

Exploring a model of far-from-equilibrium computation

Răzvan V. Florian
Center for Cognitive and Neural Studies (Coneural)
Str. Saturn 24, 400504 Cluj-Napoca, Romania
florian@coneural.org

July 10, 2005

Abstract

We explore the possibility of implementing a conjectured far-from-equilibrium computational model, where the informational input determines the limit condition of a system kept in a nonequilibrium state by another input flux, and the output of the computation is determined by a linear readout that interprets the self-organization state of the system. We tried two implementations, one where the system is a network of threshold gates, and one where the system is a model of a physical liquid subjected to Rayleigh-Bénard convection rolls. We were not able to implement the conjectured computational model.

1 Introduction

Many physical systems that are kept in a far-from-equilibrium state by a flux of energy or matter tend to self-organize. Since self-organization is a desirable property for computational systems, we are interested to design computational systems that function analogously to nonequilibrium systems. We have previously defined several nonequilibrium computational models (Florian and Dumitrescu, 2004). In this paper, we will explore the feasibility of the previously proposed N_4 computational model.

2 The conjectured N_4 computational model

We consider a computational system that is kept far-from-equilibrium by a continuous flux. If this flux is steady, the system may converge to a nonequilibrium steady state (NESS). In physical systems, the NESS is often a self-organized state, with long range spatial correlations or temporal oscillations. Since a computational system (e.g., a randomly connected neural network) does not always have a topology

that allows the definition of spatial correlations, we conjectured that in such systems the state of self-organization is observer dependent (Florian and Dumitrescu, 2004). This conjecture is supported by the liquid state machine, the N_3 nonequilibrium computational model, where a linear readout is trained to discriminate between different far-from-equilibrium states of a complex nonlinear system (Maass et al., 2002; Jaeger, 2001), thus effectively defining different observer-dependent NESSs of the system.

The long-range spatial correlations that appear in self-organized physical systems make the systems sensitive to macroscopic spatial constraints. Thus, macroscopic constraints can influence the structure of the NESS. By analogy, we conjectured the N_4 computational model, based on a nonequilibrium informational system, where the input of the computation corresponds to macroscopic constraints imposed to the system, while the output is the state of the system (possibly a NESS), or a function of it. The system has an input flux that keeps it out of equilibrium, but which carries no information relevant to the computation.

3 An first implementation of the N_4 computational model

3.1 The liquid

In our implementation, we choose the computational system to be a random recurrent network of McCulloch-Pitts neurons (also called threshold gates), following the studies of Natschläger et al. (Natschläger et al., 2004; Bertschinger and Natschläger, 2004; Natschläger and Bertschinger, 2004) who have investigated and used this system as a liquid for the liquid state machine. The network contains N_a active neurons and N_s static neurons, all having states $x_i \in \{0, 1\}$. The static neurons have the role of limit constraints imposed to the system. Their states are fixed to the values of the input of the computation. The states of the active neurons are dynamic. Each active neuron receives incoming weights w_{ij} from K nodes randomly chosen between the active and the static neurons. The weights are randomly drawn from a Gaussian distribution with mean μ and variance σ . The active neurons are also driven by an external input flux u that is injected into each of them. In contrast with the experiments of Natschläger et al., where this was the input of the computation, in our case it does not carry any information and it just serves as a flux that keeps the system in a nonequilibrium state.

Hence, the dynamics of the active neurons is given by

$$x_{i,t+1} = \Theta \left(\sum_j w_{ij} \cdot x_{j,t} + u_{i,t} \right) \quad (1)$$

with

$$\Theta(h) = \begin{cases} 1, & \text{if } h \geq 0 \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

The input flux $u_{i,t}$ is random, having the value $u_0 + 1$ with probability r or the value u_0 with probability $1 - r$. In some cases, we used the same input for all the neurons, $u_{i,t} = u_t$, as in the studies of Natschläger et al.

3.2 The readout

As a readout, we have used a parallel perceptron (henceforth named paralon) (Auer et al., 2002). A paralon consists of an odd number n_p of perceptrons, and the output of the paralon is given by the “vote” of the majority of its composing perceptrons. It was shown that the performance of paralons is comparable to that of backpropagation, but their implementation and computational needs are more favourable. We used them because they were already used as readouts for liquid state machines (Maass et al., 2002) and because we can interpret them as giving the result of the “vote” of several different linear observers that assess whether the liquid has reached their preferred state of self-organization. In our implementation we have used the version of the algorithm that does batch updates of the weights, adaptive η and fixed $\gamma = 0.05$ (see Auer et al. (2002) for details). We considered the learning to converge when the relative change in the value of the error function of the p-delta learning rule was less than 0.001.

