ESANN'2005 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 27-29 April 2005, d-side publi., ISBN 2-930307-05-6.

Two or three things that we (intend to) know about Hopfield and Tank networks

Miguel Atencia[1], Gonzalo Joya[2] and Francisco Sandoval[2] *

1- Universidad de Málaga - Dept. Matemática Aplicada
Campus de Teatinos, s/n. 29071-Málaga - Spain
e-mail: matencia@ctima.uma.es
2- Universidad de Málaga - Dept. Tecnología Electrónica
Campus de Teatinos, s/n. 29071-Málaga - Spain

Abstract. This work aims at reviewing some of the main issues that are under research in the field of Hopfield networks. In particular, the feasibility of the Hopfield network as a practical optimization method is addressed. Together with the current results, the main directions that deserve ongoing analysis are shown. Besides, some suggestions are provided in order to identify lines that are at an impasse point, where there is no evidence that further research will be fruitful, or topics that nowadays can just be considered as historically interesting.

1 Introduction

Since its inception in 1982 [16], the neuronal paradigm generically known as the Hopfield network has literally been the subject of thousands of papers. Indeed, it is worth realizing that many important contributions do not even mention the name "Hopfield", since there are several related models, extensions and particularizations, such as Cohen-Grossberg networks [12], the Takeda-Goodman model and so on. However, assuming some lack of rigour, we freely use the term "Hopfield networks", as a synonym for recurrent, fully-connected networks. Further, when these recurrent networks are considered from the viewpoint of dynamical systems theory, their main feature is the existence of a Lyapunov function. The presented study of the Hopfield network is certainly restricted and biased since a comprehensive review would probably fill the available pages tenfold. Admittedly, the selection of topics is guided by our own interests and can be concisely described as the application of Hopfield networks to optimization. This is the reason for the surname "and Tank" of the title, from the other author of the first reported optimization usage of this paradigm [41], where an instance of the Travelling Salesman Problem (TSP) is solved. Again, this is not a restriction but an illustrative term, and some of the presented models are far apart from the original work by Tank and Hopfield. Hence, we are ready to state one of the things that we *intend to know* about Hopfield networks:

1. Is the Hopfield network capable of performing combinatorial optimization?

*This work has been partially supported by Project No. TIN2004-05961, funded by the Spanish Ministerio de Educación y Ciencia.

Despite the time passed since the model was first presented and the intense research effort, this is still an open question. The controversy on the optimization efficiency of Hopfield networks arose soon after its outset [47], and has accompanied the paradigm since then. The main objection of Wilson and Pawley was the impossibility to reproduce Tank and Hopfield's results. It is now agreed that this was partly due to the omission of a number of important implementation details in the original proposal of the algorithm. Anyway, this criticism persuaded many researchers to lose interest in the Hopfield paradigm, both in the optimization and the neural network scientific communities. Yet, a significant number of contributions have been filled with partly affirmative answers to the above question. Most of them suggest additional heuristics to the Hopfield methodology, so that more or less successful solutions to the TSP have been provided. This progress can be tracked in the review [36] (and references therein), since no dramatic advance has occurred since its publication. We will not pursue a taxonomy of these empirical advances since the TSP itself has been severely questioned as the appropriate benchmark for Hopfield networks [35]. Besides, our interest is focused on methodological issues that are applicable to a wide range of problems and have been somewhat disregarded in the literature. In particular, we emphasize the need for explicitly describing the model that is used in each study, since there are several formulations with different dynamical properties: the discrete formulation [16], the continuous Hopfield formulation [17] and the continuous Abe formulation [1]. We explain the advantages and limitations of each model, when applied to combinatorial optimization, based upon a rigourous theoretical analysis. In particular, it is shown that the discrete formulation is severely limited, thus in the rest of the paper only continuous models are addressed. This continuity is twofold, since, on one hand, states (neuron outputs) extend over a continuous interval; and, on the other hand, continuous networks are modelled by Ordinary Differential Equations (ODEs), thus time is also continuous.

