# A Discrete Neural Network Model for Conceptual Representation and Reasoning

Ron Sun
Computer Science Dept.
Brandeis University
Waltham, MA 02254

## ABSTRACT

Current connectionist models are oversimplified in terms of the internal mechanisms of individual neurons and the communication between them. Although connectionist models offer significant advantages in certain aspects, this oversimplification leads to the inefficiency of these models in addressing issues in explicit symbolic processing, which is proven to be essential to human intelligence. What we are aiming at is a connectionist architecture which is capable of simple, flexible representations of high level knowledge structures and efficient performance of reasoning based on the data. We first propose a discrete neural network model which contains state variables for each neuron in which a set of discrete states is explicitly specified instead of a continuous activation function. A technique is developed for representing concepts in this network, which utilizes the connections to define the concepts and represents the concepts in both verbal and compiled forms. The main advantage is that this scheme can handle variable bindings efficiently. A reasoning scheme is developed in the discrete neural network model, which utilizes the inherent parallelism in a neural network model, performing all possible inference steps in parallel, implementable in a fine-grained massively parallel computer.

## 1. INTRODUCTION

The advances in neurobiology and connectionist modeling provide a whole set of possibilities in terms of implementing and extending AI ideas in conceptual representation and reasoning. Current connectionist models are only crude approximations of the real neural network. They are oversimplified in terms of the internal mechanisms of individual neurons and the communication between neurons. This oversimplification leads to the failure and/or inefficiency (at least as I have seen so far) of the models in addressing issues such as modeling biological neural networks([Selverston 1987]), representing high level data structures, rules, or concepts, developing inferential schemes, and extending the application domain of the connectionist models to domains involving symbolic processing (see [Pinker and Mehler 1988] for the necessity of explicit symbolic processing). What we are aiming at is a computational neural network model which is capable of simple, flexible representations of high level data structures and efficient performance of reasoning based on the data, i.e. an AI architecture based on a neural network model, drawing ideas from biological mechanisms in real neural networks.

Little progress has been made toward such an architecture. Among the few works that are reported are [Touretzky and Hinden 1987], [Touretzky 1986], [Barnden 1988], [Shastri and Feldman 1987], and [Smolensky 1987]. But in each of these schemes, parallelism is lost in some way, either because of the matching process of harhwired rules or a centralized working memory. For example, in [Touretzky and Hinton 1987], an elaborate pull-out network is designed to pick up a rule from a rule network and to match the data (triples) in the working memory. Although the mechanism is very elegant, it hinders the speed of reasoning by doing one match at a time. And it is possible to travel deep down a wrong path. In [Barnden 1988] scheme, the rules are wired in symbolic forms into a network in a grid form. Thus the problem is the symbolic manipulation necessary to match the rule against data, which is a slow and complicated process in a connectionist model. Because of that, only one rule can be matched at a time. The inherent parallelism is not fully utilized as a result. In [Shastri and Feldman 1987], a mathematical formalism is developed, and a network architecture is designed to implement the formalism. Many different types of neurons are devised and each has a special activation function specifically designed for that neuron. The scheme can handle property inheritance in a conceptual hierarchy but not rule encoding and rule based reasoning. Besides these, there are other shemes that employ different techniques for high level data or knowledge representation, for example, [Ballard 1986], [Fanty 1988], [Ackley 1987], and [Derthick 1988].

## 2. THE DISCRETE MODEL

Our aim is to devise a model more general than conventional PDP models and capable of explaining many intricate phenomena found in real neural networks. The generalization goes along several dimensions: internal states, different synaptic outputs, and temporal response. The resulting model can be used to attack several important problems in developing a reasoning scheme, i.e. rule matching, variable binding, and certainty factor propagation. The model seems to be a reasonable basis for an inference system and it is presented here as a first step towards a full fledged conceptual representation and reasoning system.

Basically, a discrete neural model is a 2-tuple

$W = <N, M>$

where

$N = \{<S, A, I, IF, T, C>\}$

S= the set of all the possible states of a neuron,
A= the set of all the actions to be taken by the neuron,
I= inputs,
IF= input manipulation function: $I \dashrightarrow I'$,
T= State transition function: $S \times I' \dashrightarrow S$,
C= action function: $S \times I' \dashrightarrow 2^A$,

and M is the connectivity among neurons in the set N.

