Numerical implementation of gradient algorithms*

Miguel Atencia[1], Yadira Hernández[2], Gonzalo Joya[2,3], and Francisco Sandoval[3]

[1] Departamento de Matemática Aplicada. Universidad de Málaga (Spain)
[2] Facultad de Matemática y Computación. Universidad de La Habana (Cuba)
[3] Departamento de Tecnología Electrónica. Universidad de Málaga (Spain)
Campus de Teatinos, 29071 Málaga, Spain
matencia@ctima.uma.es

Abstract. A numerical method for computational implementation of gradient dynamical systems is presented. The method is based upon the development of geometric integration numerical methods, which aim at preserving the dynamical properties of the original ordinary differential equation under discretization. In particular, the proposed method belongs to the class of discrete gradients methods, which substitute the gradient of the continuous equation with a discrete gradient, leading to a map that possesses the same Lyapunov function of the dynamical system, thus preserving the qualitative properties regardless of the step size. In this work, we apply a discrete gradient method to the implementation of Hopfield neural networks. Contrary to most geometric integration methods, the proposed algorithm can be rewritten in explicit form, which considerably improves its performance and stability. Simulation results show that the preservation of the Lyapunov function leads to an improved performance, compared to the conventional discretization.

Keywords: Gradient Systems, Hopfield Neural Networks, Geometric Integration

1 Introduction

Algorithms that include some sort of gradient computation are pervasive in machine learning literature and, in fact, in all branches of mathematics and computer science. A non-exhaustive list could include backpropagation [7], recurrent neural networks [8, 9] or parameter estimation [3, 4]. The performance of such algorithms is a direct consequence of their dynamical behaviour, i.e. the fact that states follow a trajectory such that the gradient of some function, usually a target function of some optimization problem, is forced to decrease. Many algorithms have a natural description in continuous time, i.e. they are formally defined as

* This work has been partially supported by FEDER funds (through project no. TIN2010-16556 from the Spanish Government and project no. P08-TIC-04026 from the Junta de Andalucía), and the Agencia Española de Cooperación Internacional para el Desarrollo (project no. A2/038418/11).

Ordinary Differential Equations (ODEs), so that they should be discretized in order to be implemented as a practical algorithm on a digital computer, which is intrinsically a discrete system. In general, numerical methods used to discretize a continuous system will destroy the dynamical properties.

The recent approach of *geometric numerical integration* aims at designing numerical methods for ODEs that preserve the qualitative properties of the original ODE [13, 6]. This approach has been rather fruitful in providing numerical methods for Hamiltonian systems, whereas few methods have been proposed that preserve the Lyapunov function of gradient systems. To the best of our knowledge, the main proposals in this regard comprise the *discrete gradient* methods [12] and the projection methods [5]. On one hand, projection methods are too involved and, although they are formally explicit, they require solving a nonlinear equation at every step. Projection methods have the advantage that arbitrary order can be achieved but, when preservation of the gradient structure is the main aim and accuracy is not crucial, discrete gradient methods are more appealing. However, on the other hand, no practical implementations of discrete gradient methods have been pursued. Therefore, in this contribution we construct a discrete gradient method for the implementation of Hopfield neural networks.

In Section 2 we briefly recall the definition of discrete gradient methods. Hopfield neural networks are presented as an interesting case in point in Section 3. The main contribution of this paper is the construction of a discrete gradient method for continuous Hopfield networks, which is the subject of Section 4. Some experimental results shown in Section 5 support the performance of the proposed implementation. Finally, Section 6 gathers the conclusions and suggest some directions for further research.

