Zeitschrift: Helvetica Physica Acta
Band: 62 (1989)
Heft: 5

Artikel: Introduction to cellular automata computing, neutral network computing and transputer based special purpose computers
Autor: Würtz, Diethelm / Hartung, Georg
DOI: https://doi.org/10.5169/seals-116044

Helvetica Physica Acta
Vol. 62 (1989)

0018-0238/89/050461-28$1.50+0.20/0

INTRODUCTION TO CELLULAR AUTOMATA COMPUTING, NEURAL NETWORK COMPUTING AND TRANSPUTER BASED SPECIAL PURPOSE COMPUTERS

Diethelm Würtz[1] and Georg Hartung[2]

[1]Institut für Theoretische Physik, Ruprecht-Karls Universität Heidelberg
D-6900 Heidelberg, Philsophenweg 19, FRG

[2]ABB Asea Brown Boveri, Corporate Research Heidelberg
D-6900 Heidelberg, Eppelheimerstrasse 82, FRG

Abstract: *We give a brief introduction and review the basic aspects in the theory of cellular automata and neural network computing from the viewpoint of a physicist. We also present the main applications for physical problems, like Ising dynamics and hydrodynamics simulations by cellular automata evolution. In the case of neural networks we discuss the Hopfield model with its relation to the theory of spin glasses and report on multilayer networks. In addition we show how a reconfigurable network of transputers can be used as a highly flexible and fast simulation machine. Hardware configuration and software implementation for a neural network model are presented.*

Table of Contents

---

[1] Permanent address after November 1988: Interdisziplinäres Projektzentrum für Supercomputing and Institut für Theoretische Physik, ETH Zürich, Switzerland

Invited talk presented at the 12th Gwatt-Workshop on Complex Systems, October 13-15, 1988, Gwatt/Thuner See, Switzerland

Zur Selbstorganisation ebenso fähig wie zur Selbstreproduktion sind Zelluläre Automaten ein Modell der Welt und doch zugleich auch eine Welt für sich.[18]

1. Cellular Automata Computing

Cellular Automata (CA) are simple mathematical objects introduced in the early fifties by John von Neumann and Stanislaw Ulam to *abstract the logical structure of Life*. Thus beside many pure mathematical papers on the theory of automata, one of the earliest applications was for biological systems. Physicists become active in this field only relatively recently, motivated in part by the expected proliferation of special purpose computers built for stochastic simulations of very highly time consuming problems and partly by the study of models for neural networks and failure-resistant computing machines. The popularity in the physical community arises later from the work of Stephan Wolfram on deterministic CA and from the *Game of Life* discussed extensively about 1970 in the columne *Mathematical Games* of *Scientific American* by Martin Gardner and in the famous book *Winning Ways for Your Mathematical Plays* of Elwyn Berlekamp, John Conway and Richard Guy. Since then, CA have been established as unique tools to analyze the emergence of organization, complexity and pattern formation, including applications like dendritic crystal growth, reaction diffusion systems, Ising models, and fluid patterns in hydrodynamics. CA are versatile enough to offer analogies with almost all the topics presented at this workshop, especially in complex behaviour, chaos and neural networks.

1.a. Basic Theory and Phenomenology

CA may be used as simple models for mathematical idealizations for a wide variety of physical, biological and computational systems. They can be considered as discrete dynamical systems, constructed in a very simple way, however, able to show very complex self-organizing behaviour arising from cooperative effects. Their main features are:

- ***Discrete in space:*** CA consist of spatial cells or sites arranged on a discrete grid or array.
- ***Discrete in time:*** The values of the cells evolve in a sequence of discrete time steps.
- ***Discrete states:*** Each cell can take on only a finite number of possible values.
- ***Homogeneous states:*** All cells are constructed identically and are arranged in a regular array.
- ***Synchronous updating:*** All cells are updated in parallel.
- ***Deterministic rule:*** Each cell is updated according to a fixed, deterministic rule, depending in space only on the values of a local neighbourhood of cells around it and depending in time only on the values for a fixed number of preceding steps.

In the most simple case, a CA consists of a line of cells with each cell carrying a value either 0 or 1. For example, the rule could take the value of a cell at a given time step to be the sum modulo two of the values of its two nearest neighbour cells on the previous time step. E.g. if we start from a single seed, then the pattern is found to be selfsimilar with fractal dimension $\log_2 3$. Even starting with a random sequence of initial cell values, say each with equal probability, the dynamics leads to complex patterns due to correlations. For the considered one dimensional case we can generalize the rules, allowing for the cell values integers in the range 0 through $k-1$ with a spatial neighbourhood of at most $2r+1$ cells. These systems may evolve by iteration of the mapping

$$a_i(t) = f\{\sum_{j=-r}^{j=r} \alpha_j a_{i+j}(t-1)\} \quad , \tag{1}$$

where $a_i(t)$ is taken to denote the value of the cell at local position i at time step t. The α_j are integer constants, taking on $\alpha_j = k^{r-j}$ for the above discussed situation and the function f takes a single integer argument. Special rules giving equal weight to all cells in a neighbourhood, can be defined by setting $\alpha_j = 1$ and are termed *totalistic rules*. Let us return to our example from the beginning with $k = 2$ and $r = 1$. We find the eight possible states at time t of the three adjacent states at time $t-1$ as:

$$\frac{111}{0} \quad \frac{110}{1} \quad \frac{101}{0} \quad \frac{100}{1} \quad \frac{011}{1} \quad \frac{010}{0} \quad \frac{001}{1} \quad \frac{000}{0} \quad . \tag{2}$$

This is one of $k^{k^{2r+1}} = 256$ possible rules, coded binaer as 01011010 and thus in decimal termed *rule 90*. Another example *rule 22*, coded as 00010110, belongs to the subclass which can be characterized by a *totalistic rule:* If according to eq. (1) the sum of the adjacent three sites of the previous time step is 0, 2 or 3, we have $a_i(t) = 0$ and on the other hand, if the sum is 1, we have $a_i(t) = 1$. Thus this situation can be characterized by the (totalistic) binaer code 0010, termed *totalistic rule 2*. Stephan Wolfram has investigated all the possible rules for different situations in one dimension with given small k and r. First he found empirical evidence for the existence of four basic different classes of behaviour in CA, shown in typical examples in FIG. 1.

- ***Class 1:*** The chain becomes a homogeneous state.
- ***Class 2:*** The dynamics forms simple localized time periodic structures.
- ***Class 3:*** The evolution exhibits chaotic behaviour.
- ***Class 4:*** The evolution leads to complex structures.

Patterns generated for the simplest one-dimensional CA with $k = 2$ and $r = 1$ do not exhibit *Class 4* behaviour. This complex behaviour is displayed only by CA that involve more than 2 states per cell or a wider neighbourhood than 3 adjacent cells. E.g. the approximate fraction of totalistic CA rules with $f(0) = 0$ belonging to *Class 4* is 6% for $k = 2, r = 2$ and $k = 2, r = 3$ and 7% for $k = 3, r = 1$.

Beside this phenomenological description one needs quantitative statistical measures of order and chaos in patterns generated by CA evolution to distinguish the four classes of behaviour, identified qualitatively above. *Dynamical systems theory* gives us objective measures of complexity, like entropies and dimensions.

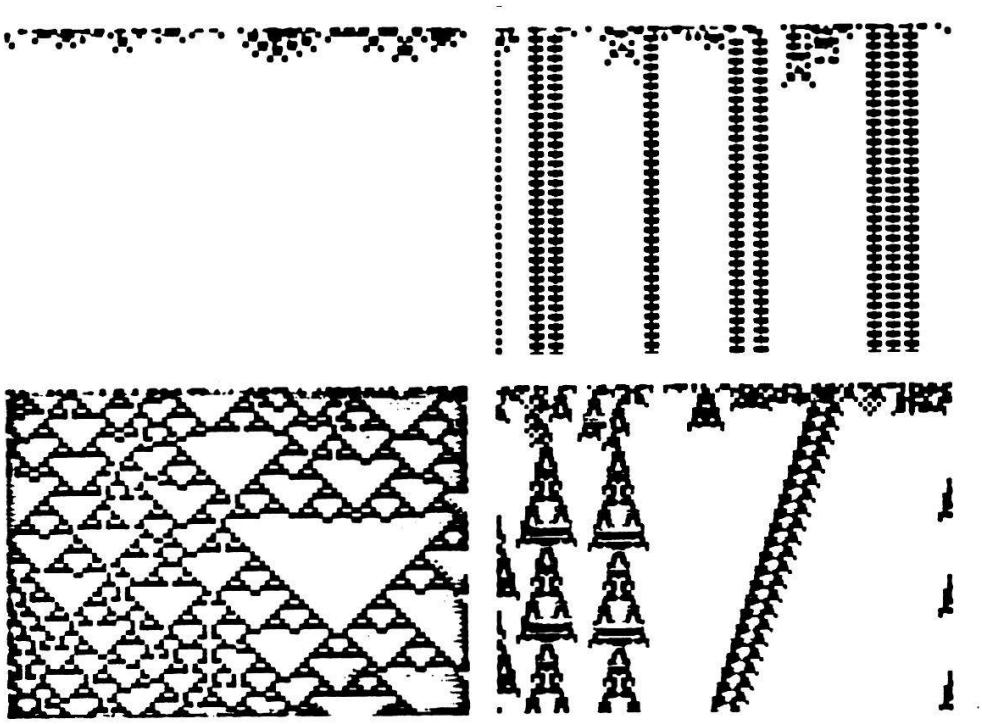

FIG. 1. Typical examples for the four classes of CA behaviour for evolution from a random startconfiguration. The cases shown are specified by totalistic rules 1302, 1005, 444 and 792, respectively, for $k = 3$ (0 blank, 1 grey, 2 black) and $r = 1$. (From S. Wolfram /7/)