Even when restricting to theoretical studies on the Hopfield network, the list of references is still overwhelming, most of them dealing with the stability of the network as a dynamical system. As mentioned above, this theoretical interest on the Hopfield network is not coherent with its limited reputation as a practical computational algorithm, which is in sharp contrast to other neuronal paradigms. Consider, for instance, the Self Organizing Map (SOM), proposed by Kohonen [24, 25] the same year as Hopfield networks were introduced. The practical usefulness of the SOM is unarguable [26], whereas its theoretical analysis has faced great difficulties. Some advances have indeed been presented [13] that support the empirical results, and this "samosien" approach[1] has provided considerable inspiration both to the title and the content of the present contribution. Therefore, it is important to determine lines of theoretical research that contribute to attain efficient optimization techniques by means of Hopfield networks. In contrast, some other directions may be less fertile for practical applications. Hence we can state another aspect that we *intend to know*:

[1]The authors of [13] belong to the SAMOS research group.

2. Which theoretical stability analysis is still worth undertaking?

Needless to say, it is not our intention to suggest that some published work is useless. Rather, our aim is to draw the attention of researchers who are interested in optimization to recent theoretical advances that could reveal fruitful.

The application of Hopfield networks is not limited to combinatorial optimization. As mentioned above, the Hopfield network presents some difficulties when dealing with the TSP. It can be argued that these objections are extensible to the whole domain of combinatorial optimization. However, this is not a reason to disdain the usefulness of Hopfield networks, since other interesting applications certainly exist. Indeed, based upon previous work, we can affirm that we *know* that:

3. The Hopfield network is competent at solving other classes of optimization problems.

In particular, we present a reformulation of parametric identification of dynamical system as an optimization problem, which is solved by a modified Hopfield network. The resulting network possesses variable weights so that existing stability analysis are not applicable. Both theoretical and empirical results support the efficiency of the presented neural estimator.

The success of an algorithm is not separable from the feasibility of its implementation. Concerning neural networks, hardware implementation has witnessed some remarkable developments [48], but software simulation is still the first-choice method for practical application. In the case of Hopfield networks, some transformation is needed in order to represent continuous networks, modelled by ODEs, on digital computers. Discretization of continuous models is the subject of numerical analysis, thus a successful implementation of Hopfield networks will be served by the choice of an appropriate method, among numerical methods for ODEs, together with the correct selections of step size. In other words, we *intend to know*:

4. Which numerical method preserves the optimization capability after discretization in the best possible way?

Numerical analysts have determined that some families of methods can preserve qualitative properties of the underlying ODE, such as stability or hamiltonian structure. These results are within the framework of an ongoing research that joins numerical analysis and dynamical systems theory [38]. Since optimization is a direct consequence of stability, this theory provides a hint for the search of an optimal numerical method for implementing Hopfield networks.

In view of the persistent bad reputation of Hopfield networks, some researchers have considered that the original model, based on ODEs, is exhausted, and several extensions have been proposed. With this respect, we *intend to know*:

5. Which alternative proposals will result in practical optimization algorithms?

Generally speaking, most of these extended models result from replacing ODEs by functional differential equations, whose solutions are formed by functions, rather than vectors. In particular, this infinite-dimensional framework leads to Stochastic Differential Equations, Delay Differential Equations and Partial Differential Equations. We can only speculate on the practical applicability of these models, since their computational usage is almost unexplored.

Section 2 deals with the first three questions of the enumeration above: we describe the diverse formulations of Hopfield networks and present some theoretical results that suggest their relative merit, as long as optimization is concerned. The application of Hopfield networks to parametric identification is presented too. In Section 3, the discrete-time network is studied, considered as a numerical method that discretizes the continuous network, i.e. the fourth question is addressed. Several extensions of the original model, as stated in the fifth question, are the subject of Section 4. Finally, a summary of the things that we *know*, and those that we *intend to know* in the future, puts an end to the paper in Section 5.