In this model a set of discrete states is explicitly specified instead of a continuous activation function. Hopefully this can capture more accurately the biological information processing mechanisms built into a real neuron. The idea came from the modeling study of lobster stomatogastric ganglion neural networks (see [Sun et al. 1988]). Evidence from physiological data observed by biologist overwhelmingly points to a more powerful neural network model, which is capable of accounting for more phenomena than conventional models. In a real neuron, unlike in conventional connectionist models, there is no continuous input or output through synapses. Instead, an all-or-none action potential is generated if the cell is depolarized to a certain degree, which in turn causes the release of neurotransmitters. The input to the postsynaptic cell is dependent upon two factors: the type and the amount of neurotransmitters released ([Kandel and Schwartz 1984] and [Edelman 1987]). This powerful mechanism can not be captured by conventional neural network models. In a conventional neural network model, the continuous output is meant to represent the frequency in which the action potentials are generated, it is doubtful that the firing frequency is a primitive feature (not an emergent feature that is caused by other more primitive activities) in the neuronal information processing mechanism. On the contrary, we have shown in a simulation study [Sun et al 1988] that the firing frequency, as well as phase relationship, is an emergent property of the network created by the complex interaction of the components of the network, at least in lobster stomatogastric ganglions. The proposed discrete model can easily capture the neural information processing mechanisms through action functions and state transition functions by specifying a sequence of states to go through, and specifying actions associated with each state, namely

$s(t) = s(t\text{-}1) + 1 \bmod n$ ,
$C(t) = f(s(t), I1, I2, \ldots, Ik)$ ,
where f is a predetermined function such as weighted sum or Goldman Equation.

This formalism can explain many intricate phenomena found in real neural networks such as phase reponses, neuronal modulation and phasic relationship. These properties are important in terms of the functional capability and versitility of a network, as seen in many different domains (e.g. [Richmond & Optican 1987]).

Another issue is the importance of the membrane properties and, therefore, the endogenous firing of individual cells. According to our study [Sun et al 1988], the dynamics and emergent properties of a neural network can mostly be attributed to two factors: the endogenous firing (determined by membrane properties of the cell, which could be affected by current inputs and the input history) and the synaptic connectivity. Because of the physiological properties of the cell membrane, each cell is capable of firing endogenuously even when it is insulated from any external influence. The endogenous firings are important as a source of influences that help to shape the behavior of a network. This fact is indicated in many biological papers (e.g. [Selverston and Moulin 1987]). However, the importance of membrane properties and endogenous firings is overlooked in conventional connectionist models, because of the highly approximate nature of these models. In the discrete neural model, this feature can be captured by a state variable that represents the particular moment of internal changes. The mechanism works this way: using the formula specified above, s(t) now determines a particular endogenous firing curve, for example,

Suppose the weighted sum model is used (see [Sun et al 1988]),
C(t) = f(s(t), I1, I2, .... ,Ik) = w0*E(s(t)) +w1*I1(t) +w2*I2(t) + ....... +wk*Ik(t) ,
where E(s(t)) = sin(s(t)) .

Yet another issue is the different presynaptic actions performed by the same cell at differnet sites of the axon. Different sites on the same axon can release different types of neurotransmitter (thus cause different types of reactions in postsynaptic cells) or different amount of transmitters of the same type. Some types of neurotransmitters may have long lasting effect, while others may act instantaneously. Each can cause a different reaction in a postsynaptic cells. The "action" taken by an individual postsynaptic cell is determined mainly, but not exclusively, by the following factors: the endogenuous properties of the cell, the type(s) and amount of transmitters it received, and the current that is injected into it in case of electric synapsis. The issue of different postsynaptic actions is not dealt with in conventional connectionist models either. In my model, the variaty in presynaptic actions can be modeled by A (the set of actions) and C (the action functions).

The equivalence property of this model to the more conventional models is studied. It is at least capable of the same computational power as well as expressive power. Beyond that, it has the advantage of generality and versatility. It is more general because it can accormodate the conventional connectionist models as special cases as discussed below. It is also more versatile because, by introducing state variables and a set of synaptic actions, the model can handle more elaborate processing at neuronal level.

To see how my model simulates other connectionist models, look at various neural network models. In general, neural network models can be classified into four classes:

continuous input/discrete activation models (e.g. linear threshold unit model),

discrete input/discrete activation models (e.g. Feldman and Ballard model),

continuous input/continuous activation models (e.g. McClelland and Rumelhart's interactive activation and competition model),

discrete input/continuous activation models (as another possibility).