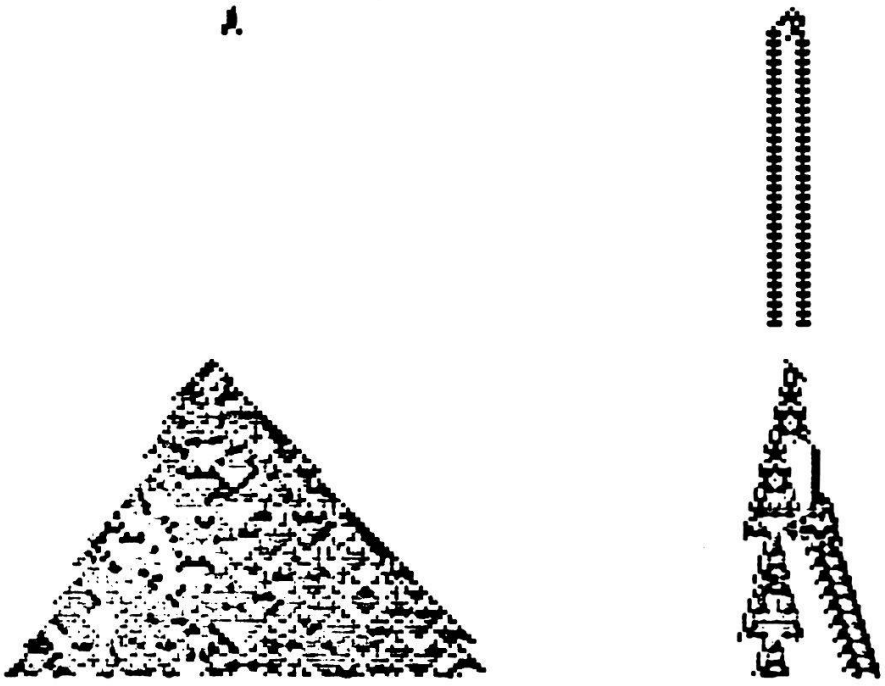

FIG. 2. Information propagation generated by difference patterns obtained by changing a single initial site value in the CA of FIG. 1. (From S. Wolfram /7/)

Topological entropies provide characterizations of the number of possible configurations $N(n)$ of length n that occur in the system

$$S(n) = \frac{1}{n} \log_k N(n) \quad , \tag{3}$$

measure entropies reflect those that are probable

$$S(n) = -\frac{1}{n} \sum_i^{k^n} p_i \log_k p_i \tag{4}$$

with probability p_i. If we consider the CA configurations as elements of a Cantor set, then the limiting entropies give respectively the Hausdorff and measure dimensions of this set

$$d^{(n)} = \lim_{n \to \infty} S(n) \tag{5}$$

If the sequences are considered as *messages* we can make contact with the capacity and the Shannon information content. A tentative definition of the four classes of CA behaviour may now be given in terms of measure dimensions. Investigating the number of possible configurations of all values in some space $N(n = X)$ or time region $N(n = T)$ we find for *Class 1* CA $d^{(X)} = d^{(T)} = 0$ since the limit set contains only a finite number of configurations. *Class 2* CA have zero temporal measure dimension, since it almost always yields periodic structures, but has a spatial measure dimension $d^{(X)} > 0$. For most *Class 3* CA $d^{(X)}$ decreases with time, giving $0 < d^{(X)} < 1$, and suggesting that a fractal subset of all possible configurations occurs. *Class 4* CA give rise to patterns with complicated structure, that typically expands highly irregular with time, prohibiting the definition of dimensions. Thus dynamical system theory provides analogues to CA: simple rules exhibit simple limit points *(Class 1)* or limit cycles *(Class 2)*, while complex rules exhibit chaotic behaviour analogous to that found with strange attractors *(Class 3)*. No direct analogue can be identified for *Class 4* CA among continuous dynamical systems.

Information propagation, characterizing the stability or predictability of CA behaviour under small perturbations in initial configurations, is another important property of CA. FIG. 2 shows examples for patterns of differences generated by changing a single cell value in the initial CA configuration. The characteristic effects obtained allow a further classification scheme of the four different classes of CA.

- ***Class 1:*** No effect on the final state.
- ***Class 2:*** Changes only in a finite space region.
- ***Class 3:*** Expands in space and time at an asymptotically constant rate.
- ***Class 4:*** Shows irregular changes, effectively unpredictable.

This behaviour, describing the speed of information propagation can be related to the Lyapunov exponents for the CA evolution, which measures the rate of divergence of trajectories in the space of configurations. A number of inequalities can be derived between entropies and Lyapunov exponents implying an important connection between the static properties of CA and their dynamic behaviour. Here we have only described the most simple examples for direct statistical measurements of complex

behaviour in CA configurations. Beyond entropies and Lyapunov exponents, correlation functions, power spectra or Fourier transforms provide other statistical measures.

Such quantities suggested by *information theory* allow only a rough characterization of CA. However, *formal language theory* and *computation theory* give a more complete description of complex behaviour in CA. These theories may in general be expected to play a role in the theory of non-equilibrium and self-organizing systems analogous to information theory in conventional statistical mechanics. Sets of configurations generated by a CA rule are described as formal languages in computation theory terms. Each configuration corresponds to a word in a language formed by definite grammatical rules. Its evolution is considered as a computation, which processes information specified as the initial state. The output of the CA evolution for *Class 4* CA, however, cannot be predicted by an effective short cut, that allows a more efficient computation than the evolution itself. This means, there is no effective way to determine the evolution from a given initial state beside the explicit simulation. Or in other words, no finite formula can be given to describe *Class 4* CA behaviour, and many infinite time limiting properties would be formally undecidable. As a consequence, the long time behaviour $t \to \infty$ of the entropy of the limiting set of CA configurations is in general not finitely computable. The question whether a *Class 4* CA with a given finite initial configuration dies out or not, may be equivalent to the halting problem for *Turing Machines*. Some of these finite structures can have very long histories before their demise. From the formal equivalence between CA and Turing Machines it is strongly suspected that *Class 4* CA would be capable of universal computation. However, this has only be proved in some simple cases, the most prominent one is the 2-dimensional CA *Game of Life*. In this context Stephan Wolfram writes:

> The possibility of undecidable questions in mathematical models for physical systems can be viewed as a manifestation of Gödel's theorem on undecidability in mathematics, which was proved by Kurt Gödel 1931. The theorem states that in all but the simplest mathematical systems there may be propositions that cannot be proved or disproved by any finite mathematical or logical process.

And von Neumann said in his laudatio on the occasion of the presentation of the 1951 Einstein Award for Kurt Gödel: Kurt Gödel hat gezeigt,

> daß gewisse mathematische Theoreme mit den akzeptierten exakten Methoden der Mathematik weder bewiesen noch widerlegt werden können. Mit anderen Worten hat er das Vorhandensein von unentscheidbaren mathematischen Sätzen bewiesen.

Then he continues:

> Es muß der wichtige Punkt betont werden, daß dies kein philosophisches Prinzip oder eine einleuchtende intellektuelle Einstellung, sondern das Resultat eines strengen mathematischen Beweises von besonders raffinierter Art ist.

This shows that the understanding of complexity and its origins is a fundamental challenge for modern science. The considered CA models, which are as simple as

possible in their basic construction, yet capture all essential mathematical features necessary to reproduce the complexity that is seen. Their overall behaviour may be complex enough to reproduce the complex phenomena observed in many physical and other systems, like e.g. partial differential equations (fluid dynamics), finite difference equations (spin systems, lattice dynamical systems), feedback shift registers, systolic arrays, Turing Machines, random Boolean networks and neural network models. Only two examples of this incomplete list will be considered very briefly, the simulation of Ising spin systems in the microcanonical ensemble and pattern formation in 2-dimensional hydrodynamic systems. In the next Chapter in some more detail neural network models will be presented.

1.b. Ising Dynamics Simulations by CA Evolution

Numerical simulations of statistical systems has usually been carried out by stochastic methods, especially the *Metropolis Monte Carlo* algorithm, and they are a major tool in the study of phase transitions and critical phenomena. Fast simulations of Ising models are here of special interest in the context of spin glasses, nucleation and random field systems. A new deterministic method was originally proposed by E. Fradkin, G. Vichniac and Y. Pomeau, as an example for a CA. Since their algorithm is deterministic, random numbers or only required for setting up the initial spin configuration. Furthermore the algorithm avoids exponential Boltzmann factors required by the Metropolis Monte Carlo method and permits integer programming with multispin coding, leading to a consequent increase in computer speed. The largest Ising system ever studied with 15.130.968.192 spins was done with this method by J.G. Zabolitzky and H.J. Herrmann with updating rates of about 4300 million of spins per second on a CRAY-supercomputer.

The idea of the Ising dynamics simulation by CA evolution, is based on reversible automata, which are backward as well as forward deterministic, that means that each configuration has a unique predecessor. This subclass of CA has an exact time invariant quantity, which plays the role of energy. With the ergodic assumption, one gets a model for Ising spin systems. Let us become more specific. Devide a square or cubic lattice into two independent sublattices A and B with spins $\sigma_i^{A,B}$, having Boolean values 0 or 1 on each lattice site or cell. Then at alternating time steps a whole sublattice is simultaneously updated in the following way:

> If and only if the spin i has as many up as down neighbours it is flipped, but only on one sublattice at a time.

Thus this is a description for the dynamics of an Ising model in the microcanonical ensemble with conserved energy, called Q2R-rule.

The crucial question whether the Q2R-CA can really simulate the Ising model was considered very recently. We know from CA theory that the system, or a part of it, can settle into periodic motion, without ever reaching a time independent equilibrium. Later it was shown, that the Q2R algorithm is indeed not fully ergodic.

