On the Computational Power of Continuous-Time Symmetric Hopfield Nets

Šíma, Jiří
2000

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

On the Computational Power of Continuous-Time Symmetric Hopfield Nets

Jiří Šíma

Technical report No. 815

October 2000

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic
phone: (+420 2) 6605 3030 fax: (+420 2) 85 85 789
e-mail: sima@cs.cas.cz

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

On the Computational Power of Continuous-Time Symmetric Hopfield Nets

Jiří Šíma [1]

Technical report No. 815
October 2000

Abstract

The computational power of continuous-time symmetric Hopfield nets is investigated. As is well known, such networks have very constrained, Lyapunov-function controlled dynamics. Nevertheless, it is shown that they are universal and efficient computational devices, in the sense that any convergent fully parallel computation by a network of n discrete-time binary neurons, with in general asymmetric interconnections, can be simulated by a symmetric continuous-time Hopfield net containing only $O(n)$ units using the saturated-linear sigmoid activation function. In terms of standard discrete computation models this result implies that any polynomially space-bounded Turing machine can be simulated by a polynomially size-increasing sequence of continuous-time Hopfield nets.

Keywords

neural networks, continuous-time Hopfield net, Lyapunov function, analog computation, computational power

[1] Research partially supported by GA ČR Grant No. 201/98/0717 and GA AS CR Grant B2030007. This work has been completed during my stay at the Department of Mathematics, University of Jyväskylä, Finland, that also partially supported my research. I am very grateful to Pekka Orponen who originally inspired this work for his many valuable comments.

1 Introduction

In this paper, the computational power of the continuous-time symmetric recurrent neural network model popularized by John Hopfield (1984) is studied. The dynamics of these networks were actually already analyzed earlier by Cohen and Grossberg (1983) in a more general setting, but because of the affinity to the very influential discrete-time binary-state version of the model (Hopfield, 1982), this additive special case of the Cohen-Grossberg equations has become known as the "continuous-time Hopfield network model". Part of the appeal of Hopfield's continuous-time model stems from its efficient implementations in analog electrical (Hopfield, 1984) and optical (Farhat et al., 1985; Stoll and Lee, 1988) hardware. Besides associative memory, proposed uses of continuous-time Hopfield nets include, e.g., fast approximate solution of combinatorial optimization problems (Hopfield and Tank, 1985; Aarts and Korst, 1989; Liao, 1999; Mańdziuk, 2000; Sheu et al., 1991; Wu and Tam, 1999).

As is well known (Cohen and Grossberg, 1983; Hopfield, 1984), the dynamics of any network adhering to this model is governed by a Lyapunov function defined on its state space. At first sight, this would appear to severely limit the capabilities of such networks for general computation, because the Lyapunov property implies that a network always converges from any initial state towards some stable final state. Thus e.g. nondamping oscillations, which seem to be an essential prerequisite of general computation, cannot be created in such networks.

The existence of a Lyapunov function is a characteristic of networks whose interconnection weight matrix is *symmetric*, as required for both continuous- and discrete-time Hopfield nets. More general asymmetric networks usually do not behave in the simple manner guaranteed by this property. E.g. one can easily create an oscillator out of two asymmetrically connected continuous-time units (or even one discrete-time neuron with a negative feedback), but even this simple device cannot be simulated by any symmetric continuous-time network.

Nevertheless, it will be shown that infinite oscillations are the *only* feature of general-purpose (digital) computation that cannot be reproduced in symmetric continuous-time networks. More precisely, it will be proved in sections 3, 4 that any converging fully parallel computation by a network of n discrete-time binary neurons, with in general asymmetric interconnections, can be simulated by a symmetric continuous-time Hopfield net containing $18n+7$ units using the saturated-linear sigmoid activation function.

Observe, namely, that any converging computation by a discrete-time deterministic network of n binary neurons must terminate within 2^n steps. A basic technique used in our proof is the construction of an $(n+2)$-bit symmetric continuous-time *clock* network (a simulated binary counter) that, using $8n+7$ units, produces a sequence of 2^n well-controlled oscillations (generated by the second least significant counter bit) before it converges. This sequence of clock pulses is used to drive the rest of the network where each discrete neuron is simulated by a symmetrically interconnected subnetwork of 10 continuous-time units.

The clock network is already by itself of some interest from a dynamical systems perspective, because it provides an explicit example of a Lyapunov-type continuous-time

system whose convergence time grows exponentially in the system dimension. More precisely, it will be shown in section 5 that the convergence time for an $(n+1)$-bit clock network, which consists of $d = 6n + 1$ units, is $\Omega(2^n/\varepsilon) = \Omega(2^{d/6}/\varepsilon)$, where ε is a parameter controlling the convergence rate of the system. In terms of bit representations this bounds translates to a convergence time of $2^{\Omega(g(M))}$ for a network with an encoding size of M bits, where $g(M)$ is an arbitrary continuous function such that $g(M) = o(M)$, $g(M) = \Omega(M^{2/3})$, and $M/g(M)$ is increasing.

This result can be compared to a general convergence time upper bound of $2^{O(\sqrt{N})}$ for discrete Hopfield networks with N-bit representations (Šíma et al., 2000). Thus, the continuous-time implementation actually yields better bounds than the discrete-time one, providing that the time interval between two subsequent discrete updates corresponds to a continuous time unit. This result suggests that continuous-time analog models of computation may be worth investigating more for their efficiency gains than for their (theoretical) capability for arbitrary-precision real number computation (Balcázar et al., 1997; Siegelmann and Sontag, 1994; Siegelmann and Sontag, 1995).

The predecessors of the present work are: a similar, but considerably simpler, construction used by Orponen (1996) to prove the computational equivalence of symmetric and convergent asymmetric *discrete-time binary* networks[2], and the simulation of discrete-time networks by *asymmetric* continuous-time networks proposed by Orponen (1997a). The original idea for the discrete-time clock network used by Orponen (1996), and on which our current construction is based, stems from the work by Goles and Martínez (1989). A general survey of topics in continuous-time computation is presented by Orponen (1997b).

As pointed out by Orponen (1996), increasing sequences of discrete networks are computationally equivalent to (nonuniform) polynomially space-bounded Turing machines, more precisely, they compute the complexity class PSPACE/poly (Balcázar et al., 1995, p. 105). By the result in the present paper, it is now known that continuous-time symmetric networks are at least as powerful, that is, given any polynomially space-bounded Turing machine, a polynomial-size sequence of continuous-time Hopfield nets can be constructed for simulating it. This is to our best knowledge the first result concerning the computational power of symmetric continuous-time networks, and it is somewhat surprising that they turn out to be computationally universal in this complexity-limited sense.

A related line of study concerns the computational power of *finite discrete-time analog-state* neural networks (Siegelmann, 1999). Here it is known that the computational power of asymmetric networks using the saturated-linear sigmoid activation function increases with the Kolmogorov complexity of the weight parameters (Balcázar et al., 1997). With integer weights such networks are equivalent to finite automata (Horne and Hush, 1996; Indyk, 1995; Šíma and Wiedermann, 1998), while with rational weights arbitrary Turing machines can be simulated (Indyk, 1995; Siegelmann and

[2]The present construction can actually also be used to improve the discrete-time simulation by Orponen (1996), which requires a symmetric network of $\Omega(n^2)$ units to simulate a convergent asymmetric network of size n. Using the technique presented in section 3, the simulation overhead can be reduced to $6n + 2$ units in the discrete case (Šíma et al., 2000).

Sontag, 1995). With arbitrary real weights the networks can even have "super-Turing" computational capabilities (Siegelmann and Sontag, 1994). Similar results also apply to *symmetric* neural networks to some extent since the computational power of finite analog Hopfield nets augmented with an external clock providing them with an "energy source" has been proved to be the same as that of asymmetric analog networks (Šíma et al., 2000). On the other hand, it is known that any amount of analog noise reduces the computational power of these models to that of finite automata (Casey, 1996; Maass and Orponen, 1998).

An experimental evidence of the presented results also appeared in an extended abstract (Šíma and Orponen, 2000). This article is organized as follows. After a brief review of the basic definitions in section 2, our main construction of the underlying continuous-time Hopfield network is outlined in section 3 where its operation is also informally explained. The formal verification of this construction, which has the form of a rather tedious case analysis, is given in section 4. Section 5 contains the corresponding convergence time analysis of continuous-time Hopfield nets. Finally, some open problems are mentioned in section 6.

2 Preliminaries

The model of a finite *discrete recurrent neural network* will first be briefly specified. Such a network consists of n simple computational *units* or *neurons*, indexed as $1, \ldots, n$, that are connected into a generally cyclic oriented graph or *architecture*, in which each edge (i, j) leading from neuron i to j is labeled with an integer *weight* $w(i, j) = w_{ji}$. The absence of a connection within the architecture indicates a zero weight between the respective neurons, and vice versa.

Only the *fully parallel* dynamics of such networks will be considered here, in which the evolution of the network *state* $\mathbf{y}^{(t)} = (y_1^{(t)}, \ldots, y_n^{(t)}) \in \{0, 1\}^n$ is determined for discrete time instants $t = 0, 1, \ldots$, as follows. At the beginning of a computation the network is placed in an *initial state* $\mathbf{y}^{(0)}$ which may include an external input. At discrete time $t \geq 0$, each neuron $j = 1, \ldots, n$ collects its binary *inputs* from the *states* (*outputs*) $y_i^{(t)} \in \{0, 1\}$ of incident neurons i. Then its integer *excitation*

$$\xi_j^{(t)} = \sum_{i=0}^{n} w_{ji} y_i^{(t)}, \quad j = 1, \ldots, n \tag{2.1}$$

is computed as the respective weighted sum of inputs. The sum includes an integer *bias* w_{j0} local to neuron j, modeled as a weight from a formal constant unit input $y_0^{(t)} = 1$, $t \geq 0$. At the next instant $t + 1$, an *activation function*, which in this case is the *hard limiter* or *threshold function* s, is applied to $\xi_j^{(t)}$ for all neurons $j = 1, \ldots, n$ in order to determine the new network state $\mathbf{y}^{(t+1)}$ by the following rule:

$$y_j^{(t+1)} = s\left(\xi_j^{(t)}\right), \quad j = 1, \ldots, n \tag{2.2}$$

where

$$s(\xi) = \begin{cases} 1 & \text{for } \xi \geq 0 \\ 0 & \text{for } \xi < 0\,. \end{cases} \tag{2.3}$$

In addition, the *maximum weight*

$$w_{\max} = \max_{j=1,\ldots,n;\, i=0,\ldots,n} |w_{ji}| \tag{2.4}$$

is defined as the maximum absolute value over all weight parameters in the network.

Similarly, a finite *continuous-time analog neural network* is composed of m analog units $P = \{1, \ldots, m\}$ which operate (in our case) with the *saturated-linear* sigmoid activation function

$$\sigma(\xi) = \begin{cases} 1 & \text{for } \xi \geq 1 \\ \xi & \text{for } 0 < \xi < 1 \\ 0 & \text{for } \xi \leq 0\,. \end{cases} \tag{2.5}$$

Hence, the states of analog units are real numbers within the interval $[0, 1]$, and the weights (including biases), denoted by $v(p, q)$ (for neurons $p, q \in P$) are reals as well. The computational dynamics of a continuous-time network is defined for every real $t > 0$ by the following system of differential equations, with the initial network state $\mathbf{y}(0) = (y_1(0), \ldots, y_m(0)) \in [0, 1]^m$ providing its initial conditions:

$$\begin{aligned} \frac{dy_p}{dt}(t) &= -y_p(t) + \sigma(\xi_p(t)) \\ &= -y_p(t) + \sigma\left(\sum_{q=0}^{n} v(p, q) y_q(t)\right) \quad p = 1, \ldots, m\,. \end{aligned} \tag{2.6}$$

In addition, an analog unit p is called *saturated* at 0 or 1 at time t if its excitation $\xi_p(t) \leq 0$ or $\xi_p(t) \geq 1$, respectively, and p is *unsaturated* when $0 < \xi_p(t) < 1$ (see equation 2.5).

A *Hopfield* (*symmetric*) *network* has as an architecture an undirected graph, and weights that satisfy $v(p, q) = v(q, p)$ for every $p, q \in P$. By a Lyapunov function argument (Cohen and Grossberg, 1983; Hopfield, 1984), it can be shown that a Hopfield network converges from any initial state $\mathbf{y}(0)$ to some stable state satisfying $dy_p/dt = 0$ for all $p = 1, \ldots, m$. The set of stable states of the discrete system from equation 2.2 coincides with that of the continuous-time system in equation 2.6.

3 Constructing the Continuous-Time Network

In this section a linear-size construction of the continuous-time symmetric Hopfield net simulating a given convergent asymmetric discrete-time recurrent network is outlined. The representation of binary states within the simulating Hopfield net is realized by saturating specific analog-state units at the appointed time instants corresponding to discrete-time updates. An optional parameter ε of the construction is introduced to control a scalable length of the continuous-time interval between two subsequent simulated discrete updates. The main result is summarized in the following theorem.

Theorem 1 *Any fully parallel computation by a recurrent neural network of n binary neurons with asymmetric weights of the maximum size $w_{\max}$, converging within*

$t^\star$ discrete update steps, can be simulated by a symmetric analog Hopfield net with $m = 18n + 7$ continuous-time units, within continuous time $\Theta(t^\star/\varepsilon)$ for any positive real $0 < \varepsilon < 0.0025$ such that

$$w_{\max} 2^{3n} \leq \varepsilon 2^{1/\varepsilon} \,. \tag{3.1}$$

Proof: The underlying continuous-time Hopfield net is composed of a simulated $(n+2)$-bit binary counter (clock) subnetwork consisting of $8n+7$ units whose construction will be described in section 3.1, and n other subnetworks, each containing 10 analog units for the purpose of simulating one discrete neuron which are defined in section 3.2. The simulation is first explained here informally while a formal verification of its correctness is deferred to section 4.

3.1 The Clock Network

Any converging computation by a discrete-time deterministic network of n binary neurons must terminate within $t^\star \leq 2^n$ steps. Thus, an $(n+2)$-bit symmetric continuous-time clock subnetwork of size $8n + 7$ units, each starting at the zero initial state, is constructed which is a simulated binary counter whose second least significant counter bit produces a sequence of 2^n well-controlled oscillations before the network converges. Each of these clock pulses is exploited for one simulated discrete update as will be shown in section 3.2.

The construction of the simulated $(n + 2)$-bit binary counter will be described by induction on n. The induction starts with a 2-bit counter network which is presented in Figure 3.1, where the symmetric connections between units are labeled with the corresponding weights and the biases are indicated by the edges drawn without an originating unit. The bias $v(0, c_0) = \varepsilon > 0$ of the least significant counter unit c_0 of order 0 corresponds to its initial positive excitation. Because of its feedback weight $v(c_0, c_0) = 1 + \varepsilon$ the state of c_0 gradually grows towards value 1. After c_0 is saturated at 1 (i.e. its excitation reaches 1) its state is "sufficiently close" to 1 when c_0 is called to be *active* or to *fire*. This trick of gradual transition from 0 to 1 is used repeatedly throughout the continuous-time clock network construction.