2 The continuous Hopfield network and its application to optimization

2.1 Different formulations for combinatorial optimization

The application of Hopfield networks to optimization problems is based upon the existence of a Lyapunov (also called energy) function, which decreases while the network spontaneously evolves (see [15, 22] for background on dynamical systems and stability). Hence, the stable states of the network coincide to the minima of the energy function. Therefore, optimization is achieved by making the target function identical to the Lyapunov function. Although the original Lyapunov function presented by Hopfield was quadratic, the addition of higher-order terms was later proposed [34, 32]. Nowadays, limiting a contribution to first order networks is a severe restriction, so that general multilinear functions should be considered:

$$V(\mathbf{s}) = -\sum_{q=1}^{r} \sum_{(i_1,i_2\ldots i_q,i_{q+1})\in\mathcal{C}_{\mathcal{N}_n}^{q+1}} w_{i_1\,i_2\ldots i_q\,i_{q+1}}\, s_{i_1}\, s_{i_2}\, \ldots\, s_{i_q}\, s_{i_{q+1}} + \sum_{i=1}^{n} b_i\, s_i \quad (1)$$

where $\mathcal{C}_{\mathcal{N}_n}^{q}$ represent the set of all combinations of q elements chosen from the set $\mathcal{N}_n$ of the first n natural numbers. Then, a particular combinatorial optimization problem is stated by the target function V, together with the constraints $|s_i| = 1$. The crucial contribution by Hopfield consists in constructing a gradient system by defining the network input as the gradient of the target function:

$$net_i(\mathbf{s}) = \frac{\partial\, V}{\partial\, s_i} = \sum_{q=1}^{r} \sum_{(i_1,i_2\ldots i_q)\in\mathcal{C}_{\mathcal{N}_n-\{i\}}^{q}} w_{i\,i_1\,i_2\ldots i_q}\, s_{i_1}\, s_{i_2}\, \ldots\, s_{i_q} - b_i \quad (2)$$

However, from this basic setting, there are several ways to formulate the dynamics of the network.

The discrete Hopfield formulation [16] proceeds both in discrete time and with discrete states $s_i \in \{-1, 1\}$:

$$s_i\,(t+1) = \operatorname{sgn}\left(u_i\,(t+1)\right)\;; \qquad u_i\,(t+1) = net_i(\mathbf{s}(t))$$

The fact that the states of this system always satisfy the constraints $|s_i| = 1$ is, at first sight, an advantage. However, it has been proved both with theoretical and empirical arguments, that the non-convexity of the state space results in a high probability of falling into local minima [45, 21]. Therefore, to the best of our knowledge, the discrete formulation should be considered exhausted, unless a dramatic advance occurs.

The continuous Hopfield formulation [17], used in the first optimization application [41], is defined by an ODE:

$$\frac{d\,u_i}{d\,t} = -u_i + net_i\;; \qquad s_i(t) = \tanh\left(\frac{u_i(t)}{\beta}\right) \tag{3}$$

where β is an adjustable parameter. The Lyapunov function of this system is not exactly equal to that given in Equation (1), but rather it needs the addition of an integral term: $\beta \sum_i \int_0^{s_i} \operatorname{argtanh}(x)\,dx$. This is a significant drawback, since the stable states no longer fulfill the constraint $|s_i| = 1$. Methods to overcome this difficulty have been proposed [44, 21], based upon some strategy to drive β towards zero, while at the same time preserving the smoothness of the hyperbolic tangent or, what is the same, the convexity of the state space. These strategies result in an additional computational cost, and further research is needed in order to determine whether they are computationally affordable in large-size problems.

The continuous Abe formulation [1] is also defined by an ODE:

$$\frac{d\,u_i}{d\,t} = net_i\;; \qquad s_i(t) = \tanh\left(\frac{u_i(t)}{\beta}\right) \tag{4}$$

Many successful applications make use of this formulation (e.g. [40, 29]) although sometimes the Hopfield formulation is misleadingly referenced instead. The Abe formulation has the advantage that its Lyapunov function exactly matches the multilinear form of Equation (1), hence the vertices of the unitary hypercube (i.e. the points $\mathbf{s}$ that satisfy $|s_i| = 1$) are stable. However, there exist interior equilibria that, if stable, would result in spurious solutions. It has been proved that an interior fixed point is unstable, unless the hessian matrix of the function V evaluated at such a point is singular, thus resulting in a non-hyperbolic equilibrium [8]. Therefore, two directions of further research are open: determining whether these non-hyperbolic equilibria actually appear in practical applications and, if they do so, completing the study of their stability.

