On the Computational Complexity of Binary and Analog Symmetric Hopfield Nets

Jiří Šíma
Department of Theoretical Computer Science, Institute of Computer Science, Academy of Sciences of the Czech Republic, P.O. Box 5, 182 07 Prague 8, Czech Republic

Pekka Orponen
Teemu Antti-Poika
Department of Mathematics, University of Jyväskylä, FIN-40351 Jyväskylä, Finland

We investigate the computational properties of finite binary- and analog-state discrete-time symmetric Hopfield nets. For binary networks, we obtain a simulation of convergent asymmetric networks by symmetric networks with only a linear increase in network size and computation time. Then we analyze the convergence time of Hopfield nets in terms of the length of their bit representations. Here we construct an analog symmetric network whose convergence time exceeds the convergence time of any binary Hopfield net with the same representation length. Further, we prove that the \textit{MIN ENERGY} problem for analog Hopfield nets is NP-hard and provide a polynomial time approximation algorithm for this problem in the case of binary nets. Finally, we show that symmetric analog nets with an external clock are computationally Turing universal.

1 Introduction

In his 1982 paper (Hopfield, 1982), John Hopfield introduced an influential associative memory model that has since come to be known as the discrete-time Hopfield (or symmetric recurrent) network. The fundamental characteristic of this model is its well-constrained convergence behavior as compared to arbitrary asymmetric networks. Part of the appeal of Hopfield nets also stems from their connection to the much-studied Ising spin glass model in statistical physics (Barahona, 1982) and their natural hardware implementations using electrical networks (Hopfield, 1984) or optical computers (Farhat, Psaltis, Prata, & Paek, 1985). Hopfield originally proposed that this type of network could be used for removing noise from large binary patterns (Chiueh & Goodman, 1988; Gold, 1986; Park, Cho, & Park, 1999; Pawlicki, Lee, Hull, & Srihari, 1988), but they have also been applied to, for example, the fast approximate solution of combinatorial optimization problems (Hopfield & Tank, 1985; Aarts & Korst, 1989; Liao, 1999; Mańdziuk, 2000; Sheu, Lee, & Chang, 1991; Wu & Tam, 1999). Because of their simple

Neural Computation 12, 2965–2989 (2000) © 2000 Massachusetts Institute of Technology
structure, Hopfield nets are also often used as a reference model for investigating new analytical or computational ideas, in the same way as Ising spin systems are used in statistical physics or Turing machines in theoretical computer science. Although the basic Hopfield model per se is of limited practical applicability, it has inspired such other important neural network architectures as the bidirectional associative memory (BAM) (Kosko, 1988), and Boltzmann machines and their further stochastic variations (Ackley, Hinton, & Sejnowski, 1985; Haykin, 1999; Rojas, 1996). The accumulated knowledge concerning Hopfield nets has often been an essential prerequisite for understanding the capabilities of these other models. (For instance, the convergence behavior of the BAM model was analyzed in Kosko, 1988, along the lines first established for the Hopfield model in Hopfield, 1982.)

In this article, we investigate a number of issues in the computational analysis of Hopfield networks, complementing the existing literature in this area (Florén & Orponen, 1994; Parberry, 1994; Siegelmann, 1999; Siu, Roychowdhury, & Kailath, 1995; Wiedermann, 1994). After a brief review of the basic definitions in section 2, our first result in section 3 establishes a size- and time-optimal (up to constant factors) simulation of arbitrary discrete-time binary-state neural networks (with in general asymmetric interconnections) by symmetric binary-state Hopfield nets.

It is a fundamental property of symmetric Hopfield nets that their computations always lead to a stable network state (Hopfield, 1982) or possibly an oscillation between two different network states when the neurons are updated in parallel (Goles, Fogelman, & Pellegrin, 1985; Poljak & Súra, 1983). On the other hand, general asymmetric networks can have arbitrarily complicated limit behavior. Thus, the asymptotic dynamics of symmetric Hopfield nets are considerably more constrained than those of arbitrary networks. However, it was established in Orponen (1996) that when only convergent computations are considered, an arbitrary network of $n$ discrete-time binary neurons can be simulated by a Hopfield net with $O(n^2)$ symmetrically interconnected binary units. In section 3 we strengthen this result by showing that in fact, a convergent computation that takes $t$ steps on a general network of $n$ neurons can be simulated in $4t$ steps on a Hopfield network consisting of only $6n + 2$ symmetrically interconnected units.

This tight converse to Hopfield’s (1982) convergence theorem thus shows that in binary networks, it holds in a quite strong sense that “convergence $\equiv$ symmetry,” that is, not only do all symmetric networks converge, but also all convergent computations can be implemented efficiently in symmetric networks. The result also has some practical implications, because it guarantees that recurrent networks with arbitrary asymmetric interconnections can always be replaced, without much overhead in network size or computation time, by symmetric networks with their guaranteed convergence properties, and in some technologies more efficient implementations.

In section 4 we study the convergence time of both binary- and analog-state Hopfield networks in terms of the length of their bit representations.
This is to our knowledge the first analysis that takes into account the actual representation size of the networks. We obtain the curious result that there exist analog-state symmetric networks whose convergence time is greater than that of any binary-state symmetric network with the same representation size. The result shows that although analog-state networks with limited-precision states are not computationally more powerful than binary-state ones (Casey, 1996; Maass & Orponen, 1998), they may still be more efficient.

In section 5 we investigate the NP-complete MIN ENERGY or GROUND STATE problem of finding a network state with minimal energy for a given symmetric network. (For precise definitions, see section 2.) This problem forms the basis of all applications of Hopfield nets to combinatorial optimization (Hopfield & Tank, 1985; Aarts & Korst, 1989; Liao, 1999; Mańdziuk, 2000; Sheu, Lee, & Chang, 1991; Wu & Tam, 1999), and is also of importance in the Ising spin glass model (Barahona, 1982). In section 5.1 we derive a new polynomial time approximation algorithm for this problem in the case of binary-state networks. In section 5.2 we provide the first rigorous proof that the MIN ENERGY problem is NP-hard also for analog-state Hopfield nets, the model actually used in practical optimization applications.

Section 6 deals with the computational power of finite analog-state recurrent neural networks, a topic that has recently attracted considerable attention (Siegelmann, 1999). At a general level, it is known that finite asymmetric analog networks are computationally universal, i.e., equivalent to Turing machines (Siegelmann & Sontag, 1995), but that any amount of analog noise reduces their computational power to that of finite automata (Casey, 1996; Maass & Orponen, 1998).

For reasons discussed more fully in section 6, one cannot reasonably expect finite symmetric analog networks to be computationally universal. However, we show that universality can be achieved even in symmetric networks, provided that they are augmented with an external oscillator that produces an appropriately sequenced infinite stream of binary pulses. Thus, we obtain a full characterization of the computational power of finite analog-state discrete-time networks in the form of “Turing universality ≡ asymmetric network ≡ symmetric network + oscillator.” Moreover, we give in section 6 necessary and sufficient conditions that the external oscillator needs to satisfy in order to qualify for this equivalence.

An extended abstract of this article appeared in ˇS´ıma, Orponen, and Antti-Poika (1999).