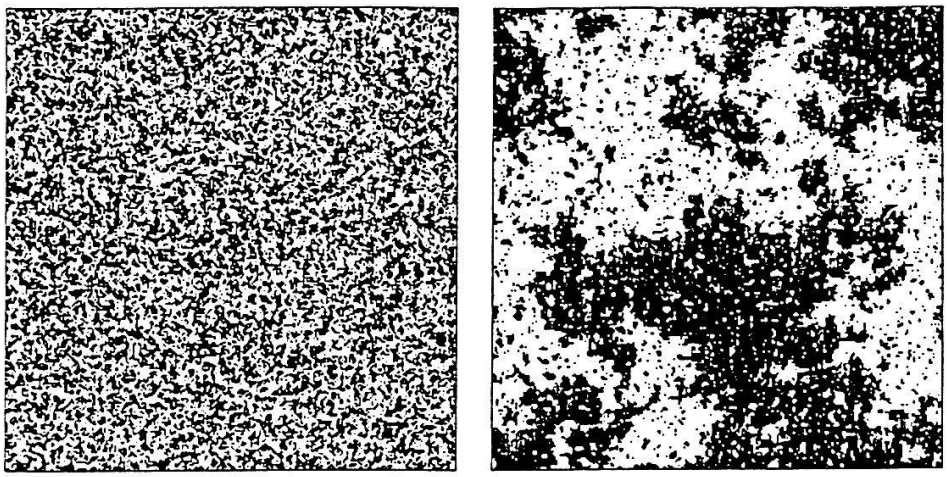

FIG. 3. Ising dynamics simulation by CA evolution: Equilibrium configuration of the 2-D Ising model above the critical energy (left) and at the critical energy (right). (From T. Toffoli and N. Margolus /12/)

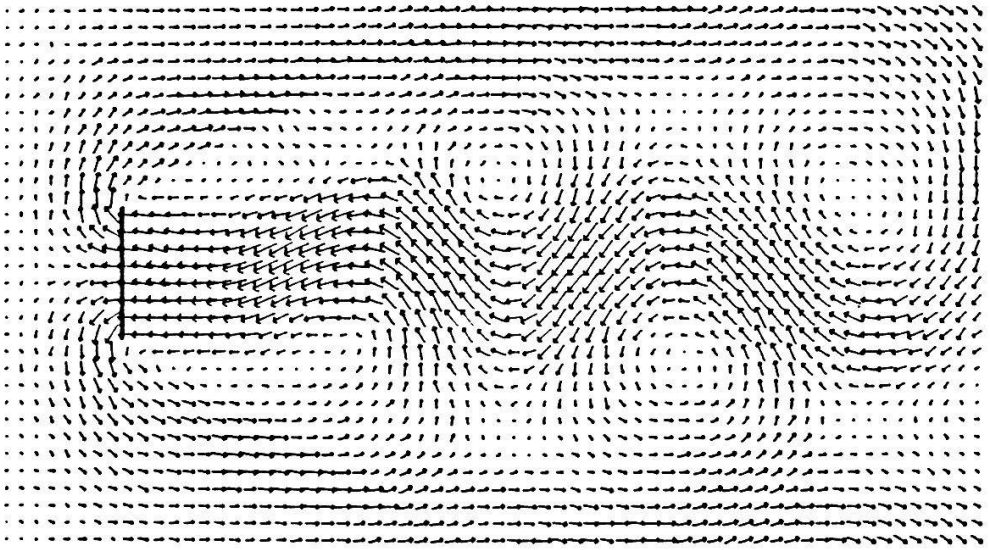

FIG. 4. Hydrodynamics simulation by CA evolution: Von Karman street behind a flat plate at 5000 time steps ($Re \sim 90$) with *wind tunnel* boundary conditions, the mean flow has been subtracted. (From D. d'Humières and P. Lallemend /58/)

In other words, Q2R does not explore the whole phase space and certain parts are not accessible. To overcome such problems, at least one has to introduce a certain degree of randomness into the algorithm e.g. by randomly flipping some spins over long time intervals. A generalization of this Q2R algorithm was introduced by M. Creutz. The basic idea in the CREUTZ and Creutz-like algorithms is to introduce one (or a small number of) demon(s) which act as a movable heath bath. The updating of the local spins conserves the total energy of the system plus demon. Thus for a small number of demons compared to the total number of degrees of freedom in the system, again we have a simulation in the microcanonical ensemble.

1.c. Hydrodynamics Simulations by CA Evolution

On a microscopic scale physical fluids consist of discrete particles, forming large assemblies of atoms or molecules, which are described by the laws of continuum mechanics, e.g. such as the Navier-Stokes and continuity equations. Thus from the dynamics on a microscopic level we can derive rules, equivalent to those for CA, describing discrete approximation to hydrodynamics. The most simple fluid model is the HPP lattice gas, which has an underlying 2-dimensional square lattice with unit bonds or links. At the most 4 particles with unit speed can be located at the vertices having velocities of unit modulus pointing in one of the four possible link directions. In addition we have the exclusion rule, that no two particles occupying the same vertex with the same velocity are allowed. The evolution proceeds synchronously at each vertex in discrete time steps, where the particles first propagate and then collide: Each particle moves one link to the next vertex to which its velocity was pointed. Then any configuration of exactly two molecules moving in opposite directions (head-on-collisions) is replaced by another one, deflected 90° from the path they were following. This HPP-automaton has been shown numerically to relax to thermodynamic equilibrium. However, the underlying lattice produces a symmetry which is insufficient to insure the isotropy of the fourth degree tensor, relating momentum flux to quadratic terms in the velocity. This unisotropy, a ghost of the lattice, yields via coarse graining macroscopic hydrodynamics which departs from the Navier-Stokes equation. This blemish is removed by the FHP-automaton, a triangular lattice gas model with hexagonal symmetry and hexagonal lattice gas rules. The setup is the same as in the HPP-automaton except for different collision rules, which can be defined in different versions, each conserving particle number and momentum. It is worth to note, that this model can be made nondeterministic, choosing the outcome defined by a given collision rule at random.

These CA models for lattice gas hydrodynamics compete with the traditional computational methods. The novelty of the CA approach is the discrete version of the theory, using only binary arithmetic instead of high-precision floating point arithmetic, required by conventional approaches. Moreover, complicate boundaries, barriers and container walls are more easily implemented by special collision rules. Finally, macroscopic hydrodynamics is obtained by coarse graining the lattice gas over a few hundred particles. In this context, however, the particles cannot interpreted as real atoms or molecules, moreover they play the role of idealized

super-particles, accurately enough to reproduce hydrodynamics in the limit of large systems and for incompressible flows. So far we have considered lattice gas models for 2-dimensional hydrodynamics. The extension of lattice gas models to three dimensions is not straightforward since there exist no regular lattices with the required symmetries. The study of appropriate models is under current investigation.

2. Neural Network Computing

Neural network computing and connectionism denote a new multidisciplinary research paradigm concerned with the analysis and construction of simplified models of the humain brain. The idea to construct intelligent machines is relatively old and goes back to the late fifties to John von Neumann's uncompleted work *Die Rechenmaschine und das Gehirn*, which was also triggered by principles of neural information processing. Another classical field in theoretical informatics, the theory of finite automata which makes the contact to the previous Chapter, has its origin in a neural network model: The classical McCulloch-Pitts-neuron. Such simple networks define finite automata and are related to regular languages. Later, about 1960, F. Rosenblatt published his famous perceptron concept, which was investigated intensively by M. Minsky and S. Papert. They pointed out key limitations of perceptrons that led to mass abandonment of this line of research. However, in the 70's, a renewed interest was borne and many researchers were working in this field, e.g. Amari, Anderson, Feldman, Fukushima, Grossberg, Hinton, Kohonen, McClelland and Rumelhart. Their work lead currently to an explosion of interest: We estimate that research in this area has increased at least by two orders of magnitude over the last five years. Both literature and number of meetings focusing on artificial neural systems are growing at an amazing rate.

Neurobiologists, neurophysiologists, mathematiciens, physicists, psychologists, computer scientists and engineers are studying and formulating theories with the goal to understand better how brains work and are designing devices and computers that emulate some of their typical computing attributes. Most neuroscientists accept that the brain's power arises from the huge number of relatively simple neural cells, that are highly interconnected and process information in parallel. The cortex contains over 10^{10} such neurons, each connected to thousands of others. All of our knowledge is probably stored in the strengths of these synaptic connections. The inputs make contact with the receiving neuron at the synapses and releases neurotransmitters, which influence the electrical excitability of the receiving cell at this point. The neuron collects all the different inputs and either decides to fire an impulse or not, depending on a certain threshold value. *Patterns* that may occur repeatedly will eventually become stable representations or associative memories of the inputs that give rise to them. Thus the network produces a dynamical system of firing patterns. Furthermore the network needs to be able to store new information. A widely accepted assumption is, that learning modifies the efficiency

of the synapses. Thus the learning process is described by rules about how the connections in a network should change as a result of experience.

As shown by examples, electronic and optical realizations of neural networks are capable of performing various and surprising processing functions, such as associative memory and learning systems, combinatorial optimization, source encoding and channel decoding, pattern recognition and image restoration, speech recognition and synthesis, control of robot motion, and many others. The basic elements of such artificial neural networks are: *Network properties, cell properties, dynamic properties,* and *learning properties:*

- ***Network Properties:*** A neural network model consists of autonomous processing units called neural cells or neurons which are joined by lines having a given weight called synaptic efficiencies. The network structure can be classified according to its topology: Simple networks, e.g. of the Hopfield type, with only input and output units, and hierarchically structured networks with cells between intermediate layers, called *hidden units*, which become necessary for more complex processing functions.
- ***Cell properties:*** Neural cells compute their activation or output from the activations of the other cells directly connected to them and the corresponding synaptic efficiencies for these connections. Cell activations may be discrete, taking on values $\{0,1\}$ or $\{-1,0,1\}$, or continuous, assuming values in the interval $[0,1]$ or $[-1,1]$.
- ***Dynamic properties*** The dynamics of neural cells can be chosen in different ways, e.g. parallel or synchronous and sequential or asynchronous. In the first case all cells are updated at the same time, and in the second case one after another according to a given rule.
- ***Learning properties:*** Learning is an adaptive self-organizing process, which makes it very intriguing, complex and difficult to understand. It is widely accepted that learning is achieved by changing the efficiency of the synapses. Hopfield-like models start with a given input information, whereas networks with hidden units learn from examples.

2.a. Hopfield Model and Spin Glass Analogy

To become more specific let us consider the Hopfield-Little model, a thermodynamic extension of the McCulloch-Pitts program for the realization of the basic features of a neural network. The model consists of $N \times N$ mutually interconnected neurons with a phase space of 2^N states represented by outputs $\{\sigma_i\}$ taking on values $+1$ (active) or -1 (inactive neurons). The dynamics of these neurons can be assumed to be parallel *(Little dynamics)* or sequential *(Hopfield dynamics)*, whereas the performance of both dynamics appears to be similar. There are still other possibilities and we like to mention the dynamics proposed by Horner, where that neuron is updated with the largest local field pointing in opposite direction. The local fields h_i are evaluated from the inputs $J_{ij}\sigma_j$, which each neuron σ_i receives from all the other neurons σ_j, and a bias input θ_i associated with itself

$$h_i(t) = \sum_j^N J_{ij}\sigma_j(t) + \theta_i \quad , \tag{6}$$

where the synapses J_{ij} describe the strengths or weights of the connections. At given times switches are turned on or remain off, and the inputs h_i are fed back to corresponding neurons σ_i to change their states or to leave them fixed, according to a given threshold rule $g(h)$, so that

$$\sigma_i(t + \delta t) = g(h_i(t)) \quad , \tag{7}$$

where $g(h) = -1$ for $h \leq 0$ and $+1$ for $h < 0$. Thus neurons take binary values either ± 1 and the binary outputs are sent again out or distributed through the interconnection network to regenerate new inputs. Learning is performed by an appropriate choise of the synapses J_{ij}. The relation between the J_{ij} 's and stored patterns or memories, which is the encoded information in the system, will be termed the learning rule. Hopfield's model employs the so called Hebb-rule, where the p −memories $\{\xi_i^\mu\}$, $i = 1, \ldots N$, $\mu = 1, \ldots p$, $\xi_i^p = \pm 1$, are learned at the very beginning after the recipe

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu \quad . \tag{8}$$

It is evident that this is a rather simple form of learning and more sophisticated learning rules can be constructed.