Despite the need for rigourous theoretical analysis, any eventual methodological advance should be confirmed by simulations and comparison to other

techniques. Hence an useful companion of theory would be the construction of a problem repository, which serves as benchmark, instead of the deprecated TSP [35]. Many interesting applications are reported in [36] and also the solution of diophantine equations has been attempted with Hopfield networks [19].

2.2 Alternative directions of theoretical analysis

Apart from the mentioned directions of theoretical research (evolution strategies for the Hopfield formulation and non-hyperbolic equilibria of the Abe formulation) other aspects of Hopfield networks have also been addressed in a huge number of contributions, e.g. [11]. Most of them aim at weakening the conditions for stability given by Hopfield for first order networks: no self-weights and symmetry of the weight matrix. Undoubtedly, these contributions provide significant advances to the mathematical theory of Hopfield networks. However, their profitability for optimization applications is questionable, due to the limited practical importance of the *symmetry + no self-weights* restriction. Consider that the weights that appear in the network input, Equation (2), are the coefficients of the multilinear target function V, given by Equation (1). For each set of indices $\{i_1, i_2 \dots i_q\}$, there exists only one term $w_{i_1\, i_2 \dots i_q}\, s_{i_1}\, s_{i_2}\, \dots\, s_{i_q}$. Then, two different components of the gradient have different sets of variables, but the same weight $w_{i_1\, i_2 \dots i_q}$, thus justifying the symmetry, even in higher-order networks. With regard to the absence of self-weights, it is explained by the fact that no variable is raised to an exponent greater than one in V. This is due to the constraints $|s_i| = 1$, since for every power s_i^n with $n > 1$, $s_i^n = s_i^{n-2}$ holds. Therefore, the *symmetry + no self-weights* restriction leads to no loss of generality, as long as combinatorial optimization is concerned.

A severe drawback of Hopfield networks for optimization is the appearance of local minima. Even worse, since the constraints are often included as penalty terms of an unconstrained target function, local minima often lead to unfeasible points of the problem. Local minima are an unavoidable consequence of the gradient nature of Hopfield networks, since the states follow a steepest descent trajectory. Hence, a way to overcome this difficulty is the inclusion of some hill-climbing heuristic [36]. In Section 4 below, some stochastic techniques, which could result in local minima avoidance, will be commented on. Within a deterministic setting, the computation of the basins of attraction of stable states is a promising approach, although still at its infancy. The relation of these contributions to optimization stems from the random selection of the initial state: the larger the domain of attraction of a stable fixed point, the more probable the convergence towards that point. In other words, if successive runs are executed with random initial states, the global minimum is achieved sooner if its basin is largest. Some explorations have been presented that aim at relating the basins to the networks weights [5, 10], but no applicable relation has been obtained up to now, let alone a practical algorithm to enlarge the basin of the global optimum. Hence, in our opinion, no evidence supports the optimism of the authors that claim that the local minima problem is solved [39].

An interesting idea is the search for Lyapunov functions different from the

standard one, given by Equation (1). This advance could be aimed at either expanding the range of possible applications or solving the same problem with less computational cost. The paths opened by some contributions [23, 20] have not been continued, and this field remains mostly unexplored.

2.3 Optimization beyond combinatorics: parametric identification

In view of the difficulties that the application of Hopfield networks to combinatorial optimization encounters, one can wonder whether other optimization problems can be more easily solved. Among the long list of other interesting problems that can be thought of, Hopfield networks have been applied to parametric identification of dynamical systems [9], which is formulated as optimization. A remarkable feature of the resulting network is that weights vary with time, so the conventional stability proofs are no longer valid, and a new stability analysis must be performed [6]. A brief and somewhat simplified description of this application follows. The interested reader will found a related model applied to non-parametric estimation in [31].