3.3 The problem

We have tested the proposed computational model on an image classification problem, using the ATT database of faces (Samaria and Harter, 1994; Paredes et al., 2001), a dataset that can be freely downloaded from the internet¹. The faces belong to 40 persons / classes, and there are 10 photos / instances per class, for a total of 400 instances. The first 5 images from each class were used as the training sample, and the other images as the test sample. The size of each image is 92 x 112 pixels, with 256 grey levels per pixel. In our experiments, we have monochromatized the images by setting to 1 all pixels with a grey level greater than 122, and to 0 the other pixels. This was needed because the liquid was composed of units with binary output.

3.4 Experiments

In the following, we used as a liquid a network with $K = 5$, $\mu = -0.2$, $\sigma = 1.14$, and separate input for each neuron. The readout was composed of 40 paralons with $n_p = 3$ perceptrons, each paralon corresponding to a class. For each classification, the active neurons of the liquid were initialized to random states and the static neurons were clamped to the values of the pixels of the image. The structure of the network, although random, is kept the same during each learning experiment. The liquid was then updated for a number of $t_s = 100$ timesteps. This was a so-called shuffling period, when the limit conditions established by the informational input

¹<http://www.uk.research.att.com/facedatabase.html>

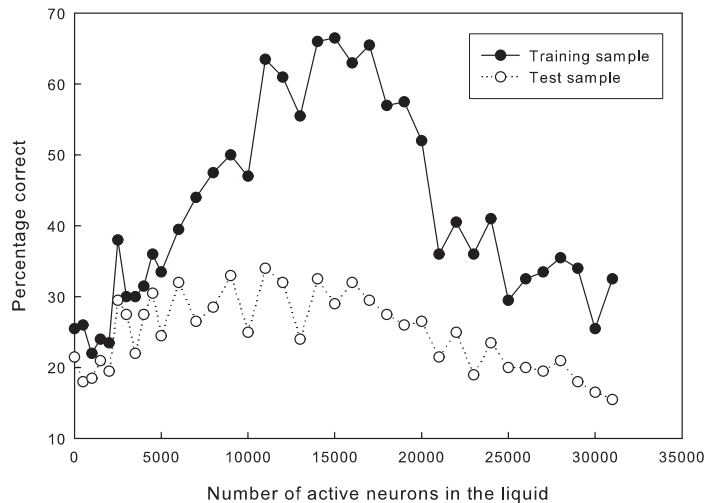


Figure 1: The percentage of the instances that were classified correctly by the system, as a function of the liquid size (the number of active neurons, N_a). The initial value of the learning rate of the paralons was set to $\eta = 0.1$.

were allowed to influence the dynamics of the liquid, and thus its state. During this time, the liquid was also under the influence of the input flux that kept it out of equilibrium. The readout was then fed with the state of the entire network (the states of both the static and the active neurons). We considered that the system classified correctly an image if the paralon corresponding to that image, and only it, had the highest number of perceptrons with output 1, among all paralons. This sequence was performed both during learning and during testing. During learning, this sequence was performed for each example instance, repeatedly for several cycles, until convergence. During testing, this sequence was performed for each test instance, to count the number of instances that were classified correctly.

Some initial experiments, where the initial value of the learning rate of the paralons was set to $\eta = 0.1$, seemed to confirm the efficacy of the proposed computational model. Fig. 1 illustrates the percentage of the instances that were classified correctly by the system, as a function of the liquid size (the number of active neurons). Further investigations revealed, however, that a much smaller initial value of η , 0.0005, led to better overall recognition, but that in this case the recognition is not much better when the size of the liquid increases. This is illustrated in Fig. 2. Moreover, the input flux that keeps the system out of equilibrium does not contribute favourably to the computational performance. Instead of leading to self-organization, it seems just to act as noise injected in the system, that disturbs it. This is illustrated in Fig. 3.

