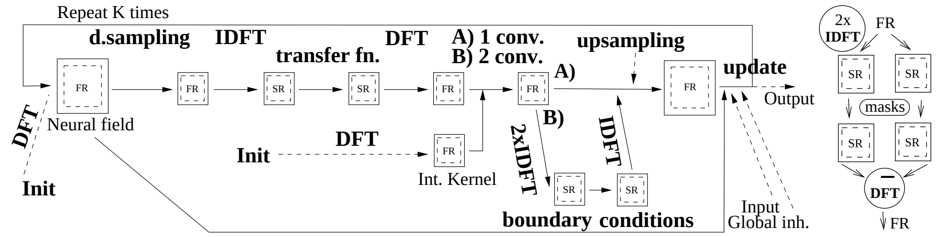

Fig. 1: Flowcharts of the MAD simulation algorithm. Left: One complete iteration, distinguishing between periodic (case **A**) and zero-padding (case **B**) boundary conditions. Solid/dashed boxes show the neural field either in the FR or the SR and indicate downsampled or original neural field size. The embedding of the neural field into a larger area (case **B**) is symbolized by dashed boxes. Right: details about case **B**. The two masks applied to the convolution results are obtained by filtering a zero-padded unity field, $a(\vec{x}, t) \equiv 1$, with the first and the second term in the interaction kernel of eqn.(3), and then inverting the results. The application of these masks counteracts the edge sensitivity of the kernel at the zero-padded boundaries.

applied outside the neural field dimensions, zero values are used instead of "missing" field entries (see, e.g., [9]). In order to apply these boundary conditions, certain operations must be performed at initialization of the field as well as at each iteration. First of all, the initial state of the neural field must be enlarged in each spatial dimension, initializing undefined regions to zero values ("zero-padding"). The enlargement can be done at initialization and must be, at each border, more than half of the size of the convolution filter. Secondly, every K iterations, the (new) input to the neural field must be equally enlarged. Thirdly, at each iteration, the neural field must be convolved in a way that corrects the creation of an artificial edge at the boundaries between neural field and zero-filled regions. It must be ensured that the convolution filter does not "see" this edge: we require that the convolution with a zero-padded identity field, $f(\vec{x}) * I$, results in a homogeneous value everywhere within the area of the field. This is achieved by convolving (i.e., multiplying) the size-reduced field in the FR by both summands in eqn. (3) separately; the results are transformed back into the SR, where each is point-wise multiplied by a separate pre-computed mask. The results are re-transformed to the size-reduced FR and subtracted from each other, giving the final convolution result. For details, please see Fig. 1 (right).

Please note that the application of boundary conditions in the described way does not prevent MAD simulations to be more efficient than SR-based ones.

3 Theoretical analysis

In order to assess the advantage gained by iterating eqn. (2), it is necessary to make an estimate of the complexity of the algorithm and compare it to an estimate derived for the simulation of dynamics following eqn. (1). Furthermore, a proof is required for the claim that the qualitative behavior of eqn. (2) reproduces that of eqn. (1). In all calculations in this section, zero-padding boundary conditions are assumed since they are the "worst case" in terms of efficiency.

3.1 Complexity analysis

We first introduce some notation: assume a discretized neural field of quadratic layout having M^2 discretized elements ("neurons"). Let furthermore K designate the number of time steps during which the afferent input is constant (it is commonly assumed that the field is on a faster time scale than its input), and let $\eta <= 1$ stand for the factor that the size of each neural field dimension is reduced by (see also Fig. 1). The size-reduced field therefore has $\eta^2 M^2$ elements. Adding up the contributions from all steps shown in Fig. 1 for MAD (see section 2), using zero-padding boundary conditions (case **B** in Fig. 1, left) and a linear rectified transfer function, we arrive at a complexity per iteration of

$$C_{\text{MAD}}(M, K, \eta) = O(M^2\eta^2\{\frac{2}{K\eta^2}\log_2(M^2) + 5\log_2(\eta^2 M^2) + 6 + \eta^{-2}\}) \quad (4)$$

if a fast Fourier transform can be used which has a complexity of $M^2 \log_2 M^2$[10]. When trying to do the same thing for eqn. (1), we obtain $C_{\text{AD,FFT}}(M) = O(M^2\{3\log_2(M^2) + 5\})$, assuming that the calculation of the convolution term in eqn. (1) is done in the FR; calculation in the SR is disregarded since it depends unfavorably on the convolution filter size. The advantage of MAD, as can be seen from Fig. 2 (left), is already quite pronounced for $\eta = 0.5$ and grows as the application allows to use even smaller values of η than assumed in Fig. 2. This is supported by experimental results as can be seen in Fig. 2 (right).

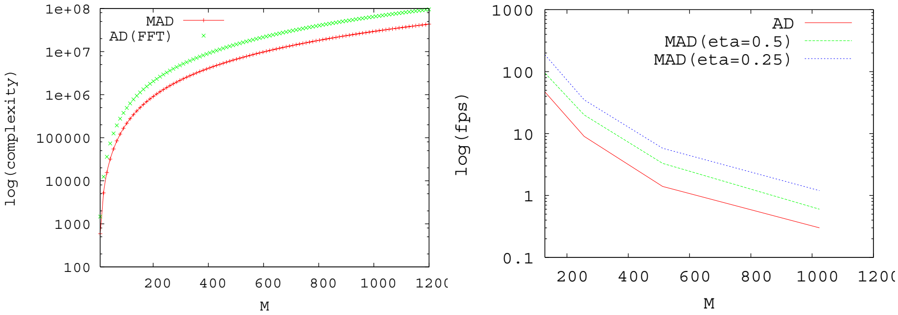

Fig. 2: Theoretical and experimental performance measures. Left: Plot of the complexity measures C_{MAD} ($\eta = 0.5$) and $C_{\text{AD,FFT}}$ defined in the text against the neural field size M. The values on the complexity axis are irrelevant since only their relative magnitude matters. Right: Experimental performance (on a 2,4 GHz Pentium Processor running Linux), given in frames per second (fps). For producing both figures, a value of $K = 10$ was used (see text).