The operation of the remaining 6 units in 2-bit counter network (see Figure 3.1), which are of order 1, is the same as that of the corresponding units of higher orders $k > 1$ whose inductive description and explanation follows below. Only the weight $v(a_1, x_1) = W + 4$ actually depends on an additional positive integer parameter

$$W = \frac{3}{2}\, nu + 42 \sum_{j=1}^{n} |w_{j0}| + 42n > 0 \,, \tag{3.2}$$

where

$$u = 28 \max_{j=1,\ldots,n} \left(- \sum_{i=1;\, w_{ji} \leq 0}^{n} w_{ji} \,,\; \sum_{i=1;\, w_{ji} \geq 0}^{n} w_{ji} \right) \geq 0 \,, \tag{3.3}$$

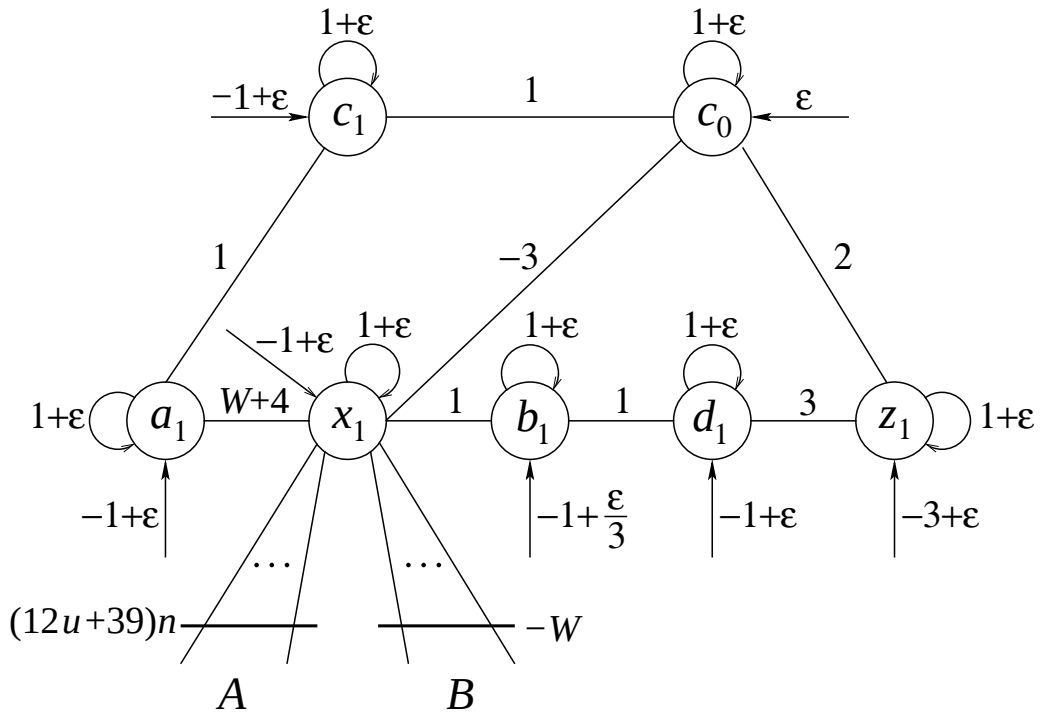

Figure 3.1: A continuous-time 2-bit counter network

and w_{ji} are the original asymmetric weights from the discrete network. It is because x_1 represents an interface unit which is connected to the simulating subnetwork (see section 3.2) to transfer the clock pulses and its operation within the clock may not be affected by the rest of the network.

For the induction step which is depicted in Figure 3.2, suppose that the counter has been constructed up to the first k ($2 \le k < n+2$) counter bits $c_0, \ldots, c_{k-1}$, and denote by P_k the set of all its $m_k = 8k-9$ units, including the auxiliary ones labeled $a_\ell, x_\ell, b_\ell, d_\ell, z_\ell$, for $\ell = 1, \ldots, k-1$, and for $k > 2$ also x'_ℓ, b'_ℓ for $\ell = 2, \ldots, k-1$. Then the counter unit c_k with a feedback weight $v(c_k, c_k) = 1+\varepsilon$, is connected to all m_k units $p \in P_k$ via unit weights $v(p, c_k) = 1$, which, together with its bias $v(0, c_k) = -m_k + \varepsilon$, make c_k to fire shortly after all these units are active. This is because the activity of the first k counter bits $c_0, \ldots, c_{k-1}$ means that counting from 0 to $2^k - 1$ has been accomplished, and hence, the next counter bit c_k must fire. In addition, unit c_k is connected to a sequence of seven auxiliary units $a_k, x'_k, b'_k, x_k, b_k, d_k, z_k$, all having feedbacks $1+\varepsilon$, which are being, one by one, activated after c_k fires. This is implemented by the following weights $v(c_k, a_k) = m_k$, $v(a_k, x'_k) = -v(x'_k, x_1) + 1$, $v(x'_k, b'_k) = v(x_k, b_k) = v(b_k, d_k) = 1$, $v(b'_k, x_k) = V_k + v(x'_k, x_1)$, $v(d_k, z_k) = V_k - m_k$, and biases $v(0, a_k) = -m_k + \varepsilon$, $v(0, x'_k) = v(0, x_k) = v(0, d_k) = -1 + \varepsilon$, $v(0, b'_k) = v(0, b_k) = -1 + \varepsilon/3$, $v(0, z_k) = m_k - V_k + \varepsilon$ where $v(x'_k, x_1) < 0$ is a sufficiently negative weight between units x'_k and x_1, while $V_k > -v(x'_k, x_1) > 0$ is a sufficiently large positive parameter. The exact values of $v(x'_k, x_1)$ and V_k will be determined below by

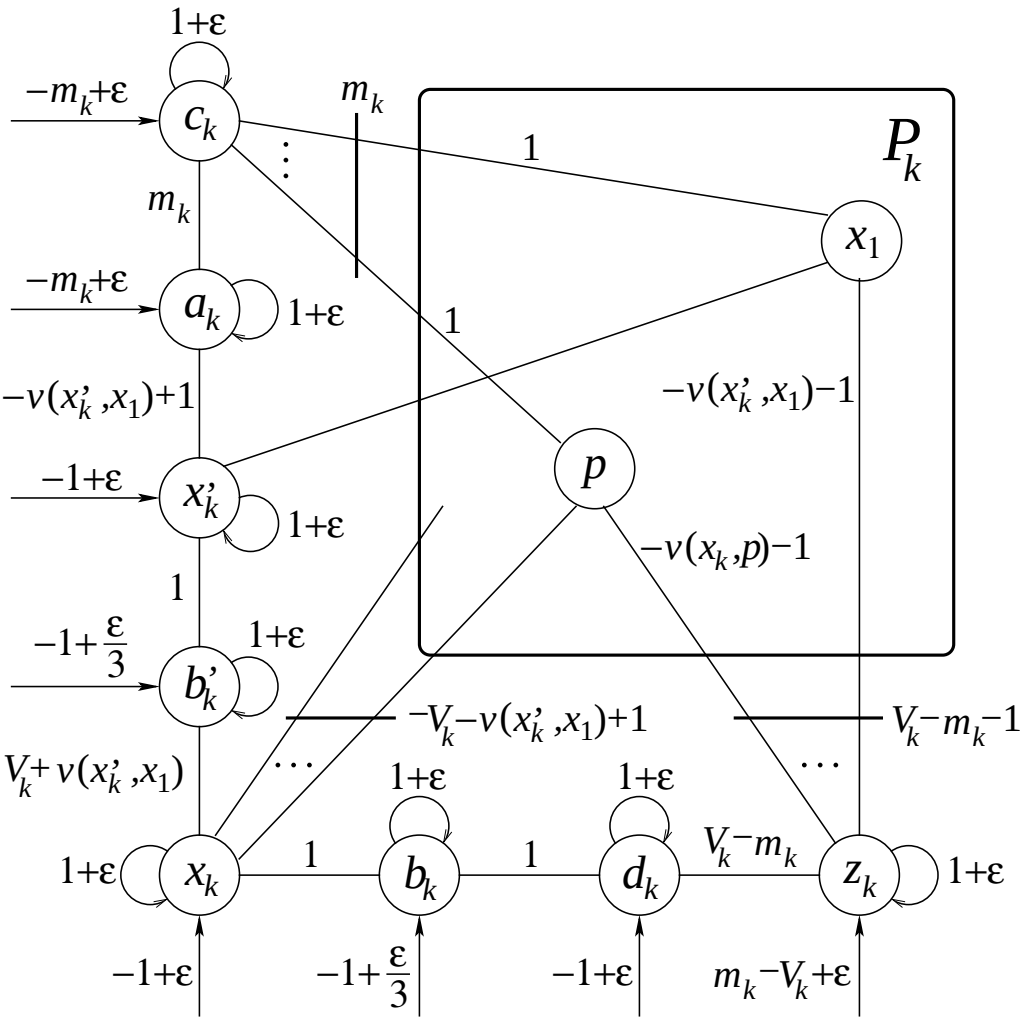

Figure 3.2: Inductive construction of continuous-time counter network

equations 3.4 and 3.7, respectively, in order to guarantee that neurons x'_k, x_k, z_k are not directly influenced by a computation of units from P_k except via c_k.

The only purpose of units a_k, b'_k, b_k, d_k is to slow down the continuous-time state flow in order to synchronize the computation.

Units x'_k, x_k reset all the neurons in P_k to their initial states "sufficiently close" to 0 when they are called *passive* which means they are saturated at 0 (i.e. their excitations are nonpositive). This is consistent with the correct counter computation when c_k fires. In particular, the interface unit $x_1 \in P_k$ is first treated separately by x'_k so that its continuous-time state decrease is not affected by the remaining units in $P_k \setminus \{x_1\}$ which, in the meantime, remain active as can be checked in Figure 3.1, especially for those neurons incident to x_1. Only after x_1 is passive, unit x_k is activated to reset the rest of units from $P_k \setminus \{x_1\}$. For this purpose, unit x'_k is connected to x_1 via sufficiently large negative weight

$$v(x'_k, x_1) = -S_{kx_1} - (12u + 39)n\,, \tag{3.4}$$

and similarly, x_k is linked with each $p \in P_k \setminus \{x_1\}$ via

$$v(x_k, p) = -S_{kp} \tag{3.5}$$

so that

$$S_{kp} = 3 + \left(\sum_{q \in P_k \setminus \{p\};\, v(q,p)>0} v(q,p) \right) \tag{3.6}$$

for $p \in P_k$ exceeds weight $v(c_k, p) = 1$ and their mutual positive influence where $3 = v(c_k, p) + \lceil v(p,p) \rceil$ in equation 3.6 also includes the feedback weight $v(p,p) = 1 + \varepsilon$ whose rounding up covers the only positive bias $v(0, c_0) = \varepsilon > 0$ when $p = c_0$. In addition, term $-(12u + 39)n$ in definition 3.4 balances the total positive influence on the interface unit x_1 originating from the weights $v(\alpha_j, x_1) = v(\beta_j, x_1) = 12u + 39$ for $j = 1, \ldots, n$ (units α_j and β_j will never be active for the same j simultaneously) of the simulating subnetwork (see section 3.2). The previous weight definitions 3.4, 3.5 also determine the value of parameter

$$V_k = 1 - v(x'_k, x_1) - \sum_{p \in P_k \setminus \{x_1\}} v(x_k, p)\,, \tag{3.7}$$

which makes the state of x_k (similarly for z_k below) independent of the outputs from $p \in P_k$.

Finally, unit z_k balances the negative influence of both x'_k and x_k on P_k so that the first k counter bits can again count from 0 to $2^k - 1$ but now with c_k being active. This is achieved by the exact weights

$$v(z_k, p) = \begin{cases} -v(x'_k, x_1) - 1 & \text{for } p = x_1 \\ -v(x_k, p) - 1 & \text{for } p \in P_k \setminus \{x_1\}\,. \end{cases} \tag{3.8}$$

in which $-v(x'_k, x_1)$ and $-v(x_k, p)$ eliminate the influence of x'_k and x_k on p, respectively, whereas -1 compensates $v(c_k, p) = 1$ for each $p \in P_k$. Clearly, neurons $p \in P_k$ cannot reversely affect z_k since their maximal contribution

$$\sum_{p \in P_k} v(z_k, p) = -m_k - v(x'_k, x_1) - \sum_{p \in P_k \setminus \{x_1\}} v(x_k, p) = V_k - m_k - 1 \tag{3.9}$$

to the excitation of z_k according to equations 3.7, 3.8, cannot overcome its bias $v(0, z_k) = m_k - V_k + \varepsilon$. This completes the induction step of the clock construction.

3.2 The Simulating Subnetwork

The high-level scheme of the underlying simulation of discrete asymmetric network by a symmetric continuous-time Hopfield net is depicted in Figure 3.3. The simulating subnetwork consists of a sequence of five layers, each layer being linked to the subsequent one, through which a signal is transmitted only in one direction (from top to bottom in Figure 3.3). As it will be seen below this is implemented by a decreasing

sequence of absolute values of symmetric weights and biases which prevent the signal from propagating backwards; a technique that has originally been introduced for symmetric implementation of threshold circuits in discrete case (Parberry, 1994). In addition, the last fifth layer is further connected to the first one via weights of only small absolute values that need a support of large positive weights from the clock in order to transmit a signal and create a simulation cycle.