2 Numerical methods based on discrete gradients

In this section we recall, for the sake of completeness, the process of construction of a numerical method that preserves the Lyapunov function of an ODE, by means of a discrete gradient. Further details can be found in references [12]. First of all, consider an ODE $\frac{d\,y}{d\,t} = f(y)$ with the usual assumptions about existence and uniqueness of solutions, so that a trajectory $y(t)$ is defined for each initial value $y_0 = y(0)$. The trajectories $y(t)$ are defined on the real n-dimensional vectorial space $\mathbb{R}^n$. The interesting case for our exposition is the existence of—at least—one equilibrium y_f, i.e. a point where $f(y_f) = 0$ holds. We assume further that such equilibrium is asymptotically stable, which amounts to both stability in the sense of Lyapunov *and* attractiveness; in other words, all trajectories $y(t)$ that start in a neighbourhood of y_f remain in a—possibly different—neighbourhood and tend to the equilibrium, i.e.: $\lim_{t\to\infty} y(t) = y_f$. It is well known from the converse Lyapunov function theorems [11] that if an equilibrium is asymptotically stable, then a Lyapunov function exists. Then we assume the explicit knowledge of a function $V(y)$ that satisfies the conditions

for being a Lyapunov function in a neighbourhood of the equilibrium y_f. Recall that a Lyapunov function, apart from usual smoothness assumptions, fulfils the condition $\frac{d\,V}{d\,t} \leq 0$ and y_f must be a—possibly local—minimum of V. Then, the ODE $\frac{d\,y}{d\,t} = f(y)$ can be written in the *linear-gradient* form:

$$\frac{d\,y}{d\,t} = L(y)\,\nabla V(y) \tag{1}$$

where $L(y)$ is a symmetric and negative-semidefinite matrix function and $\nabla V(y)$ is the gradient of the function V. Arguably the fact that the ODE is explicitly written in the linear-gradient form could have been taking as the starting assumption for the studied systems, however it is worth emphasizing that the linear-gradient form given by Equation (1) is not unique [10].

An ODE that has been rewritten in the form of linear gradient, as in Equation (1), can be discretized by considering the following integration method:

$$\frac{y_{k+1} - y_k}{h} = \widetilde{L}(y_k, y_{k+1}, h)\,\overline{\nabla} V(y_k, y_{k+1}) \tag{2}$$

where $\widetilde{L}$ is an approximation of L such that $\widetilde{L}(y_k, y_k, 0) = L(y)$ and $\overline{\nabla} V$ is a *discrete gradient*, i.e. the following conditions are met:

$$\begin{gathered} \overline{\nabla} V\left(y_k, y_{k+1}\right) \cdot \left(y_{k+1} - y_k\right) = V\left(y_{k+1}\right) - V\left(y_k\right) \\ \overline{\nabla} V\left(y, y\right) = \nabla V\left(y\right) \end{gathered} \tag{3}$$

It can be proved that the numerical method given by Equation (2) has the same Lyapunov function as the original ODE, regardless the step size h, thus the approximate discretized trajectories will converge to the same equilibria, provided that initial values are close enough.

Note that there is considerable freedom in the choice of both $\widetilde{L}$ and $\overline{\nabla} V$ and, in general, the method given by Equation (2) is implicit. In this paper we propose a suitable choice of these functions so that an explicit method for discretization of Hopfield networks is obtained.

3 Continuous Hopfield Networks

Hopfield networks comprise a well known neural paradigm that was originally defined as a discrete time recurrent system [8]. Then, a continuous version [9] was proved to be useful for solving, among others, optimization problems. In the Abe formulation [1], the continuous Hopfield network is defined by the following system of ODEs:

$$\frac{d\,u}{d\,t} = W\,y - b\,; \quad y = \tanh u \tag{4}$$

where u and y are n-dimensional vectors, W is a zero-diagonal matrix of weights, and b is a bias vector. Note that the computation of the vector y is componentwise, i.e. $y_i = \tanh u_i$ for $i = 1 \dots n$ is meant. From a biological point of view, the vector u represents the post-synaptic potentials of the neurons whereas the vector y represents the action potential, thus y can be considered an output and u comprises the internal neuron states. However, mathematically the variables y are fed back into the ODE so it is convenient to rewrite the model depending only on this vector, which results from the chain rule:

$$\frac{d\,y}{d\,t} = \frac{d\,y}{d\,u}\frac{d\,u}{d\,t} = (1 - \tanh^2 u)\ (W\,y - b) = (1 - y^2)\ (W\,y - b) \tag{5}$$