System identification can be defined as establishing a model of the system, by observing its behaviour. In robotics, the model stems from physical laws, where the numeric values of some parameters may be unknown. Besides, in the models of rigid mechanical systems, the parameters enter linearly, so the model equation can be written as $y = A\,\theta$, where θ is the unknown -possibly time variable- vector of parameters, y is a vector and A is a matrix. Both y and A depend on the eventual inputs p, the states x of the system and their derivatives $\frac{dx}{dt}$. Then, identification is accomplished by parameter estimation, which, in turn, can be considered as the minimization of the prediction error $e = A\,\hat{\theta} - y$ as a function of the estimated parameters $\hat{\theta}$. A straightforward calculation shows that this target function is, indeed, quadratic:

$$V = \frac{1}{2}\,|e|^2 = \frac{1}{2}\,\hat{\theta}^\top\,\mathbf{A}^\top\,\mathbf{A}\,\hat{\theta} - \hat{\theta}^\top\,\mathbf{A}^\top\,\mathbf{y} + V_1 \tag{5}$$

where V_1 is neglected, since it does not depend on $\hat{\theta}$. Therefore, this equation is identical to the standard Lyapunov function of a first order Hopfield network, Equation (1) with $q = 1$. The network states $\mathbf{s}$ represent the estimations $\hat{\theta}$ and the weights are obtained by comparing Equations (1) and (5):

$$\mathbf{W} = -\mathbf{A}^\top\,\mathbf{A}\,; \qquad \mathbf{b} = -\mathbf{A}^\top\,\mathbf{y}$$

In the cited references [9, 6], both theoretical and empirical results are presented to show the efficiency of this *neural estimator*. Yet we are engaged in further research in order to build a complete adaptive control module for robotic systems.

As mentioned above, the Hopfield network that is applied to parameter estimation possesses time-variable weights. The stability analysis of such non-autonomous systems with Lyapunov methods is far from easy. Instead, some advances could result from the application of input-output methods, well known in control engineering and systems theory [43]. The input-output approach has

been successfully applied to recurrent learning algorithms [37] and its application to Hopfield networks is promising, although much work remains to be done.

3 The discrete Hopfield network for optimization

First of all, it must be emphasized that our usage of the term "discrete" only stands for "discrete-time", whereas the neuron states are not discrete since, as mentioned above, networks with binary-valued neurons are severely limited. Hence, the basic model results by replacing the differential $\frac{du}{dt}$ by a finite difference Δu, either in the Hopfield formulation, Equation (3):

$$u_i(t+1) = net_i(\mathbf{s}(t)) \,; \qquad s_i(t+1) = \tanh\left(\frac{u_i(t+1)}{\beta}\right)$$

or in the Abe formulation, Equation (4):

$$u_i(t+1) = u_i(t) + net_i(\mathbf{s}(t)) \,; \qquad s_i(t+1) = \tanh\left(\frac{u_i(t+1)}{\beta}\right)$$

Both practical and theoretical results have been presented concerning the discrete network. On one hand, among papers with problem-dependent heuristics and empirical results, the discrete model is often disregarded or even unmentioned. This is rather disappointing, since computer simulation of a continuous model always implies some discretization process. For instance, the fact that the energy may increase has been attributed to a mixture of discretization, updating rule and presence of self-weights [14], whereas the influence of these factors should be clearly distinguished. On the other hand, some theoretical contributions are inspiring regarding mathematical methods [42] but, as in the continuous case, we are not particularly interested in generalizations such as those concerning non-symmetric weights.

The discrete network can be considered as a numerical method that solves the ODE of the continuous model. This is a promising approach, since the application of the continuous network to optimization is better studied. Generally speaking, the discrete model would then have the structure:

$$u_i(t+1) = f(\mathbf{s}(t); \Delta\, t) \,; \qquad s_i(t+1) = \tanh\left(\frac{u_i(t+1)}{\beta}\right) \tag{6}$$

where the function f depends on the choice of numerical method and Δt is the step size. Both f and Δt should be adjusted so that the stable states of the dynamical system defined by Equation (6) are the same as those of the continuous network. Interestingly, the design of numerical methods that mirror qualitative properties of ODEs is an active line of research [38], but no efficient numerical method is yet known that always preserves a Lyapunov function. When the Hopfield formulation is discretized by means of the Euler rule, the eventual appearance of periodic solutions that destroy convergence has been proved [46] (see also references therein). Stability can be guaranteed when neurons are updated