Hopfield has shown that if $J_{ij} = J_{ji}$, neurons always change theire states in such a manner that they minimize an energy function defined by

$$E = -\frac{1}{2} \sum_{i,j} \sigma_i J_{ij} \sigma_j - \sum_i \theta_i \sigma_i \tag{9}$$

and stop at minima of this function. Now we are coming closely related to the extensively studied problem of spin glasses. In fact we have precisely the same energy function and the same equation describing relaxational dynamics of Ising spins at zero temperature. The dynamic rule (7) in spin glass theory means, that the energy (9) is described by flipping spins σ_i aligning with their internal field h_i , eq. (6), until a stable state or local minimum is reached. Translating this back to neural networks, we see that stable states are associated with stored patterns. Building an associative memory with high storage capacity, one has to construct an energy function E with a connection matrix J_{ij}, leading to many local minimas, precisely the property of spin glasses. However, the typical question asked is rather different in neural network and spin glass theory. In spin glasses the connection matrix J_{ij} according to a given probability distribution $P(J_{ij})$ is given and one investigates the phase space. For associative memories the situation is just the opposite. For a given set of pattern pixels $\{\xi_i^\mu\}$ in phase space, one has to construct the connection matrix J_{ij} via a learning rule in such a way, that the resulting energy function will have local minimas at positions ξ_i^μ. Nevertheless, methods of statistical mechanics and spin glass theory can be applied to such neural network models and can help to answer questions e.g. about the memory capacity and the basin of attractions for the stored patterns. The first question can be answered by analytical methods, showing that

$M \sim 0.14N$ patterns are memorized only with minor errors, where in the limit $N \to \infty$ the system has a sharp transition at $M/N = 0.144$. The second question was investigated numerically, demonstrating that the size of the basin of attractions depends on the amount of stored patterns and increases with decreasing M/N .

Elements of the presented model go back to the work of Cooper, Hebb, Kohonen, Steinbuch and many others. Since its introduction in 1982 the considered Hopfield model has been analyzed, modified and improved in many different directions by people of the spin glass industry. Estimates of the storage capacity, memories for correlated and hierarchically structured information, sequences of patterns, learning by clipping off and dilution of synapses, asymmetric connection matrix elements, learning and forgetting, etc. were investigated. This is only a very incomplete list of recent advances by physicists in this field and we apologize the developments which we have not mentioned so far.

2.b. Multilayer Networks

The most simple case, that of a single layer neural network, known as the perceptron model, is well investigated and a simple theory exists. Perceptrons consist of an array of input units, taking values 0 or 1, and another array of output units, which have in between a single layer of connections J_{ij}. For a given connection matrix J_{ij} and values for the input units σ_i, the net activity h_i (identified as local fields in the previous section) into output unit i is defined as $h_i = \Sigma_{ij} J_{ij} \sigma_j$, from which the state of unit i can be determined according to $\sigma_i = \Theta(h_i)$, where $\Theta(x)$ is the unit step function. Learning results from a simple learning heuristic , also called supervised learning process, which gives the network the ability to form and modify its own connections in ways that often rapidly approach a performance optimum. This is done by the *delta learning rule* (or Widrow-Hoff rule), which determines the appropriate values of the connection matrix J_{ij} in the following way: On a given trial the network first generates an output pattern $\sigma^{(O)}$ in response to the input pattern $\sigma^{(I)}$ of a training pair. This defines an error signal (or delta) with respect to the desired output or training target $\sigma^{(T)}$, which is used to adjust the connection matrix, in that way to reduce the error by a simple gradient or steepest descent procedure

$$\Delta J_{ji} = \varepsilon \, (\sigma_j^{(T)} - \sigma_j^{(O)}) \, \sigma_i^{(I)} \quad , \tag{10}$$

where ε represents a learning rate, controlling the size of the discrete learning steps. Thus perceptrons can learn to recognize patterns by adjusting the strengths of connections between the arrays of input and output units. However, certain patterns, e.g. the exclusive disjunction or parity function cannot be computed by single layer perceptrons. This can easily be understood. As an example we consider the computation for an exclusive-or (XOR) on two inputs A and B, linked to a single output with synapses J_A and J_B , respectively. We can consider four different cases: *i)* A=B=0, *ii)* A=0, B=1, *iii)* A=1, B=0, and *iv)* A=B=1. Case *i)* requires $\Theta(0) = 0$, case *ii)* requires $J_A > 0$ so that $\Theta(J_A) = 1$, *iii)* similarly with J_B, and case *iv)* $J_A + J_B < 0$, so that $\Theta(J_A + J_B) = 0$. This contradicts the conditions for case *ii)* and

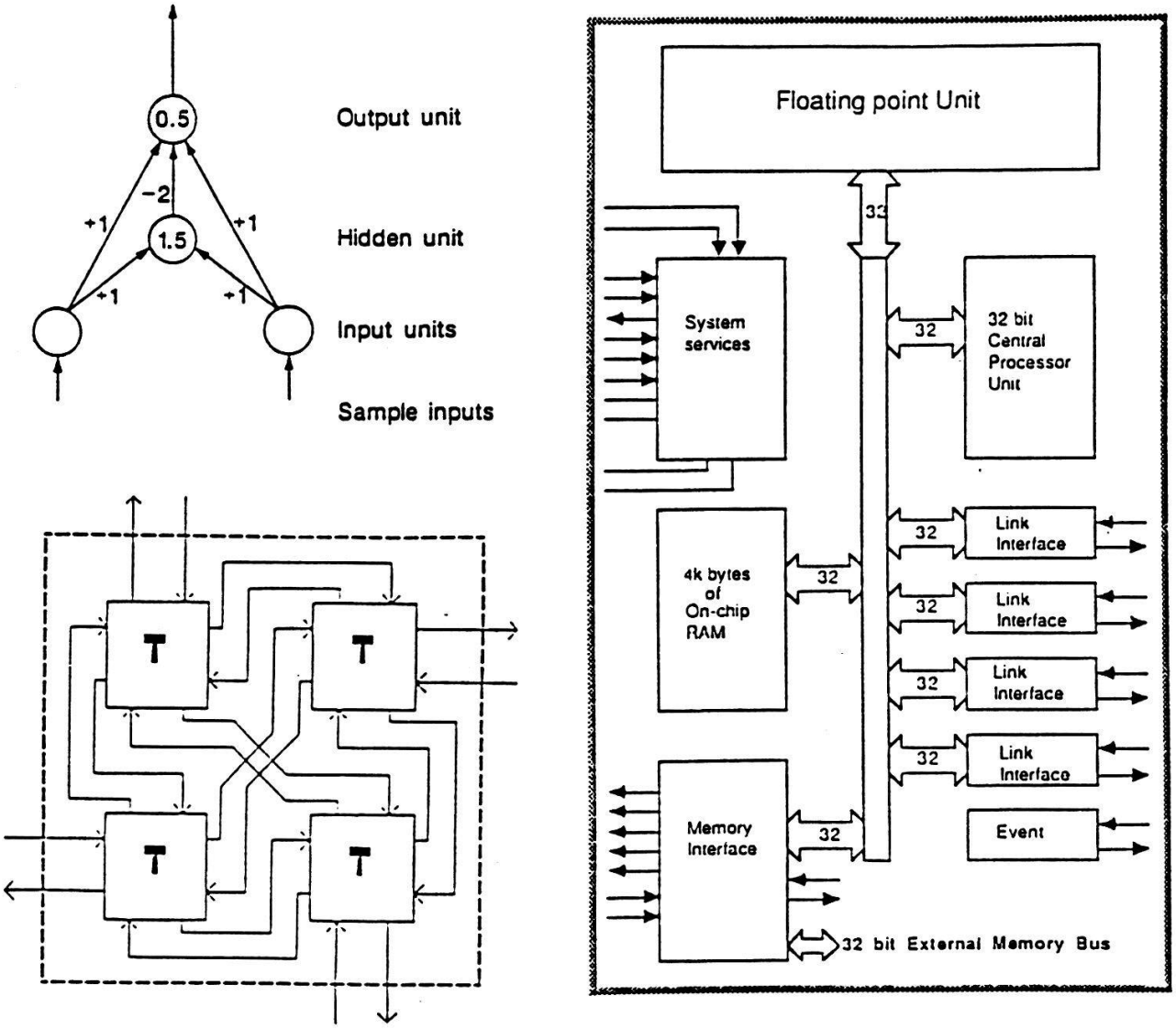

FIG. 5. A neural network with a *hidden unit* that recognizes an exclusive disjunction (XOR). Each unit fires when its threshold (given within the unit) is exceeded by the sum of its inputs.

FIG. 6. Block diagram of the INMOS T800 transputer, a 32bit CMOS microprocessor with a 64bit floating point unit. It has 4Kbytes on-chip RAM for high speed processing and for communication links. The instruction set achieves efficient implementation of high level languages and provides direct support for the OCCAM model of concurrency. (From INMOS /120/)

FIG. 7. Example for a link configuration of 4 Transputers. Note the selfsimilar structure. (From D. Würtz and G. Hartung /127/)

iii). From this Minsky and Papert concluded, if the perceptron could not compute such simple functions it could hardly perform the complex tasks required for perception and intelligence.

However, if we add one more layer or multiple layers of *hidden units* between the array of input and output units, the computational power of the artificial neural network rises abruptly, and Minsk and Papert's critique no longer applies. The network solution for the XOR-problem with one hidden unit is shown in FIG. 5. Thus through hidden units the network is able to simulate logic functions, like e.g. XOR, OR, AND and the MAJority rule. Note, however, the strengths of the synapses, necessary to implement these funtions are not unique. Feedforward neural nets of the 80's are the best known examples for adaptive multilayer networks and the *back-propagation* method of D. Rumelhart and coworkers is the most prominent member of this class of models. The equation from which we can evaluate the synaptic weights is similar in form to that of the delta-rule, but the back-propagation rule provides a more general means of computing the delta of a neural unit. The process of back-propagation consists essentially of two parts: First, the input into the network is forward-propagated until the output is created. Secondly, the output will be compared with the learning input, and the resulting error will be back-propagated through the system, so that the synaptic weights will be adjusted starting from the output array in direction of the input array.

$$\Delta J_{ji} = \varepsilon\,(\sigma_j^{(T)} - \sigma_j^{(O)})\, g'(net_j)\, \sigma_i^{(I)} \quad , \tag{11}$$

where g' is the derivative of the activation function, operating on the sum of the inputs in order to determine the unit's output. The real power of this back-propagation rule arises of its assignments of deltas to hidden units that receive no direct feedback from training or learning patterns in the outside world. Thus, like the basic delta-rule, back-propagation is a gradient descent heuristic, however, modified for internal units again based on discrepancies between the values of output units and a training pattern.

