Neurocomputing 164 (2015) 45–55

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A discrete gradient method to enhance the numerical behaviour of Hopfield networks ☆

Yadira Hernández-Solano [a], Miguel Atencia [b,*], Gonzalo Joya [c], Francisco Sandoval [c]

[a] Universidad de La Habana, Facultad de Matemática y Computación, Cuba
[b] Universidad de Málaga, Campus de Excelencia Internacional Andalucía Tech, Departamento de Matemática Aplicada, Spain
[c] Universidad de Málaga, Departamento de Tecnología Electrónica, Spain

ARTICLE INFO

Article history:
Received 18 May 2014
Received in revised form
11 October 2014
Accepted 13 October 2014
Available online 14 March 2015

Keywords:
Hopfield neural networks
Dynamical systems
Numerical methods for ODEs
Discrete gradients

ABSTRACT

This paper presents the construction of a numerical method for implementing algorithms that are based on a gradient flow. In particular, continuous Hopfield networks for solving optimization problems are considered as a case in point. The focus is the preservation of the favourable properties of the continuous system under discretization. Firstly, the conventional discretization is formulated as a non-standard numerical method that solves the continuous equation, depending on the step size. A rigourous theoretical analysis shows that it is a consistent method, but it fails to preserve the gradient nature, since periodic solutions occur, so no Lyapunov function can exist. Next we present the construction of a method that preserves the Lyapunov function of the original differential equation regardless of the step size. This procedure, based upon discrete gradients, yields an algorithm that preserves the Lyapunov function of the continuous system, thus reproducing the same qualitative behaviour, namely stability of equilibria and convergence to solutions. A remarkable property of the proposed technique is that it can be computed explicitly as long as the Lyapunov function is multi-linear, which is the case of Hopfield neural networks. This results in enhanced performance, compared to the conventional discretization, as shown in a comprehensive set of numerical experiments.

1. Introduction

The concept of gradient is a key element of many algorithms that are formulated in *continuous* time, both in machine learning and in conventional mathematics. Most often, the field of application of such *gradient algorithms* involves some sort of optimization, which is hardly surprising since, in the end, evolving according to the gradient of a function amounts to tracking a downwards trajectory of that function, thus performing its minimization. Some examples include back-propagation [1], conjugate gradient [2], recurrent neural networks [3], parameter estimation [4,5], and optimization in manifolds [6]. The computer implementation of these continuous algorithms, usually formulated as Ordinary Differential Equations (ODEs), requires them to undergo discretization, which will possibly destroy the dynamical properties. Therefore we propose that research on continuous algorithms should proceed through three stages: analysis of the continuous system, discretization, and finally, implementation issues, choice of design parameters, and problem-dependent heuristics. In this regard, continuous Hopfield networks are paradigmatic: on one hand, they were applied to combinatorial optimization shortly after their inception [7] and their performance for particular problems has been extensively analysed (e.g. [8] and references therein); on the other hand, a great many theoretical results have been published on the continuous network [9], as well as different variants, e.g. systems with delays [10]. This paper is dedicated to analyse the discretization stage, which we postulate that, despite being critical, has often been disregarded.

As a proof of concept, the results here stated are applied to the continuous Hopfield network and, in particular, we consider the so-called Abe formulation [11], which has been shown to present favourable performance in optimization [9]. When self-weights are assumed to vanish, stable equilibria occur at binary states, which correspond to feasible solutions of combinatorial optimization problems. This and other results about the continuous model that will be used during the paper are briefly recalled in Section 2. The conventional discretization of this model results from replacing the derivative by the finite difference, although this process often goes unmentioned. We study this discretization from the point of view of numerical methods, thus we firstly require to explicitly formulate the discrete system as an one-step map, yielding a—non-standard—numerical method. Finally, the dynamical properties of this discrete map are analysed, leading to a major result: the Lyapunov function

☆The useful suggestions of the anonymous reviewers are gratefully acknowledged. This work has been partially supported by the Asociación Universitaria Iberoamericana de Postgrado (AUIP).

* Corresponding author.
E-mail address: matencia@ctima.uma.es (M. Atencia).

http://dx.doi.org/10.1016/j.neucom.2014.10.091
0925-2312/

of the continuous network cannot be preserved in general by the conventional discretization, since periodic solutions appear. To summarize, in Section 3 we prove that *discretization destroys the gradient-like nature*. In addition to the value of this result for the Hopfield model in particular, we emphasize that the analysis of algorithms as the discretization of continuous systems is a fruitful methodology of general validity.

In view of the limitations of the conventional discretization of Hopfield networks, we propose using a numerical method that has been specifically designed for preserving the qualitative properties of the original ODE. This aim presents remarkable similarities to the recent approach of *geometric numerical integration* [12,13], which has provided interesting results for Hamiltonian systems. However, these results cannot be trivially extended to deal with the preservation of the Lyapunov function of a gradient system, because in this case there is no explicit manifold that the system resides on, thus the classification of these methods as "geometric" is questionable. It is noticeable that this approach has been studied from different perspectives, for instance, within the optimization framework the discretization of appropriate continuous dynamical systems, whose trajectory minimizes the target, has resulted in a promising "ODE-approach to nonlinear programming" [14,15], as well as machine learning by optimization on Riemannian manifolds [16,17]. Also, promising results have been presented for learning systems endowed with either a Lie-group structure [18] or a pseudo-Riemannian metric [19], and machine learning by dynamical systems on manifolds [20,21]. In Section 4 we present *discrete gradient* methods [22], which by construction respect the energy-diminishing feature of gradient systems regardless of the step size. Then, we construct a discrete gradient integrator for the implementation of Hopfield neural networks, with a remarkable property: the method can be explicitly computed, which is not often the case in algorithms that preserve qualitative properties. Incidentally, let us note that there is limited experience in the practical implementation of discrete gradient methods to particular applications, so we expect that this paper also sheds light on the range of applicability of these methods.