In conclusion, this experiment did not confirm the validity of the proposed

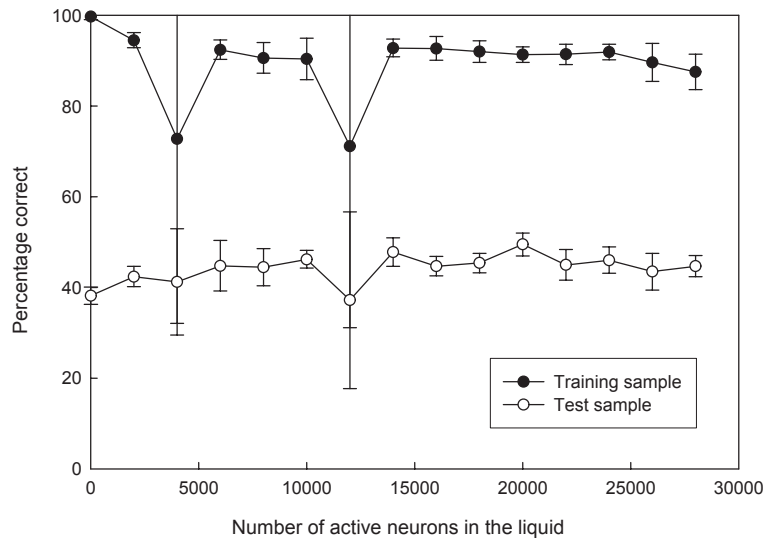


Figure 2: The percentage of the instances that were classified correctly by the system, as a function of the liquid size (the number of active neurons, N_a). The initial value of the learning rate of the parolons was set to $\eta = 0.0005$. Data is averaged over 5 experiments; the error bars represent standard deviations. The large deviations for the two data points may indicate some resonance phenomena.

computational model.

4 Other experiments

We have tried to improve the performance of the system by varying its different parameters. The previous experimental setup required much computing time for investigation (this was due to the large size of the liquid, required by the size of the input - the number of the pixels in the image; and by the large number of parolons, determined by the number of classes), for the order of 1 day of continuous running on a standard PC for obtaining a graph like Fig. 2. This did not allow rapid experimentation with different parameters. We then explored a different problem, computation of 3-parity. In this case, the input is a vector of just 3 values and we have to train a single paralon. However, in this case the problem is statistically neutral, and we cannot separate the input into a training sample and a test sample; hence, the learning efficacy is assessed just on the training sample, that includes all instances.

We have tested the influence of our choice of $u_{i,t}$ varying with i , versus the choice of $u_{i,t} = u_t$. We found no particular influence on the computational performance.

We have also tested the influence of having a fixed (although randomly gener-

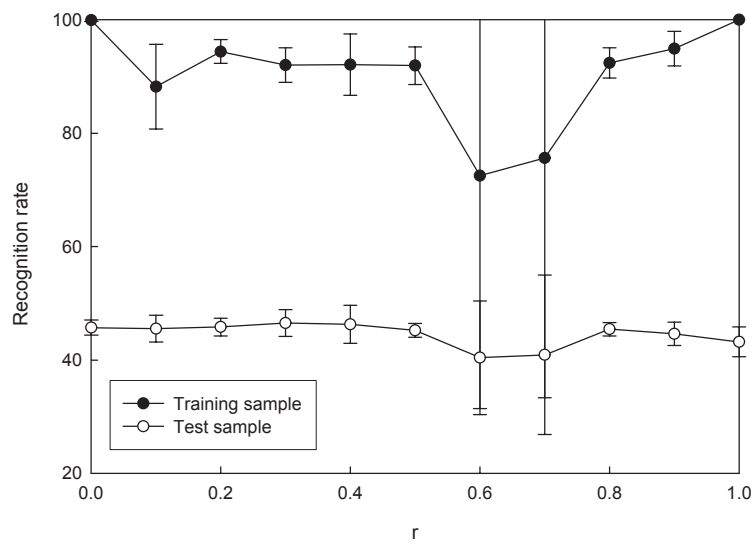


Figure 3: The percentage of the instances that were classified correctly by the system, as a function of the probability r that the input flux has the value $u_0 + 1$. Data is averaged over 5 experiments; the error bars represent standard deviations. It can be seen that the input flux acts more like disturbing noise, rather than helping the self-organization of the liquid and thus the computational performance of the system.

ated initially) u_t time series for all the shuffling periods, instead of regenerating it randomly for each shuffling. This was to verify the hypothesis that a fixed input flux would not have the effect of just noise, and would contribute to information processing. We found no significant influence on the computational performance.

We have also modified the afferent connectivity of the active neurons of the liquid. Instead of having 5 afferent connections from neurons randomly chosen between the static and active neurons, we established that at least 2 incoming connections would be from the static neurons, i.e. the input. In this case, the recognition performance increased significantly. This means that previously the input information was not redistributed efficiently to all active neurons through self-organization and long-range correlations, as hypothesised. If the input information was distributed through direct connections, the liquid acted efficiently as a nonlinear kernel that allowed a linear readout to perform the computation. However, this implies that our initial hypothesis that long-range correlations, due to nonequilibrium self-organization, would distribute information in the whole liquid, was wrong.