2 Basic Notions

A finite discrete recurrent neural network consists of \( n \) simple computational units or neurons, indexed as \( 1, \ldots, n \), which 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
a connection within the architecture corresponds to a zero weight between the respective neurons. Special attention will be paid to Hopfield (symmetric) networks, whose architecture is an undirected graph with symmetric weights \( w(i, j) = w(j, i) \) for every \( i, j \).

We shall mostly consider the synchronous computational dynamics of the network, working in fully parallel mode, which determines the evolution of the network state \( \mathbf{y}(t) = (y_1(t), \ldots, y_n(t)) \in \{0, 1\}^n \) for all discrete time instants \( t = 0, 1, \ldots \) as follows. At the beginning of the 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 \( \xi_j(t) = \sum_{i=0}^{n} w_{ji} y_i(t) \) \( (j = 1, \ldots, n) \) is computed as the respective weighted sum of inputs, including an integer bias \( w_{0j} \), which can be viewed as the weight of the formal constant unit input \( y_0(t) = 1, t \geq 0 \). At the next instant \( t+1 \), an activation function \( \sigma \) is applied to \( \xi_j(t) \) for all neurons \( j = 1, \ldots, n \) in order to determine the new network state \( \mathbf{y}(t+1) \) as follows:

\[
y_j(t+1) = \sigma \left( \xi_j(t) \right) \quad j = 1, \ldots, n
\]

where a binary-state neural network employs the hard limiter (or threshold) activation function

\[
\sigma(\xi) = \begin{cases} 
1 & \text{for } \xi \geq 0 \\
0 & \text{for } \xi < 0.
\end{cases}
\]

Alternative computational dynamics are possible in Hopfield nets. For example, under sequential mode, only one neuron updates its state according to equation 2.1 at each time instant, while the remaining neurons do not change their outputs. Or we shall also deal with the finite analog-state discrete-time recurrent neural networks, which, instead of the threshold activation function of equation 2.2, employ some continuous sigmoid function, for example, the saturated-linear function

\[
\sigma(\xi) = \begin{cases} 
1 & \text{for } \xi > 1 \\
\xi & \text{for } 0 \leq \xi \leq 1 \\
0 & \text{for } \xi < 0.
\end{cases}
\]

Hence the states of analog neurons are real numbers within the interval \([0, 1]\), and similarly the weights (including biases) are allowed to be reals.

The fundamental property of symmetric nets is that on their state-space, a bounded Lyapunov, or energy function can be defined, whose values are properly decreasing along any nonconstant computation path (productive computation) of such a network. For example, consider a sequential computation of a binary symmetric Hopfield net with, for simplicity, zero biases.
$w_{ij} = 0$ and feedbacks $w_{ii} = 0$ (negative feedbacks are actually not allowed for sequential Hopfield nets to have a Lyapunov property), and without loss of generality (Parberry, 1994) also assume nonzero excitations $\xi_j^{(0)} \neq 0$, $j = 1, \ldots, n$. Then one can associate an energy with state $y^{(t)}$, $t \geq 0$, as follows:

$$E(y^{(t)}) = E(t) = -\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} w_{ji} y_j^{(t)} y_j^{(t)}.$$  \hspace{1cm} (2.4)

Hopfield (1982) showed that for this energy function, $E(t) \leq E(t - 1) - 1$ for every $t \geq 1$ of a productive computation. Moreover, the energy function is bounded, that is, $|E(t)| \leq W$, where

$$W = \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} |w_{ji}|$$  \hspace{1cm} (2.5)

is called the weight of the network. Hence, the computation must converge to a stable state within time $O(W)$. An analogous result can be shown for parallel updates, where a cycle of length at most two different states may appear (Goles, Fogelman, & Pellegrin, 1985; Poljak & Šura, 1983).

3 An Optimal Simulation of Asymmetric Networks

The computational power of symmetric Hopfield nets is properly less than that of asymmetric networks due to their different asymptotic behavior. Because of the Lyapunov property, Hopfield networks always converge to a stable state (or a cycle of length two for parallel updates), whereas asymmetric networks can have limit cycles of arbitrary length. However, Orponen (1996) showed that this is the only feature that cannot be reproduced, in the sense that any converging fully parallel computation by a network of $n$ discrete-time binary neurons, within general asymmetric interconnections, can be simulated by a Hopfield net of size $O(n^2)$. More precisely, given an asymmetric network to be simulated, there exists a subset of neurons in the respective Hopfield net whose states correspond to the original convergent asymmetric computation in the course of the simulation—possibly with some constant time overhead per each original update. The idea behind this simulation is that each asymmetric edge to be simulated is implemented by a small symmetric subnetwork that receives “energy support” from a symmetric clock subnetwork (a binary counter; Goles & Martínez, 1989) in order to propagate a signal in the correct direction.

In the context of infinite families of neural networks, which contain one network for each input length (a similar model is used in the study of boolean circuit complexity; Wegener, 1987), this simulation result implies that infinite sequences of discrete symmetric networks with a polynomially increasing number of binary neurons are computationally equivalent.
to (nonuniform) polynomially space-bounded Turing machines. Stated in standard complexity theoretic notation (Balcázar, Díaz, & Gabarró, 1995), such sequences of networks thus compute the complexity class PSPACE/poly, or P/poly when the weights in the networks are restricted to be polynomial in the input size (Orponen, 1996).

In the following theorem, the construction from Orponen (1996) is improved by reducing the number of neurons in the simulating symmetric network from quadratic to linear as compared to the size of the asymmetric network; this result is asymptotically optimal. The improvement is achieved by simulating the individual neurons (as opposed to edges, as in Orponen, 1996), whose states are updated by means of the clock technique. A similar idea was used for a corresponding continuous-time simulation in Šima and Orponen (2000). Also a quantitatively similar result has been achieved for sequential Hopfield nets with negative feedbacks in Goles and Matamala (1996). However, such networks lose the Lyapunov property, and thus they can simulate any in general nonconvergent asymmetric network relatively easily.

**Theorem 1.** Any fully parallel computation by a recurrent neural network of \(n\) binary neurons with asymmetric weights, converging within \(t^*\) discrete update steps, can be simulated by a Hopfield net with \(6n + 2\) neurons, converging within \(4t^*\) discrete-time steps.

**Proof.** Observe, first, that any converging computation by an asymmetric network of \(n\) binary neurons must terminate within \(t^* \leq 2^n\) steps. A basic technique used in our proof is the exploitation of an \((n + 1)\)-bit symmetric clock subnetwork (a binary counter), which, using \(3n + 1\) units being initially in the zero state, produces a sequence of \(2^n\) well-controlled oscillations before it converges. This sequence of clock pulses will be used to drive the underlying simulation of an asymmetric network in the remaining part of the Hopfield net.

The construction of the \((n + 1)\)-bit binary counter will be described by induction for \(n\). An example of a 3-bit counter network is presented in Figure 1, where the symmetric connections between neurons are labeled with corresponding weights and the biases are indicated by the edges drawn without an originating unit. In the sequel the symmetric weights in the Hopfield net will be denoted by \(w\), whereas \(w'\) denotes the original asymmetric weights.