All of them can be easily handled by my general formalism. To simulate a continuous input/discrete activation neural model (suppose using a uniform activation function a), let a dicrete neural network model be

<S,A,I,IF,T,C>
where
S={0,1},
A={do-nothing, output-1-to-all-postsynaptic-cells},
IF = $\sum w_i I_i$,
T= the original acivation function a,
C= a table specifying which action in A to perform (see Figure 1).

This model can simulate the original model and produce the same output: 0 if I'<ths and 1 if I'>=ths. The model will carry out the computation exactly as its conventioal counterparts.

In case of simulating a continuous activation model, we will have to discretize the output, i.e. C will be a table specifying a sequence of points sampled from the responce curve the neuron in the original model. If we sample enough points on the continuous output curve, we can approximate the behavior of the original model closely enough for any practical purpose. For example, Figure 2 shows the approximation of the model: output= potential+ $\sum(w_i i_i)$ .

We also looked at ways of implementing this model with a multilayer conventional PDP model. The question is how to implement the state variable and the state transition function of a discrete neuron. It has been shown that a three layer network could do the job: the hidden layer represents the current state and each input/ouput value is explicitly represented by individual cells (cf. [Servan-Schreiber 1988] and [Allen 1988]). See Figure 3. The output from the hidden layer is fedback into the input layer to help decide, together with current inputs, which state to enter next.

There is one aspect of the model that may raise some questions. Usually in real neuron, one synapse can only release a certain type of transmitters (or a certain group of transmitters). But in my model each synaptic site can release different transimitters (i.e. different messages or synaptic actions). How do we resolve this contradiction? This contradiction can be easily resolved by realizing the fact that we can implement this dicrete neural network model using only the type of neurons in which each synaptic site can only release one type of transmitters, by adding a group of intermediate cells each of which represents a particular message and hook them together. See Figure 4.

The communication between cells can be viewed this way: each message sent is coded as (state, strength), where state means output state or symbol. This can be implemented, in the same sense as above, with the same kind of intermediate layer of units, connecting to source and target cells with certain strengths. See Fig 5.

## 3. LOGICAL OPERATIONS

A number of researchers have cited logical operations as an important factor in determining the adequency of a neural network model as a universal computational model ([Abu 1986],[Rumelhart 1986] etc.).

The discrete model can handle all logical operations very efficiently because of the nature of the state transition function and the action function. An AND operation of two inputs can be modeled by the following state transition/action function (see Figure 6). The other operations such as OR and NOT can be modeled exactly the same way.

Another advantage of the model is the ease with which we can model multiplicative connections. There is no need for an extra type of connections in the network. All of the connections can be accormodated in the same general framework. Yet this framework is kept simple and directly implementable.

## 4. DUALITY-CONNECTION ENCODING

A technique, called duality-connection encoding or DCE, has been developed for representing concepts in a neural network of the type mentioned above, which utilizes the connections to define the concepts and represents the concepts in both verbal and compiled forms. The main advantage is that this scheme can handle variable binding efficiently.

The main dilemma of reasoning in connectionist models is at which level we should incorporate symbols into the schemes. If we perform pure symbolic reasoning, the cost for representing symbols and performing the reasoning is too high, such as in Barnden's scheme (see [Barnden 1988]) or Touretzsky&Hinton's scheme (see [Touretzsky and Hinton 1985]) (even though it handles only a much simplified case). But if we eliminate symbols from the scheme, it will not be suitable for performing high level cognitive task. This model resolved this dilemma by introducing a dual coding technique, That is, encoding a concept by using two cell assemblies: one for linguistic(symbolic) representation and the other for non-linguistic representation suitable for reasoning and variable binding. This scheme ensures the efficiency of reasoning processes. Coarse coding can be used here to have the advantage of fault-tolerance. Another mechanism for fault tolerance is the replication of identical units, which is particularly suitable in this model. This mechanism is found in some small neural circuits in crustacean stomatogastric systems ([Selverston 1986]).

A concept is encoded in the network by the connections it has to the other cells. Those uni-directional connections help shape the concept as well as guide the reasoning. For example, Figure 7 shows how a concept is wired into a network.