We have also tested the influence of the various regimes of the liquid (ordered / critical / chaotic; see (Natschläger et al., 2004; Bertschinger and Natschläger, 2004; Natschläger and Bertschinger, 2004)). We found that recognition performance was better for greater values of σ , but it was not influenced by the critical regime (at the edge of chaos) of the liquid, corresponding to some intermediate values of σ .

In conclusion, these experiments did not manage to confirm the efficiency of the conjectured computational model.

5 An second implementation of the N_4 computational model

The previous implementation of the N_4 computational model, which assumed that the input flux would induce a state of self-organization with long range correlations, was not confirmed by the experiments. We then tried an implementation of the N_4 model closer to the physical systems that inspired this model. We used a physical model of the self-organization of a liquid in nonequilibrium that models the formation of the Bénard rolls. According to the N_4 computational model, the informational input could be coded as limit conditions — in this case, the shape of the container holding the liquid — and the computation is performed by the self-organized liquid and by a linear readout that gets information from the whole liquid.

A model for the formation of Rayleigh-Bénard rolls is the Swift-Hohenberg equation

$$\partial\phi/\partial t = [\varepsilon - (\nabla^2 + 1)^2]\phi - \phi^3, \quad (3)$$

where $\phi(\mathbf{r}, t)$ can be interpreted as a fluid field in a given horizontal plane, e.g., the vertical velocity component in the midplane of the convection cell (Greenside and Coughran, 1984; Cross and Hohenberg, 1993).

Numerical simulations from physics usually use special geometries (rectangular, annular, periodic conditions) that permit special numerical algorithms for

the simulation of the Swift-Hohenberg equation, e.g. (Chiam et al., 2003; Chiam, 2004; Cross et al., 2001; Greenside and Coughran, 1984; Sensoy, 1999). These algorithms are not useful for our purpose, since we would like to vary the shape of the container holding the liquid. Another approach is to model the fluid with molecular dynamics (Rapaport, 1988); this is not suitable for our purpose because this simulation is time consuming. There are a few studies that simulated the dynamics of the Swift-Hohenberg equation on a discrete lattice, however the information in the corresponding papers is not sufficient for reproducing accurately the same simulation (Viñals et al., 1991; Xi et al., 1991; Elder et al., 1992).

We modeled the Swift-Hohenberg dynamics on a square lattice of size $N \Delta l$, with timestep Δt . The only tricky issue in the simulation is the way in which the term $(\nabla^2 + 1)^2 \phi$ is discretized. We have

$$\begin{aligned} (\nabla^2 + 1)^2 \phi &= \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + 1 \right)^2 \phi \\ &= \left[\frac{\partial^4}{\partial x^4} + \frac{\partial^4}{\partial y^4} + 2 \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial x^2} \frac{\partial^2}{\partial y^2} \right) + 1 \right] \phi. \end{aligned} \quad (4)$$

We first made the discretization of this equation by expanding in a Taylor series the values of ϕ in the 12 lattice neighbours of the point of coordinates (i, j) . This led to the following formulas:

$$\left(\frac{\partial^4}{\partial x^4} + \frac{\partial^4}{\partial y^4} \right) \phi_{i,j} \approx \frac{c_2 - 4c_1 + 12\phi_{i,j}}{\Delta l^4} + \mathcal{O}(\Delta l^2) \quad (5)$$

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \phi_{i,j} \approx \frac{-c_2 + 16c_1 - 60\phi_{i,j}}{12\Delta l^2} + \mathcal{O}(\Delta l^4) \quad (6)$$

$$\frac{\partial^2}{\partial x^2} \frac{\partial^2}{\partial y^2} \phi_{i,j} \approx \frac{d_1 - 2c_1 + 4\phi_{i,j}}{\Delta l^4} + \mathcal{O}(\Delta l^2) \quad (7)$$

where

$$c_1 = \phi_{i,j+1} + \phi_{i,j-1} + \phi_{i+1,j} + \phi_{i-1,j} \quad (8)$$

$$c_2 = \phi_{i,j+2} + \phi_{i,j-2} + \phi_{i+2,j} + \phi_{i-2,j} \quad (9)$$

$$d_1 = \phi_{i+1,j+1} + \phi_{i+1,j-1} + \phi_{i-1,j+1} + \phi_{i-1,j-1}. \quad (10)$$