The induction starts with the least significant counter unit \(c_0\). Its bias \(w(0, c_0) = (5U + 4)n + 1\) is also denoted by \(B\) in Figure 1 and the parameter \(U\) is defined as follows:

\[
U = \max_{j=1,...,n} \sum_{i=0}^{n} |w'_{ij}|. \tag{3.1}
\]

As it will be seen later, this large bias prevents the rest of the network from
affecting the counter computation. Neuron \( c_0 \) is initially \emph{passive} (its state is 0). However, at the next time instant, \( c_0 \) will \emph{fire} or be \emph{active} (its state is 1) according to equation 2.2 since its excitation is positive. Hence, \( c_0 \) simply implements counting from 0 to 1.

For the induction step, suppose that the counter has been constructed up to the first \( k \) \((0 < k < n + 1)\) counter bits \( c_0, \ldots, c_{k-1} \), and denote by \( V_k \) the set of all its \( n_k = 3(k - 1) + 1 \) neurons, including the auxiliary ones labeled \( a_\ell, b_\ell \) for \( \ell = 1, \ldots, k - 1 \). Then the counter unit \( c_k \) is connected to all \( n_k \) neurons \( v \in V_k \) via unit weights \( w(v, c_k) = 1 \), which, together with its bias \( w(0, c_k) = -n_k \), make \( c_k \) to fire if and only if all these units are active. This includes the first \( k \) active counter bits \( c_0, \ldots, c_{k-1} \) which means that counting from 0 to \( 2^k - 1 \) has been accomplished, and hence, the next counter bit \( c_k \) must fire. In addition, the unit \( c_k \) is connected to \( a_k \) which is further linked to \( b_k \), so that these auxiliary neurons are, one by one, activated after \( c_k \) fires. This is implemented by the following weights \( w(c_k, a_k) = W_k, w(a_k, b_k) = W_k - n_k \), and biases \( w(0, a_k) = -1, w(0, b_k) = n_k - W_k \), where \( W_k > 0 \) is a sufficiently large, positive parameter whose value will optimally be determined below so that neurons \( a_k, b_k \) are not directly influenced by a computation of units from \( V_k \) except via \( c_k \).

The neuron \( a_k \) resets all the units in \( V_k \) to their initial zero states, which is consistent with the correct counter computation when \( c_k \) fires. For this purpose, \( a_k \) is further connected to each \( v \in V_k \) via a sufficiently large negative weight \( w(a_k, v) < 0 \) such that \(-w(a_k, v) > 1 + \sum_{v \in V_k \cup \{0\}: w(a_k, v) > 0} w(u, v) = S_{kv} \) exceeds their mutual positive influence (including the weight \( w(c_k, v) = 1 \))

Figure 1: A 3-bit counter network.
and its possibly positive bias \( w(0, v) > 0 \). Thus, we can choose the integer weight \( w(a_k, v) = -S_k - 1 \) less only by 1 than the minus right side of the preceding inequality suggests in order to optimize the weight size. This also determines the minimal value of the large, positive weight parameter \( W_k = 1 - \sum_{v \in V_k} w(a_k, v) \), which makes the state of \( a_k \) (similarly for \( b_k \) below) independent on the outputs from \( v \in V_k \).

Finally, the unit \( b_k \) balances the negative influence of \( a_k \) on \( V_k \) so that the first \( k \) counter bits can again count from 0 to \( 2^k - 1 \) but now with \( c_j \) being active. This is achieved by the exact weight \( w(b_k, v) = -w(a_k, v) \) for each \( v \in V_k \) in which \(-w(a_k, v)\) eliminates the influence \( w'(a_k, v)\) of \( a_k \) on \( v \) and \(-1 \) compensates \( w(c_k, v) = 1 \). Again, neurons \( v \in V_k \) cannot reversely affect \( b_k \) since their maximal contribution \( \sum_{v \in V_k} w(v, b_k) = -n_k - \sum_{v \in V_k} w(a_k, v) = W_k - n_k - 1 \) to the excitation of \( b_k \) cannot beat its bias \( w(0, b_k) = n_k - W_k \). This completes the induction step. It follows from the preceding description that the sizes of integer weights in the counter network are as minimal as this construction allows.

Now the symmetric clock subnetwork— in particular, the counter unit \( c_0 \), which outputs the state sequence \( (0111)^2 \) during the clock operation—will be used to continue the underlying simulation of an asymmetric network. Note that the corresponding weights in the counter have been chosen sufficiently large that the clock is not influenced by the subnetwork it drives. In addition, a neuron \( c_0 \) is added, which computes the negation of the \( c_0 \) output. Then for each neuron \( j \) from the asymmetric network, three units \( p_j \), \( q_j \), \( r_j \) are introduced in the Hopfield net so that \( p_j \) represents the new (current) state \( y_j^{(t)} \) of \( j \) at time \( t \geq 1 \) while \( q_j \) stores the old state \( y_j^{(t-1)} \) of \( j \) from the preceding time instant \( t - 1 \), and \( r_j \) is an auxiliary neuron realizing the update of the old state. The corresponding symmetric subnetwork simulating one neuron \( j \) is depicted in Figure 2. The total number of units simulating the asymmetric network is \( 3n + 1 \) (including \( c_0 \)), which, together with the clock size \( 3n + 1 \), gives a total of \( 6n + 2 \) neurons in the Hopfield net.

At the beginning of the simulation, all the neurons in the Hopfield net are passive, except for those units \( q_j \) corresponding to the original initially active neurons \( j \), that is, \( y_j^{(0)} = 1 \). Then an asymmetric network update at time \( t \geq 1 \) is simulated by a cycle of four steps in the Hopfield net as follows.

In the first step, unit \( c_0 \) fires and remains active until its state is changed by the clock, since its large, positive bias makes it independent of all the \( n \) neurons \( p_j \). Also the unit \( c_0 \) fires because it computes the negation of \( c_0 \) that was initially passive. At the same time, each neuron \( p_j \) computes its new state \( y_j^{(t)} \) from the old states \( y_j^{(t-1)} \), which are stored in the corresponding units \( q_j \). Thus, each neuron \( p_j \) is connected with units \( q_j \) via the original weights \( w(q_j, p_j) = w'(i, j) \), and also its bias \( w(0, p_j) = w'(0, j) \) is preserved. So far, unit \( q_j \) keeps the old state \( y_j^{(t-1)} \) due to its feedback.
In the second step, the new state $y_{j}^{(t)}$ is copied from $p_j$ to $r_j$, and the active neuron $c_0$ makes each neuron $p_j$ passive by means of a large, negative weight, which exceeds the positive influence from units $q_i$ $(i = 1, \ldots, n)$ including its bias $w_j(0, p_j)$ according to equation 3.1. Similarly, the active neuron $c_0$ erases the old state $y_{j}^{(t-1)}$ from each neuron $q_j$ by making it passive with the help of a large, negative weight, which, together with the negative bias, exceeds its feedback and the positive influence from units $p_i$ $(i = 1, \ldots, n)$. Finally, neuron $c_0$ also becomes passive since $c_0$ was active.

In the third step, the current state $y_{j}^{(t)}$ is copied from $r_j$ to $q_j$ since all the remaining incident neurons $p_i$ and $c_0$ are and remain passive due to $c_0$ being active. Therefore unit $r_j$ also becomes passive.