sequentially, but this asynchronous activation is not computationally efficient. Other proposed conditions so that the Lyapunov function is preserved [46, 30] are not applicable when β is too small. Hence, the analysis of the discretization of the Hopfield formulation is hindered by the usage of strategies that drive β towards zero, which are needed in combinatorial optimization. In the case of the Abe formulation, also the Euler rule leads to periodic solutions, unless a small enough step size is selected, and the analysis has been extended to higher-order networks [7]. It seems that the only way to guarantee the stability of the discrete model, regardless the step size, is to select an alternative numerical method. Some explorations with implicit methods have been reported in this direction [2, 3, 4], however the design of a numerical method that both is computationally efficient and preserves the Lyapunov function is an open fundamental question.

4 Generalized models of Hopfield networks

Several extensions of the original continuous model have been proposed, among which the stochastic networks are presented first. The introduction of randomness has long been known in the form of Boltzmann machines, where the sigmoid function is replaced by a stochastic decision. The aim of randomness was to provide some way to escape from local minima, but the convergence of these models turns out to be too slow. Instead, we here consider the addition of stochastic noise to the weights [18]. In the case of the Hopfield formulation, Equation (3), the following model results from considering randomness:

$$d\,u_i = (-u_i + net_i)\,dt + \sigma(u)\,d\,W_t$$

where σ is a noise intensity matrix and W_t is a Brownian motion. The stochastic network preserves the Lyapunov function of the continuous network and it has been suggested as a model of hardware implementations, but it could also serve as a hill-climbing algorithm to escape from local minima. The stochastic extensions of the Abe formulation and higher order networks deserve further research.

Another proposed extension results from the introduction of delays in the first-order Hopfield formulation, leading to the following model:

$$\frac{d\,u_i}{d\,t} = -u_i(t) + \sum_j w_{ij}\,s_j(t-\tau_{ij}) + b_i$$

where τ_{ij} is a delay associated to the connection from neuron j to neuron i. A complete stability analysis of models with delays has recently been published [28], whereas the discretization of delayed networks has also been studied [30].

Finally, a sort of network with infinite neurons results from the introduction of partial derivatives [33]:

$$\frac{\partial\,u(x,t)}{\partial\,t} = -u(x,t) + \tanh\left(\int_\Omega w(x,y)\,u(y,t)\,dy\right)$$

for x belonging to some domain Ω. The stability of this and a similar model [27] has been proved, but its computational usefulness is unexplored.

5 Conclusions

This paper reviews the main topics concerning the application of Hopfield networks to optimization. Both current results and promising directions for further research have been presented. With regard to the continuous Hopfield network, in principle, it is capable of performing combinatorial optimization, but some issues still need study: evolution strategies for the Hopfield formulation and non-hyperbolic equilibria of the Abe formulation. Besides, the determination of the basins of attraction and its relation to local minima, as well as the obtention of different Lyapunov functions, are research programmes that are worth undertaking. The Hopfield network is also capable of solving other classes of optimization problems, and an efficient application to parametric identification of dynamical systems is presented. The analysis of the resulting non-autonomous network could be enhanced with the use of input-output methods. The discrete network can be attained by the application of a numerical method to the continuous model. Since the Euler rule does not preserve stability, it has been suggested that implicit numerical methods are more appropriate, but results are far from conclusive. Proposed extensions of the original network include stochastic differential equations, models with delays and partial differential equations, whose importance for optimization applications remains unexplored.

Hopfield networks comprise a variety of mathematically appealing systems, which highlight the most interesting dynamical and numerical aspects of neural computing. Through an interdisciplinary research, they can also become an efficient computational method for optimization.

References

[1] S. Abe. Theories on the Hopfield Neural Networks. In *Proc. IEE International Joint Conference on Neural Networks*, volume I, pages 557–564, 1989.

[2] M. A. Atencia, G. Joya, and F. Sandoval. Numerical implementation of continuous Hopfield networks for optimization. In *Proc. European Symposium on Artificial Neural Networks (ESANN 2001)*, pages 359–364, 2001.

[3] M. A. Atencia, G. Joya, and F. Sandoval. Continuous-state Hopfield Dynamics Based on Implicit Numerical Methods. In J. R. Dorronsoro, editor, *Artificial Neural Networks (ICANN 2002). Lecture Notes in Computer Science*, volume 2415, pages 1365–1370. Springer, 2002.