This results in the following update equation:

$$\begin{aligned} \phi_{i,j}(t + \Delta t) &= \phi_{i,j}(t) + \Delta t \left[(\varepsilon - 1) \phi_{i,j}(t) - \phi_{i,j}(t)^3 - \right. \\ &\quad \left. \left(\frac{2d_1 + c_2 - 8c_1 + 20\phi_{i,j}(t)}{\Delta l^4} + \frac{-c_2 + 16c_1 - 60\phi_{i,j}(t)}{6\Delta l^2} \right) \right] \end{aligned} \quad (11)$$

where c_1 , c_2 , d_1 are evaluated at coordinates (i, j) and time t . These equations in 2D are consistent with the 1D discrete equation reported in (Viñals et al., 1991).



Figure 4: Examples of the patterns generated by the Swift-Hohenberg equation, modeling Rayleigh-Bénard convection rolls in a liquid. We simulated them in a container having a shape modulated by 4 binary inputs. The square marked with a thick line in the outline on the right represents the area that was connected to the readout.

Alternatively, we have also used the following discretization method:

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \phi_{i,j} \approx \frac{c_1 + 2d_1 - 12\phi_{i,j}}{12\Delta l^2} \equiv \delta_{i,j} \quad (12)$$

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \phi_{i,j} \approx \frac{c_{1,\delta} + 2d_{1,\delta} - 12\delta_{i,j}}{12\Delta l^2}, \quad (13)$$

following (Elder et al., 1992), where

$$c_{1,\delta} = \delta_{i,j+1} + \delta_{i,j-1} + \delta_{i+1,j} + \delta_{i-1,j} \quad (14)$$

$$d_{1,\delta} = \delta_{i+1,j+1} + \delta_{i+1,j-1} + \delta_{i-1,j+1} + \delta_{i-1,j-1}. \quad (15)$$

Thus,

$$\begin{aligned} \phi_{i,j}(t + \Delta t) = & \phi_{i,j}(t) + \Delta t \left[(\varepsilon - 1)\phi_{i,j}(t) - \phi_{i,j}(t)^3 - \right. \\ & \left. \left(\frac{c_{1,\delta} + 2d_{1,\delta} - 12\delta_{i,j}}{12\Delta l^2} + 2\delta_{i,j} \right) \right]. \end{aligned} \quad (16)$$

We have tested the proposed computational model in a simulation inside a lattice with $N = 100$; the parameters of the simulation were $\Delta l = 0.8$, $\Delta t = 0.1$, $\varepsilon = 0.3$. The problem we proposed was the computation of 4-parity. The 4 binary inputs modulated the shape of the container in which the liquid was simulated, and a linear, trained readout (a paralon) that was connected to part of the liquid was supposed to detect the parity of the input. Each input changed the shape of one of the 4 sides of the container; depending on the value of the input, the side was either a rectangle or part of a circle (see Fig. 4). The liquid was set in a random initial state (ϕ was generated randomly in the interval $[-0.05, 0.05]$), and then it was left to evolve for 2000 timesteps according to the Swift-Hohenberg equation. Then the readout was connected to the liquid and was trained to generate the correct signal, given the input.

Unfortunately, the readout was not able to converge during training to learn to solve the proposed problem.

6 Conclusion

We were not able to implement the hypothesized N_4 nonequilibrium computational model. Possible reasons are that the limit conditions do not influence the NESS in a consistent manner, detectable by a linear readout; and, in the case of the first implementation, that the network does not enter in a state of self-organization where the limit conditions influence the bulk of the network. We have used, in the first implementation, a network that was successfully used as a liquid in the framework of the liquid state machine. We have also used, in the second implementation, a classical model of self-organization far-from-equilibrium. Since the conjectured behavior of the computational system did not appear in these implementations, it is hard to see how the proposed computational model can be implemented in another way. It is possible that the self-organization of the nonequilibrium steady states that appear in some systems (e.g., the convection rolls) does not appear in a meaningful way in systems without spatial topology (e.g., neural networks), or appears in a way that is not detectable by a linear readout in a generic system.