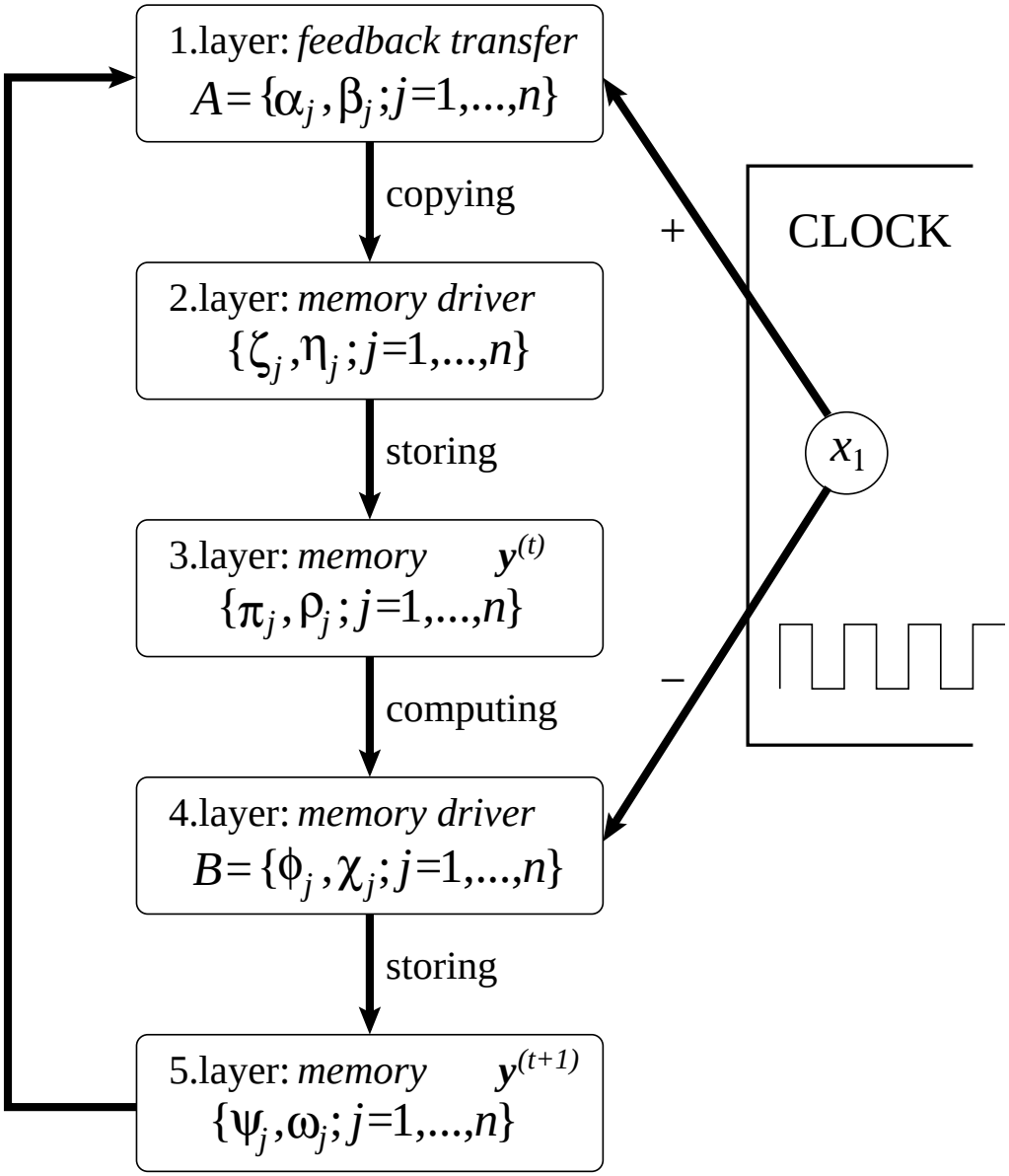

Figure 3.3: A high-level scheme of the simulation

Thus, Figure 3.3 also shows a clock subnetwork that has been constructed in section 3.1. In particular, the interface unit x_1 from the clock is connected to the first layer A via large positive weights and similarly to the fourth layer B via large negative weights in order to drive the simulation. The clock unit x_1 of the second least

significant order in the $(n+2)$-bit simulated counter fires 2^n times. Each of these 2^n clock pulses is exploited for implementing one discrete-time update which is sufficient to simulate any convergent computation on a discrete network of size n binary neurons. Furthermore, each of these simulation steps consists of two phases: one for computing a new discrete-network state and the second one for updating an old network state by this new one.

In the first phase, unit x_1 is passive and the first layer gates A are locked since they miss the respective support from the clock. The third layer of the simulating subnetwork serves as a memory and stores the current binary state $\mathbf{y}^{(t)} \in \{0,1\}^n$ of the simulated network from discrete time instant t. This also means that at the beginning of the simulation, the third-layer units must properly be initialized by the initial discrete-network state $\mathbf{y}^{(0)}$. Note these are the only units in the underlying continuous-time Hopfield net that may initially be active. The new state $\mathbf{y}^{(t+1)}$ at next time instant $t+1$ of the simulated discrete network is computed in the subsequent fourth layer via original weights from the asymmetric discrete network. This layer is also used as a memory driver for storing $\mathbf{y}^{(t+1)}$ in the subsequent fifth layer that implements a memory. This completes the first phase of computing the new discrete-network state and the simulating subnetwork is stabilized until x_1 fires.

In the second phase, unit x_1 is active and the first layer gates A are unlocked by means of large positive weights from x_1 while the fourth layer gates B are locked due to the large negative weights from x_1. Thus, the new state $\mathbf{y}^{(t+1)}$ is transmitted from the memory in the last fifth layer back to the first layer which further transfers $\mathbf{y}^{(t+1)}$ to the subsequent second layer that serves as a memory driver for the third layer. Hence, the current network state which is stored in the memory implemented by the third layer is properly updated by the new one. The simulation step is completed and the simulating subnetwork is stabilized until x_1 becomes passive again.

Now, the detailed implementation of the simulating subnetwork will be presented. As indicated in Figure 3.3 each of the five layers in the simulating subnetwork is composed of n pairs of continuous-time units. Each pair in one layer is associated with one discrete neuron j $(j = 1, \ldots, n)$ from the simulated network. The corresponding five pairs, each from one layer, that are associated with the same j create a 10-unit symmetric continuous-time subnetwork for simulating one discrete neuron j $(j = 1, \ldots, n)$ whose operation can be described separately. This 10-unit network for discrete neuron j is depicted in Figure 3.4 including all the necessary symmetric weight definitions. The figure again presents the interface clock unit x_1 which is linked to the simulating subnetwork providing it with 2^n driving oscillations.

At the beginning of the simulation, all the units in the underlying continuous-time Hopfield net are placed into the initial zero states, except for those units π_j, ϱ_j from the third layer that represent the initially active discrete neurons j (i.e. $y_j^{(0)} = 1$), and their states are set to 1.

The first phase of simulating one computational step of discrete network corresponds to a continuous-time interval when unit x_1 is passive. During this period the new state of the simulated discrete network is computed. The current output $y_j^{(t)} \in \{0,1\}$ of the simulated binary-state neuron j $(j = 1, \ldots, n)$ at discrete time instant $t \geq 0$ is doubly represented by the analog states of two third-layer continuous-time units π_j, ϱ_j which

are saturated at value $y_j^{(t)}$. Indeed, $y_j^{(t)}$ is stored in units π_j, ϱ_j which are either both passive, if $y_j^{(t)} = 0$, due to their negative biases, or they are saturating each other at 1 via mutual positive symmetric weight $v(\pi_j, \varrho_j)$ if $y_j^{(t)} = 1$, while neurons $\alpha_j, \beta_j, \zeta_j, \eta_j$ from the first two layers are momentarily passive.

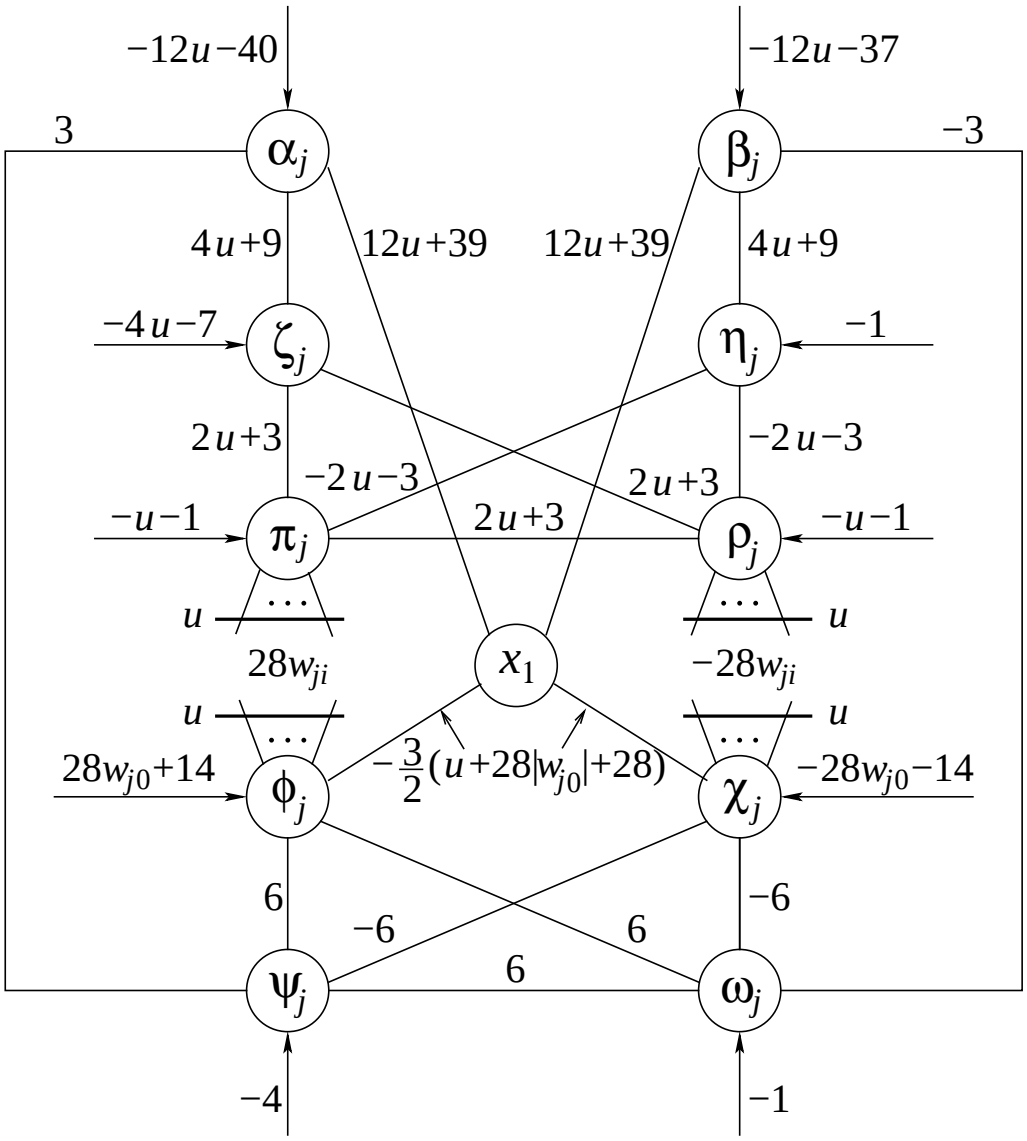

Figure 3.4: A continuous-time simulation of discrete neuron j

The new binary output $y_j^{(t+1)}$ of simulated neuron j ($j = 1, \ldots, n$) at discrete time $t + 1$ is computed in the fourth-layer unit ϕ_j which is connected to the appropriate third-layer units π_i of the respective subnetworks that store the states $y_i^{(t)}$ ($1 \leq i \leq n$) from the previous discrete-time instant t, following the original discrete network architecture. Also the bias $v(0, \phi_j) = 28w_{j0}+14$ and symmetric weights $v(\pi_i, \phi_j) = 28w(i, j)$ associated with these connections correspond to the original integer asymmetric weights

$w(i,j)$ of the discrete network. In fact, the weights are appropriately adjusted so that the state of ϕ_j is not affected by the fifth-layer units ψ_j, ω_j while the original function of j is preserved (Parberry, 1994). Similarly, the fourth-layer unit χ_j computes the negation of $y_j^{(t+1)}$ from the respective third-layer units ϱ_i via the opposite symmetric weights $v(\varrho_i, \chi_j) = -28w(i,j)$ and bias $v(0, \chi_j) = -28w_{j0} - 14$ which also prevents χ_j from being influenced by fifth-layer units ψ_j, ω_j. In addition, parameter u given by equation 3.3 ensures that the fourth-layer units ϕ_i and χ_i $(i = 1, \ldots, n)$ cannot reversely affect the third-layer units π_j and ϱ_j, respectively. Note also that units ϕ_j, χ_j are never simultaneously active for the same j since either they are both passive or each of them represents the negation of the binary state stored in the other unit. Hence, the positive parameter W defined in equation 3.2 which is included in weight $v(a_1, x_1)$ is sufficient for activating the clock unit x_1 when required regardless of the total negative backward influence of $B = \{\phi_j, \chi_j;\ j = 1, \ldots, n\}$ on x_1 (see also Figure 3.1).

Further, the fourth-layer units ϕ_j, χ_j store the new state $y_j^{(t+1)}$ into the fifth-layer memory units ψ_j, ω_j in such a way that either ϕ_j ensures activating ψ_j, ω_j through positive weights if $y_j^{(t+1)} = 1$ or χ_j arranges their reseting by means of negative weights when $y_j^{(t+1)} = 0$. In fact, state $y_j^{(t+1)}$ is doubly stored in the continuous-time units ψ_j, ω_j which are either both passive, if $y_j^{(t+1)} = 0$, due to their negative biases, or they are saturating each other in 1 via mutual positive symmetric weight $v(\psi_j, \omega_j)$ if $y_j^{(t+1)} = 1$. Notice that in spite of the symmetric weights the computation and flow of signal $y_j^{(t+1)}$ (and its negation) in this part of the continuous-time network goes only in the direction from third-layer units π_j, ϱ_j via fourth-layer units ϕ_j, χ_j up to fifth-layer units ψ_j, ω_j due to the decreasing sequence of the absolute values of weights and biases (see Figure 3.4). However, after the signal $y_j^{(t+1)}$ reaches units ψ_j, ω_j, the continuous-time simulating subnetwork becomes temporarily stable since the subsequent first-layer gates α_j, β_j are locked by their large negative biases which cannot be overcome by the respective small weights $v(\psi_j, \alpha_j)$, $v(\omega_j, \beta_j)$ until the interface clock unit x_1 is activated.

The second phase of simulating the discrete computational step corresponds to a continuous-time interval when unit x_1 is active. During this period the old state $y_j^{(t)}$ of the simulated discrete-time neuron j $(j = 1, \ldots, n)$ stored in the third-layer memory units π_j, ϱ_j is being replaced by the new one $y_j^{(t+1)}$ stored in the fifth-layer memory units ψ_j, ω_j. The interface clock unit x_1 when being activated, first locks the fourth-layer gates ϕ_j, χ_j by large negative weights $v(x_1, \phi_j) = v(x_1, \chi_j) = -3(u + 28|w_{j0}| + 28)/2$ that saturate them at 0 whereas the fifth-layer memory units ψ_j, ω_j keep the original state $y_j^{(t+1)}$.

Only after neurons ϕ_j, χ_j are locked, unit x_1 unlocks the first-layer gates α_j and β_j. Unit α_j receives the state $y_j^{(t+1)}$ from the fifth-layer memory unit ψ_j since the large positive weight $v(x_1, \alpha_j)$ from x_1 balances its negative bias $v(0, \alpha_j)$ so that the small positive weight $v(\psi_j, \alpha_j)$ from ψ_j may play its role. Similarly, unit β_j computes the negation of state $y_j^{(t+1)}$ stored in the fifth-layer memory unit ω_j when the large positive weight $v(x_1, \beta_j)$ from x_1 overcomes its negative bias $v(0, \beta_j)$ so that the small negative weight $v(\omega_j, \beta_j)$ from ω_j may influence the computation. Further, the state $y_j^{(t+1)}$ and its negation are propagated from the first-layer units α_j and β_j to the second-layer

units ζ_j and η_j, respectively, following again the decreasing sequence of the absolute values of symmetric weights and biases (see Figure 3.4) up to the third-layer memory units π_j, ϱ_j that doubly store the current output of simulated discrete neuron j which is now rewritten by the new value $y_j^{(t+1)}$.