In the fourth step, $c_0$ becomes passive, and the state $y_{j}^{(t)}$, being called old from now on, is stored in $q_j$. Thus, the Hopfield net finds itself at the starting condition of the time $t + 1$ asymmetric network simulation step, which proceeds in the same way. Altogether, the whole simulation is achieved within $4^n$ discrete-time steps.

4 Convergence Time Analysis

In this section, the convergence time in Hopfield networks—the number of discrete updates required before the network converges—will be analyzed. We shall consider only worst-case bounds (an average-case analysis can be found in Komlós and Paturi, 1988.) Since a network with $n$ binary neurons has $2^n$ different states, a trivial $2^n$ upper bound on the convergence time in symmetric networks of size $n$ holds. On the other hand, the symmetric clock
network (Goles & Martínez, 1989), which is used in the proof of theorem 1, provides an explicit example of a Hopfield net whose convergence time is exponential with respect to $n$. More precisely, this network yields a $\Omega(2^{n/3})$ lower bound on the convergence time of Hopfield nets, since a $(k + 1)$-bit binary counter can be implemented using $n = 3k + 1$ neurons.

However, these bounds do not take into account the size of the weights in the network. An upper bound of $O(W)$ follows from the characteristics of the energy function (see section 2), and this estimate can be made more accurate by using a slightly different energy function (Floréen, 1991). This yields a polynomial upper bound on the convergence time of Hopfield nets with polynomial weights. Similar arguments can be used for fully parallel updates.

In theorem 2, these results will be translated into convergence time bounds with respect to the length of bit representations of Hopfield nets. More precisely, for a binary-state symmetric network, which is described within $M$ bits, convergence-time lower and upper bounds $2^{\Omega(M/3)}$ and $2^{O(M/2)}$, respectively, will be shown. It is an open problem whether these bounds can be improved. This is an important issue since the convergence-time results for binary-state Hopfield nets can be compared with those for analog-state symmetric networks in which the precision of real weight parameters (i.e., the representation length) plays an important role. For example, in theorem 2, we shall also introduce an analog-state version of the symmetric clock network from theorem 1 whose computation terminates later than that of any other binary-state Hopfield net of the same representation size.

To the best of our knowledge, this provides the first known lower bound on the convergence time of analog-state Hopfield nets. A similar result can be achieved even for continuous-time symmetric networks (Šima & Orponen, 2000). Note also that in this approach, we express the convergence time with respect to the full descriptional complexity of the Hopfield net, not just the usual measure of the number of neurons, which captures its computational resources only partially. This suggests that analog Hopfield nets whose parameters are limited in the size may gain efficiency over binary ones.

**Theorem 2.** There exist both binary- and analog-state Hopfield nets with encoding size of $M$ bits that converge after $2^{\Omega(M/3)}$ and $2^{O(g(M))}$ updates, respectively, where $g(M)$ is an arbitrary continuous function such that $g(M) = \Omega(M^{2/3})$, $g(M) = o(M)$, and $M / g(M)$ is increasing. On the other hand, any computation of a symmetric binary-state network with a binary representation of $M$ bits terminates within $2^{O(M/2)}$ discrete computational steps.

**Proof.** For the underlying lower bounds, the clock network with parameter $B = 1$ from the proof of theorem 1 can again be exploited. It is sufficient to estimate its representation length. It can be shown by induction on $k$ that the
maximum integer weight in the \((k + 1)\)-bit counter with \(n = 3k + 1\) neurons is of order \(2^\Omega(n)\). This corresponds to \(O(n)\) bits per weight, which is repeated \(O(n^2)\) times, and thus yields at most \(M = O(n^3)\) bits in the representation. Hence, the convergence-time lower bound \(2^\Omega(n)\) of the binary-state clock network can be expressed as \(2^\Omega(M^{1/3})\).

In the analog-state case, the underlying weights of the clock with the parameter \(B = 1\) will be adjusted further, as follows. For every analog unit \(v\) in the counter network a feedback weight \(w(v, v) = 1 + \varepsilon\) is introduced, and also each bias is increased by \(\varepsilon\) except for neuron \(c_0\), whose bias is explicitly set to \(w(0, c_0) = \varepsilon\), where \(\varepsilon > 0\) is a small (e.g., \(\varepsilon < 0.1\)) optional parameter controlling the convergence rate of the analog clock network.

Now, starting the counter computation at the zero initial state, the state of unit \(c_0\) gradually grows toward saturation at value 1 because of its bias \(w(0, c_0) = \varepsilon\) and feedback \(w(c_0, c_0) = 1 + \varepsilon\). It can be verified that at the least discrete-time instant greater than \(\log(2 - \varepsilon)/\log(1 + \varepsilon)\), its output is greater than 1, which makes the excitation of the next counter unit \(c_1\) positive due to its bias \(w(0, c_1) = -1 + \varepsilon\). Now, \(c_1\) starts to grow because of its feedback \(w(c_1, c_1) = 1 + \varepsilon\), which shortly completes the saturation of \(c_0\) at 1. This trick of gradual state transition from 0 to 1 is applied repeatedly throughout the analog clock computation by using the introduced feedbacks 1 + \(\varepsilon\).

It follows that the respective transition for \(c_0\) takes at least \(\Omega(1/\log(1 + \varepsilon))\) discrete-time steps, which, together with the fact that the unit \(c_0\) fires \(2^k\) times before \((k + 1)\)-bit clock converges, provides the lower bound \(\Omega(2^b/\log(1 + \varepsilon)) \geq \Omega(2^{n^2/3}/\varepsilon)\) on the convergence time of analog symmetric networks with \(n\) neurons. We shall express this bound in terms of the size \(M\) in bits of the network representation. The length of the integer part of the weight parameter representation excluding fractions \(\varepsilon\) has already been upper-bounded by \(O(n^2)\) bits in the above-considered binary-state case. In addition, the biases and feedbacks of the \(n\) units include the fraction \(\varepsilon\), and taking this into account requires \(\Theta(n \log(1/\varepsilon))\) additional bits, say, at least \(\kappa n \log(1/\varepsilon)\) bits for some constant \(\kappa > 0\). By choosing an appropriate \(\varepsilon = 2^{-f(n)/(\kappa n)}\) in which \(f\) is a continuous increasing function whose inverse is defined as \(f^{-1}(\mu) = \mu/g(\mu)\), where \(g\) is an arbitrary function such that \(g(\mu) = \Omega(\mu^{2/3})\) (implying \(f(n) = \Omega(n^3)\)) and \(g(\mu) = o(\mu)\), we get \(M = \Theta(f(n))\), especially \(M \geq f(n)\) from \(M \geq \kappa n \log(1/\varepsilon)\). Finally, the convergence time \(\Omega(2^{n^2/3}/\varepsilon)\) can be translated to \(\Omega(2^{f(n)/(\kappa n)} + n/3) = 2^{\Omega(f(n)/n)}\), which can be rewritten as \(2^{\Omega(M/f^{-1}(M))} = 2^{\Omega(g(M))}\) since \(f(n) = \Omega(M)\) from \(M = \Theta(f(n))\) and \(f^{-1}(M) \geq n\) from \(M \geq f(n)\).