Then, from the dynamical point of view, Hopfield networks are stable systems because a Lyapunov function can be defined as a function of the states y:

$$V(y) = -\frac{1}{2}\,y'\,W\,y + b'\,y \tag{6}$$

where the apostrophe $'$ denotes matrix transpose. We emphasize that the construction of a practical optimization algorithm from Hopfield networks stems from the dynamical properties [2]: since a Lyapunov function is decreasing over time, thus the evolution of the states of the network heads towards a stable equilibrium, where the Lyapunov function presents a—possibly local—minimum. Therefore, an optimization problem where the target function has the same structure of the Lyapunov function given by Equation (6) is solved by matching the target and the Lyapunov function, so that the weights and biases are obtained. Finally, the network is "constructed", which usually means that some numerical method is used to integrate Equation (4) until the states reach an equilibrium, which provides the minimum. Therefore, using a numerical method that preserves the dynamical properties of the continuous ODE under discretization is a crucial step, since a solution is only obtained by an algorithm that mimics the *correct* dynamical behaviour.

4 Discretization of Hopfield Networks

In this section, for the sake of comparison, we describe the conventional discretization that is usually adopted to implement Hopfield networks. Then, we apply the results of Section 2 to define a discrete gradient method that numerically integrates a Hopfield network while preserving the Lyapunov function, regardless of the step size h.

The usual discretization of Hopfield networks results from replacing the derivative $\frac{d\,u}{d\,t}$ by the linear approximation $\frac{u_{k+1} - u_k}{h}$ in Equation (4) for a chosen h that should be small enough. Therefore, a two-step iteration that involves both the internal potential u and the state y is obtained:

$$u_{k+1} = u_k + h\ (W\,y_k - b)\ ;\ \ y_{k+1} = \tanh u_{k+1} \tag{7}$$

The linear approximation is well founded—indeed it is the basis for the well known Euler rule—but it does not guarantee that the resulting numerical method will preserve the dynamical properties of the ODE and, in particular, that it will possess the same Lyapunov function as the continuous dynamical system.

We now present the main contribution of the paper: designing a discrete gradient method for the Hopfield network. First of all, recall that the Hopfield network given by Equation (4) can be cast into the linear gradient form, by eliminating the internal variable u, as shown in Equation (5). Then, the functions $\widetilde{L}$ and $\overline{\nabla}$ are chosen with a suitable definition:

$$\begin{aligned} \left(\widetilde{L}(y,z,h)\right)_{ii} &= 1 - y_i\, z_i \\ \left(\overline{\nabla} V(y,z)\right)_i &= \frac{V(z_1 \dots z_i, y_{i+1} \dots y_n) - V(z_1 \dots z_{i-1}, y_i \dots y_n)}{z_i - y_i} \end{aligned} \tag{8}$$

and $\widetilde{L}$ is a diagonal matrix, i.e. $\left(\widetilde{L}(y,z,h)\right)_{ij} = 0$ if $i \neq j$. It is straightforward to prove that this choice is consistent with the definition of discrete gradient presented in Section 2. A crucial point of this definition is the fact that the implicit numerical method given by Equation (2) can be rewritten as an explicit method in the particular case of Hopfield networks. After straightforward algebra, the proposed iteration results:

$$(y_{k+1})_i = \frac{(y_k)_i + h\, \left(W\, v^{(i)} - b\right)_i}{1 + h\, (y_k)_i\, \left(W\, v^{(i)} - b\right)_i} \tag{9}$$

where $v^{(i)}$ is a vector with mixed components: $\left(v^{(i)}\right)_j = (y_{k+1})_j$ if $j < i$ and $\left(v^{(i)}\right)_j = (y_k)_j$ if $j \geq i$. Note that Equation 9 is an explicit map because the matrix W has zeros on the diagonal.