After that, the simulating subnetwork is temporarily stabilized until x_1 is passive again when the new discrete state $y_j^{(t+2)}$ is to be computed. Recall also that units α_j, β_j are never simultaneously active for the same j since either they are both passive or each of them represents the negation of the binary state stored in the other unit. Therefore, term $-(12u + 39)n$ included in the negative weights $v(x'_k, x_1)$ $(k > 1)$ defined in equation 3.4 for reseting unit x_1 within the clock computation (see section 3.1) suffices to suppress the total positive backward influence of weights $v(\alpha_j, x_1) = v(\beta_j, x_1) = 12u + 39$ from units in $A = \{\alpha_j, \beta_j;\ j = 1, \ldots, n\}$ on x_1 in order to prevent the clock from being affected by the simulating subnetwork (see also Figure 3.1). As a result of saturating x_1 at 0 also the first- and second-layer units $\alpha_j, \beta_j, \zeta_j, \eta_j$ become passive whereas the third-layer memory units π_j, ϱ_j keep the current state $y_j^{(t+1)}$ which is used for computing the next discrete state $y_j^{(t+2)}$, etc. Finally, recall that the interface clock unit x_1 in the $(n+2)$-bit simulated counter fires 2^n times, which is sufficient to simulate any convergent computation on a discrete neural network of size n.

4 Formal Verification

In section 3, the construction of the continuous-time symmetric Hopfield net simulating a discrete asymmetric neural network has been described and some insight has been given into its operation. However, in order to formally prove theorem 1, the correct state evolution of the continuous-time units in the underlying Hopfield net during the simulation has to be checked by analyzing the corresponding system of differential equations 2.6. The following sections examine five different issues of the presented simulation which altogether provide its formal verification.

In section 4.1 the continuous-time state evolution will explicitly be expressed for a saturated unit and its intended saturation stability will be proved. Section 4.2 analyzes in detail the transfer of the activity from a continuous-time unit to a subsequent one within the clock network when incident neurons are saturated, and estimates its duration which represents a simulated discrete time. In section 4.3 the continuous-time Hopfield net is shown to approximate any discrete-time computation in general which is proved to be consistently integrated with the special case of transferring the activity within the clock. Section 4.4 gives an evidence that the interface clock unit provides a correct synchronization for the simulation. Finally, in section 4.5 correct timing of the whole simulation is checked ensuring fast decrease of errors in the binary-state representation.

Before these issues will be analyzed, the following technical lemma is first proved concerning the maximum sum of absolute values of input weights to one unit in the constructed continuous-time Hopfield net:

Lemma 1 *For any unit p $(1 \leq p \leq m)$ in the continuous-time Hopfield net that has been constructed in section 3 for simulating an asymmetric discrete neural network of size n and maximum weight $w_{\max}$, the sum of absolute values of its input weights (excluding bias) can be upper bounded as follows:*

$$\Xi_p = \sum_{q=1}^{m} |v(q,p)| < \varepsilon 2^{1/\varepsilon} \,. \tag{4.1}$$

Proof: The parameter V_{n+1} will first be computed that has been introduced in equation 3.7 which can further be rewritten for $k > 2$ as

$$V_k = 1 - v(x'_k, x_1) - \sum_{p \in P_{k-1} \setminus \{x_1\}} v(x_k, p) - \sum_{p \in P_k \setminus P_{k-1}} v(x_k, p) \,. \tag{4.2}$$

From definitions 3.5, 3.6 the weight $v(x_k, p)$ for $p \in P_{k-1} \setminus \{x_1\}$ $(k > 2)$ reduces to

$$\begin{aligned} v(x_k,p) &= v(x_{k-1},p) - \left(\sum_{q \in P_k \setminus P_{k-1};\, v(q,p)>0} v(q,p) \right) \\ &= v(x_{k-1},p) - v(c_{k-1},p) - v(z_{k-1},p) = 2v(x_{k-1},p) \end{aligned} \tag{4.3}$$

by using equation 3.8. In the same way, a recursive formula is obtained for weight $v(x'_k, x_1)$ $(k > 2)$:

$$v(x'_k, x_1) = 2v(x'_{k-1}, x_1) \,. \tag{4.4}$$

By introducing formulas 4.3, 4.4 into 4.2 a recursive formula for V_k $(k > 2)$ is derived

$$V_k = 2V_{k-1} - 1 - \sum_{p \in P_k \setminus P_{k-1}} v(x_k, p) \tag{4.5}$$

from definition 3.7.

It remains to compute the weights from sum $-\sum_{p \in P_k \setminus P_{k-1}} v(x_k, p)$ in formula 4.5 which can be achieved from definitions 3.5, 3.6 by using Figure 3.2 as follows:

$$\begin{aligned} -v(x_k, c_{k-1}) &= 3 + \left(\sum_{q \in P_{k-1};\, v(q,c_{k-1})>0} v(q, c_{k-1}) \right) + v(a_{k-1}, c_{k-1}) \\ &= 2m_{k-1} + 3 \qquad (4.6) \\ -v(x_k, a_{k-1}) &= 3 + v(c_{k-1}, a_{k-1}) + v(x'_{k-1}, a_{k-1}) \\ &= -v(x'_{k-1}, x_1) + m_{k-1} + 4 \qquad (4.7) \\ -v(x_k, x'_{k-1}) &= 3 + v(a_{k-1}, x'_{k-1}) + v(b'_{k-1}, x'_{k-1}) \\ &= -v(x'_{k-1}, x_1) + 5 \qquad (4.8) \\ -v(x_k, b'_{k-1}) &= 3 + v(x'_{k-1}, b'_{k-1}) + v(x_{k-1}, b'_{k-1}) \\ &= V_{k-1} + v(x'_{k-1}, x_1) + 4 \qquad (4.9) \end{aligned}$$

$$
\begin{aligned}
-v(x_k, x_{k-1}) &= 3 + v(b'_{k-1}, x_{k-1}) + v(b_{k-1}, x_{k-1}) \\
&= V_{k-1} + v(x'_{k-1}, x_1) + 4 && (4.10) \\
-v(x_k, b_{k-1}) &= 3 + v(x_{k-1}, b_{k-1}) + v(d_{k-1}, b_{k-1}) = 5 && (4.11) \\
-v(x_k, d_{k-1}) &= 3 + v(b_{k-1}, d_{k-1}) + v(z_{k-1}, d_{k-1}) \\
&= V_{k-1} - m_{k-1} + 4 && (4.12) \\
-v(x_k, z_{k-1}) &= 3 + v(d_{k-1}, z_{k-1}) + \left(\sum_{q \in P_{k-1};\, v(q, z_{k-1}) > 0} v(q, z_{k-1}) \right) \\
&= 2V_{k-1} - 2m_{k-1} + 2 && (4.13)
\end{aligned}
$$

where formula 3.9 has been applied in equation 4.13. The opposite values of weights 4.6–4.13 are summed up as

$$
- \sum_{p \in P_k \setminus P_{k-1}} v(x_k, p) = 5V_{k-1} + 31 \qquad (4.14)
$$

which is plugged in formula 4.5:

$$
V_k = 7V_{k-1} + 30\,. \qquad (4.15)
$$

Hence, for $n \geq 1$

$$
V_{n+1} = 7^{n-1}\,(V_2 + 5) - 5\,. \qquad (4.16)
$$

Furthermore, the value of V_2 is determined from definition 3.7

$$
\begin{aligned}
V_2 &= 1 - v(x'_2, x_1) - \sum_{p \in P_2 \setminus \{x_1\}} v(x_2, p) \\
&= 1 + 3m_2 + \sum_{p \in P_2} \left(\sum_{q \in P_2 \setminus \{p\};\, v(q,p) > 0} v(q, p) \right) + (12u + 39)n \\
&= 2W + (12u + 39)n + 48 && (4.17)
\end{aligned}
$$

by using formulas 3.4–3.6 and Figure 3.1. By introducing equation 4.17 into 4.16 the explicit formula for V_{n+1} is derived for $n \geq 1$:

$$
V_{n+1} = 7^{n-1}\,(2W + (12u + 39)n + 53) - 5\,. \qquad (4.18)
$$

The sum of absolute values of input weights

$$
\begin{aligned}
\Xi_{x_1} &= W + 9 + \varepsilon + 2W + 2(12u + 39)n \\
&\quad + \sum_{k=2}^{n+1} \left(|v(c_k, x_1)| + |v(x'_k, x_1)| + |v(z_k, x_1)| \right) && (4.19)
\end{aligned}
$$

to the interface clock unit x_1 including contribution $2W + 2(12u + 39)n$ from units $A \cup B$ in the simulating subnetwork is computed from Figures 3.1, 3.2, and 3.4 which reduces to

$$
\Xi_{x_1} = 3W + 2(12u + 39)n + 9 + \varepsilon + 2 \sum_{k=2}^{n+1} |v(x'_k, x_1)| \qquad (4.20)
$$

by using definition 3.8. Further, the absolute value of weight

$$|v(x'_k, x_1)| = 2^{k-2}\,|v(x'_2, x_1)| = 2^{k-2}(W + (12u + 39)n + 8) \tag{4.21}$$

in equation 4.20 is expressed for $k \geq 2$ from recursive formula 4.4 and Figure 3.1 which leads to

$$\Xi_{x_1} = 2^{n+1}(W + (12u + 39)n + 8) + W - 7 + \varepsilon\,. \tag{4.22}$$

Observe from Figure 3.4 that units α_j, β_j $(1 \leq j \leq n)$ have the maximum value

$$\Xi_{\alpha_j} = \Xi_{\beta_j} = 16u + 51 < \Xi_{x_1} \tag{4.23}$$

of Ξ_p in the simulating network which is still less than Ξ_{x_1} of the interface clock unit x_1 according to equation 4.22.

It can be checked in the Figure 3.2 that the maximum value of Ξ_p among the units p in the clock except for interface neuron x_1, is reached by unit z_{n+1} of the highest order $n + 1$, that is

$$\Xi_{z_{n+1}} = 2V_{n+1} - 2m_{n+1} + \varepsilon = 2V_{n+1} - 16n + 2 + \varepsilon < 2V_{n+1} \tag{4.24}$$

according to equation 3.9. For example,

$$\begin{aligned} \Xi_{x_{n+1}} &= 2V_{n+1} - 2v(x'_{n+1}, x_1) + 1 + \varepsilon \\ &= 2V_{n+1} - 2^n(W + (12u + 39)n + 8) + 1 + \varepsilon < \Xi_{z_{n+1}} \end{aligned} \tag{4.25}$$

by using equation 4.21. Thus, it follows from inequalities 4.23-4.25 that

$$\Xi_p < \max\left(2V_{n+1}, \Xi_{x_1}\right) \tag{4.26}$$

for every unit $p \in P$ in the underlying continuous-time Hopfield net. Actually, $\Xi_{x_1} < 2V_{n+1}$ for $n \geq 2$ which is not valid only for $n = 1$ according to equations 4.18, 4.22. In addition, the following two upper bounds

$$W \leq 42n^2 w_{\max} + 42n w_{\max} + 42n \tag{4.27}$$

$$u \leq 28n w_{\max} \tag{4.28}$$

are obtained from definitions 3.2 and 3.3, respectively, which together with formulas 4.18, 4.22 rewrite inequality 4.26 as

$$\Xi_p < 12 \cdot 7^n \left(10n^2 w_{\max} + 2n w_{\max} + 3n + 2\right) \tag{4.29}$$

For $n \geq 110$, inequality 4.29 implies $\Xi_p < w_{\max} 2^{3n} < \varepsilon 2^{1/\varepsilon}$ according to condition 3.1. For $n < 110$ the proposition 4.1 follows from assumption $\varepsilon < 0.0025$ of theorem 1.

4.1 Saturated Units

Consider a saturated continuous-time unit $p \in P$ whose state $y_p(t)$ is independent of outputs from the remaining neurons according to equation 2.5. Its continuous-time dynamics is described by a differential equation from system 2.6 depending on whether p is saturated at 0 or 1:

$$\frac{dy_p}{dt}(t) = \begin{cases} -y_p(t) & \text{for } \xi_p(t) \leq 0 \\ -y_p(t) + 1 & \text{for } \xi_p(t) \geq 1\,. \end{cases} \tag{4.30}$$

However, the saturated units may reversely influence excitations of other neurons since their analog states do not often exactly coincide with the binary values which they represent but they are only converging to them as the continuous time proceeds. Thus, an *error* $\delta_p(t) \geq 0$ of saturated unit p at time $t \geq 0$ can be defined as a difference between its state $y_p(t)$ and the binary value (0 or 1) at which p is saturated, that is

$$\delta_p(t) = \begin{cases} y_p(t) & \text{for } \xi_p(t) \leq 0 \\ 1 - y_p(t) & \text{for } \xi_p(t) \geq 1\,. \end{cases} \tag{4.31}$$

Assume that the saturated unit p has an initial error $\delta_p(t_0) = \delta_p$, $0 \leq \delta_p \leq 1$, measured at some point $t_0 \geq 0$ after p is saturated. Then the continuous-time evolution of its state $y_p(t)$ for $t \geq t_0$ can explicitly be expressed by solving equation 4.30 as follows:

$$y_p(t) = \begin{cases} \delta_p e^{-t'} & \text{for } \xi_p(t) \leq 0 \\ 1 - \delta_p e^{-t'} & \text{for } \xi_p(t) \geq 1 \end{cases} \tag{4.32}$$

where $t' = t - t_0$ is a *local time* starting at t_0. This confirms the fast convergence of state $y_p(t)$ towards its saturation.