[4] M. A. Atencia, G. Joya, and F. Sandoval. Combinatorial Optimization via Discrete Gradient Implementation of Hopfield Neural Networks. In *6th International Conference of Operations Research (OR 2003), Havana (Cuba)*, 2003.

[5] M. A. Atencia, G. Joya, and F. Sandoval. Spurious minima and basins of attraction in higher-order Hopfield networks. In J. Mira and J. R. Álvarez, editors, *Computational Methods in Neural Modeling. Lecture Notes in Computer Science (IWANN 2003)*, volume 2686, pages 350–357. Springer, 2003.

[6] M. A. Atencia, G. Joya, and F. Sandoval. Parametric Identification of Robotic Systems with Stable Time-varying Hopfield Networks. *Neural Computing & Applications*, 13(4):270–280, 2004.

[7] M. A. Atencia, G. Joya, and F. Sandoval. Discretization of Continuous Higher Order Hopfield Networks for Combinatorial Optimization. *Neural Processing Letters (submitted)*, 2005.

[8] M. A. Atencia, G. Joya, and F. Sandoval. Dynamical Analysis of Continuous Higher Order Hopfield Networks for Combinatorial Optimization. *Neural Computation (in press)*, 2005.

[9] M. A. Atencia, G. Joya, and F. Sandoval. Hopfield Neural Networks for Parametric Identification of Dynamical Systems. *Neural Processing Letters (in press)*, 2005.

[10] J. Cao. An estimation of the domain of attraction and convergence rate for Hopfield continuous feedback neural networks. *Physics Letters A*, 325:370.374, 2004.

[11] T. Chen and S.-I. Amari. New theorems on global convergence of some dynamical systems. *Neural Networks*, 14:251–255, 2001.

[12] M. Cohen and S. Grossberg. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Trans. on Systems, Man and Cybernetics*, 13(5):815–826, 1983.

[13] M. Cottrell, J. Fort, and G. Pagès. Two or three things that we know about the Kohonen algorithm. In *Proc. European Symposium on Artificial Neural Networks*, pages 235–244, 1994.

[14] G. Feng and C. Douligeris. On the convergence and parameter relation of discrete-time continuous-state Hopfield networks with self-interaction neurons. *IEICE Trans. Fundamentals*, E84-A(12):3162–3173, 2001.

[15] M. W. Hirsch and S. Smale. *Differential Equations, Dynamical Systems, and Linear Algebra*. Academic Press, 1974.

[16] J. Hopfield. Neural Networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA*, 79:2554–2558, 1982.

[17] J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci. USA*, 81:3088–3092, 1984.

[18] S. Hu, X. Liao, and X. Mao. Stochastic Hopfield neural networks. *Journal of Physics A: Mathematical and General*, 36:2235–2249, 2003.

[19] G. Joya, M. A. Atencia, and F. Sandoval. Application of High-order Hopfield Neural Networks to the Solution of Diophantine Equations. In A. Prieto, editor, *Artificial Neural Networks. Lecture Notes in Computer Science (IWANN 1991)*, volume 540, pages 395–400. Springer, 1991.

[20] G. Joya, M. A. Atencia, and F. Sandoval. Associating arbitrary-order energy functions to an artificial neural network. Implications concerning the resolution of optimization problems. *Neurocomputing*, 14:139–156, 1997.

[21] G. Joya, M. A. Atencia, and F. Sandoval. Hopfield Neural Networks for Optimization: Study of the Different Dynamics. *Neurocomputing*, 43(1-4):219–237, 2002.

[22] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, 2002.

[23] Y. Kobuchi. State Evaluation Functions and Lyapunov Functions for Neural Networks. *Neural Networks*, 4:505–510, 1991.

[24] T. Kohonen. Analysis of a Simple Self-Organizing Process. *Biological Cybernetics*, 43(1):59–69, 1982.

[25] T. Kohonen. Self-Organized Formation of Topologically Correct Feature Maps. *Biological Cybernetics*, 44(2):135–140, 1982.