3. System of Parallel Organized Transputers (SPOT) for Cellular Automata and Neural Network Computing

Large systems of CA and neural network models have a large number of possible parameters and their simulation requires powerful computers. In addition they process immense quantities of information in parallel, which is handled by serial computers less efficiently. On the other hand if you have your own special purpose computer you have not to share the machine with other users and you will have no limited access, resulting e.g. in a restricted number of CPU hours. The financial aspect is also important, compared with 2×10^7 US$ for a CRAY-2, you need only tools and components for your special purpose computer, which are usual less than 10^5 US$. The third argument concerns the speed. Now there are highly integrated ultrafast chips on the market, so that special purpose computers can be as fast or

even faster than a supercomputer. These three points are the reason why special purpose computers were developed during the last couple of years. Now the situation is again changing, and so called *Neurocomputers*, hardware on which neural nets can be implemented efficiently, are becoming commercially available. These machines are usually coprocessor boards that plug into conventional personal computers or work stations.

One of the first special purpose computers that have been built was the Cellular Automata Machine (CAM) developed at MIT in Cambridge. For the simulation of hydrodynamic problems a similar machine was constructed at ENS in Paris. On the side of neural network computing many prototype machines were constructed based on usual microprocessor concepts. Recently there appeared a new RISC-microprocessor on the market, the Transputer, which seems to be very suited for powerful low-cost special purpose mini-supercomputers. The Parsytec and Meiko Computing Surfaces are machines which are based on the Transputer technology and which are commercially available. CA and neural nets were succesfully implemented on these computing surfaces. Our machine SPOT, a system of parallel organized transputers, makes also use of the transputer microprocessor concept, and is a highly flexible and reconfigurable personal mini-supercomputer system, which allows both, CA and neural network computing in a very efficient way. Flexibility and reconfigurability guarantee that several type of models of CA and neural nets can be implemented which is important in research, application studies and early applications.

3.a. Hardware Configuration

We have investigated hardware configuration and software implementation, which allow for simulations on a massive parallel computer architecture. To realize this we were looking for new routes and have chosen the transputer as the central unit for our simulation machine. The INMOS T414 transputer is a powerful 32-bit multiprocessor capable of performing about 10 million instructions per second and carrying 2-Kbytes of fast on chip memory for high speed processing. The block diagram of the new T800 transputer is shown in FIG. 6. In contrast to the T414 transputer this microprocessor has in addition a 64-bit floating point unit and a doubled on chip fast memory. The main advantage of these chips in contrast to usual microprocessors are the four bidirectional communication links with a capacity up to 20-Mbits/sec supporting communication between programs running on different transputers. A simple possible configuration for 4 transputers is shown in FIG. 7. and many other topologies can be constructed like square and toroid arrays, trees, etc., depending on the considered model. The transputers allow for a direct point-to-point communication between these links without any data bus. So in contrast to usual multiprocessor systems connected by a shared bus, problems by bus contention are unknown in multi-transputer systems. Also no additional control logic is required to control sharing of the bus. Each transputer can address 4-Gbytes and is connected via an own and independent data bus to additional off-chip memory. It is worth to note, that in contrast to a large global memory, the overall channel capacity is

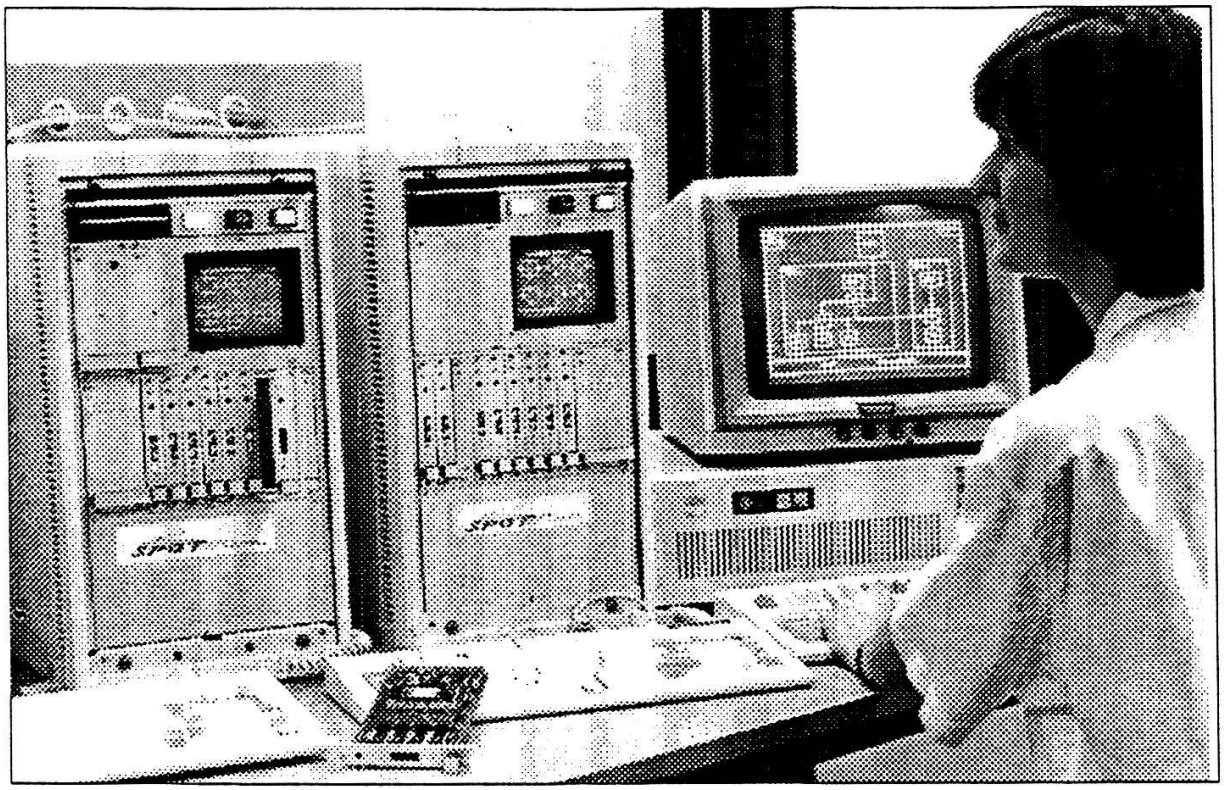

FIG. 8. Computing surface module SPOT. Two connected SPOT's can be seen operating together with an additional external host environment. (From D. Würtz and G. Hartung /127/)

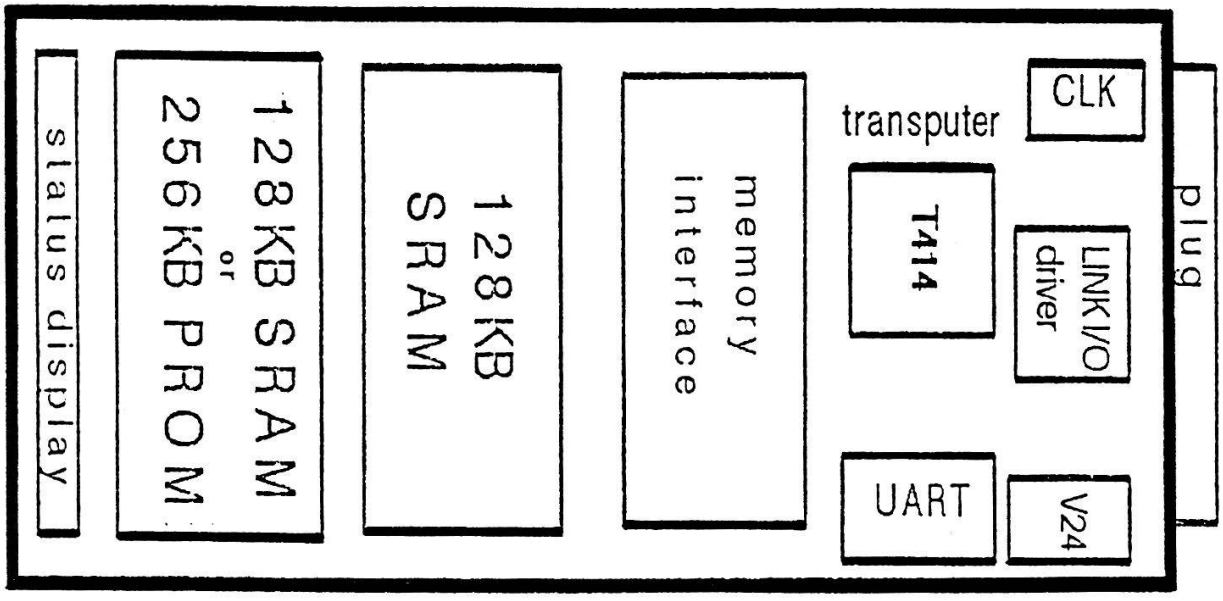

FIG. 9. Standard computing element with one transputer, 256Kbyte external memory and other device support. (From D. Würtz and G. Hartung /127/)

proportional to the total number of transputers preventing an overtax of the data bus capacity.

The INMOS transputer was designed to implement the *OCCAM model* of concurrency. *OCCAM* is a high level language and is the transputer's native tongue. The language enables a system to be described as a collection of concurrent *processes*, which communicate with each other through channels. Programs are built from primitive processes and then are combined to form sequential or parallel constructs. Each channel, which can be either a software construct or mapped onto physical links between transputers, provides a one way connection between two concurrent processes. Independent processes can be executed in parallel, each having its own physical region of program and data storage. So the transputer processor and the *OCCAM* language are ideally suited to one another.