Now, the effect of errors by saturated neurons from a given subset Q on the excitation $\xi_p(t)$ of any unit $p \in P$ in the network will be explored in local time starting at $t_0 \geq 0$ after all the neurons in Q are saturated. Note that only the units $q \in P$ incident to p count whereas the remaining non-incident ones are implicitly assumed to have zero weights $v(q,p) = 0$. First, $\xi_p(t)$ is split among the contributions from units outside Q and from those within Q saturated at 0 and 1 whose continuous-time state dynamics is described by equation 4.32 for $t \geq t_0$, that is

$$\begin{aligned} \xi_p(t) \;\; = \;\; & v(0,p) + \sum_{q \notin Q} v(q,p) y_q(t) + \sum_{q \in Q;\, \xi_q(t) \leq 0} v(q,p) \delta_p e^{-t'} \\ & + \sum_{q \in Q;\, \xi_q(t) \geq 1} v(q,p) \left(1 - \delta_p e^{-t'}\right)\,. \end{aligned} \tag{4.33}$$

Denote by

$$\Delta_{pQ} = \sum_{q \in Q;\, \xi_q(t_0) \leq 0} v(q,p) \delta_p - \sum_{q \in Q;\, \xi_q(t_0) \geq 1} v(q,p) \delta_p \tag{4.34}$$

an initial *total weighted error* of saturated neurons from Q affecting $\xi_p(t_0)$ at time t_0. Thus, equation 4.33 reduces to

$$\xi_p(t) = v(0,p) + \sum_{q \in Q;\, \xi_q(t) \geq 1} v(q,p) + \sum_{q \notin Q} v(q,p) y_q(t) + \Delta_{pQ} e^{-t'}\,. \tag{4.35}$$

Clearly, the error term $\Delta_{pQ}e^{-t'}$ in equation 4.35 vanishes quite quickly as the time proceeds. For example, for local time $t' \geq t_1$ where

$$t_1 = \frac{\ln 2}{\varepsilon} \tag{4.36}$$

the respective total error of saturated neurons from Q affecting $\xi_p(t)$ can simply be bounded by using lemma 1 in the following way:

$$\left|\Delta_{pQ}e^{-t'}\right| \leq \Xi_p e^{-t_1} < \varepsilon\,. \tag{4.37}$$

Finally, it can be checked that a unit p which is supposed to be saturated during some continuous-time interval of the underlying simulation, remains really saturated. According to the Hopfield net construction in section 3, such unit p is characterized by the existence of a subset Q of neurons saturated at the latest at $t_0 \geq 0$ which satisfy either

$$v(0,p) + \sum_{q\in Q;\, \xi_q(t_0)\geq 1} v(q,p) + \sum_{q\notin Q;\, v(q,p)>0} v(q,p) \leq -1 + 2\varepsilon \tag{4.38}$$

when p is saturated at 0 or

$$v(0,p) + \sum_{q\in Q;\, \xi_q(t_0)\geq 1} v(q,p) + \sum_{q\notin Q;\, v(q,p)<0} v(q,p) > 1 + \varepsilon \tag{4.39}$$

when p is saturated at 1. Hence, by introducing the error bound 4.37 into equation 4.35, the excitation of p in local time $t' \geq t_1$ starting at t_0 is either $\xi_p(t) < -1 + 3\varepsilon < 0$ (recall $\varepsilon < 0.0025$) in the former case or $\xi_p(t) > 1$ in the latter case which assures its saturation.

4.2 Activity Transfer in the Clock Network

In the clock subnetwork the discrete signal is typically propagated by gradual activating only one unit p at a continuous-time interval within which its incident neurons are saturated. Thus, only after p is saturated at 1 the next neuron r is ready to activate (unsaturate from 0). In this way, the activity is transmitted from one unit to another providing a lower bound on the transfer time for the purpose of synchronization.

Consider a situation depicted in Figure 4.1 where a clock unit $p \in P$ with feedback weight $1 + \varepsilon$ and a bias including fraction $0 < \varepsilon' \leq \varepsilon$ should transfer its activation to a subsequent clock unit $r \in P$ with bias fraction ε via a positive integer weight $v(p,r) \geq 1$. The possible pairs of units from the clock introduced in section 3.1 which p, r may represent are listed in table 4.1. Note that typically $\varepsilon' = \varepsilon$ except for p being b_k or b'_k when $\varepsilon' = \varepsilon/3$ is employed.

Further, suppose that all the neurons incident to unit p are saturated when p ceases its saturation at 0 at time $t_0 \geq 0$, that is $\xi_p(t_0) = 0$, so that the weights from the clock network (see Figures 3.1, 3.2) satisfy

$$v(0,p) + \sum_{q\in Q;\, \xi_q(t_0)\geq 1} v(q,p) = \varepsilon'\,. \tag{4.40}$$

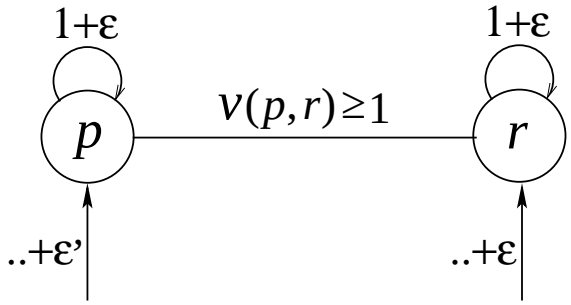

Figure 4.1: Activity transfer from p to r in the clock network

p	r	unit order
c_0	c_k	$1 \leq k \leq n+1$
c_k	a_k	$1 \leq k \leq n+1$
a_1	x_1	1
a_k	x'_k	$2 \leq k \leq n+1$
b'_k	x_k	$2 \leq k \leq n+1$
b_k	d_k	$1 \leq k \leq n+1$
d_k	z_k	$1 \leq k \leq n+1$
z_k	c_0	$1 \leq k \leq n+1$

Table 4.1: Activity transfer from p to r in the clock network

Thus, the excitation

$$\xi_p(t) = \varepsilon' + (1+\varepsilon)y_p(t) + \Delta_p e^{-t'} \tag{4.41}$$

of p in local time starting at t_0 is determined by equation 4.35 for $Q = P \setminus \{p\}$, where condition 4.40 is employed, and Δ_p stands for Δ_{pQ} in this special case representing the initial error of saturated units from Q that affects $\xi_p(t_0)$. Hence, the continuous-time state dynamics of p under local time is determined by the following differential equation from system 2.6:

$$\frac{dy_p}{dt}(t) = -y_p(t) + \varepsilon' + (1+\varepsilon)y_p(t) + \Delta_p e^{-t'} \tag{4.42}$$

together with the initial condition

$$y_p(t_0) = \frac{-\varepsilon' - \Delta_p}{1+\varepsilon} = \delta_p \tag{4.43}$$

which comes from $\xi_p(t_0) = 0$. Equation 4.43 also implies the following error bounds

$$-1 - \varepsilon - \varepsilon' \leq \Delta_p \leq -\varepsilon' < 0 \tag{4.44}$$

due to $1 \geq \delta_p \geq 0$.

The state evolution of unit p as the time $t \geq t_0$ grows can explicitly be expressed by solving equation 4.42 with initial condition 4.43 as follows:

$$y_p(t) = \frac{\varepsilon' \left(e^{\varepsilon t'} - 1\right)}{\varepsilon(1+\varepsilon)} - \frac{\varepsilon' + \Delta_p e^{-t'}}{1+\varepsilon} \tag{4.45}$$

from which an explicit formula for its derivative can be calculated:

$$\frac{dy_p}{dt}(t) = \frac{e^{-t'}\left(\varepsilon' e^{(1+\varepsilon)t'} + \Delta_p\right)}{1+\varepsilon}\,. \tag{4.46}$$

Hence, before the state $y_p(t)$ starts to grow, it is initially nonincreasing for the time interval $t_0 \leq t \leq t_0 + t_g$ where

$$t_g = \frac{\ln \frac{-\Delta_p}{\varepsilon'}}{1+\varepsilon} \tag{4.47}$$

since its derivative 4.46 is nonpositive here according to inequality 4.44.

By substituting 4.45 into equation 4.41 for $y_p(t)$, the excitation evolution of p can also be determined explicitly:

$$\xi_p(t) = \frac{\varepsilon' \left(e^{\varepsilon t'} - 1\right)}{\varepsilon} > 0\,. \tag{4.48}$$

Hence, $\xi_p(t)$ is clearly positive for local time $t' > 0$ confirming the state dynamics of p is controlled by equation 4.42 until p is saturated at 1, that is $\xi_p(t) = 1$, which happens exactly at local time

$$t_1' = \frac{\ln\left(1 + \frac{\varepsilon}{\varepsilon'}\right)}{\varepsilon} \geq t_1 \tag{4.49}$$

according to equation 4.48 (compare with inequality 4.36). Clearly, $t_1' > t_g$ for small $\varepsilon < 0.0025$ which shows the actual state growth of unit p. Notice that the length t_1' of time interval within which p is unsaturated (when being activated) is *constant* depending here only on parameters $\varepsilon, \varepsilon'$ while not being affected by any error. This important property is exploited for introducing the discrete time into the clock operation which is based on t_1 (see section 4.5). Finally, the state

$$y_p(t_0 + t_1') = \frac{1 - \varepsilon' - \Delta_p \left(1 + \frac{\varepsilon}{\varepsilon'}\right)^{-1/\varepsilon}}{1+\varepsilon} \tag{4.50}$$

of unit p at local time t_1' is determined from equation 4.45.

Note that unit c_0 at time $t = 0$ when the simulation starts, is already unsaturated and represents an exception from the state dynamics of p described above for which the respective formulas must slightly be adapted. In particular, $\Delta_p = 0$ in equations 4.41, 4.42 and 4.50. In addition, equations 4.43, 4.45, 4.46 and 4.48 are replaced by $y_p(t_0) = \varepsilon' = \varepsilon$, $y_p(t) = (1+\varepsilon)e^{\varepsilon t'} - 1$, $(dy_p)/(dt)(t) = \varepsilon(1+\varepsilon)e^{\varepsilon t'} > 0$, and $\xi_p(t) = (1+\varepsilon)^2 e^{\varepsilon t'} - 1$, respectively. Hence, unit c_0 starting with initial state ε at $t = 0$

immediately grows until it saturates at 1 at time $(1/\varepsilon)\ln(2/(1+\varepsilon)^2)$ which means that the first activating phase is, in fact, a little bit shorter than usual t_1'. One can check that also the following analysis is still valid in this case and actually simplifies since there are no saturation errors at the beginning.

Now, consider the next unit r with a bias including fraction ε which should receive the signal from p. From the clock network construction, the respective input weights to r are now assumed to satisfy

$$v(0,r) + \sum_{q\in Q;\, \xi_q(t_0)\geq 1} v(q,r) = \varepsilon - v(p,r) \tag{4.51}$$

where $v(p,r) \geq 1$ is a positive integer weight associated with the connection between p and r. Hence, its excitation $\xi_r(t)$ for $t \geq t_0$ can again be expressed by using equation 4.35 as

$$\xi_r(t) = \varepsilon - v(p,r) + v(p,r)y_p(t) + \Delta_{rQ}e^{-t'} \tag{4.52}$$

where Δ_{rQ} is the initial error of saturated units from the same Q as above (including r) that affects $\xi_r(t_0)$ exactly at time t_0 after which p becomes unsaturated. In the following,

$$\Delta_{rQ} + \delta_p < \varepsilon \tag{4.53}$$

(corresponding to inequality 4.37) will be assumed. In order to approve that the state dynamics of unit p is indeed controlled by equation 4.42 during the whole continuous-time interval $t_0 < t < t_0 + t_1'$ (i.e. until p is saturated at 1) it must be checked that unit r remains saturated at 0 in this period, that is $\xi_r(t) \leq 0$ justifying assumption 4.40. Since $v(p,r) \geq 1$ in equation 4.52 it suffices to prove

$$\xi_r(t) \leq \varepsilon + y_p(t) - 1 + \Delta_{rQ}e^{-t'} \leq 0 \tag{4.54}$$

which can be rewritten by substituting 4.45 for $y_p(t)$ where $-\Delta_p$ is replaced by $\varepsilon' + (1+\varepsilon)\delta_p$ according to equation 4.43, as follows:

$$\varepsilon\left(\varepsilon' + (1+\varepsilon)(\Delta_{rQ} + \delta_p)\right)e^{-t'} + \varepsilon'\left(e^{\varepsilon t'} - 1\right) - \varepsilon \leq \varepsilon(\varepsilon' - \varepsilon^2)\,. \tag{4.55}$$

Hence, it is sufficient to show that

$$\varepsilon\left(\varepsilon' + \varepsilon(1+\varepsilon)\right)e^{-t'} + \varepsilon'\left(e^{\varepsilon t'} - 1\right) - \varepsilon \leq \varepsilon(\varepsilon' - \varepsilon^2) \tag{4.56}$$

because condition 4.53 has been assumed. Consider first the local time interval $0 \leq t' \leq t_\varepsilon = \ln((\varepsilon' + \varepsilon(1+\varepsilon))/(\varepsilon' - \varepsilon^2))$ (recall $\varepsilon' \geq \varepsilon/3$) where term $e^{\varepsilon t'}$ has its maximum value at $t' = t_\varepsilon$ and $e^{-t'} \leq 1$ which imply inequality 4.56 for small $\varepsilon < 0.0025$. Within the remaining local time interval $t_\varepsilon \leq t' \leq t_1'$, term $\varepsilon(\varepsilon' + \varepsilon(1+\varepsilon))e^{-t'}$ in inequality 4.56 has its maximum value $\varepsilon(\varepsilon' - \varepsilon^2)$ at $t' = t_\varepsilon$ while $\varepsilon'(e^{\varepsilon t'} - 1) - \varepsilon$ is nonpositive here reaching zero at $t' = t_1'$. This completes the proof that unit r is saturated when p is unsaturated.

Furthermore, after p is saturated at 1 at local time t_1' (when all the neurons incident to p are also saturated) its state dynamics for $t' \geq t_1'$ is given by equation 4.32, that is

$$y_p(t) = 1 - \delta_p(t_0 + t_1')e^{-(t'-t_1')} \tag{4.57}$$

where

$$\delta_p(t_0 + t_1') = \frac{\varepsilon + \varepsilon' + \Delta_p \left(1 + \frac{\varepsilon}{\varepsilon'}\right)^{-1/\varepsilon}}{1 + \varepsilon} \tag{4.58}$$

comes from equation 4.50. By substituting expressions 4.57, 4.58 into equation 4.41, the corresponding excitation evolution

$$\xi_p(t) = 1 + (\varepsilon + \varepsilon')\left(1 - e^{-(t'-t_1')}\right) \geq 1 \tag{4.59}$$

is derived for $t' \geq t_1'$ which confirms that unit p remains saturated at 1 after $t_0 + t_1'$ at least until r becomes unsaturated. Also the excitation $\xi_r(t)$ of unit r after p saturates at 1 can be expressed for $t' \geq t_1'$ from equations 4.52, 4.57 as follows:

$$\xi_r(t) = \varepsilon - v(p,r)\delta_p(t_0 + t_1')e^{-(t'-t_1')} + \Delta_{rQ}e^{-t'} \tag{4.60}$$

which reaches value 0 at local time

$$t_2 = \ln \frac{v(p,r)\delta_p(t_0 + t_1')\left(1 + \frac{\varepsilon}{\varepsilon'}\right)^{1/\varepsilon} - \Delta_{rQ}}{\varepsilon}\,. \tag{4.61}$$

Thus, unit r is unsaturated after time $t_0 + t_2$ and its state dynamics is further described by differential equation of the form 4.42 which means that r saturates at 1 typically (excluding the case when r represents x_k' or x_k which will be analyzed in section 4.3) at time $t_0 + t_2 + t_1'$. In this way, the activity of unit p is transferred to neuron r.

