N87-26540

NEURAL NETS ON THE MPP

Harold M. Hastings Stefan Waner
Department of Mathematics
Hofstra University
Hempstead, NY 11550 USA

ABSTRACT

The Massively Parallel Processor (MPP) is an ideal machine for computer experiments with simulated neural nets as well as more general cellular automata. The purpose of this paper is to describe our experiments using the MPP with a formal model neural network. Our results on problem mapping and computational efficiency apply equally well to the neural nets of Hopfield, Hinton et al., and Geman and Geman.

Keywords: Neural Network, Annealing System, Stochastic Cellular Automaton

INTRODUCTION

This paper is a preliminary report on a major component of the research proposal of M. Conrad and the authors, entitled "Applications of stochastic and reaction-diffusion cellular automata." These types of automata are a natural formal setting for theoretical investigations in brain and ecosystem modelling. Most of the proposal was concerned with brain modelling. A significant part of the proposed activity in that area has been completed, and will be discussed here.

Hastings and Pekelney (Ref. 4), observed that many of the properties of the brain seemed to be natural consequences of the working hypothesis that the brain was a large network of McCulloch-Pitts neurons (threshold devices) connected by synapses with stochastic conduction thresholds. In particular, such networks display both gradualism (small changes in inputs cause small changes in outputs (Ref. 1)) and modification-based learning (structural changes as a result of history, Conrad, Ref. 2).

Later, the authors developed a model neural network, implemented the network on a VAX11-780, and conducted experiments in basic learning principles. They also defined (Ref. 5) three postulates which characterized evolutionary learning (for example, by simulated or real neural networks).

Evolutionary Learning

An evolutionary learning system is a formal dynamical system in which the states correspond to modes of information processing, while the suitability of each state is measured by a potential function, the most desirable states possessing least potential. The dynamics of such a system are determined by an annealing process (Refs. 8-9), so that desirable modes are attained by a gradual lowering of the amount of thermal noise.

(The prototype example of an annealing process consists of a gas molecule confined in a potential well, in which the goal is the location of a global potential minimum. If the ambient temperature is lowered sufficiently slowly, the molecule will become trapped in the global minimum with a probability arbitrarily close to one. It is the random behavior of the molecule, "thermal noise," which accounts for its ability to escape from local minima during the cooling process. Simulated annealing then entails simulation of these dynamics in the solution of combinatorically large-scale minimization problems such as the "travelling salesperson"

problem. The essential role played by random noise in such techniques places them outside the realm of algorithmic strategies.)

Further, the potential energy function depends on the environment, so that it is the environment which indirectly determines the equilibria and evolution of the system. We refer to this indirect process of control as soft programming. An evolutionary learning system may then be thought of as a dynamical system which behaves according to three principles:

ergodicity - the use of chaotic behavior to search a state space,

annealing - the regulation of thermal noise by means of (local) lowering of ambient temperature, and

soft programming - the indirect control of the evolution of the system by the environment.

More complex learning regimes were shown to follow the same basic principles (Waner and Hastings, Ref. 10). We also remark that gradualism in annealing systems is a consequence of the annealing dynamics: small changes in the starting point or the shape or potential surface usually cause small changes in the dynamics. Modification of the potential surface through feedback in learning corresponds to Conrad's modification based learning. The annealing dynamics are considered to be internal and inaccessible in detail compared to the feedback dynamics of any learning scheme.

In late 1985 and early 1986 the neural network programs were transported to the MPP. The relationship between the theoretical dynamics and the neural net models will be briefly discussed below.

The rest of this paper is divided into three main parts. The first of these summarizes our neural network models. The second part summarizes our experiments to date on the MPP, and should be understood in the context of a preliminary report. The last part describes conclusions for the application of the MPP and similar massively parallel architectures for our model and similar models. Most of our qualitative conclusions may be readily applied to other neural networks (Hopfield (Ref. 7), Hinton, Sejnowski, and Ackley (Ref. 6), Geman and Geman (Ref. 3)).