Since the transputer was introduced by INMOS this processor has found many applications as well in universities and industry. Edinburgh is one of the most famous universities where highly concurrent computers for physics are exploited and a machine is planned containing over a thousand of T800's. To our knowledge smaller projects are under current investigation at University of Southampton, Mainz, and Bonn, essentially all planning to solve physical problems which were not accessible so far to available supercomputers. Also large supercomputers of the fifth generation are under construction on the basis of transputer networks. In industrial projects the transputer may offer new routes in areas like molecular modelling, radar signal recognition, helicopter rotor control systems, robotics and artificial intelligence based language research. These are only a few examples. At ABB Corporate Research Heidelberg the transputer was investigated for distributed real time systems in process control based on flexible multiprocessor systems. A prototype system SPOT *(System of Parallel Organized Transputers)* was developped consisting of up to 25 transputers, to investigate the possibilities for applications of this rather new technology. Therefore the architecture of the transputer based computer system SPOT was kept highly flexible, which makes it possible to reconfigurate it easily for special purposes like neural network simulations.

All computing elements are housed in a computing surface module, see FIG. 8, which consists essentially of two parts: the host environment and an assembly of standard computing elements with a capacity of 24 boards. The computing elements of different modules can be configurated together as if they were in the same module. So the module can be considered as a 200-MIPS *personal supercomputer!* The host environment consists of an integrated IBM-XT (compatible) personal computer with display and mass storage element. Furthermore it involves the INMOS standard board B004 with the master transputer and 2-Mbyte on board memory. The standard computing elements, as shown in FIG. 9 are boards, consisting of one transputer T414 or T800 with 256-Kbyte of local storage, an interface to the configurable communication network and other device support. Thus each board can be regarded as a powerful 32-bit computer in its own right. The links of the individual boards are connected together via lines on a switchboard, providing a full crossbar switch between the links of different transputers.

3.b. Software Implementation for a Neural Network Model

Our aim was to construct a simulation machine that, allows to investigate memories with following properties: content addressable, associative and fault tolerant. From our machine we expect, that it can solve wide classes of problems. Finally we demand for fast retrieval from which we conclude that the network should operate in a massively parallel fashion. In the following we investigate essentially the last point and present as an example the software implementation for Hopfield like neural network models.

The main task required for neural network simulations is the individual updating of the spins or neurons according to eqs. (6,7). First let us discuss the parallel Little-dynamic. The calculations performed for each spin are completely independent so they can be done in any order and on any number of processors. One way of distributing the work to a number of 21 processors is schematically illustrated in FIG. 10. The whole system consists of the master transputer in the host environment, 2 control transputers and 18 work transputers. Their link connections are shown in FIG. 11. As already described, neural network models are intrinsically parallel and map well onto the presented processor topology. Each work transputer needs only the data access to 1/18 of all the elements of the synaptic matrix J_{ij} and also updates 1/18 of the neurons according to

$$\sigma_{m\alpha} = g\left(\sum_{n\beta} J_{m\alpha n\beta}\,\sigma_{n\beta}\right) \quad , \tag{12}$$

where the latin letters label the processors and the greek their part of the connection matrix elements, which are needed for updating their part of neurons. So we keep in the local memory of each transputer 1/18 of the J_{ij} matrix elements and in addition the whole set of neurons.

The information flow between the workers and the two controllers is shown in FIG. 11. After updating of 1/18 of the neurons in each work transputer (a0,a1,a2, ..., f0,f1,f2) the information is sent around in each group (a,b, ..., f) over the links as shown in FIG. 11. After this we have in each work transputer group the information of 3/18 of the updated neurons. How the information flow will then proceeded, will be explained for the work transputer group (a0,a1,a2). Parallel to this the remaining 5 worker groups work in the same way. (a0) is sending his datas to the controller (C2) and to the worker (d0), (a1) and (a2) were sending to (b2), (c2) and (e1), (f1), respectively. On the other hand (a0), (a1) and (a2) get simultaneously datas from the same processors to whom they were sending their datas. After this parallel data exchange we have in each work transputer group the whole information of the updated neurons. Again sending the datas around in each group leads to the whole information available in each transputer. Worker '0' in each group has also sent 3/18 of the updated neurons either to the controller (C1) or (C2). The controllers supervise the workers, but are themselves inspected by the master transputer (M). The controllers have to process the incoming datas from the work transputers and to supply the master transputer with the results, which are managed and interpreted in the host environment. In the case of the maximum field dynamic the information flow

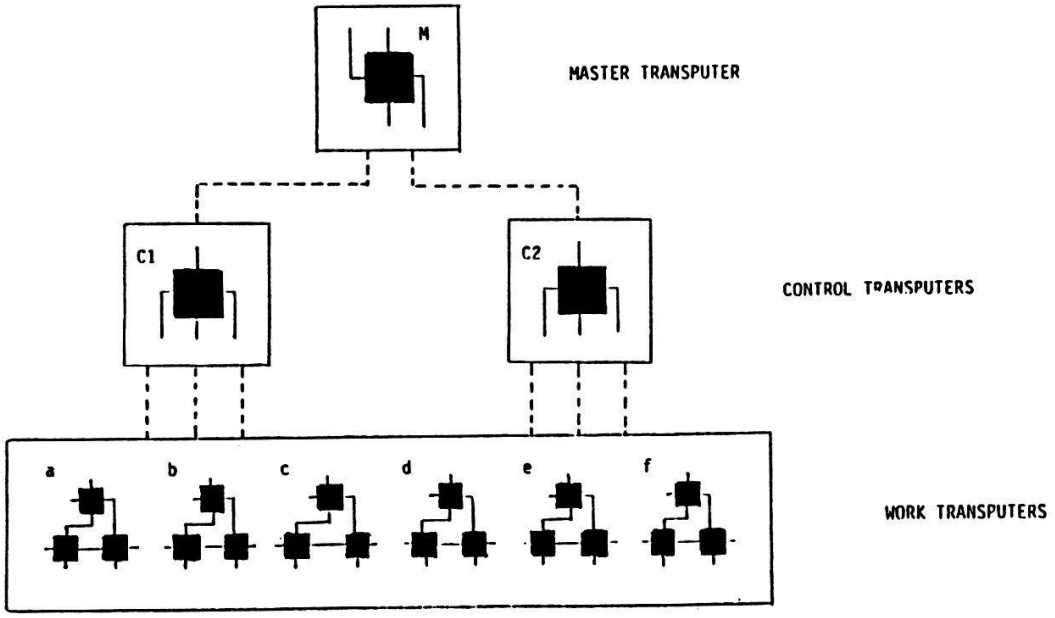

FIG. 10. Schematic illustration how the work is organized and distributed for a 21 neural network transputer array. (From D. Würtz and G. Hartung /127/)

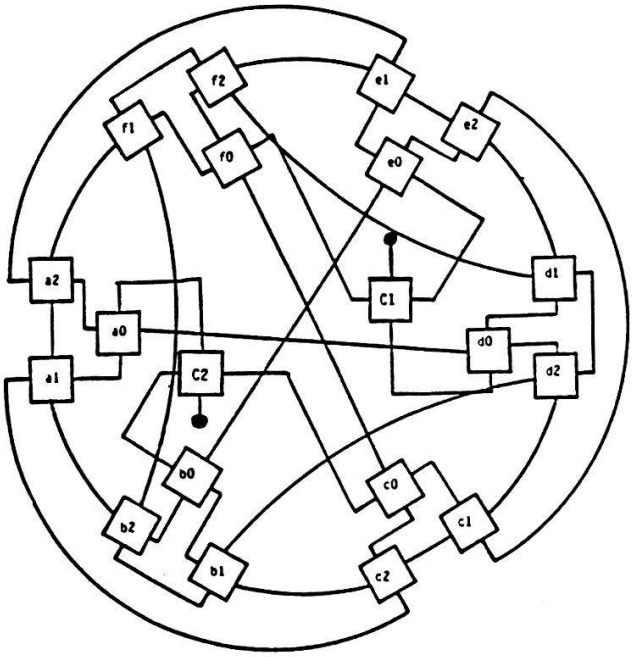

FIG. 11. Link connections for the neural network transputer array. The links with the two dots at the end are the connections to the master transputer in the host environment. (From D. Würtz and G. Hartung /127/)

is rather similar. The actual values h_i, σ_i are distributed in the same way over the work transputer array as it was done for the neurons. The maximum value of $-h_i\sigma_i$ is detected in the following way. Each worker is looking for the largest value in his table. Then the largest value is sent around in each work transputer group, where the largest value of a group is determined in each work transputer group. These values then are distributed over all transputers as already indicated and finally the largest value is sorted out. Keeping the index of the values during the data exchange we can determine the whole set of updated fields according to $h_i^{new} = h_i^{old} - 2J_{ik}\sigma_k$, where k is the index of the maximum value and σ_k the neuron changed from $\sigma_k \rightarrow -\sigma_k$. At the moment different demonstration programs simulating neural networks are available and a network for storing correlated and hierarchical organized patterns is being implemented. Results of the simulations will be presented elsewhere.

Summary

We have given an introduction to cellular automata and neural network computing. In addition hardware configuration and software implementation for a transputer based simulation machine were presented.

Acknowledgement

Discussions with the Heidelberg 'Neural Network Group', J. Bernasconi, and the 'ABB Transputer Group' were acknowledged. One of us (DW) thanks for partial support from the SFB123 (Stochastic Mathematical Models), ABB and INMOS.

References

This bibliography is a collection of books and papers which is thought as an introduction to cellular automata and neural network computing. It includes most of the literature on cellular automata, neural networks and special purpose computers that we were aware. However, the collection has to be limited, and so we apologize to those collegues we have given no reference.