In the reasoning assembly, there are k+1 cells: CF,c1,c2, .... ,ck. CF cell contains the certainty factors (or confidence, possibility, etc.) used in reasoning and connects to all other concepts related to it. The other k cells take care of variable bindings for a maximun of k variables. The set of states in the variable cell represents all possible bindings. The signal from CF cell tells the variable cell which input to take. See Figure 8. The formulas used are summarized below (just one simple case as an example):

For CF, S(t) = f(CF(t-1), I1(t), I2(t), ..... Ik(t) ).

f here is a mapping to a value which encodes two things: the activation level and the activation source. Because several other assemblies are connected to this one, f has to distinguish different sources by encoding it in its resulting value. CF(t-1) here is for keeping certain historical contextual information and $I_i s$ are outputs of other CF cells in other assemblies.

And for Ci, i=1,2,....k, S(t) = fi( CF(t), I1(t), I2(t), ..... In(t) ).

$f_i$ here is a table specifying the state of $C_i$ at time t based on inputs at that moment from the CF cell and $c_i$ cells in other assembly. CF(t) is actually an instruction to $c_i s$ , telling them which input to take and make that input state its activation state.

The linguistic (symbolic) assembly of a concept representation records the verbal form of a concept or a predicate which defines that concept. The information recorded can be recalled when the concept is invoked in reasoning.

The representation scheme actually can handle two things: concept representation and predicate representation. Predicate representation is a cell assembly containing one main connection cell determining CF and a set of variable cells for variable binding as described above. On the other hand, concept representation is a cluster (implemented with a cell assembly) with slots to be filled just like variable binding. Each cluster is a frame like structure consisting several parts: a control cell (CF) and a set of role cells. Cells in the latter two groups send signals to the control cell. Control cell also receive signals from other cell assemblies(spreading activation). The conceptual hierarchy is traversed based on spreading activation.

## 5. REASONING SCHEME

A reasoning scheme is developed in the discrete neural network model, which maximizes the inherent parallelism in a neural network model and performs all possible inference steps in parallel.

One of the major drawbacks of conventional connectionist model is its inability in handling deductive reasoning, i.e. deriving conclusions from existing facts. The explicit deductive reasoning (not intuition or subconscious reasoning) requires one to establish rules, store facts in working memory, and use rules to deduce new facts. In real world, facts are usually known with certain uncertainty. So a connectinoist inference engine has to take that into consideration too. Another problem is variable binding. In order to avoid crosstalk betweeen rules, we have to have an efficient mechanism for handling variable binding.

The basic architecture consists of three layers: input, processing and output, with additional modules attachable to them, each of which can handle learning and pre- and post-processing correspondingly.

The input information is processed in input layer and passed on to the processing layer. The processing layer can have complicated internal structures. The information passed to each cell is processed and propagated to all the post synaptic cells from each pre-synaptic cells. This scheme guaranttees a high degree of parallelism.

The calculus for dealing with uncertainty factor propagation was proposed in [Sun 1984, 1985]. The soundness and completeness under certain constraints were proven. It is capable of dealing with certain types of default reasoning. For exampple, if C1(X) and C2(X) and C3(X) then A(X), where X is the vector of variable bindings, can be coded as shown in Figure 8 in DCE. Cell A then combines the evidences by doing CF=w* $\sum I_i$ . In case C3 is undetermined, based on partial information available, the system can still deduce A, with activation less strong than it would be if C3 is known. For contradictory propositions, there can be inhibitory connections between each pair of them, with strengths corresponding to the degrees of contradiction. When strict logical operations (AND, OR, and NOT) are required in the reasoning, the method described in section 3 is applied to achieve the desired effect. An interesting thing is that this scheme fits the 100 step rule well ( see [Feldman 1986]). Besides this formalism, it can also implement schemes proposed in [Zadeh 1983] or [Shastri and Feldman 1987].

ATTENTIONAL MECHANISMS There are two attentional mechanisms in the model: A-area, the reasoning trace, and C-area, a mechanism for controlling the reasoning process directed by the goal of the system.

We assume that the activation is calculated with $\sum w_i i_i$ , where $w_i = s_i m_i + l_i$ . Usually $m_i = 0$. But when we want to concentrate on one area in the network (forming a C-area), the external intention control module can increase $m_i$ . $s_i$ is a predetermined value. So we might come up with results that can not be deduced otherwise (because of strong inhibitions or high thresholds).