Finally, it must be checked that unit p remains still saturated at 1 when r becomes unsaturated after time $t_0 + t_2$. The excitation $\xi_p(t)$ of p for local time $t' \geq t_2$ can be described as

$$\begin{aligned} \xi_p(t) \;=\; & \varepsilon' + 1 + \varepsilon + v(p,r)y_r(t) \\ & -(1+\varepsilon)\delta_p(t_0 + t_2)e^{-(t'-t_2)} + \left(\Delta_p - v(p,r)\delta_r\right)e^{-t'} \end{aligned} \tag{4.62}$$

according to equation 4.35 in which new $Q = P \setminus \{r\}$ (including saturated p) is actually used now while the error constants δ_r, Δ_p in equation 4.62 are still related to time t_0. The error $\delta_p(t_0 + t_2)$ of saturated unit p at local time t_2 can be calculated from formulas 4.57, 4.58, 4.61:

$$\delta_p(t_0 + t_2) = \frac{\varepsilon}{v(p,r) - \frac{(1+\varepsilon)\Delta_{rQ}}{(\varepsilon+\varepsilon')\left(1+\frac{\varepsilon}{\varepsilon'}\right)^{1/\varepsilon} + \Delta_p}}\,. \tag{4.63}$$

In order to prove that $\xi_p(t) \geq 1$ for local time $t' \geq t_2$, the corresponding negative error terms in equation 4.62 having the least value for $t' = t_2$ will be lower bounded by

$-\varepsilon' - \varepsilon$ whereas $v(p,r)y_r(t) > 0$. Thus, it is sufficient to prove

$$\begin{aligned} &-(1+\varepsilon)\delta_p(t_0+t_2) + (\Delta_p - v(p,r)\delta_r)\, e^{-t_2} \\ &\quad = \frac{-\varepsilon(1+\varepsilon)\left((\varepsilon+\varepsilon')\left(1+\frac{\varepsilon}{\varepsilon'}\right)^{1/\varepsilon} + v(p,r)\delta_r\right)}{v(p,r)\left((\varepsilon+\varepsilon')\left(1+\frac{\varepsilon}{\varepsilon'}\right)^{1/\varepsilon} + \Delta_p\right) - (1+\varepsilon)\Delta_{rQ}} > -\varepsilon' - \varepsilon \end{aligned} \tag{4.64}$$

where formulas 4.58, 4.61, and 4.63 have been applied. Inequality 4.64 further reduces to

$$\begin{aligned} &(1+\varepsilon)(\varepsilon+\varepsilon')\left(\varepsilon\left(1+\frac{\varepsilon}{\varepsilon'}\right)^{1/\varepsilon} + \Delta_{rQ}\right) \\ &\quad < v(p,r)\left((\varepsilon+\varepsilon')^2\left(1+\frac{\varepsilon}{\varepsilon'}\right)^{1/\varepsilon} + (\varepsilon+\varepsilon')\Delta_p - \varepsilon(1+\varepsilon)\delta_r\right) . \end{aligned} \tag{4.65}$$

According to conditions 4.43, 4.53 it suffices to show inequality 4.65 with Δ_{rQ} and Δ_p replaced by ε and $-\varepsilon' - (1+\varepsilon)\varepsilon$, respectively. In addition, $v(p,r) \geq 1$ and $\delta_r \leq 1$ which lead to

$$\begin{aligned} &(1+\varepsilon)(\varepsilon+\varepsilon')\left(\varepsilon\left(1+\frac{\varepsilon}{\varepsilon'}\right)^{1/\varepsilon} + \varepsilon\right) \\ <\ &(\varepsilon+\varepsilon')^2\left(1+\frac{\varepsilon}{\varepsilon'}\right)^{1/\varepsilon} - \varepsilon'(\varepsilon+\varepsilon') - \varepsilon(1+\varepsilon)(1+\varepsilon+\varepsilon') . \end{aligned} \tag{4.66}$$

For small $\varepsilon < 0.0025$ inequality 4.66 follows from $\varepsilon' \geq \varepsilon/3$. This completes the argument for unit p to be saturated when r is unsaturated. In addition, after local time $t' \geq t_2 + t_1'$, the error affecting $\xi_p(t)$ is bounded by inequality 4.37 and condition 4.39 suffices for proving that p is further saturated at 1.

4.3 The Continuous-Time Approximation of Discrete Computation

For a general continuous-time computation when more incident units in the Hopfield net are unsaturated at the same time, the network state evolution described by system 2.6, in which more differential equations are now mutually dependent, is too complicated to be analyzed in detail as in sections 4.1, 4.2 where, in fact, only one isolated differential equation has been considered at one time. Therefore, the continuous-time dynamics will be proved here only to simulate the discrete computation properly without using exact formulas for the state evolution.

Suppose a unit $p \in P$ from the underlying continuous-time Hopfield net has been unsaturated whose further saturation depends on a subset Q of incident neurons that saturate one after another so that the last one from Q saturates at time $t_0 > 0$. The excitation of unit p in local time starting at t_0 can be expressed by equation 4.35. The weights in the Hopfield net construction have been designed in section 3 so that either 4.38 or 4.39 is satisfied when p is supposed to become passive or active, respectively. Hence, p saturates as required at the latest at local time t_1 (see equation 4.36)

o	p	r	unit order	R
a_1	x_1	b_1	1	$\{c_0\} \cup A \cup B_{t_0}$
a_k	x'_k	b'_k	$2 \leq k \leq n+1$	$\{x_1\}$
b'_k	x_k	b_k	$2 \leq k \leq n+1$	$P_k \setminus \{x_1\}$

Table 4.2: Activity transfer from p to r when $q \in R$ are updated

according to inequality 4.37. In this way, the Hopfield net is shown to approximate a general discrete update 2.2 within continuous-time period of length at most t_1.

Furthermore, the issue of combining the special one-unit state dynamics from section 4.2 with the previous general case which come together in the clock units p listed in table 4.2 (see also section 3.1) will be analyzed. In addition, this table shows contextual units, namely, the preceding neuron o from which p receives its activation and the subsequent unit r to which the signal is further sent from p via unit weight $v(p,r) = 1$. This situation is depicted in Figure 4.2. Recall that units p, r have feedback weights $v(p,p) = v(r,r) = 1 + \varepsilon$ and the bias $v(0,p) = -1 + \varepsilon$ associated with p includes fraction ε while that $v(0,r) = -1 + \varepsilon/3$ with unit r employs fraction $\varepsilon/3$. The activity transfer is realized in a similar way as described in section 4.2. However, unit p is now additionally connected to other neurons $q \in R$ that may become unsaturated after p ceases its saturation at 0 at time $t_0 > 0$. For each possible p, the corresponding set R of these units from the Hopfield net (see section 3) is also presented in table 4.2 where $B_{t_0} = \{q \in B;\, \xi_q(t_0) \geq 1\}$ when p represents the interface clock unit x_1. Note that $\phi_j \in B_{t_0} \longleftrightarrow \chi_j \notin B_{t_0}$ for the same $j = 1, \ldots, n$ since neuron χ_j from the simulating subnetwork stores the negation of discrete state kept by unit ϕ_j. Thus, the state dynamics for activating p that has originally been determined by equation 4.45 is not valid anymore and the analysis must be tailored to this case accordingly.

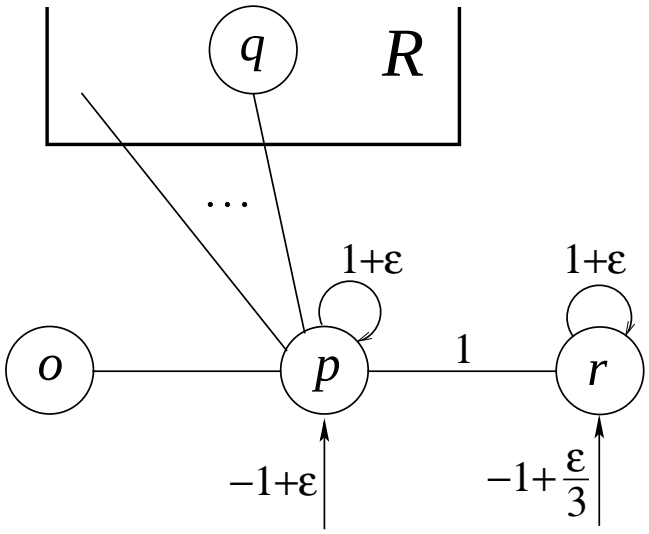

Figure 4.2: Activity transfer from p to r when $q \in R$ are updated

Notice that unit o has been activated exactly by the process described in section 4.2 in which p, r are now replaced by o, p, respectively, and hence, o saturates at 1 before p is unsaturated from 0 at time t_0. The excitation $\xi_p(t)$ of unit p for local time t' starting at t_0 can be expressed as

$$\begin{aligned} \xi_p(t) &= v(0,p) + \sum_{q \in Q;\, \xi_q(t_0) \geq 1} v(q,p) + (1+\varepsilon) y_p(t) \\ &\quad + \sum_{q \in R} v(q,p) y_q(t) + \Delta_{pQ}\, e^{-t'} \end{aligned} \tag{4.67}$$

by using equation 4.35 for $Q = P \setminus (R \cup \{p\})$ (including o, r) where Δ_{pQ} is the initial weighted error of saturated neurons from Q affecting $\xi_p(t_0)$. It follows from the Hopfield net construction (see section 3) that the corresponding weights satisfy

$$v(0,p) + \sum_{q \in Q;\, \xi_q(t_0) \geq 1} v(q,p) + \sum_{q \in R;\, v(q,p) < 0} v(q,p) = \varepsilon \tag{4.68}$$

which gives

$$\xi_p(t) \geq \varepsilon + (1+\varepsilon) y_p(t) + \Delta_{pQ}\, e^{-t'} \tag{4.69}$$

for $t \geq t_0$. According to the dynamics 2.6 this also provides the following bound on its state derivative

$$\frac{dy_p}{dt}(t) \geq \varepsilon y_p(t) + \varepsilon + \Delta_{pQ}\, e^{-t'}\,. \tag{4.70}$$

Also observe that in the beginning the state evolution of unit p is determined by the dynamics 4.45 before the first $q \in R$ is unsaturated since equation 4.68 coincides with 4.40 (recall $\varepsilon' = \varepsilon$ for p now) due to $\xi_q(t_0) \geq 1$ for all $q \in R$ with $v(q,p) < 0$. Therefore by definition 4.34, the error Δ_{pQ} can be expressed in terms of the initial total weighted error Δ_p of saturated neurons incident to p (excluding p) affecting $\xi_p(t_0)$ at time t_0 as follows:

$$\Delta_{pQ} = \Delta_p - \sum_{q \in R;\, \xi_q(t_0) \leq 0} v(q,p)\delta_q + \sum_{r \in R;\, \xi_q(t_0) \geq 1} v(q,p)\delta_q \tag{4.71}$$

In the following, assume that the initial weighted error of units from $R \cup \{p\}$ (outside Q) incident to p is bounded:

$$(1+\varepsilon)\delta_p + \sum_{q \in R;\, \xi_q(t_0) \leq 0} v(q,p)\delta_q - \sum_{q \in R;\, \xi_q(t_0) \geq 1} v(q,p)\delta_q \leq \varepsilon 2^{-1/\varepsilon} \tag{4.72}$$

which actually strengthens inequality 4.37 (for complement of current Q) by applying $2t_1$ instead of t_1. Further, $-\varepsilon - (1+\varepsilon)\delta_p$ is substituted for Δ_p in formula 4.71 according to 4.43 and condition 4.72 is employed in order to bound the error

$$\Delta_{pQ} \geq -\varepsilon \left(1 + 2^{-1/\varepsilon}\right)\,. \tag{4.73}$$

By introducing inequality 4.73 into 4.70, the state derivative of unit p is further lower bounded for $t \geq t_0$ as follows:

$$\frac{dy_p}{dt}(t) \geq \varepsilon y_p(t) + \varepsilon - \varepsilon \left(1 + 2^{-1/\varepsilon}\right) e^{-t'} . \tag{4.74}$$

Thus, for local time $t' \geq t_d$ where

$$t_d = \ln \frac{1 + 2^{-1/\varepsilon}}{\varepsilon} , \tag{4.75}$$

the state derivative of p is at least

$$\frac{dy_p}{dt}(t) \geq \varepsilon - \varepsilon^2 > 0 \tag{4.76}$$

since $\varepsilon y_p(t) \geq 0$ in equality 4.74. Hence, the state $y_p(t)$ of unit p for local time $t' \geq t_d$ grows at least as fast as the straight line described by equation $(\varepsilon - \varepsilon^2)(t' - t_d) - y = 0$ until p saturates at 1. Also $\xi_p(t) > y_p(t)$ for $t \geq t_0 + t_d$ from 2.6 due to the derivative 4.76 is positive. Therefore, unit p is certainly saturated at 1 after local time $t_d + t_s$ where

$$t_s = \frac{1}{\varepsilon - \varepsilon^2} \tag{4.77}$$

corresponds to the point at which the underlying straight line crosses value 1. Hence, the time within which p saturates can here be upper bounded by $2t_1$ (see equation 4.36) which is greater than $t_d + t_s$ for small $\varepsilon < 0.0025$.

In addition, it will be proved that the subsequent unit r is saturated at 0 until p saturates at 1. The excitation $\xi_r(t)$ of saturated unit r for $t \geq t_0$ can be upper bounded similarly as in equation 4.54 where the bias fraction $\varepsilon/3$ is now considered:

$$\xi_r(t) \leq \frac{\varepsilon}{3} + y_p(t) - 1 + \Delta_{rQ}\, e^{-t'} \leq 0 \tag{4.78}$$

which implies the following bound on the state $y_p(t)$ of unit p

$$y_p(t) \leq 1 - \frac{\varepsilon}{3} - \Delta_{rQ}\, e^{-t'} \tag{4.79}$$

ensuring r is saturated at 0. Also assume

$$\Delta_{rQ} < \varepsilon 2^{-1/\varepsilon} \tag{4.80}$$

which strengthens inequality 4.53 again by applying $2t_1$ instead of t_1 in inequality 4.37. Consider the least local time instant $t_y > 0$ at which the equality in condition 4.79 holds, that is

$$y_p(t_0 + t_y) = 1 - \frac{\varepsilon}{3} - \Delta_{rQ}\, e^{-t_y} \tag{4.81}$$

which means that r is still saturated at 0. In order to prove that p is already saturated at 1 at local time t_y it suffices to show that

$$\varepsilon + (1 + \varepsilon) \left(1 - \frac{\varepsilon}{3} - \Delta_{rQ}\, e^{-t_y}\right) + \Delta_{pQ}\, e^{-t_y} \geq 1 \tag{4.82}$$

according to inequality 4.69. By substituting error bounds 4.73, 4.80 for Δ_{pQ}, Δ_{rQ}, respectively, this can further be rewritten as

$$5 - \varepsilon - 3\left(1 + (2+\varepsilon)2^{-1/\varepsilon}\right) e^{-t_y} \geq 0 \tag{4.83}$$

which follows from $e^{-t_y} < 1$ for small $\varepsilon < 0.0025$. Thus, p is indeed saturated at 1 at local time t_y when r is saturated at 0.