Cellular Automata Computing - General references

1. J. von Neumann, *Theory of self reproducing automata*, (completed and edited by A.W. Burks), University of Illinois Press, Urbana 1966
2. E.F. Codd, *Cellular automata*, Academic Press, New-York 1968
3. A.W. Burks (ed.), *Essays on cellular automata*, University of Illinois Press, Urbana 1970

4. S. Wolfram, *Statistical mechanics of cellular automata*, Rev. Mod. Phys. 55, 601 (1983)
5. D. Farmer, T. Toffoli and S. Wolfram (eds.), *Cellular automata*, Physica 10D, nos. 1&2, North-Holland Publishing, Amsterdam 1984
6. K. Preston Jr. and M.J.B. Duff, *Modern cellular automata, theory and application* MIT Press, Cambridge 1984
7. S. Wolfram, *Twenty problems in the theory of cellular automata*, Physica Scripta T9, 170 (1985)
8. J. Demongeot, E. Goles and M. Tchuente (eds.), *Dynamical systems and cellular automata*, Academic Press, New-York 1985
9. D. Farmer, A. Lapedes, N. Packard and B. Wendroff (eds.), *Evolution, games and learning - models for adaptation in machines and nature*, Physica 22D, North-Holland Publishing, Amsterdam 1985
10. S. Wolfram (ed.), *Theory and applications of cellular automata*, World Scientific Publ., Singapore 1986
11. L. Bienenstock, F. Fogelman and G. Weisbuch (eds.), *Disordered systems and biological organization*, Springer Verlag, Berlin 1986
12. T. Toffoli and N. Margolus, *Cellular automata machines: a new environment for modelling*, MIT Press, Cambridge 1987
13. P. Rujan, *Cellular automat and statistical mechanical models*,J. Stat. Phys. 49, 139 (1987)

Game of Life

14. M. Gardner, *Mathematical Games*, Sci. Amer. 223#4, 120; 223#5, 118; 223#6, 114 (1970); 224#1, 108, 224#2, 112; 224#3, 108; 224#4, 116 (1971); 225#5, 120 (1971); 226#1, 107 (1972); 233#6, 119 (1975)
15. L.S. Schulman and P.E. Seiden, *Statistical mechanics of a dynamical system based on Conway's Game of Life*, J. Stat. Phys. 19, 293 (1978)
16. E.R. Berlekamp, J.H. Conway and R.K. Guy, *Winning ways for your mathematical plays*, Academic Press, New-York 1982, Vol. 2, Chapter 25
17. M. Gardner, *Wheels, life and other mathematical amusements*, Freeman, San Francisco 1983
18. Spektrum der Wissenschaft, *Computer Kurzweil II*, Sonderheft 1988

Dynamical systems Theory and Information Theory

19. C. Shannon and C. Weaver, *The mathematical theory of communication*, University of Illinois Press, Urbana 1949
20. P. Billingsley, *Ergodic theory and information*, Miley, New-York 1965
21. B. Mandelbrot, *Fractals: form, chance and dimension*, Freeman, San Francisco 1977
22. R. Shaw, *Strange attractors, chaotic behaviour and information flow*, Z. Naturforsch. 36a, 80 (1981)
23. H. Haken (ed.), *Evolution of order and chaos in physics, chemistry, and biology*, Springer Verlag, Berlin 1982
24. J.D. Farmer, *Dimension, fractal measures and chaotic dynamics* in: Evolution of order and chaos in physics, chemistry, and biology, H. Haken (ed.), Springer Verlag, Berlin 1982, p. 228; *Information dimension and the probabilistic structure*

of chaos, Z. Naturforsch. 37a, 1304 (1982); J.D. Farmer, E. Ott and J.A. Yorke, *The dimension of chaotic attractors*, Physica 7D, 153 (1983)
25. J.P. Crutchfield and N.H. Packard, *Symbolic dynamics of one-dimensional maps: entropies, finite precision and noise*, Int. J. Theoret. Phys. 21, 433 (1982) and *Symbolic dynamics of noisy chaos*, Physica 7D, 201 (1983)
26. M.S. Waterman, *Some applications of information theory to cellular automata*, Physica 10D, 45 (1984)
27. G. Mayer-Kress (ed.), *Dimensions and entropies in chaotic systems: quantification of complex behaviour*, Springer Verlag, Berlin 1985
28. P. Grassberger, *Toward a quantitative theory of self-generated complexity*, Int. J. Theoret. Phys. 25, 907 (1986); *How to measure self generated complexity*, Physica 140A, 319 (1986) and *Problems in quantifying self-generated complexity*, this conference
29. T.S. Parker and L.O. Chua, *Chaos: a tutorial for engineers*, Proc. IEEE 75, 982 (1987)

Formal Language Theory and Computation Theory

30. M. Minsky, *Computation: finite and infinite machines*, Prentice Hall, Englewood Cliffs 1967
31. J.E. Hopcroft and J.D. Ullman, *Introduction to automata theory, languages, and computation*, Addison-Wesley, Reading 1979
32. S. Wolfram, *Cellular automata as models of complexity*, Nature 311, 419 (1984); *Computation theory of cellular automata*, Comm. Math. Phys. 96, 15 (1984) and *Origins of randomness in physical systems*, Phys. Rev. Lett. 55, 49 (1985)

Undecidability and Gödel's Theorem

33. D.R. Hofstadter, *Gödel, Escher, Bach: an eternal golden braid*, Basic Books, New York 1979
34. G. J. Chaitin, *Gödel's theorem and information*, Int. J. of Theoret. Phys. 21, 941 (1982)
35. S. Wolfram, *Computer software in science and mathematics*, Sci. Amer. September 1984, 140 (1984) and *Undecidability and intractability in theoretical physics*, Phys. Rev. Lett. 54, 735 (1985)
36. P. Weibel and E. Köhler, *Gödel's Unentscheidbarkeitsbeweis: Ideengeschichtliche Konturen eines berühmten mathematischen Satzes*, in: Gödel-Satz, Möbius-Schleife und Computer-Ich, F. Deuticke Verlag, Wien 1986, p. 72

Ising Dynamics Simulation by CA Evolution, Q2R-Automaton

37. G.Y. Vichniac, *Simulating physics with cellular automata*, Physica, 10D, 96 (1984)
38. Y. Pomeau, *Invariant in cellular automata*, J. Phys. A17, L415 (1984)
39. J. Viñals and J.D. Gunton, *Pattern formation in reversible cellular automata*, J. Phys. A19, L933 (1986)
40. E. Goles and G.Y. Vichniac, *Invariants in automata networks*, J. Phys. A19, L961 (1986)
41. G.Y. Vichniac, *Cellular automata models of disorder and organization*, in: Disordered Systems and Biological Organization, Springer Verlag, Berlin 1986

42. H.J. Herrmann, *Fast algorithm for the simulation of Ising models,* J. Stat. Phys. 45, 145 (1986); H.J. Herrmann, H.O. Carmesin and D. Stauffer, *Periods and clusters in Ising cellular automata,* J. Phys. A20, 4939 (1987) and W.M. Lang and D. Stauffer, *Test of 3-dimensional Q2R Ising algorithm,* J. Phys. A20, 5413 (1987)
43. M. Schulte, W. Stiefelhagen and E.S. Demme, *Period in the chaotic case of Q2R automata,* J. Phys. A20, L1023 (1987)
44. J.G. Zabolitzky and H.J. Herrmann, *Multitasking case study on the Cray-2: The Q2R algorithm,* Preprint (1988)

CREUTZ-Automaton

45. M. Creutz, *Microcanonical Monte Carlo simulation,* Phys. Rev. Lett. 50, 1411 (1983); G. Bhanot, M. Creutz and H. Neuberger, *Microcanonical simulation of Ising systems,* Nucl. Phys. B235, 417 (1984); M. Creutz, P. Mitra and K.J.M. Moriarty, *A fast algorithm for investigations on the three-dimensional Ising model,* Comp. Phys. Comm. 33, 361 (1984) and M. Creutz, A. Gocksch and M. Ogilvie, *Microcanonical renormalization group,* Phys. Rev. Lett. 53, 875 (1984)
46. R. Harris, *A study of first- and second-order phase transitions using Monte Carlo simulations in the micro-canonical ensemble,* Phys. Lett. 111A, 299 (1985)
47. M. Creutz, *Deterministic Ising dynamics,* Ann. of Phys. NY 167, 62 (1986)
48. T. Toffoli and N. Margolus, *Cellular automata machines,* MIT Press 1987, Chapter 17
49. D.W. Heerman and R.C. Desai, *Ising model, transputers and dynamical correlations using a microcanonical ensemble,* Comp. Phys. Comm. 50, 297 (1988)
50. R.C. Desai, D.W. Heermann and K. Binder, *Finite size scaling in a microcanonical ensemble,* Preprint 1988

see als reference 12)

For Monte Carlo Simulations see e.g.

51. N. Metropolis, A.W. Roesenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller, *Equation of state calculations by fast computing machines*, J. Chem. Phys. 21, 1087 (1953)
52. K. Binder (ed.), *Monte Carlo methods in statistical physics*, Springer Verlag, Berlin 1979 and *Applications of the Monte Carlo method in statistical physics*, Springer Verlag, Berlin 1984
53. M.H. Kalos and P.A. Whitlock, *Monte Carlo methods* Wiley, New-York 1986

Hydrodynamics Simulation by CA Evolution

54. J. Hardy, Y. Pomeau and O. de Pazzis, *Time evolution of a two-dimensional model system - Invariant states and time correlation functions*, J. Math. Phys. 14, 1746 (1973) and J. Hardy, O. de Pazzis and Y. Pomeau, *Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions*, Phys. Rev. A13, 1949 (1976)
55. D. d'Humières, Y. Pomeau and P. Lallemand, *Simulation d'allées de von Karman bidimensionelles a l'aide d'un gaz sur réseau*, C.R. Acad. Sc. Paris, t301, 1391 (1985)
56. J.-P. Rivet and U. Frisch, *Automates sur gaz de reseau d'ans l'approximation de Boltzmann*, C. R. Acad. Sc. Paris, t302, 267 (1986)

57. U. Frisch, B. Hasslacher and Y. Pomeau, *Lattice gas automata for the Navier Stokes equation*, Phys. Rev. Lett. 56, 1505 (1986)
58. D. d'Humieres and P. Lallemand, *Lattice gas automata for fluid mechanics*, Physica 140A, 326 (1986)
59. J. Salem and S. Wolfram, *Thermodynamics and hydrodynamics with cellular aotomata*, in: Theory and Applications of Cellular Automata, S. Wolfram (ed.), World Scientific, Singapore 1986, p. 362
60. S.A. Orszag and V. Yakhot, *Reynolds number scaling of cellular-automation hydrodynamics*, Phys. Rev. Lett. 56, 1691 (1986
61. D. d'Humieres, P. Lallemand and U. Frisch, *Lattice gas models for 3D hydrodynamics*, Europhys. Lett. 2, 291 (1986)
62. S. Wolfram, *Cellular automaton fluids: Basic theory*, J. Stat. Phys. 45, 471 (1986)
63. V. Yakhot, B. Bayley and S. Orszag, *Analogy between hyperscale transport and cellular automaton fluid dynamics*,
64. D. d'Humières and P. Lallemend, Complex Systems 1, 599 (1987)
65. A.J.C. Ladd, M.E. Colvin and D. Frenkel, *Application of lattice gas cellular automata to the Brownian motion of solids in suspension*, Phys. Rev. Lett. 60, 975 (1988)
66. P. Clavin, P. Lallemend, Y. Pomeau and G. Searby, *Simulation of free boundaries in flow systems by lattice gas models*, J. Fluid Mech. 188, 437 (1988)
67. H.A. Lim, *Lattice gas automata of fluid dynamics for unsteady flow*, Comlex Systems 2, 45 (1988)