[26] T. Kohonen. *Self-Organizing Maps*. Springer, 2001.

[27] K. Leblebicioglu, U. Halici, and O. Celebi. Infinite dimensional Hopfield neural networks. *Nonlinear Analysis-Theory Methods & Applications*, 47(9):5807–5813, 2001.

[28] X. Liao, C. Li, and K.-w. Wong. Criteria for exponential stability of Cohen-Grossberg neural networks. *Neural Networks*, 17(10):1401–1414, 2004.

[29] S. Matsuda. "Optimal" Neural Representation of Higher Order for Traveling Salesman Problems. *Electronics and Communications in Japan,Part 2*, 85(9):32–42, 2002.

[30] S. Mohamad and K. Gopalsamy. Dynamics of a class of discrete-time neural networks and their continuous-time counterparts. *Mathematics and Computers in Simulation*, 53(1-2):1–39, 2000.

[31] A. S. Poznyak, E. Sanchez, and W. Yu. *Differential Neural Networks for Robust Nonlinear Control*. World Scientific Publishing, 2001.

[32] T. Samad and P. Harper. High-order Hopfield and Tank optimization networks. *Parallel Computing*, 16:287–292, 1990.

[33] E. C. Segura and R. P. Perazzo. Associative Memories in Infinite Dimensional Spaces. *Neural Processing Letters*, 12(2):129–144, 2000.

[34] T. J. Sejnowski. Higher-Order Boltzmann Machines. In J. S. Denker, editor, *Neural Networks for Computing*, pages 398–403. American Institute of Physics, 1986.

[35] K. A. Smith. An argument for abandoning the travelling salesman problem as a neural-network benchmark. *IEEE Transactions on Neural Networks*, 7(6):1542–1544, 1996.

[36] K. A. Smith. Neural networks for combinatorial optimization: A review of more than a decade of research. *INFORMS Journal on Computing*, 11(1):15–34, 1999.

[37] J. J. Steil. Local stability of recurrent networks with time-varying weights and inputs. *Neurocomputing*, 48(1-4):39–51, 2002.

[38] A. Stuart and A. Humphries. *Dynamical systems and numerical analysis*. Cambridge University Press, 1996.

[39] Y. Takahashi. A neural network theory for constrained optimization. *Neurocomputing*, 24(1-3):117–161, 1999.

[40] Y. Takefuji and K. C. Lee. Artificial Neural networks for Four- Coloring Map Problems and K-Colorability Problems. *IEEE Trans. on Circuits and Systems*, 38(3):326–333, 1991.

[41] D. Tank and J. Hopfield. 'Neural' Computation of Decisions in Optimization Problems. *Biological Cybernetics*, 52:141–152, 1985.

[42] P. Tiňo, B. Horne, and C. Giles. Attractive Periodic Sets in Discrete-Time Recurrent Networks (with Emphasis on Fixed-Point Stability and Bifurcations in Two-Neuron Networks). *Neural Computation*, 13(6):1379–1414, 2001.

[43] M. Vidyasagar. *Nonlinear systems analysis*. Prentice Hall, 1993.

[44] M. Vidyasagar. Minimum-seeking properties of analog neural networks with multilinear objective functions. *IEEE Trans. on Automatic Control*, 40(8):1359–1375, 1995.

[45] M. Vidyasagar. Are Analog Neural Networks Better Than Binary Neural Networks? *Circuits, Systems and Signal Processing*, 17(2):243–270, 1998.

[46] X. Wang, A. Jagota, F. Botelho, and M. Garzon. Absence of Cycles in Symmetric Neural Networks. *Neural Computation*, 10:1235–1249, 1998.

[47] G. Wilson and G. Pawley. On the Stability of the Travelling Salesman Problem Algorithm of Hopfield and Tank. *Biological Cybernetics*, 58(1):63–70, 1988.

[48] J. Zhu and P. Sutton. FPGA Implementations of Neural Networks - A Survey of a Decade of Progress. In P. Y. Cheung, G. A. Constantinides, and J. T. de Sousa, editors, *Field-Programmable Logic and Applications. Lecture Notes in Computer Science (FPL 2003)*, volume 2778, pages 1062–1066. Springer-Verlag, 2003.