On the other hand, for the upper bound, consider a binary-state Hopfield network with an \(M\)-bit representation that converges after \(T(M)\) updates. A major part of this \(M\)-bit representation consists of \(m\) binary encodings of weights \(w_1, \ldots, w_m\) of the corresponding lengths \(M_1, \ldots, M_m\), where \(\sum_{r=1}^{m} M_r = \Theta(M)\). Clearly there must be at least \(T(M)\) different energy
levels corresponding to the states visited during the computation. Thus, the underlying weights must produce at least \( S \geq T(M) \) different sums \( \sum_{r \in A} w_r \) for \( A \subseteq \{1, \ldots, m\} \) where \( w_r \) for \( r \in A \) agrees with \( w_j \) for \( y_j = y_i = 1 \) in equation 2.4. So it is sufficient to upper-bound the number of different sums over \( m \) weights whose binary representations form a \( \Theta(M) \)-bit string altogether.

Observe that all the \( S = 2^m \) sums over subsets of \( m \) weights 1, 2, 4, 8, \ldots, \( 2^{m-1} \) are different. At the same time these weights altogether have the least length \( m(m+1)/2 \) of binary representation over all the weights that generate \( 2^m \) different sums. Since the representation length is at most \( M \), we get \( m = O(M^{1/2}) \) upper bound on the number of weights, which yields \( T(M) \leq \frac{M}{2} \).

5 The Minimum Energy Problem

In this section we investigate the important MIN ENERGY or GROUND STATE problem of finding a network state with minimal energy for a given symmetric network. This problem is of special interest because many hard combinatorial optimization problems have been heuristically solved by minimizing the energy in Hopfield nets (Hopfield & Tank, 1985; Aarts & Korst, 1989; Liao, 1999; Ma´ndziuk, 2000; Sheu, Lee, & Chang, 1991; Wu & Tam, 1999). This issue is also important for the Ising spin glass model in statistical physics (Barahona, 1982).

It follows from these heuristic reductions of optimization problems to MIN ENERGY that the problem is computationally hard. In particular, for binary networks, the decision version of MIN ENERGY—that is, the question whether there exists a network state having an energy less than a prescribed value, is NP-complete (Barahona, 1982). On the other hand, it is known that MIN ENERGY has polynomial time algorithms for binary Hopfield nets whose architectures are planar lattices (Bieche, Maynard, Rammal, & Uhry, 1980) or planar graphs (Barahona, 1982).

We address here two aspects of the MIN ENERGY problem. In section 5.1 we provide a polynomial time approximation algorithm for the problem in the case of binary nets. In section 5.2 we analyze the complexity of the MIN ENERGY problem for analog Hopfield nets, which is the model whose continuous-time version is usually used for practical optimizations. Surprisingly, to the best of our knowledge this problem has been unresolved so far. This is possibly because there is a slight technical complication to the issue, related to the fact that unlike binary networks, analog networks can also converge to an interior point of their state-space. To prove the NP-hardness result, we need to ensure that the minimum energy values of the constructed networks are actually reached close to extremum points of their respective state-spaces. Our proof applies to all continuous nondecreasing sigmoidal neuron activation functions of a practical interest.
5.1 Polynomial Time Approximation for Binary Nets. Recall that in definition 2.4 of the energy function for binary nets, we assumed, for simplicity, that \( w_{ij} = 0 \) and \( w_{ii} = 0 \) for \( j = 1, \ldots, n \). In addition, without loss of generality (Parberry, 1994), we shall work throughout this section with bipolar states \([-1, 1]\) of neurons instead of binary ones \([0, 1]\) from equation 2.2, where 0 is now replaced by \(-1\).

Perhaps the most direct and frequently used reduction to MIN ENERGY is from the MAX CUT problem (see, e.g., Bertoni & Campadelli, 1994). In MAX CUT we are given an undirected graph \( G = (V, A) \) with an integer edge cost function \( c : A \rightarrow \mathbb{Z} \), and we want to find a cut \( V_1 \subseteq V \) that maximizes the cut size,

\[
c(V_1) = \sum_{\{i, j\} \in A, i \notin V_1, j \in V_1} c(i, j) - \sum_{\{i, j\} \in A, c(i, j) < 0} c(i, j). \tag{5.1}
\]

Note that the standard MAX CUT problem is here generalized by allowing also the negative edge weights that are needed for the opposite reduction from MIN ENERGY to MAX CUT. Recently, a new, randomized approximation algorithm with a high-performance guarantee \( \alpha = 0.87856 \) for this MAX CUT formulation has been proposed (Goemans & Williamson, 1995) and later derandomized (Mahajan & Ramesh, 1995), a fact we shall exploit for approximating the MIN ENERGY problem. We shall observe that MIN ENERGY can be approximated in a polynomial time within absolute error less than \( 0.243W \) where \( W \) is the network weight (see equation 2.5).

Theorem 3. The MIN ENERGY problem for binary Hopfield nets can be approximated in polynomial time within absolute error less than \( 0.243W \) where \( W \) is the network weight (see equation 2.5).

Proof. We first recall the well-known simple reduction between MIN ENERGY and MAX CUT problems. For a binary Hopfield network with architecture \( G \) and weights \( w(i, j) \), we can easily define the corresponding instance \( G = (V, A) \), \( c \) of MAX CUT with edge costs \( c(i, j) = -w(i, j) \) for \( \{i, j\} \in A \). We shall show that any cut \( V_1 \subseteq V \) of \( G \) corresponds to a Hopfield net state \( y \in \{-1, 1\}^n \) where \( y_i = 1 \) if \( i \in V_1 \) and \( y_i = -1 \) for \( i \in V \setminus V_1 \), so that the respective cut size \( c(V_1) \) is related to the underlying energy \( E(y) \). Thus, the energy function of equation 2.4 can be expressed in terms of the
cut size of equation 5.1 as follows:

\[
E(y) = -\frac{1}{2} \sum_{j \in V} w(i, j)y_iy_j \\
= -\frac{1}{2} \sum_{y_i = y_j} w(i, j) + \frac{1}{2} \sum_{y_i \neq y_j} w(i, j) = -\frac{1}{2} \sum_{w(i, j) < 0} w(i, j) - \frac{1}{2} \sum_{w(i, j) > 0} w(i, j) \\
+ \sum_{y_i \neq y_j} w(i, j) + \frac{1}{2} \sum_{w(i, j) < 0} w(i, j) - \frac{1}{2} \sum_{w(i, j) > 0} w(i, j) \\
= -\frac{1}{2} \sum_{w(i, j) < 0} w(i, j) + \frac{1}{2} \sum_{w(i, j) > 0} w(i, j) - \sum_{y_i \neq y_j} w(i, j) + \sum_{y_i \neq y_j} w(i, j) \\
= W + 2 \sum_{\{i, j\} \in A, c(i, j) < 0} c(i, j) - 2 \sum_{\{i, j\} \in A, i \in V_1, j \notin V_1} c(i, j) \\
= W - 2c(V_1). \tag{5.2}
\]

It follows from equation 5.2 that the minimum energy state corresponds to the maximum cut.