For other applications we refer the reader to the bibliography given in reference 7.) and to references given the paper

68. M. Droz and B. Chophard, *Cellular automata approach to physical problems*, Helvetica Physica Acta 61, 801 (1988)

Neural Network Computing - General references

69. D.O. Hebb, *The organization of behaviour*, Wiley, New-York 1949
70. J. von Neumann, *Die Rechenmaschine und das Gehirn*, Oldenbourg Verlag, München 1960
71. F. Rosenblatt, *Principles of Neurodynamics*, Spartan, New-York 1962
72. N. Nilson, *Learning machines*, MIT Press, Cambridge 1965
73. M.L. Minsky and S. Papert, *Perceptrons*, MIT Press, Cambridge 1969
74. B. and S. Lundquist (eds.), *Collective properties of physical systems*, Academic Press, New York 1974
75. T. Kohonen, *Content addressable memories*, Springer Verlag, Berlin 1980
76. G.E. Hinton and J.A. Anderson (eds.), *Parallel models of associative memory*, Erlbaum, Hillsdale 1981
77. S. Grossberg, *Studies of Mind and Brain*, Reidel Publ., Dordrecht 1982
78. T. Kohonen, *Self organization and associative memory*, Springer Verlag, Berlin 1984
79. J.L. van Hemmen and I. Morgenstern (eds.), *Heidelberg colloquium on glassy dynamics*, Springer Verlag, Berlin 1986
80. D.E. Rumelhart and J.L. McClelland (eds.), *Parallel distributed processing: explorations in the microstructure of cognition*, Vol. 1&2, MIT Press, Cambridge 1987

81. T. Hogg and B.A. Huberman, *Artificial intelligence and large scale computation: a physics perspective*, Physics Reports 156, 228 (1987)
82. Proceedings of the IEEE first international conference on neural networks, Vol. I-IV, San Diego 1987
83. G.A. Carpenter and S. Grossberg (eds.), Applied Optics 26, No. 23, December 1987, special issue on neural networks
84. E.R. Caianiello (ed.), *Physics of cognitive processes*, Amalfi 1986, World Scientific, Singapore 1987
85. S. Grossberg, *Neural Networks and Natural Intelligence*, MIT Press, Cambridge 1988
86. C. Kemke, *Der Neuere Konnektionismus*, Informatik Spektrum 11, 143 (1988)
87. J.A. Anderson and E. Rosenfeld (eds.), *Neurocomputing: a reader*, MIT Press, Cambridge 1988
88. *Neural Networks*, a new quarterly journal, began publication in january 1988

Hopfield Model and Spin Glass Analogy

89. W.A. Little, *Existence of persistent states in the brain*, Math. Biosc. 19, 101 (1974)
90. J.J. Hopfield, *Neural networks and physical systems with emergent collection computational abilities*, Proc. Natl. Acad. Sc. USA 79, 2554 (1982) and *Neurons with graded response have collective computational properties like those of two-state neurons*, Proc. Natl. Acad. Sc. USA 81, 3088 (1984)
91. P. Peretto, *Properties of neural networks, a statistical physics approach*, Biol. Cybern. 50, 51 (1984)
92. J.J. Hopfield and D.W. Tank, *Neural computation of decisions in optimization problems*, Biol. Cybern. 52, 141 (1985) and *Computing with neural circuits: a model*, Science 233, 625 (1986)
93. D.J. Amit, H. Gutfreund and H. Sompolinsky, *Spin glass models of neural networks*, Phys. Rev. A32, 1007 (1985) and Phys. Rev. Lett. 55, 1530 (1985); *Information storage in neural networks with low levels of activity*, Phys. Rev. A35, 2293 (1987) and *Statistical mechanics of neural networks near saturation*, Ann. Phys. N.Y. 173, 30 (1987)
94. W. Kinzel, *Learning and pattern recognition in spin glass models*, Z. Phys. B60, 205 (1985)
95. L. Personnaz, I. Guyon and G. Dreyfus, *Information storage and retrieval in spin-glass like neural networks*, J. Physique Lett. 46, L359 (1985)
96. V.S. Dotsenko, *Hierarchical model of memory*, Physica 140A, 410 (1986) and J. Phys. C18, L1017 (1985)
97. J.P. Nadal, G. Toulouse, J.P. Changeux and S. Dehaene, *Networks of formal neurons and memory palimpsests*, Europhys. Lett. 1, 535 (1986), G. Toulouse, S. Dehaene and J.P. Changeux, *Spin glass model of learning by selection*, Proc. Natl. Acad. Sc. USA 83, 1695 (1986)
98. J.L. vanHemmen and R.Kühn, *Nonlinear neural networks*, Phys. Rev. Lett. 57, 913 (1986); J.L. vanHemmen, *Spin glass models of a neural network*, Phys. Rev. A35, 3435 (1986); *Nonlinear neural networks near saturation*, Phys. Rev. A36, 1959 (1987); J.L. vanHemmen, D. Grensing, A. Huber and R. Kühn, *Nonlinear neural networks: I General Theory, II Information processing*, SFB Preprints Nos. 411&412 (1987); S. Bös, R. Kühn and J.L. vanHemmen, *Martingale approach to*

neural networks with hierarchically structured information, SFB Preprint No. 435 (1987)
99. N. Parga and M.A. Virasoro, *The ultrametric organization of memories in a neural network*, J. Physique 47, 1857 (1986)
100. S. Diederich and M. Opper, *Learning of correlated patterns in spin-glass networks by local learning rules*, Phys. Rev. Lett. 58, 949 (1987)
101. G. Pöppel and U. Krey, *Dynamical Learning process for recognition of correlated patterns in symmetric spin-glass models*, Europhys. Lett. 4, 979 (1987)
102. M.V. Feigelman and L.B. Ioffe, *The augmented models of associative memory asymmetric interaction and hierarchy of patterns*, Int. J. Mod. Phys. B1, 51 (1987)
103. F. Fogelman-Soulie and G. Weisbuch, *Random iterations of threshold networks and associative memory*, SIAM J. Comput. 16, 203 (1987)
104. H. Horner, *Spingläser und Hirngespinste: Physikalische Modell des Lernens und Erkennens*, Phys. Bl. 44, Nr. 2, 29 (1988)
105. B. Derrida, *Statistical mechanics of neural networks*, this conference

Multilayer networks

106. D.E. Rumelhart and D. Zipser, *Feature discovery by competitive learning*, Cogn. Sci. 9, 75 (1985); D.E. Rumelhart, G.E. Hinton and R.J. Williams, *Learning internal representations by error propagation*, in: Parallel distributed processing, MIT Press, Cambridge 1986 and *Learning representations by backpropagating errors*, Nature 323, 533 (1986)
107. T.J. Sejnowski, P.K. Kienker and G.E. Hinton, *Learning symmetry groups with hidden units: beyond the perceptron*, Physica 22D, 260 (1986)
108. B. Widrow and R. Winter, *Neural nets for adaptive filtering and adaptive pattern recognition*, Computer, March 1988, 25 (1988)
109. F. Fogelman-Soulie, *Backpropagation learning*, in: Proc. of the Workshop on chaos and complexity, Torino 1987, World Scientific, Singapore 1988
110. R.P. Gorman and T.J. Sejnowski, *Analysis of hidden units in a layered network trained to classify sonar targets*, Neural Networks 1, 75 (1988)
111. S.I. Gallant, *Connectionist expert systems*, Commun. ACM 31, 152 (1988)
112. J. Bernasconi, *Analysis and comparison of different learning algorithms for pattern association problems*, IEEE Conf. on neural information processing systems, Denver 1987, AIP Conf. Proc. 1988, and *Learning algorithms*, this conference
see also reference 73)

Special Purpose Computers

113. T. Toffoli and N. Margolus, *The CAM-7 multiprocessor: a cellular automata machine*, MIT/LCS/TM-289 Preprint 1985
114. H.J. Herrman, *Special purpose computers in statistical physics*, Physica 140A, 421 (1986)
115. R. Hauser, A. Genthner, H. Horner, R. Lange and R. Männer, *NERV - Ein Simulationssystem für neuronale Netzwerke*, Preprints 1988
116. R. Hecht-Nielsen, *Neurocomputing: picking the human brain*, IEEE Spectrum, March 1988, 36 (1988)
see also references 12 and 127)

Transputer - Hardware and Software

117. P. Eckelmann, F.D. Kübler, O. Krämer, W. Hahn, articles in: *Transputer*, Chip plus, Beilage August 1988
118. R. Steinmetz, *Occam 2*, Hüthig Verlag, Heidelberg 1987
119. *Edinburgh Concurrent Supercomputer Newsletter,*, #1 May 1987, #2 August 1987, #3 November 1987
120. *Product informations*, INMOS Gmbh, Dachau/München; Meiko Ltd, Bristol; Parsytec GmbH, Achen
121. J.Ch. Jesshope (ed.), *Applying the transputer,* forthcoming special issue of 'Micropressors and Microsystems'

Transputer - Applications in Physics

122. C.R. Askew, D.B. Carpenter, J.T. Chalker, A.J.G. Hey, D.A. Nicole and D.J. Pritchard, *Simulation of statistical mechanical systems on transputer arrays*, Comp. Phys. Comm. 42, 21 (1986)
123. K.C. Bowler, A.D. Bruce, R. Kenway, G.S. Pawley and D.J. Wallace, *Exploiting highly concurrent computers for physics*, Physics Today, Oct. 1987, 40 (1987)
124. A. Johaunet, G. Loheac, L. Personnaz, I. Guyon and G. Dreyfus, *A transputer-based neurocomputer*, Proc. of the 7th OCCAM Users Group Meeting, Grenoble (1987)
125. C.R. Askew, D.B. Carpenter, J.T. Chalker, A.J.G. Hey, M. Moore, D.A. Nicole and D.J. Pritchard, *Monte Carlo Simulation on transputer arrays*,
126. B.M. Forrest, D. Roweth, N. Stroud, D. J. Wallace and G.V. Wilson, *Implementing neural network models on parallel computers*, The Computer Journal 30, 413 (1987)
127. D. Würtz and G. Hartung, *Neural computing on a system of parallel organized transputers: software implementation and hardware configuration*, Comp. Phys. Comm. (in press)