3.2 Mathematical analysis

In this section, we will prove that a key property of AD also holds for the proposed MAD model: the existence of stable, static "disk" solutions characterized by $a(\vec{x}, t) > 0$ for $\vec{x} \in D$ and $a(\vec{x}, t) < 0$ otherwise, where D is a circularly symmetric region of radius R. Following the analyses given in [3, 11], we choose the Heaviside function $\theta(x)$ as transfer function. For the static case with no afferent input to the field, the differential equation for MAD reduces to

$$a(\vec{x}, t) = F(\vec{x}) * f[S(\vec{x}) * a(\vec{x}, t)] + h \tag{5}$$

which is identical to the expression obtained in [3] except for the smoothing operation on the right-hand side. A way to reduce this expression to the one obtained in [3] is to look for field configurations where the smoothing has no effect, or: $S(\vec{x}) * a(\vec{x}, t) \equiv a(\vec{x}, t)$. In that case, the proof from [3] applies which states the stability of disk solutions depending on their radius and the global inhibition term h. Obviously, smoothing has no effect if the field contains only low spatial frequencies unaffected by S. This can be trivially fulfilled, and thus we have found a sufficient condition for the existence of symmetric disk solutions.

3.3 Choice of η

MAD can be iterated in a very efficient manner if small values of η (see section 3.1) can be justified. To determine this, let us recall that this parameter governs the smoothing of the neural field before computing the lateral interaction term in eqn. (2). Depending on η, only low spatial frequencies will contribute to the lateral interactions meaning that smaller localized activations cannot produce any lateral interaction themselves (although they "feel" it). As shown previously, stable disk solutions of a certain radius may arise, and this radius is roughly of the order of the Gaussian variances in the interaction kernel, see [11]. A simple heuristic to determine feasible values of η is therefore to choose η such that no stable disk solutions are eliminated by smoothing. Thus, η can be determined from the variance parameters in eqn. (3). Another way to see this is to consider that a downsampled version of the convolution kernel $F(\vec{x})$ is used in order to perform the convolution during a MAD iteration, see section 2. Clearly, an η must be chosen which ensures that $F(\vec{x})$ can be downsampled without being corrupted, which gives us a lower bound for η.

4 Experimental tests

We conducted experiments using the same setting and parametrization as used in the complexity analysis of section 3.1, but instead of the complexity measures $C_{\mathrm{AD,FFT}}$ and C_{MAD} we calculated the number of iterations per second that could be achieved by using the different simulation approaches on standard computer hardware. All simulation algorithms make use of strongly optimized block operations using the Intel IPP library. The results are shown in Fig. 2 (right). Keeping in mind that $K = 10$ iterations are computed for each frame, real-time capability of MAD simulation is evident even though the more complex (see Fig. 1) zero-padding boundary conditions were used. Although we did not test this fully, we always observed stable localized activations even when the field initially contained high frequencies, violating the condition for stable solutions derived in section 3.2.

5 Discussion

Summarizing, we have proposed a modified version of the neural field equation proposed by Amari, and shown experimentally as well as theoretically that it can be solved numerically (simulated) more efficiently than previous approaches if certain conditions are met. We have presented a simple way to estimate whether this is the case, and what kind of speed-up may be expected in a given scenario. We believe that in many application scenarios such as bio-inspired image processing [7, 6], cognitive modeling[5, 2, 4] or scene interpretation[8], the necessary requirements are met to gain strong performance gains in the way presented here. It is our hope that this may contribute to a greater acceptance of neural fields in general, and to the researchers' capability to simulate much larger cognitive systems and models than has been possible up to now.

References

[1] Shun-ichi Amari. Mathematical foundations of neurocomputing. *Proceedings of the IEEE*, 78(9):1441–1463, 1990.

[2] Wolfram Erlhagen and Gregor Schöner. Dynamic field theory of movement preparation. *Psychol Rev*, 109(3):545–572, Jul 2002.

[3] JG Taylor. Neural 'bubble' dynamics in two dimensions: foundations. *Biological Cybernetics*, 80:393–409, 1999.

[4] Paul Cisek. Integrated neural processes for defining potential actions and deciding between them: a computational model. *J Neurosci*, 26(38):9761–9770, Sep 2006.

[5] Fred H Hamker. A dynamic model of how feature cues guide spatial attention. *Vision Res*, 44(5):501–521, Mar 2004.

[6] J Conradt, P Simon, M Pescatore, and P Verschure. Saliency maps operating on stereo images detect landmarks and their distance. In *Proceedings of the International Conference on Neural Networks*, 2002.

[7] L. Itti and C. Koch. Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203, Mar 2001.

[8] U Handmann, I Leefken, and W von Seelen. Scene interpretation and behavior planning for driver assistance. In *Enhanced and Synthetic Vision 2000 at AEROSENSE 2000*, volume 4023 of *SPIE Proceedings Series*, pages 201–212, 2000.

[9] B Jähne. *Digital image processing.* Springer Verlag Berlin, Heidelberg, New York, 6 edition, 2005.

[10] *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, 3 edition, 2007.

[11] Inna Mikhailova and Christian Goerick. Conditions of activity bubble uniqueness in dynamic neural fields. *Biol Cybern*, 92(2):82–91, Feb 2005.