Now, the approximate polynomial time algorithm from Goemans and Williamson (1995) can be employed to solve an instance \( G = (V, A), c \) of the MAX CUT problem, which provides a cut \( V_1 \) whose size \( c(V_1) \geq \alpha c^* \) is guaranteed to be at least \( \alpha = 0.87856 \) times the maximum cut size \( c^* \). Let cut \( V_1 \) correspond to the Hopfield network state \( y \), which implies \( c(V_1) = 1/2(W - E(y)) \) from equation 5.2. Hence, we get a guarantee \( W - E(y) \geq \alpha(W - E^*) \) where \( E^* \) is the minimum energy corresponding to the maximum cut \( c^* \), which leads to \( E(y) - E^* \leq (1 - \alpha)(W - E^*) \). Since \( |E^*| \leq W \), we obtain the desired guarantee for the absolute error \( E(y) - E^* \leq (1 - \alpha)2W < 0.243W \).

5.2 Hardness Result for Analog Nets. In this section we consider the MIN ENERGY(\( \sigma \)) problem for analog Hopfield nets with general activation function \( \sigma \). The energy function of equation 2.4 can be generalized for analog networks with nonnegative feedbacks \( w_{ij} \geq 0 \) (j = 1, \ldots, n) as follows (Koiran, 1994):

\[
E_{\sigma}(y) = -\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} w_{ij}y_iy_j - \sum_{j=1}^{n} w_jy_j + \sum_{j=1}^{n} \int_{0}^{y_j} \sigma^{-1}(y)dy, \tag{5.3}
\]

where the activation function \( \sigma \) is continuous and strictly increasing on an interval \([\alpha, \beta]\) (\( \alpha < \beta \), with possibly \( \alpha = -\infty \) or \( \beta = +\infty \)), and constant
outside, that is,
\[
\sigma(\xi) = \begin{cases} 
    a & \text{for } \xi \leq \alpha \\
    b & \text{for } \xi \geq \beta.
\end{cases}
\] (5.4)

For $\alpha = -\infty$ or $\beta = +\infty$, we require that $\lim_{\xi \to -\infty} \sigma(\xi) = a$ or $\lim_{\xi \to +\infty} \sigma(\xi) = b$, respectively. The inverse function $\sigma^{-1}$ in equation 5.3 may not be defined on the whole interval $[0, y]$; in that case we assign it to zero outside of $(a, b)$. In the sequel, we shall also assume that $\sigma$ is differentiable on $(\alpha, \beta)$ (thus, we know that $\sigma'(\xi) > 0$ for $\xi \in (\alpha, \beta)$) and that the respective integrals in equation 5.3 are bounded, that is,
\[
\sup_{y \in [a, b]} \left| \int_0^y \sigma^{-1}(y) dy \right| < I_\sigma,
\] (5.5)

where $0 < I_\sigma < +\infty$. These conditions are satisfied by all the commonly used continuous activation functions (e.g., the hyperbolic tangent and the saturated-linear map).

The decision version of the MIN ENERGY($\sigma$) problem for analog Hopfield nets is the question whether a given analog network possesses a state $y \in [a, b]^n$ for which the energy $E_\sigma(y)$ has value less than a given constant $K$. Similarly as in the proof of theorem 3, we shall show this problem to be NP-hard by reduction from the SIMPLE MAX CUT problem, which is known to be NP-complete (Garey, Johnson, & Stockmeyer, 1976). In SIMPLE MAX CUT, we are given an undirected graph $G = (V, A)$ and a positive integer $k$, and we want to decide whether there exists a cut $V_1 \subseteq V$ whose size $c(V_1) = |\{e \in A : e \cap V_1 \neq \emptyset\}|$ (corresponding to constant unit cost in MAX CUT) is at least $k$. The idea of the proof is to exploit the reduction from the bipolar case by forcing the analog energy of equation 5.3 to achieve its minimum at a state close to one of the extremal points $(a, b)^n$ of the state-space. For this purpose we prove the following lemma concerning a saturation tendency of analog symmetric networks with large positive feedbacks.

**Lemma 1.** Let $(\eta, \vartheta)$ be an interval in which $\delta = \inf_{\xi \in (\eta, \vartheta)} \sigma'(\xi) > 0$ (i.e., $\sigma$ is certainly not saturated on $(\eta, \vartheta)$), and let $w_{ij} > 1/\delta$ for every $j = 1, \ldots, n$. If $y^\ast = (y_1^\ast, \ldots, y_n^\ast) \in [a, b]^n$ is a local minimum of the energy function of equation 5.3, then $y_j^\ast \notin (\sigma(\eta), \sigma(\vartheta))$ for every $j = 1, \ldots, n$.

**Proof.** Suppose that $y^\ast \in [a, b]^n$ is a local minimum of the energy function of equation 5.3. Clearly, $y_j^\ast \notin (\sigma(\eta), \sigma(\vartheta))$ for $y_j^\ast \in (a, b)$. For $y_j^\ast \in (a, b)$ consider function $E_j(y) = E_\sigma(y_1^\ast, \ldots, y_{j-1}^\ast, y, y_{j+1}^\ast, \ldots, y_n^\ast)$ of one variable $y$. Thus, $E_j'(y_j^\ast) = 0$ and
\[
E_j''(y_j^\ast) = \frac{\partial^2 E_\sigma}{\partial y_j^2}(y^\ast) = -w_{ij} + \frac{1}{\sigma'(y_j^\ast)} \geq 0.
\] (5.6)
since $y^*$ is a local minimum of $E_\sigma$. It follows from equation 5.6 that $w_{ij} \leq 1/\sigma'(y^*_j)$, and hence $y^*_j \notin (\sigma(n), \sigma(\varnothing))$.

Now we shall prove the hardness result for analog MIN ENERGY($\sigma$) problem, which seems to be the first formal verification of this fact, although (continuous-time) analog Hopfield nets have often been exploited to solve hard optimization problems heuristically (Hopfield & Tank, 1985; Aarts & Korst, 1989; Liao, 1999; Mañdziuk, 2000; Sheu, Lee, & Chang, 1991; Wu & Tam, 1999).

**Theorem 4.** The MIN ENERGY($\sigma$) problem for analog Hopfield nets is NP-hard.

**Proof.** We reduce the SIMPLE MAX CUT problem to MIN ENERGY($\sigma$). Given a SIMPLE MAX CUT instance $G = (V, A)$, $k$ with $n = |V|$ vertices and $m = |A|$ edges, a corresponding MIN ENERGY($\sigma$) instance, that is, an analog Hopfield net $N$ with activation function $\sigma$ and a prescribed energy level $K$ is constructed in polynomial time. The architecture of $N$ is $G$, except that each neuron in $N$ has an additional feedback connection. The weights and biases of $N$ are defined as follows:

\[
\begin{align*}
  w_{ji} &= \begin{cases} 
    \frac{-8c}{(b-a)^2} & \text{for } [i, j] \in A \\
    \frac{u}{n} & \text{for } i = j 
  \end{cases} \\
  w_{j0} &= \frac{4c(a+b)\operatorname{deg}_G(j)}{(b-a)^2},
\end{align*}
\]

where $\operatorname{deg}_G(j) = |\{i \in V : [i, j] \in A\}|$ is the degree of vertex $j$ in $G$,

\[
C = \left| n \left( I_\sigma - \frac{u}{2} \max \left( |a|^2, |b|^2 \right) \right) \right|, 
\]

and $u$ is chosen so that

\[
u > \frac{1}{\delta} \quad \text{where} \quad \delta = \inf_{\xi \in (\sigma^{-1}(a+\varepsilon), \sigma^{-1}(b-\varepsilon))} \sigma'(<) > 0
\]

and $\varepsilon > 0$ satisfies

\[
\varepsilon < \frac{b - a}{8m}.
\]