5 Experimental results

In this section we present a summary of our simulation results, which show the superior performance of the discrete gradient numerical method, designed to preserve the Lyapunov function of an ODE. The continuous Hopfield network, as described in Section 3, is used as a case in point since, as mentioned above, its optimization ability is a result of its dynamical properties. In particular, the definition of a Lyapunov function, which is matched to the target function of the optimization problem, proves that the trajectories of the states of Hopfield networks converge to stable equilibria. We measure the performance of a discretization algorithm considering two features of the convergence to the minimum: convergence speed and avoidance of local minima.

A Hopfield network has been implemented both with the numerical method based upon discrete gradients and the conventional discretization, as presented

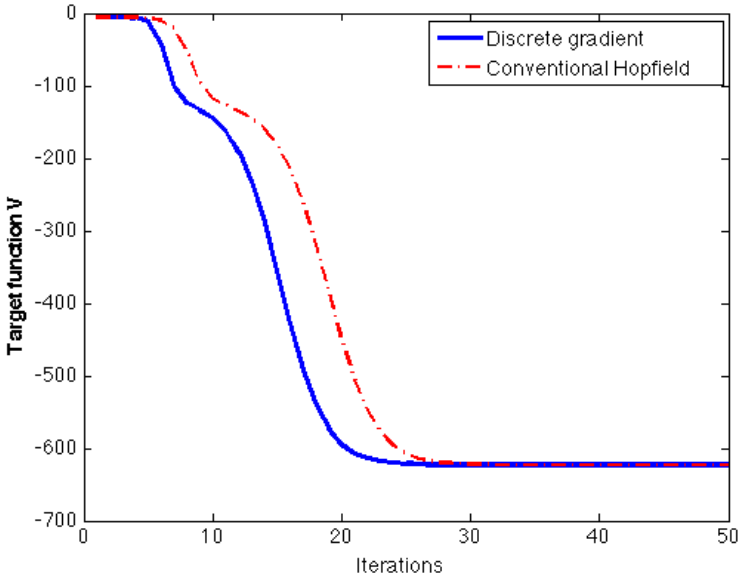

Fig. 1. Decrease in target function along trajectories for the conventional Hopfield implementation and the numerical method based upon discrete gradient.

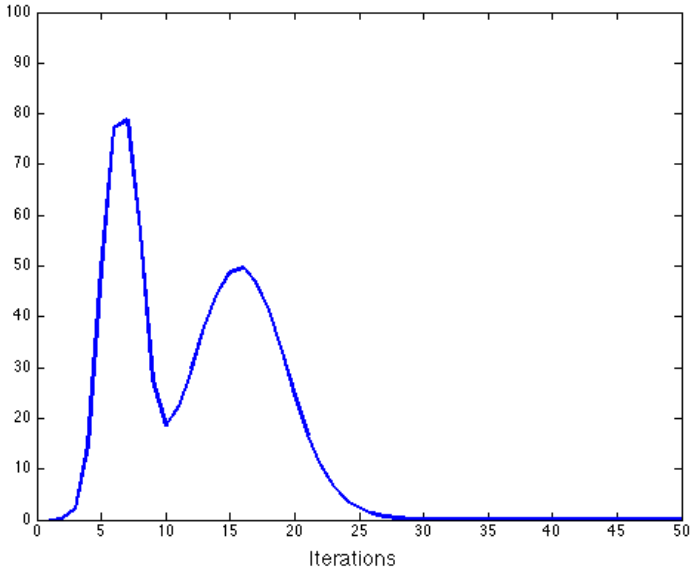

Fig. 2. Relative outperformance (percentage) of the method of discrete gradient regarding target function decreasing.