References

- Auer, P., Burgsteiner, H. M. and Maass, W. (2002), ‘The p-delta learning rule for parallel perceptrons’.
http://www.igi.tugraz.at/maass/p_delta_learning.pdf
- Bertschinger, N. and Natschläger, T. (2004), ‘Real-time computation at the edge of chaos in recurrent neural networks’, *Neural Computation* **16**, 1413–1436.
<http://www.cis.tugraz.at/igi/tnatschl/psfiles/eoc-nc-preprint.pdf>
- Chiam, K. H. (2004), Spatiotemporal Chaos in Rayleigh-Bénard Convection, PhD thesis, California Institute of Technology, Pasadena, CA.
<http://etd.caltech.edu/etd/available/etd-08062003-162208/unrestricted/thesis.pdf>
- Chiam, K. H., Lai, M. C. and Greenside, H. S. (2003), ‘Efficient algorithm on a non-staggered mesh for simulating Rayleigh-Bénard convection in a box’, *Physical Review E* **68**, 026705.
<http://arxiv.org/abs/nlin/0302057>
- Cross, M. C. and Hohenberg, P. C. (1993), ‘Pattern formation outside of equilibrium’, *Reviews of Modern Physics* **65**.
- Cross, M. C., Louie, M. and Meiron, D. (2001), ‘Finite size scaling of domain chaos’, *Physical Review E* **63**, 045201.
<http://arxiv.org/abs/nlin/0011048>
- Elder, K. R., Viñals, J. and Grant, M. (1992), ‘Dynamic scaling and quasi-ordered states in the two dimensional Swift-Hohenberg equation’, *Physical Review A*

- 46(12), 7618–7629.
<http://arxiv.org/abs/cond-mat/9205015>
- Florian, R. V. and Dumitrescu, D. (2004), Far-from-equilibrium computation, Technical report, Center for Cognitive and Neural Studies, Cluj, Romania.
<http://coneural.org/reports/Coneural-04-01.php>
- Greenside, H. S. and Coughran, W. M. (1984), ‘Nonlinear pattern formation near the onset of Raileigh - Bénard convection’, *Physical Review A* **30**, 398–428.
- Jaeger, H. (2001), ‘The echo state approach to analysing and training recurrent neural networks’, *GMD Report 148, German National Research Center for Information Technology* .
ftp://borneo.ais.fraunhofer.de/pub/indy/publications_herbert/EchoStatesTechRep.pdf
- Maass, W., Natschlaeger, T. and Markram, H. (2002), ‘Real-time computing without stable states: A new framework for neural computation based on perturbations’, *Neural Computation* **14**, 2531–2560.
<http://www.cis.tugraz.at/igi/maass/psfiles/ms2469-figs.pdf>
- Natschläger, T. and Bertschinger, N. (2004), ‘Supplementary information to the mean-field theory for randomly connected recurrent networks of threshold gates’.
<http://www.igi.tugraz.at/tnatschl/edge-of-chaos/mean-field-supplement.pdf>
- Natschläger, T., Bertschinger, N. and Legenstein, R. (2004), ‘At the edge of chaos: Real-time computations and self-organized criticality in recurrent neural networks’, *Proceedings of NIPS 2004, Advances in Neural Information Processing Systems* .
<http://www.cis.tugraz.at/igi/tnatschl/psfiles/eoc-nips04-submission.pdf>
- Paredes, R., Pérez, J. C., Juan, A. and Vidal, E. (2001), Face recognition using local representations and a direct voting scheme, in ‘Proceedings of the 9th Spanish Symposium on Pattern Recognition and Image Analysis’.
<http://www.iti.upv.es/groups/riva/papers/2001/Juan01c/Attachment00074761/Juan01c.ps.gz>
- Rapaport, D. C. (1988), ‘Molecular-dynamics study of Rayleigh-Bénard convection’, *Physical Review E* **60**(24), 2480–2483.
- Samaria, F. and Harter, A. (1994), Parameterisation of a stochastic model for human face identification, in ‘Proceedings of 2nd IEEE Workshop on Applications of Computer Vision’.
<http://www-lce.eng.cam.ac.uk/publications/files/paper.95.2.ps.Z>

- Sensoy, B. A. (1999), Pattern Formation of a Convecting Fluid in an Annular Domain, PhD thesis, Duke University.
<http://www.phy.duke.edu/undergraduate/thesis/sensoy/thesis.pdf>
- Viñals, J., Hernández-García, E., Miguel, M. S. and Toral, R. (1991), 'Numerical study of the dynamical aspects of pattern selection in the stochastic Swift-Hohenberg equation in one dimension', *Physical Review A* **44**(2), 1123–1133.
- Xi, H., Viñals, J. and Gunton, J. D. (1991), 'Numerical solution of the Swift-Hohenberg equation in two dimensions', *Physica* **177**, 356–365.