Finally, the energy level is prescribed as

\[
K = 2C \left( -2k + 1 - \frac{4abm}{(b-a)^2} \right),
\]
In the proof of theorem 3, the reduction from MIN ENERGY to MAX CUT is actually the one-to-one correspondence and the respective inverse reduction from MAX CUT to MIN ENERGY, for example, restricted even to SIMPLE MAX CUT instances, provides an NP-completeness proof for bipolar Hopfield nets. As we have mentioned, we exploit this approach by reducing the underlying analog minimum energy problem to the bipolar case. For this purpose, the following linear transformation of the analog neuron state $y_j \in [a, b]$ to $y'_j \in [-1, 1]$, which preserves the network weights except for biases (Parberry, 1994), is employed:

$$y'_j = \frac{2(y_j - a)}{b - a} - 1.$$  \hspace{1cm} (5.13)

With this transformation the energy function of equation 5.3 can be rewritten as

$$E_\sigma(y) = 2C \left( -\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1, i \neq j}^{n} w_{ji} y'_j y'_i - \left( \frac{a+b}{b-a} \right)^2 m \right) + \sum_{j=1}^{n} \int_{0}^{y'_j} \sigma^{-1}(y) dy - \frac{1}{2} \sum_{j=1}^{n} w_{jj} y'_j^2,$$  \hspace{1cm} (5.14)

where the introduced weights

$$w_{ji} = \begin{cases} -1 & \text{for } [i, j] \in A \\ 0 & \text{otherwise,} \end{cases}$$  \hspace{1cm} (5.15)

coincide with those from the bipolar case.

Furthermore, consider a local minimum $y^* = (y^*_1, \ldots, y^*_n) \in [a, b]^n$ of energy $E_\sigma$. It follows from lemma 1, whose assumptions are satisfied by equation 5.10, that $y^*_j \in [a, a+\varepsilon] \cup [b - \varepsilon, b]$ for every $j = 1, \ldots, n$. Hence, $y^*_j \in [-1, -1 + 2\varepsilon/(b - a)] \cup [1 - 2\varepsilon/(b - a), 1]$, which implies

$$\left| y^*_j y^*_i \right| > 1 - \frac{4\varepsilon}{b - a}.$$  \hspace{1cm} (5.16)

We introduce bipolar states $\tilde{y}^*_j \in \{-1, 1\}$ into equation 5.14 by rounding the corresponding analog ones, that is, $\tilde{y}^*_j = -1$ for $y^*_j \in [-1, -1 + 2\varepsilon/(b - a)]$ and $\tilde{y}^*_j = 1$ for $y^*_j \in [1 - 2\varepsilon/(b - a), 1] (j = 1, \ldots, n)$. By using equations 5.16, 5.15, and 5.11, the rounding error for the underlying term in equation 5.14 can be estimated for local minimum $y^*$ as follows:

$$\left| -\frac{1}{2} \sum_{j=1}^{n} \sum_{i=1, i \neq j}^{n} w_{ji} \left( \tilde{y}^*_j \tilde{y}^*_i - y^*_j y^*_i \right) \right| < \frac{4\varepsilon m}{b - a} < \frac{1}{2}. \hspace{1cm} (5.17)$$
Also the absolute value of the second row of equation 5.14 can be upper-bounded from equations 5.5 and 5.9:

\[ \left| \sum_{j=1}^{n} \int_{0}^{y_j} \sigma^{-1}(y)dy - \frac{1}{2} \sum_{j=1}^{n} \omega_{j}y_{j}^{2} \right| < C. \quad (5.18) \]

Moreover, the result of equation 5.2 for the bipolar case can here be applied in which the network weight \( W \) equals \( m \) now and \( V_{1}^{*} \) is the cut corresponding to bipolar network state \( \tilde{y}^{*} = (\tilde{y}_{1}^{*}, \ldots, \tilde{y}_{n}^{*}) \in \{-1,1\}^{n} \). Thus, it follows from equations 5.17 and 5.18 that the energy function value given by equation 5.14, at local minimum \( y^{*} \) belongs to the following interval:

\[ 2C \left( m - 2c(V_{1}^{*}) - \frac{1}{2} \left( \frac{a + b}{b - a} \right)^{2} m \right) - C < E_{\sigma}(y^{*}) < 2C \left( m - 2c(V_{1}^{*}) + \frac{1}{2} \left( \frac{a + b}{b - a} \right)^{2} m \right) + C, \quad (5.19) \]

which can be rewritten as

\[ 2C \left( -2c(V_{1}^{*}) - 1 - \frac{4abm}{(b-a)^{2}} \right) \]

\[ < E_{\sigma}(y^{*}) < 2C \left( -2c(V_{1}^{*}) + 1 - \frac{4abm}{(b-a)^{2}} \right), \quad (5.20) \]

According to equation 5.20, the analog energy value \( E_{\sigma}(y^{*}) \) at local minimum \( y^{*} \) determines uniquely the integer cut size \( c(V_{1}^{*}) \) of cut \( V_{1}^{*} \) associated with \( \tilde{y}^{*} \).

Now, the correctness of the reduction from SIMILE MAX CUT to MIN ENERGY(\( \sigma \)) can easily be verified. Suppose there exists a cut \( V_{1} \) of \( G \) whose size is at least \( k \). Then the energy (see equation 5.14) of the corresponding analog state \( y \in [a, b]^{n} \) of \( N \), which is defined as \( y_{j} = b \) for \( j \in V_{1} \) and \( y_{j} = a \) for \( j \in V \setminus V_{1} \), can be upper-bounded by using equations 5.2 and 5.18 as follows:

\[ E_{\sigma}(y) < 2C \left( m - 2k - \left( \frac{a + b}{b - a} \right)^{2} m \right) + C < K. \quad (5.21) \]

On the other hand, assume that an analog state \( y \in [a, b]^{n} \) of \( N \) with energy \( E_{\sigma}(y) < K \) exists. Then there is also a local minimum \( y^{*} \in [a, b]^{n} \) of \( E_{\sigma} \) with energy \( E_{\sigma}(y^{*}) \leq E_{\sigma}(y) < K \). It follows from equations 5.12 and 5.20 that \( c(V_{1}^{*}) \leq k \).
In this section we deal with the computational power of finite analog-state discrete-time recurrent neural networks with the saturated-linear activation function of equation 2.3. For asymmetric analog networks, the computational power is known to increase with the Kolmogorov complexity of their real weights (Balcázar, Gavaldà, & Siegelmann, 1997). With integer weights such networks are equivalent to finite automata (Horne & Hush, 1996; Indyk, 1995; Šíma & Wiedermann, 1998), while with rational weights, arbitrary Turing machines can be simulated (Indyk, 1995; Siegelmann & Sontag, 1995). With arbitrary real weights, such networks can even have “super-Turing” computational capabilities, so that, for example, polynomial time computations correspond to the complexity class P/poly and all languages can be recognized within exponential time (Siegelmann & Sontag, 1994). On the other hand, any amount of analog noise reduces the computational power of this model to that of finite automata (Casey, 1996; Maass & Orponen, 1998).