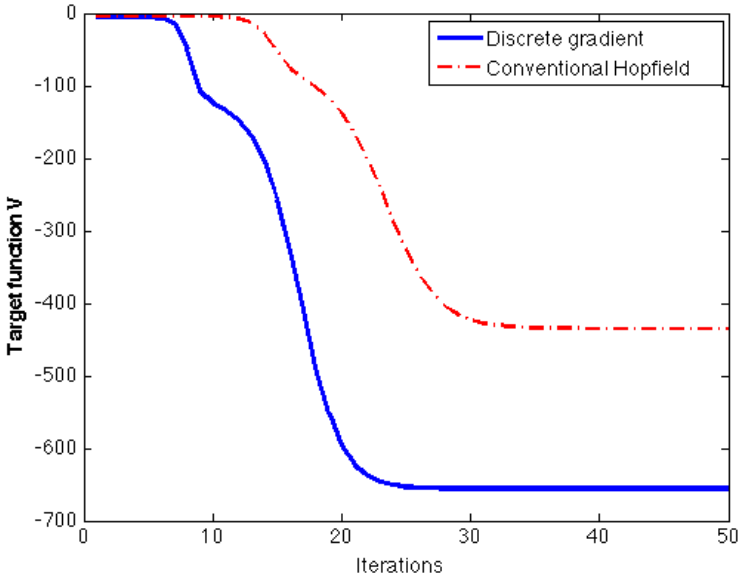

Fig. 3. Decrease in target function along trajectories, considering another initial point: each method converges to a different stable point.

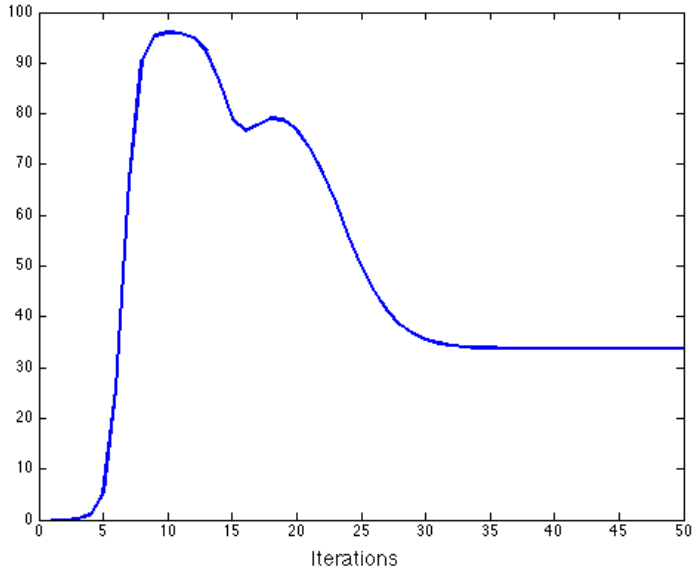

Fig. 4. Relative outperformance (percentage) of the method of discrete gradient regarding target function decreasing for the same initial point as in Figure 3.

in Section 4. The same value of the step size h was assigned for both methods. A battery of experiments has been performed, by assigning different values to all design parameters of the neural network: dimension—number of neurons—of the network, weight matrix and initial point of the trajectory. The results show a consistent outperformance of the discrete gradient method, both from the point of view of convergence speed and avoidance of local minima. There is certainly a random component, due to the choice of initial point, but in most of the experiments the discrete gradient method converges considerably faster. In a significant number of instances, the discrete gradient method attains a stable equilibrium that corresponds to a lower value of the target function, whereas the conventional Hopfield implementation is stuck at a local minimum. The discrete gradient method, as all algorithms based upon local approximations, is not immune to local minima, but the occasions in which the solution provided by the conventional network is better are exceptional and statistically not significant. Only a portion of these experiments, which is considered to illustrate the conclusions, is here shown for brevity.

In Figure 1, the implementation of an instance of a Hopfield network of dimension $n = 30$ is depicted, showing the discretization with the discrete gradient method and the conventional Hopfield network. Since the trajectories of the 30 components of the state are not particularly informative, only the values of the target function at each state are presented. The graphs show that both algorithms converge to the same equilibrium, thus the attained value of the target function is the same. However, the discrete gradient method converges significantly faster since, for instance, reaching the value $V = -600$ requires a 20 percent less in the number of computed iterations, approximately. In Figure 2, the values of the target functions along the simulated trajectories with both methods are shown as the percentage that the discrete gradient method is lower than the value achieved by the conventional method. At some stages of the network evolution, the discrete gradient algorithm provides a value of the target function that is almost 80 percent lower than that of the conventional Hopfield method, using the absolute value of the minimum of both values as a base. Note that faster convergence of a method is a significant finding, since it could mean that a quasi-optimal solution is obtained with reasonable computational cost, where is the computing time of a slower algorithm may be unaffordable.