Finally, for local time $t' \geq t_y$ when the state dynamics of saturated p is of the form 4.32, unit r may unsaturate and hence, the inequality 4.69 reads now as

$$\begin{aligned} \xi_p(t) \;\; &\geq \;\; \varepsilon + (1+\varepsilon)\left(1 - \left(\frac{\varepsilon}{3} + \Delta_{rQ}\right) e^{-(t'-t_y)}\right) + y_r(t) \\ &\quad + (\Delta_{pQ} - \delta_r)\, e^{-t'} \end{aligned} \tag{4.84}$$

(recall $v(p,r) = 1$) where $\delta_r = \delta_r(t_0)$ is the error of unit r saturated at time t_0 whose contribution to p is assumed to be bounded by

$$\delta_r < \varepsilon 2^{-1/\varepsilon} \tag{4.85}$$

at that time (compare to inequality 4.80). In order to prove that p remains saturated at 1 for local time $t' \geq t_y$, that is $\xi_p(t) \geq 1$ for $t \geq t_0 + t_y$, it suffices to prove

$$6 - (1+\varepsilon)\left(1 + 3 \cdot 2^{-1/\varepsilon}\right) e^{-(t'-t_y)} - 3\left(1 + 2^{-1/\varepsilon+1}\right) e^{-t'} \geq 0 \tag{4.86}$$

according to inequality 4.84 in which $y_r(t) \geq 0$ and error bounds 4.73, 4.80, and 4.85 have been applied for Δ_{pQ}, Δ_{rQ}, and δ_r, respectively. The inequality 4.86 follows from $e^{-t'} \leq e^{-(t'-t_y)} \leq 1$ for small $\varepsilon < 0.0025$. This completes the proof that p remains saturated at 1 when r is unsaturated.

4.4 The Synchronization of Simulation

The analysis from section 4.3 will be continued here in order to ensure the correct simulation control. In particular, the interface clock unit x_1 controls the simulating subnetwork by locking and unlocking the underlying gates $A \cup B$. It will be checked that locking always precedes unlocking which synchronizes the simulation. For this purpose, observe first that for its state value $y_{x_1}(t_\ell) = 2/3$ at some time instant $t_\ell > 0$, all the relevant incident units $\alpha_j, \beta_j, \phi_j, \chi_j \in A \cup B$ from the subnetworks simulating the discrete neurons $j = 1, \ldots, n$ are locked which means they are saturated at 0:

$$\xi_{\alpha_j}(t_\ell) \;\; \leq \;\; v(0,\alpha_j) + v(\psi_j,\alpha_j) + v(\zeta_j,\alpha_j) + v(x_1,\alpha_j)\frac{2}{3} = -2 \tag{4.87}$$

$$\xi_{\beta_j}(t_\ell) \;\; \leq \;\; v(0,\beta_j) + v(\eta_j,\beta_j) + v(x_1,\beta_j)\frac{2}{3} = -2 \tag{4.88}$$

$$\xi_{\phi_j}(t_\ell) \;\; \leq \;\; v(0,\phi_j) + u + v(\psi_j,\phi_j) + v(\omega_j,\phi_j) + v(x_1,\phi_j)\frac{2}{3} \leq -2 \tag{4.89}$$

$$\xi_{\chi_j}(t_\ell) \;\; \leq \;\; v(0,\chi_j) + u + v(x_1,\chi_j)\frac{2}{3} \leq -42\,. \tag{4.90}$$

Moreover, it must be verified that unit x_1, after crossing the state value $y_{x_1}(t_\ell) = 2/3$ at time t_ℓ, will saturate at either 1 when $y_{x_1}(t_\ell)$ is increasing or 0 when $y_{x_1}(t_\ell)$ is decreasing before reaching this state value next time.

Consider first the case when the state $y_{x_1}(t_0+t'_\ell)$ of unit x_1 is increasing at local time $t'_\ell > 0$ when it achieves value $y_{x_1}(t_0 + t'_\ell) = 2/3$ after x_1 has ceased its saturation at 0 at time t_0. Obviously, $y_{x_1}(t)$ is further increasing for $t \geq t_0 + t'_\ell$ since $(dy_{x_1})/(dt)(t) > 0$ for $y_{x_1}(t_0+t'_\ell) = 2/3$ according to 4.74 where p stands for x_1 and hence, x_1 will saturate at 1.

On the other hand, let the state $y_{x_1}(t_0 + t'_\ell)$ of unit x_1 be decreasing at local time $t'_\ell > 0$ when it achieves value $y_{x_1}(t_0 + t'_\ell) = 2/3$ after the state $y_{x'_k}(t)$ of unit x'_k $(2 \leq k \leq n+1)$ has increased ceasing its saturation at 0 at time t_0, that is $(dy_{x_1})/(dt)(t_0 + t'_\ell) < 0$. It will first be proved that the state $y_{x'_k}(t)$ of unit x'_k keeps increasing after x_1 unsaturates from 1. Thus, consider the least local time instant $t_u > 0$ at which the excitation $\xi_{x_1}(t_0 + t_u)$ of unit x_1 equals 1, that is

$$\begin{aligned} \xi_{x_1}(t_0 + t_u) &= W + (12u + 39)n + 3 + 2\varepsilon \\ &\quad + v(x'_k, x_1) y_{x'_k}(t_0 + t_u) + \Delta_{x_1 Q_1}\, e^{-t_u} = 1 \end{aligned} \tag{4.91}$$

according to equation 4.35 for $Q_1 = P \setminus \{x'_k\}$ in which

$$\begin{aligned} v(0, x_1) + \sum_{q \in Q_1;\, \xi_q(t) \geq 1} v(q, x_1) &= v(0, x_1) + v(x_1, x_1) + v(c_0, x_1) \\ + v(a_1, x_1) + v(b_1, x_1) + \sum_{q \in A_{t_0}} v(q, x_1) &= W + (12u + 39)n + 3 + 2\varepsilon \end{aligned} \tag{4.92}$$

where $A_{t_0} = \{q \in A;\, \xi_q(t_0) \geq 1\}$ (recall $\alpha_j \in A_{t_0} \longleftrightarrow \beta_j \notin A_{t_0}$ for the same $j = 1, \ldots, n$), and $\Delta_{x_1 Q_1}$ is the initial total weighted error of saturated neurons from Q_1 affecting $\xi_{x_1}(t_0)$. Further assume

$$\Delta_{x_1 Q_1} > -\varepsilon \tag{4.93}$$

which gives

$$y_{x'_k}(t_0 + t_u) \geq \frac{W + (12u + 39)n + 2 + \varepsilon}{-v(x'_k, x_1)} \tag{4.94}$$

according to equation 4.91. The state derivative of x'_k at local time $t' > 0$ is lower bounded by inequality 4.74 for p being x'_k where $e^{-t'} < 1$ as follows:

$$\frac{dy_{x'_k}}{dt}(t) \geq \varepsilon y_{x'_k}(t) - \varepsilon 2^{-1/\varepsilon}\,. \tag{4.95}$$

Clearly, in order to prove that the state $y_{x'_k}(t)$ of unit x'_k is increasing for $t \geq t_0 + t_u$ it suffices to show according to inequality 4.95 that the state derivative $(dy_{x'_k})/(dt)(t_0 + t_u) > 0$ of x'_k at local time t_u is positive since $y_{x'_k}(t)$ is continuous. By introducing inequality 4.94 into 4.95 for $t = t_0 + t_u$ it is sufficient to prove

$$\frac{dy_{x'_k}}{dt}(t_0 + t_u) \geq \frac{\varepsilon\,(W + (12u + 39)n + 2 + \varepsilon)}{-v(x'_k, x_1)} - \varepsilon 2^{-1/\varepsilon} > 0 \tag{4.96}$$

which follows from $-v(x'_k, x_1) < \varepsilon 2^{1/\varepsilon}$ by lemma 1. This completes the proof that the state of unit x'_k keeps increasing after x_1 unsaturates from 1.

The excitation $\xi_{x_1}(t)$ of unit x_1 for time $t \geq t_0$ after x'_k unsaturates from 0 can be expressed as

$$\begin{aligned} \xi_{x_1}(t) &= W + 1 + \varepsilon + (1+\varepsilon) y_{x_1}(t) + v(x'_k, x_1) y_{x'_k}(t) \\ &\quad + \sum_{q \in A_{t_0}} v(q, x_1) y_q(t) + \sum_{q \in B} v(q, x_1) y_q(t) + \Delta_{x_1 Q'_1} \, e^{-t'} \end{aligned} \tag{4.97}$$

(compare with 4.91) according to equation 4.35 for $Q'_1 = P \setminus (A_{t_0} \cup B \cup \{x_1, x'_k\})$ in which

$$\begin{aligned} v(0, x_1) + \sum_{q \in Q'_1;\, \xi_q(t) \geq 1} v(q, x_1) &= v(0, x_1) + v(c_0, x_1) + v(a_1, x_1) + v(b_1, x_1) \\ &= W + 1 + \varepsilon \,, \end{aligned} \tag{4.98}$$

and $\Delta_{x_1 Q'_1}$ is the initial total weighted error of saturated neurons from Q'_1 affecting $\xi_{x_1}(t_0)$. In the following, assume

$$\delta_{x_1}(t_0) = \delta_{x_1} < \varepsilon 2^{-1/\varepsilon} \tag{4.99}$$

$$\sum_{q \in A_{t_0}} v(q, x_1) \delta_q(t_0) < \varepsilon 2^{-1/\varepsilon} \tag{4.100}$$

$$\sum_{q \in B} v(q, x_1) \delta_q(t_0) > -\varepsilon 2^{-1/\varepsilon} \tag{4.101}$$

$$\Delta_{x_1 Q'_1} > -\varepsilon 2^{-1/\varepsilon} \,. \tag{4.102}$$

In addition, denote by $0 < t'_d < t'_\ell$ the least local time instant at which $(dy_{x_1})/(dt)(t_0 + t'_d) = 0$ and thus, $t'_d > t_u$ (see definition 4.91) according to dynamics 4.32. The state derivative $(dy_{x_1})/(dt)(t)$ of x_1 for time $t > t_0 + t_u$ can be expressed from dynamics 2.6 providing that x_1 is unsaturated (i.e. $1 > \xi_{x_1}(t) > 0$) as follows:

$$\begin{aligned} \frac{dy_{x_1}}{dt}(t) &= W + 1 + \varepsilon + \varepsilon y_{x_1}(t) + v(x'_k, x_1) y_{x'_k}(t) \\ &\quad + \sum_{q \in A_{t_0}} v(q, x_1) y_q(t) + \sum_{q \in B} v(q, x_1) y_q(t) + \Delta_{x_1 Q'_1} \, e^{-t'} \end{aligned} \tag{4.103}$$

where excitation 4.97 has been employed. Now, the change in values of particular terms in derivative 4.103 will be analyzed between time instants $t_0 + t'_d$ and $t_0 + t'_\ell$. The state decrease $y_{x_1}(t_0 + t'_d) > y_{x_1}(t_0) = 1 - \delta_{x_1} > 1 - \varepsilon 2^{-1/\varepsilon} > 2/3 = y_{x_1}(t_0 + t'_\ell)$ by assumption 4.99 implies

$$\varepsilon y_{x_1}(t_0 + t'_\ell) - \varepsilon y_{x_1}(t_0 + t'_d) < -\frac{\varepsilon}{3} + \varepsilon^2 2^{-1/\varepsilon} \,. \tag{4.104}$$

Further, for any local time $t' \geq t'_d$

$$v(x'_k, x_1) y_{x'_k}(t_0 + t') - v(x'_k, x_1) y_{x'_k}(t_0 + t'_d) < 0 \tag{4.105}$$

since $v(x'_k, x_1) < 0$ and $y_{x'_k}(t)$ is increasing for $t \geq t_0 + t_u$. In addition, $v(q, x_1) > 0$ for $q \in A_{t_0}$ and all the units in A_{t_0} are still saturated at 1 at $t_0 + t'_d$ which means

$$\sum_{q \in A_{t_0}} v(q, x_1) y_q(t_0 + t') - \sum_{q \in A_{t_0}} v(q, x_1) y_q(t_0 + t'_d) < \varepsilon 2^{-1/\varepsilon} \tag{4.106}$$

for any local time $t' \geq t'_d$ according to error bound 4.100. Similarly, $v(q, x_1) < 0$ for $q \in B$ and all the units in B are still saturated at 0 at $t_0 + t'_d$ which gives

$$\sum_{q \in B} v(q, x_1) y_q(t_0 + t') - \sum_{q \in B} v(q, x_1) y_q(t_0 + t'_d) < \varepsilon 2^{-1/\varepsilon} \tag{4.107}$$

for any local time $t' \geq t'_d$ according to assumption 4.101. Also the error term satisfies

$$\Delta_{x_1 Q'_1} e^{-t'} - \Delta_{x_1 Q'_1} e^{-t'_d} < \varepsilon 2^{-1/\varepsilon} \tag{4.108}$$

for any local time $t' \geq t'_d$ due to condition 4.102. From formula 4.103 the state derivative of unit x_1 at time $t_0 + t'_\ell$ providing that x_1 is unsaturated can be upper bounded in the following way:

$$\frac{dy_{x_1}}{dt}(t_0 + t'_\ell) = \frac{dy_{x_1}}{dt}(t_0 + t'_\ell) - \frac{dy_{x_1}}{dt}(t_0 + t'_d) < -\frac{\varepsilon}{3} + \varepsilon(3 + \varepsilon) 2^{-1/\varepsilon} \tag{4.109}$$

by using inequalities 4.104, and 4.105–4.108 for $t' = t'_\ell$. The upper bound 4.109 on the state derivative of unsaturated x_1 is further valid for $t \geq t_0 + t'_\ell$ because inequalities 4.105–4.108 still apply and 4.104 holds also for local time $t' \geq t'_\ell$ from 4.109 since $y_{x_1}(t)$ is continuous. Thus, the state $y_{x_1}(t_0 + t')$ of unsaturated x_1 for local time $t' \geq t'_\ell$ is always decreasing which is also true for x_1 saturated at 0 according to its state dynamics 4.32, implying that $y_{x_1}(t_0 + t') \neq 2/3$ for local time $t' > t'_\ell$. This completes the argument for the correct synchronization of simulation.

4.5 Timing

In this section, the correct timing of the simulation will be verified. This is an important issue since the preceding analysis in sections 4.1– 4.4 works only if the respective error bounds are satisfied. It appears that the total weighted error by saturated neurons affecting any unit in the underlying Hopfield net vanishes as the continuous time runs. According to inequality 4.37, the absolute value of this error is bounded by ε after a continuous-time interval of length t_1 (see definition 4.36) which further decreases to $\varepsilon 2^{-1/\varepsilon}$ for time $2t_1$, etc. On the other hand, t_1 represents a lower bound on time that is necessary for activating a typical unit (see p in table 4.1) in the clock network from section 4.2 according to inequality 4.49. Hence, t_1 represents a time unit that discretizes continuous time in order to cope with the errors in the analog representation of binary states during the simulation.