In Section 5 we considerably expand previous results [23] by presenting some numerical experiments that show the favourable performance of discrete gradient methods, compared to the conventional discretization. Although the theoretical results of previous sections concern the Hopfield network whose self-weights vanish, simulations also include the model with self-weights in order to illustrate the generality of the proposal. The whole point of the comparison is that enhanced performance can be expected when preservation of the Lyapunov function is born in mind, since the dynamical properties lie in the heart of the construction of continuous gradient algorithms. Finally, the main conclusions of the paper are gathered in Section 6, and some promising directions for further research are proposed.

2. Hopfield networks as gradient algorithms

In this section we recall the equations of the model that is adopted as a paradigmatic example of gradient algorithm, namely continuous Hopfield neural networks used as optimization methods [3,7]. In particular, we focus on the so-called Abe formulation [11,24,25], which is defined by the following n-dimensional system of ODEs:

$$\frac{du_i}{dt}=\mathrm{net}_i;\quad y_i(t)=\tanh\left(\frac{u_i(t)}{\beta}\right);\quad i=1,\ldots,n \tag{1}$$

where y_i is the output or state of neuron i, u_i can be regarded as a sort of internal potential, $\beta>0$ is a parameter that regulates the slope of the hyperbolic tangent function, and n is the number of neurons. As for the term net_i, it stands for the linear input to neuron i, and it can plainly be expressed in matricial form:

$$\mathrm{net}=Wy-b \tag{2}$$

where the entries w_{ij} of the matrix W are the weights of the connection from neuron j to neuron i, and b_i is the bias of neuron i. In the higher order generalization [26,27], the term net_i is multilinear, i.e. an r-th order polynomial in the n variables y_i:

$$\mathrm{net}_i=\frac{1}{q!}\sum_{q=1}^{r}\sum_{(i_1\ldots i_q)\in\{1,\ldots,n\}^q} w_{ii_1\ldots i_q}y_{i_1}\ldots y_{i_q}-b_i \tag{3}$$

where the coefficients $w_{ii_1,i_2\ldots i_q}$ are now the weights of q-th order connections. In the following, we assume that the weights are symmetric, i.e. $w_{ij}=w_{ji}$ or, in general, two weights are equal if they have the same set of sub-indices, regardless of the order. When Hopfield networks are used as optimization methods, this assumption leads to no loss of generality [28].

First of all, let us observe that the system given by Eqs. (1) and (3) is not an ODE, strictly speaking, due to the presence of the variables u_i, which do not appear in differential equations. In order to state the general results either on dynamical systems to the continuous network, or on numerical analysis to the discretization, we reformulate Eq. (1) as an ODE, with the single set of variables y_i:

$$\frac{dy_i}{dt}=\frac{dy_i}{du_i}\frac{du_i}{dt}=\frac{1}{\beta}(1-y_i^2)\mathrm{net}_i \tag{4}$$

In this form, it is clear that the parameter β only stands for a reparametrization of time, so in the rest of this paper we assume $\beta=1$ for simplicity, with no loss of generality, obtaining the ODE that defines the continuous network:

$$\frac{dy_i}{dt}=(1-y_i^2)\mathrm{net}_i\triangleq f_i(y) \tag{5}$$

A remarkable property of this system is that trajectories always remain within the hypercube $(-1,1)^n$, as long as initial values also belong to such hypercube. In other words, the hypercube is invariant, which is obvious from Eq. (1), since the range of the hyperbolic tangent function is the interval $(-1,1)$. However, if a standard numerical method is to be applied to Eq. (5), the discretization can destroy this property. Furthermore, it is obvious from Eq. (5) that the equilibria of the continuous network are either vertices $|y_i|=1$ or interior points characterized by $\mathrm{net}_i=0$.

The crucial fact about Hopfield networks given by Eq. (1) is the existence of a Lyapunov function (see e.g. [29] for Lyapunov theory), which apart from other technical conditions, possesses the distinct feature that it decreases through system trajectories. In the case of the linear dynamics of Eq. (2), the Lyapunov function is quadratic:

$$V(y)=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}w_{ij}y_iy_j+\sum_{i=1}^{n}b_iy_i=-\frac{1}{2}y^{\top}Wy+b^{\top}y \tag{6}$$

whereas the generalization to higher order—Eq. (3)—leads to a multinomial Lyapunov function:

$$V(y)=-\frac{1}{(q+1)!}\sum_{q=1}^{r}\sum_{(i_1\ldots i_{q+1})\in\{1,\ldots,n\}^{q+1}}w_{i_1\ldots i_{q+1}}y_{i_1}\ldots y_{i_{q+1}}+\sum_{i=1}^{n}b_iy_i \tag{7}$$

Regardless of the order of the model, the key observation is $\frac{\partial V}{\partial y_i}=-\mathrm{net}_i$, which leads to a simple proof for V being a Lyapunov function:

$$\frac{dV}{dt}=\sum_{i=1}^{n}\frac{\partial V}{\partial y_i}\frac{dy_i}{dt}=-\sum_{i=1}^{n}\mathrm{net}_i(1-y_i^2)\mathrm{net}_i\le 0 \tag{8}$$

due to the positivity of the derivative of the hyperbolic tangent function defined in the interval $(-1,1)$. Further, observe from Eq. (5) that the equality $\frac{dV}{dt}=0$ only occurs at equilibria, i.e. either vertices $|y_i|=1$ or interior points $\mathrm{net}_i=0$. This fact, together with