Similar results are shown in Figures 3 and 4, but now the difference between both methods is qualitative, since a different stable equilibrium is attained, i.e. the solutions obtained by the two methods are different. In Figure 3, again faster decrease of the target function achieved by discrete gradient method is observable and, besides, a significantly lower value is finally attained. This fact confirms that the conventional method has stuck at a local minimum. In Figure 4, the relative improvement in the obtained value of the target function is obtained. Not only the discrete gradient outperforms the conventional method during the network evolution, but also the solution finally obtained is a 40p̃ercent better, approximately.

An exhaustive set of experiments supports that the discrete gradient method outperforms the conventional Hopfield implementation as a rule.

6 Conclusions and future directions

We have presented the construction of a numerical method for implementation of continuous Hopfield networks. The method is based upon discrete gradients, thus it guarantees the preservation of the Lyapunov function, which is the key to the optimization ability of Hopfield networks. The suitable choice of the functions that define the method leads to an explicit method, which is a considerable advantage concerning both stability and computational cost. Simulation results show that the proposed method provides superior performance compared to the conventional implementation of Hopfield networks.

We suggest that this novel method paves the way for far-reaching advances in the construction of algorithms, beyond the particular case presented. More attention should be paid to the preservation of dynamical properties under discretization and the application of techniques from the geometric integration approach should be widespread. Therefore we are currently engaged in two main directions for expanding the presented results. On the one hand, we are generalizing the discrete gradient method to a wide range of systems, more complex than Hopfield networks. On the other hand, we aim at a deeper theoretical foundation for discrete gradients, in particular developing more accurate, higher-order methods, while preserving the qualitative properties of the dynamical system.

References

1. Abe, S.: Theories on the Hopfield Neural Networks. In: Proc. IEE International Joint Conference on Neural Networks. vol. I, pp. 557–564 (1989)
2. Atencia, M.A., Joya, G., Sandoval, F.: Dynamical Analysis of Continuous Higher Order Hopfield Networks for Combinatorial Optimization. Neural Computation 17(8), 1802–1819 (2005)
3. Atencia, M.A., Joya, G., Sandoval, F.: Hopfield Neural Networks for Parametric Identification of Dynamical Systems. Neural Processing Letters 21(2), 143–152 (2005)
4. van den Bos, A.: Parameter estimation for scientists and engineers. Wiley-Interscience (2007)
5. Calvo, M., Laburta, M.P., Montijano, J.I., Rández, L.: Projection methods preserving lyapunov functions. BIT Numerical Mathematics 50(2), 223–241 (2010)
6. Hairer, E.: Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations. Springer, Berlin (2006)
7. Haykin, S.: Neural Networks. A Comprehensive Foundation. Macmillan College Publishing Company (1994)
8. Hopfield, J.: Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. USA 79, 2554–2558 (1982)
9. Hopfield, J.: Neurons with graded response have collective computational properties like those of two-state neurons. Proc. Natl. Acad. Sci. USA 81, 3088–3092 (1984)

10. Iserles, A., Peplow, A., Stuart, A.: A unified approach to spurious solutions introduced by time discretisation. Part I: Basic theory. SIAM J. Numer. Anal. 28(6), 1723–1751 (1991)
11. Khalil, H.K.: Nonlinear Systems. Prentice Hall (2002)
12. McLachlan, R., Quispel, R., Robidoux, N.: Geometric integration using discrete gradients. Philos. Trans. of the Royal Society of London Series A 357(1754), 1021–1045 (1999)
13. Stuart, A., Humphries, A.: Dynamical systems and numerical analysis. Cambridge University Press (1996)