THE NEURAL NETWORK

In this section we describe the data structures and algorithms used in our neural network, and briefly describe the dynamics.

Data Structures

The fundamental data structure is a directed graph in which nodes correspond to formal neurons, and arrows to formal synapses. Early experiments on a VAX used a rectangular array of neurons, with nearest neighbor and second-nearest neighbor connections. This structure suggested a natural problem mapping to the MPP. The MPP model uses a 128 x 128 array of formal neurons, with connections to all neighbors in a 5 x 5 array centered at each neuron. This data structure also accords well with a $2+\varepsilon$ -dimensional structure for random access in the brain (see Ref. 4 for discussion).

The formal neurons are McCulloch-Pitts neurons. Each contains one or more inputs, has a fixed firing threshold, and fires (sends an output) if and only if the sum of inputs since the last firing is greater than or equal to the firing threshold. The sum of these inputs is called the activity of the neuron; on firing the activity is reset to 0.

The synapses are also threshold devices. Their associated thresholds are called conduction thresholds. However, there are two important differences between the use of thresholds of synapses and those of neurons. First, the conduction threshold of a synapse determines the probability of conduction along that synapse according to the rule prob(conduction) = 1 - (conduction threshold). Second, the conduction thresholds are modified according to two rules:

LEARNING BY REPETITION. Thresholds of synapses which conduct (and similar synapses) are lowered. Conduction thresholds of synapses which do not conduct (and similar synapses) are raised. In the presence of suitable learning regimens, this yields Conrad's modification based learning. The threshold modification scheme must be constructed carefully to minimize the chance of positive feedback in the internal dynamics.

LONG-TERM FORGETTING. Most conduction thresholds (all thresholds except those very near 1 or very near 0) slowly decay to a base value. Thresholds sufficiently close to 0 or 1 do not change; this corresponds to modification-based learning.

Annealing System

Recall that the temperature in an annealing system corresponds to the degree of randomness. In this sense, the entropy (Shannon information) associated with random behavior at all of the synapses corresponds to the temperature. When following a learning regimen results in reducing this entropy, this corresponds to a reduction in temperature. The use of random conduction along synapses yields the underlying diffusion in an annealing system; restricting the dimension to 2 or using a wrap-around topology would guarantee ergodicity. For all practical purposes the present system appears ergodic. Differences in thresholds yield drift terms corresponding to the gradient flow part of annealing. These differences and consequent drift become more pronounced as learning via "annealing through modes of information processing" proceeds.

We remark that classical annealing problems (Ref. 9) can be readily programmed on the MPP with an analogous problem mapping of one cell per processor.

Soft Programming

Soft programming consists of specifying the learning goals. Three types of goals have been studied theoretically. The simplest consists of structured path learning, or learning paths from "sources" to "targets." The MPP program below illustrates this case. More complex cases include associative learning and more complex route-finding (only theory so far).

Problem Mapping

We have allocated one processor to each node. This offers the advantages of simple data flow and programming, at the expense of frequently having idle processors in simple experiments.

The Program

NEURAL NET: BEGIN

1. INITIALIZE
   Net: initialize thresholds
   initialize activities
   initialize source and
   target, or learning problem

   Supervisory: initialize random
   number generator, clocks,
   maximum time allowed, etc.

   Learning regimen: specify.

2. MAIN LOOP: REPEAT until timeout or learning occurs. Increment random number generators as necessary throughout loop.

   AT each neuron: IF activity is
   greater than or equal to
   firing thresholds, THEN
   FIRE and reset activity to 0.

   AT each synapse, IF neuron at
   tail of synapse has fired, THEN
   TEST for conduction:
   synapse conducts if random
   number is greater than
   conduction threshold.

   IF synapse conducts, THEN
   increment activity of
   neuron at head of synapse.

   AT each synapse, modify
   conduction thresholds incor-
   porating learning by
   repetition and short-term
   forgetting.