The correct timing will first be inductively checked for the clock subnetwork. Consider a situation when the least significant counter unit c_0 is ready to start its activation as described in section 4.2 just before it is unsaturated from 0. Let c_k $(k \geq 1)$ also

saturated at 0 be the next counter unit to be activated which means that all the units in P_k except for c_0 are saturated at 1 at that time. The error by saturated units affecting c_k is by induction assumed to be less than ε to satisfy condition 4.53 for $r = c_k$ which guarantees the correct c_0 activation. Thus, after c_0 saturates at 1 which takes time at least t_1, assumption 4.53 is further satisfied certainly for a_k, d_k, and x'_k if $k \geq 2$ which justifies the correct activations of c_k, b_k, and a_k for $k \geq 2$, respectively, when their time comes. Further, c_k saturates at 1 after time at least t_1 during which the error of saturated c_0 decreases so that also x_k affected by c_0 satisfies condition 4.53 and either a_1 may immediately activate for $k = 1$ or b'_k for $k \geq 2$ is ready to do so later on.

After c_k, a_k have been activated which took time altogether at least $2t_1$, units p from table 4.2 may activate as described in section 4.3 since the corresponding assumptions 4.72, 4.80, and 4.85 are now satisfied. In fact, condition 4.72 when p represents x_1 must still be checked as concerns units $A \cup B_{t_0}$ from the simulating subnetwork which will be done below. Further, p is saturated at 1 at the latest in time $2t_1$ which is actually only the upper bound. Then the subsequent unit r (see table 4.2) is activated which takes time at least $2t_1$ because of fraction $\varepsilon' = \varepsilon/3$ in its bias for which $t'_1 = 2t_1$ from equation 4.49. During this period of length $2t_1$ the units in the corresponding set R which are linked with saturated neuron p first approximate general discrete computation as described at the beginning of section 4.3 and thus, also saturate in time at least t_1 as required. Within the second t_1 part of this time interval also the error of saturated units R further decreases so that particularly for $p = x_k$, unit z_k which is linked to R satisfies condition 4.53 and hence, d_k may immediately activate. In this case, assumption 4.53 is also met for $c_0 \in R$ and for the next counter unit to be activated after c_0 activates again (which is either $c_1 \in R$ for $k \geq 2$ or c_2 for $k = 1$), implying the correct activation for z_k and c_0, respectively, in the next induction step. This completes the argument for the correct timing within the clock.

Now, the correct timing will be shown for the simulating subnetwork. From the previous clock timing analysis follows that the first phase of simulated discrete update described in section 3.2 when neuron x_1 is saturated at 0 with a small error takes time at least $3t_1$ that is necessary for activating preceding units c_0, c_1, a_1 which cause x_1 to activate afterwards. Recall that at the beginning of the first phase the current binary state $\mathbf{y}^{(t)}$ of the simulated network from its discrete time t is stored in units π_j, ϱ_j $(j = 1, \ldots, n)$ which are saturated with small errors accordingly. Hence, the subsequent neurons ϕ_j and χ_j are also saturated at the new state binary state $\mathbf{y}^{(t+1)}$ and its negation, respectively, approximating discrete computation as described at the beginning of section 4.3 since the error by saturated x_1 is assumed to be already small. Thus, after time t_1, units ψ_j, ω_j are saturated and store $\mathbf{y}^{(t+1)}$. Further, during the remaining continuous-time interval of length $2t_1$, all the saturated units from the simulating network participating in the first phase have error certainly less than $\varepsilon 2^{-1/\varepsilon}$ and the respective neurons $\phi_j, \chi_j \in B_{t_0}$ satisfy the above-mentioned assumption 4.72 for the next correct activation of x_1. Also unit x_1, after being saturated at 1, has plenty of time when neurons b_1, d_1, z_1 are being activated (which takes time at least $4t_1$), to decrease its error sufficiently and be ready for the second phase.

Similarly, the second phase with x_1 being saturated at 1 with a small error takes time at least $3t_1$ that is necessary for activating preceding units c_0, c_k, a_k $(k \geq 2)$ which

results in passive x_1 afterwards. Thus, units α_j and β_j ($j = 1, \ldots, n$) are saturated so that they represent $\mathbf{y}^{(t+1)}$ and its negation, respectively, by approximating discrete computation as described at the beginning of section 4.3 since the error by saturated x_1 is assumed to be already small. In time at least t_1 these binary states are copied to ζ_j, η_j and further stored in π_j, ϱ_j after next t_1. Anyway, the errors of the participating units after the second phase of length at least $3t_1$ are small enough to meet conditions 4.93 and 4.99–4.102 for the correct synchronization. Also unit x_1, after being saturated at 0, has plenty of time when neurons b_k, d_k, z_k are being activated, to decrease its error sufficiently and be ready for the first phase again. This completes the proof that timing of the simulation is correct ensuring the required error decrease.

Finally, the duration of the simulation (see theorem 1) will be estimated. The lower bound $\Omega(t^\star/\varepsilon)$ where $t^\star$ is the number of simulated discrete updates, follows immediately from the previous time analysis by using equation 4.36. On the other hand, during the simulation every unit, after being unsaturated, saturates certainly within time at most $3t_1$. In addition, the time interval when all the units in the network are temporarily saturated takes time at most t_1 during which the error decreases within ε so that the next neuron unsaturates. This implies the corresponding upper bound $O(t^\star/\varepsilon)$ and completes the proof of theorem 1.

5 Convergence Time Lower Bound

The continuous-time clock network from section 3.1 can also be exploited to achieve lower bound $\Omega(2^{d/6}/\varepsilon)$ on the convergence time of continuous-time symmetric Hopfield nets which is exponential in the network size d. In fact, the corresponding construction of the simulated binary counter can be simplified for this purpose since no synchronization of the simulation is needed anymore. To the best of our knowledge, this provides the first known lower bound on the convergence time of continuous-time Hopfield nets.

This result can also be expressed as $2^{\Omega(g(M))}$ for $g(M) = o(M)$ where M represents the number of bits that are sufficient for encoding the underlying Hopfield net. When comparing this lower bound to a general convergence time upper bound of $2^{O(\sqrt{N})}$ for discrete Hopfield networks with N-bit representations (Šíma et al., 2000), the continuous-time implementation actually yields better bounds than the discrete-time one, assuming that the time interval between two subsequent discrete updates corresponds to a continuous time unit. This implies that continuous-time analog models of computation may be worth investigating more for their efficiency gains than for their (theoretical) capability for arbitrary-precision real number computation (Balcázar et al., 1997; Siegelmann and Sontag, 1994; Siegelmann and Sontag, 1995). The following theorem summarizes the respective convergence time result:

Theorem 2 *There exists a continuous-time symmetric Hopfield net composed of d units whose last shift between saturation at 0 and 1 appears after continuous time $\Omega(2^{d/6}/\varepsilon)$ for any $0 < \varepsilon < 0.0025$ such that $2^{d/2} \leq \varepsilon 2^{1/\varepsilon}$. In terms of bit representations this bound translates to convergence time $2^{\Omega(g(M))}$ where $g(M)$ is an arbitrary continuous function such that $g(M) = o(M)$, $g(M) = \Omega(M^{2/3})$, and $M/g(M)$ is increasing.*

Proof: As it has already been mentioned, the construction of $(n+1)$-bit simulated counter from section 3.1 will be revised in order to reduce its size to $d = 6n + 1$ units when no simulation is considered. However, this is a quite simple task since units x'_k, b'_k for $k = 2, \ldots, n$ that have treated the interface clock unit x_1 separately to synchronize the simulation can now be omitted. Thus, for $k = 2, \ldots, n$ unit a_k is directly linked with x_k (similarly as it is for $k = 1$) via weight $v(a_k, x_k) = V_k$ and x_1 is reconnected from x'_k to x_k while the original weights remain. Also parameter $W = 0$ disappears in the definition of weight $v(a_1, x_1)$ since there are no connections to a simulating subnetwork. Clearly, the operation of the clock can be checked to not be influenced by this revision. Hence, this counter network starting with its initial zero state does not converge sooner than continuous time $2^n/\varepsilon$ passes which gives the desired lower bound $\Omega(2^{d/6}/\varepsilon)$ on the convergence time of continuous-time Hopfield nets of size d units.

Now, this bound will be expressed in terms of size M in bits of the network representation. From the proof of lemma 1, the maximum integer weight parameter in the clock is of order $2^{O(d)}$. This corresponds to $O(d)$ bits per weight that is repeated $O(d^2)$ times, and thus yields at most $O(d^3)$ bits in the representation. In addition, the biases and feedbacks of the d units include fraction ε (or $\varepsilon/3$), and taking this into account requires $\Theta(d \log(1/\varepsilon))$ additional bits, say at least $\kappa d \log(1/\varepsilon)$ bits for some constant $\kappa > 0$.

By choosing $\varepsilon = 2^{-f(d)/(\kappa d)}$ in which f is a continuous increasing function whose inverse is defined as $f^{-1}(\mu) = \mu/g(\mu)$, where g satisfies $g(\mu) = \Omega(\mu^{2/3})$ (implying $f(d) = \Omega(d^3)$) and $g(\mu) = o(\mu)$, it follows that $M = \Theta(f(d))$, especially $M \geq f(d)$ from $M \geq \kappa d \log(1/\varepsilon)$. Finally, the convergence time $\Omega(2^{d/6}/\varepsilon)$ can be translated to $\Omega(2^{f(d)/(\kappa d)+d/6}) = 2^{\Omega(f(d)/d)}$ which can be rewritten as $2^{\Omega(M/f^{-1}(M))} = 2^{\Omega(g(M))}$ since $f(d) = \Omega(M)$ from $M = \Theta(f(d))$ and $f^{-1}(M) \geq d$ from $M \geq f(d)$. This completes the proof of the theorem.

6 Conclusions and Open Problems

It has been proved that an arbitrary convergent discrete-time binary network can be simulated by a symmetric continuous-time network with only a linear increase in the network size. The existence of a Lyapunov function for symmetric networks precludes the existence of undamping oscillations in the continuous-time system, but nevertheless our construction relies heavily on the finite sequence of clock pulses generated by the continuous-time counter subnetwork.

From the point of view of understanding analog computation in general this technique is somewhat unsatisfying, since the continuous-time computation is still basically discretized. It would be most interesting to develop some theoretical tools (e.g. complexity measures, reductions, universal computation) for "naturally" continuous-time computations that exclude the use of discretizing oscillations.

Another challenge for further research is to prove *upper bounds* on the power of continuous-time networks. Note that in the case of discrete-time analog-state networks a single fixed-size network with rational-number parameters can be computationally universal, i.e. able to simulate a universal Turing machine on arbitrary inputs (Siegel-

mann and Sontag, 1995). Can e.g. this strong universality result be generalized for continuous-time networks? Also, an exponential lower bound on the convergence time of symmetric continuous-time networks have been established: can a matching upper bound be proved, or the lower bound be increased?

Bibliography

Aarts, E. and Korst, J. (1989). *Simulated Annealing and Boltzmann Machines.* J. Wiley and Sons, Chichester.

Balcázar, J. L., Díaz, J., and Gabarró, J. (1995). *Structural Complexity*, volume I. Springer-Verlag, Berlin, 2nd edition.

Balcázar, J. L., Gavaldà, R., and Siegelmann, H. T. (1997). Computational power of neural networks: A characterization in terms of Kolmogorov complexity. *IEEE Transactions of Information Theory*, 43:1175–1183.

Casey, M. (1996). The dynamics of discrete-time computation, with application to recurrent neural networks and finite state machine extraction. *Neural Computation*, 8:1135–1178.

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

Farhat, N. H., Psaltis, D., Prata, A., and Paek, E. (1985). Optical implementation of the Hopfield model. *Applied Optics*, 24:1469–1475.

Goles, E. and Martínez, S. (1989). Exponential transient classes of symmetric neural networks for synchronous and sequential updating. *Complex Systems*, 3:589–597.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of the National Academy of Sciences*, volume 79, pages 2554–2558.

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. In *Proceedings of the National Academy of Sciences*, volume 81, pages 3088–3092.

Hopfield, J. J. and Tank, D. W. (1985). "Neural" computation of decisions in optimization problems. *Biological Cybernetics*, 52:141–152.

Horne, B. G. and Hush, D. R. (1996). Bounds on the complexity of recurrent neural network implementations of finite state machines. *Neural Networks*, 9:243–252.

Indyk, P. (1995). Optimal simulation of automata by neural nets. In *Proceedings of the 12th Annual Symposium on Theoretical Aspects of Computer Science (Munich, March)*, volume 900 of *LNCS*, pages 337–348. Springer-Verlag.

Liao, L.-Z. (1999). A recurrent neural network for N-stage optimal control problems. *Neural Processing Letters*, 10:195–200.

Maass, W. and Orponen, P. (1998). On the effect of analog noise in discrete-time analog computations. *Neural Computation*, 10:1071–1095.

Mańdziuk, J. (2000). Optimization with the Hopfield network based on correlated noises: Experimental approach. *Neurocomputing*, 30:301–321.

Orponen, P. (1996). The computational power of discrete Hopfield nets with hidden units. *Neural Computation*, 8:403–415.

Orponen, P. (1997a). The computational power of continuous time neural networks. In *Proceedings of the 24th SOFSEM Seminar on Current Trends in Theory and Practice of Informatics (Milovy, Czech Republic, November*, volume 1338 of *LNCS*, pages 86–103, Berlin. Springer-Verlag.

Orponen, P. (1997b). A survey of continuous-time computation theory. In Du, D.-Z. and Ko, K.-I., editors, *Advances in Algorithms, Languages, and Complexity*, pages 209–224. Kluwer Academic Publishers, Dordrecht.

Parberry, I. (1994). *Circuit Complexity and Neural Networks*. MIT Press, Cambridge.

Sheu, B., Lee, B., and Chang, C.-F. (1991). Hardware annealing for fast-retrieval of optimal solutions in Hopfield neural networks. In *Proceedings of the IEEE International Joint Conference on Neural Networks (Seattle, July)*, volume II, pages 327–332.

Siegelmann, H. T. (1999). *Neural Networks and Analog Computation: Beyond the Turing Limit*. Birkhäuser, Boston.

Siegelmann, H. T. and Sontag, E. D. (1994). Analog computation via neural networks. *Theoretical Computer Science*, 131:331–360.

Siegelmann, H. T. and Sontag, E. D. (1995). Computational power of neural networks. *Journal of Computer System Science*, 50:132–150.

Stoll, H. M. and Lee, L.-S. (1988). A continuous-time optical neural network. In *Proceedings of the IEEE International Conference on Neural Networks (San Diego, CA, July)*, volume II, pages 373–384.

Šíma, J. and Orponen, P. (2000). A continuous-time Hopfield net simulation of discrete neural networks. In *Proceedings of the NC'2000 Second International ICSC Symposium on Neural Computing (Berlin, May)*, pages 36–42, Wetaskiwin (Canada). ICSC Academic Press.

Šíma, J., Orponen, P., and Antti-Poika, T. (2000). On the computational complexity of binary and analog symmetric Hopfield nets. *Neural Computation*, 12. to appear.

Šíma, J. and Wiedermann, J. (1998). Theory of neuromata. *Journal of the ACM*, 45:155–178.

Wu, A. and Tam, P. K. S. (1999). A neural network methodology of quadratic optimization. *International Journal of Neural Systems*, 9:87–93.