For finite symmetric networks, only the computational power of binary-state Hopfield nets has been fully characterized. They recognize the so-called Hopfield languages (Šíma, 1995), which form a proper subclass of regular languages; hence such networks are less powerful than finite automata. Hopfield languages can also be faithfully recognized by analog symmetric neural networks (Maass & Orponen, 1988; Šíma, 1997), and this provides a lower bound on the computational power of such networks. A natural question then concerns improvements of this lower bound. Could this model too be Turing universal, that is, can a Turing machine simulation be achieved with symmetric networks and rational weights similarly as in the asymmetric case (Indyk, 1995; Siegelmann & Sontag, 1995)?

The main obstacle here is that under fully parallel updates, any analog Hopfield net with rational weights converges to a limit cycle of length at most two (Koiran, 1994). Thus, the only possibility of simulating Turing machines would be to exploit finer and finer distinctions among a sequence of rational network states converging to a limit cycle. Such a simulation seems to be tricky at best, if possible at all.

A more reasonable approach is to augment the network with an external clock that produces an infinite sequence of binary pulses, thus providing it with an “energy source” that can be used, for example, for simulating an asymmetric analog network similarly as in theorem 1. Indeed, we now prove that the computational power of analog Hopfield nets with an external clock is the same as that of asymmetric analog networks. Especially for rational weights, this implies that such networks are Turing universal. The following theorem also fully characterizes those infinite binary sequences by the external clock that prevent the Hopfield network from converging. In this way, we obtain a complete theoretical characterization of the symmetry of weights in analog Hopfield nets in the sense that the computational
power of analog asymmetric networks equals that of symmetric ones plus oscillator of a certain type.

**Theorem 5.** Let $N_0$ be an analog-state recurrent neural network with real asymmetric weights and $n$ neurons working in fully parallel mode. Then there exists an analog Hopfield net $N$ with $3n/8$ units whose real weights have the same maximum Kolmogorov complexity as the weights in $N_0$ and which simulates $N_0$, provided it receives as an additional external input any infinite binary sequence that contains infinitely many substrings of the form $bxb \in \{0, 1\}^3$ where $b \neq \bar{b}$. Moreover, the external input must have this property to prevent $N$ from converging.

**Proof.** First observe that an infinite binary sequence produced by the external input (clock) $c$ that does not satisfy the assumption of the theorem must be of the form $u(b_1b_2)^*$, where $b_1, b_2 \in \{0, 1\}$, and $u \in \{0, 1\}^*$ is a prefix that clearly cannot prevent the network from converging to a limit cycle of length two. On the other hand, we shall prove that if the sequence meets the respective condition, then it must contain infinitely many substrings of the form $1x0$ ($x \in \{0, 1\}$) since these strings necessarily accompany infinitely many substrings $0x1$. Thus, consider two subsequent occurrences of $0x1$. For $x = 1$, after $011$ possibly followed by several $1$s, at least one $0$ must appear due to the next $0x1$ which gives $110$ as required. For $x = 0$, the string $001$ is followed by either a $1$, which means the previous case with $011$ applies, or a $0$, possibly succeeded by several occurrences of $10$, which is followed by either a desired $0$ or $11$, which again leads to $011$.

The symmetric simulation of an asymmetric network $N'$ consisting of $n$ analog neurons is very similar to the binary one (see theorem 1). Thus, each neuron $j$ in $N'$ is simulated by three units $p_j, q_j, r_j$, which are controlled by three neurons $f, g, h$ (corresponding to $c_0, c_1$ in the binary case) whose binary states are generated by a small, symmetric subnetwork of five auxiliary units $a, d, e, r, s$, transforming the binary external input signal from $c$ to a well-controlled binary sequence. This gives the desired $3n/8 + 8$ units of the simulating analog Hopfield net $N$. The situation is depicted in Figure 3, including the definition of the symmetric weights, where $U$ is introduced in equation 3.1 and $B = (4W + 1)n + 7$ in this case.

Every binary external input bit is copied from clock $c$ to $r$ and further to $s$, and therefore the states of these neurons $s, r, c$ store the last three bits of the input sequence, respectively. Neuron $a$ detects the underlying strings of the form $1x0$ ($x \in \{0, 1\}$) at the input; that is, it is active iff $s$ is active and $c$ is passive. It may happen that two such strings follow each other immediately (also notice that it is impossible for such a string to appear again at the next step but one). This case is indicated by the activity of both neurons $a, d$, where $d$ copies the state from $a$. Thus, unit $e$ copies the state 1 from $a$ iff $d$ is passive. Hence, the neuron $e$ produces a “noiseless” sequence of zeros containing infinitely many substrings 100, each being exploited for
the simulation of one computational step of \( N' \) similarly as in the proof of theorem 1.

Namely, \( q_j \) stores the old analog state \( y_j^{(t-1)} \) by its unit feedback according to equation 2.3 and also \( h \) remains initially active preserving the stability of \( p_j, q_j, r_j \). After \( e \) fires, the unit \( h \) becomes passive and \( f \) is active, which controls the computation of a new analog state \( y_j^{(0)} \) by \( p_j \) via the original weights \( w(q_i, p_j) = w'(i, j) \) and the bias \( w(f, p_j) = w'(0, j) \). Then the control signal is further copied from \( f \) to \( g \), which enables \( r_j \) to receive the state \( y_j^{(0)} \). Finally, \( h \) becomes active and the state of \( q_j \) is updated by \( y_j^{(0)} \), and this is kept until \( e \) fires again to proceed to the next simulation step.

7 Conclusions

With the results presented in this article, we now seem to have developed quite a good understanding of the computational properties of discrete-time symmetric Hopfield nets. It is somewhat surprising that these networks turn out to be computationally essentially as powerful as general asymmetric networks, despite their Lyapunov-function constrained dynamics. As we have seen, in the binary-state case, symmetric networks can simulate asymmetric ones with only a linear increase in the network size (see section 3), and in the analog-state case, also finite symmetric networks are Turing universal, provided they are supplied with a simple external clock to prevent them from converging (see section 6). In addition, we have verified that the \textit{MIN ENERGY} problem remains NP-hard also for analog networks (see section 5), a fact that is not immediately obvious because of the possi-
ability of analog networks to converge to states in the interior of the state-space, complicating the combinatorial coding necessary for NP-hardness proofs.

It is also interesting to note that analog networks can in some cases be more efficient with respect to their encoding size than binary ones (see section 4). This result suggests that analog models of computation may be worth investigating more for their efficiency gains than for their (theoretical) capability for arbitrary-precision real number computation. Very little is currently known about the computational complexity aspects of analog computation, and the area clearly merits more intensive study.

Acknowledgments

J. Š. was supported by GA AS CR Grant No. B2030007, and T. A.-P. was supported by Academy of Finland Grant No. 37115/96.

References


Received September 20, 1999; accepted February 28, 2000.