   Did net learn? If so, then
   exit loop and print results.

   Increment clock.

END MAIN LOOP

3. OUTPUT

END NEURAL NET

One should note that the net is intrinsically parallel and stochastic. The parallel feature of the net allows a natural problem mapping: one maps one formal neuron to each processor. Other mappings are possible; for example, one could map each processor to one neuron in a neighborhood of a given neuron, and store the net in an appropriate data structure for transversal. The problem mapping we used was chosen for its simplicity, and potential to reduce the size of the program and maximize computing speed. For example, the MPP program is approximately 20-30% shorter than the VAX program, and both are programmed in similar high-level languages.

The MPP program also ran significantly faster than that of the VAX. The present improvement factor in simple experiments is about 100. However, the MPP does not slow down as the number of neurons firing is increased. This combined with a utilization factor in critical steps of about 5% in simple experiments suggests a relative speed increase in complex tasks should approach 2000.

CONCLUSIONS

Massively parallel architectures are especially appropriate and useful for neural network and similar simulations. In particular, the geometry of the MPP closely parallels the structure of our net model. This places much of the data structure in hardware, reducing computational costs. In addition, much of the computation is "strongly parallel" in the sense that the next state computations must take place simultaneously at many locations. Failing this, data structures and data movement must be developed to simulate this degree of parallelism. Furthermore, most of the VAX computation cost apparently lies in data movement, since no elaborate

function evaluations are needed. This contrasts sharply with both algorithms such as Gaussian elimination in which such tight parallelism is not necessary, and algorithms such as many finite element algorithms in which such parallelism is necessary (at least at a simulation level), but in which significant function evaluation costs far exceed data movement costs.

Much of our computing time is spent in random number generation. We are exploring the possibility of realizing random number generators or more general stochastic gates in VLSI hardware. Should this exploration prove successful, it would be possible to construct simple, rapid, and powerful evolutionary learning hardware.

Otherwise, the limited processor power and memory do not slow this type of modelling. In fact, the MPP architecture may well offer the best relative performance because much of the data structure and flow is already present in hardware.

FUTURE DIRECTIONS

At this point, the first part of our proposed research has been largely completed. We have largely developed the theory for applying our learning model to the route-finding problem, and should begin MPP investigations into this problem in early 1987. The extension of these models to more complex (reaction-diffusion) neurons will be done largely by M. Conrad with his former student K. Akingbehin. In a related direction, some successful ecosystem simulations have been performed with more work expected later this year.

REFERENCES

1. M. Conrad, "Evolutionary Learning Circuits," J. Theoret. Biol., 1974, 46, pp. 167-188.

2. M. Conrad, "Complementary Molecular Models of Learning and Memory," BioSystems, 1976, 8, pp. 119-138.

3. S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," Brown University Preprint, 1983.

4. H.M. Hastings and R. Pekelney, "Stochastic Information Processing in Biology," BioSystems, 1982, 15, pp. 155-168

5. H.M. Hastings and S. Waner, "Principles of Evolutionary Learning - Design for a Stochastic Neural Network," BioSystems, 1985, 18, pp. 105-109.

6. G.F. Hinton, T.J. Sejnowski, and D.H. Ackley, "Boltzmann Machines: Constraint Satisfaction Networks that Learn," Cognitive Science, 1984.

7. J.J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Properties," Proc. Nat. Acad. Science USA, 1982, 79, pp. 2554-2558.

8. S. Kirkpatrick, C.D. Gelatt, Jr., and M.P. Vecchi, "Optimization by Simulated Annealing," Science, 1983, 220, pp. 671-680.

9. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equations of State Calculations by Fast Computing Machines," J. Chem. Phys., 1953, 21, pp. 1087-1091.

10. S. Waner and H.M. Hastings, "Evolutionary Learning of Complex Modes of Information Processing," in Advances in Cognitive Science, ed. M. Kochen and H.M. Hastings, Amer. Assoc. for the Advancement of Science Selected Symposia, in press.