## 7. CONCLUSION

Putting these components together, it forms a coherent system with various features and modules for various purposes. This discrete neural network model has the advantages of representational flexibility and expressive power with regard to high-level conceptual representation, and massive parallelism and real-time efficiency with regard to reasoning within this representational framework. Further work is needed to specify more details of conceptual representations and various rule codings. Several learning algorithms are currently under development.

ACKNOWLEDGEMENT I would like to thank David Waltz and Lawrence Bookman for the discussions and suggestions. This work was supported in part by the Defense advanced Research Projects Agency, administered by the US Air Force Office of Scientific Research under contract #F49620-88-C-0058.

## REFERENCES

[1] Y. Abu-Mostafa, Neural Networks for Computing?, Neural Networks for Computing, 1986

[2] D. Ackley, Stochastic iterated genetic hillclimbing, Technical Report, TR CMU-CS-87-107, Carnegie-Mellon University, 1987

[3] R. Allen, Connectionist state machines, Technical Report, Bellcore, 1988

[4] D. Ballard, Parallel logical inference and energy minimization, Technical Report, TR 142, University of Rochester, 1986

[5] J. Barnden, The right of free association: relative-position encoding for connectionist data structures, 10th conf. of Cognitive Science Society, 1988

[6] M. Derthick, Mundane reasoning by parallel constraint satisfaction, Technical Report, TR CMU-CS-88-182, Carnegie-Mellon University, 1988

[7] G. Edelman et al, Synaptic Function, John Wiley & Sons, 1987

[8] M. Fanty, Learning in structured connectionist networks, Technical Report, TR 252, University of Rochester, 1988

[9] J. Feldman, Neural Representation of conceptual knowledge, Technical report 189, 1986

[10] G. Hinton, Connectionist learning procedures, Technical Report, University of Toronto, 1988

[11] J. Holland, Escaping brittleness, Machine Learning, Vol.2, 1986

[12] E. Kandel and J. Schwartz, Principle of Neural Science, 2ed. Elsevier, 1984

[13] S. Pinker and J. Mehler, Connections and Symbols. MIT Press, 1988

[14] B. Richmond and L. Optican, Temporal Encoding of Two-Dimensional Patterns by Single Units in Primate Inferior Temporal Cortex. Journal of Neurophysiology, Vol 57. No 1, January 1987

[15] D. Rumelhart and J. McClelland, Parallel Distributed Processing, MIT press, 1986

[16] A. Selverston and M. Moulins, eds. The Crustacean Stomatogastric System, Springer-Verlag, 1987

[17] D. Servan-Schreiber et al, Encoding sequential structure, Technical Report, CMU-CS-88-183, Carnegie Mellon University, 1988

[18] L. Shastri and J. Feldman, Evidential reasoning in semantic networks. 9th IJCAI, 1987

[19] P. Smolensky, On variable binding and representation of symbolic structure, Tech Report, University of Colorado, Boulder. 1987

[20] P. Smolensky, On the proper treatment of connectionism, Behavioral and Brain Sciences, 11, 1988

[21] R. Sun, The matrix representation of production rules with CF, 3rd Symp. of Discrete Mathematics in China, 1984

[22] R. Sun, The logic for approximate reasoning combining probability and fuzziness, 1st Congress of International Fuzzy System Association, 1985

[23] R. Sun, E. Marder and D. Waltz, The modeling of lobster stomatogastric ganglion, Technical Report CS-88-143, Brandeis University, 1988

[24] D. Touretzky, Representing and transforming recursive objects in a neural network, 1986

[25] D. Touretzky and G. Hinton. Symbols among neurons. 9th IJCAI, 1987

[26] L. Zadeh, Commonsense knowledge representation based on fuzzy logic, Computer, Vol.16, No.10, 1983

| S | I' | A |
|---|---|---|
| don't care | <0 | do-nothing |
| don't care | >=0 | output-1 |

Fig. 1

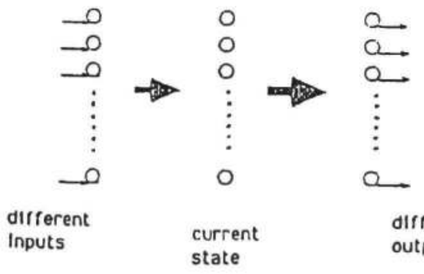


Fig. 3

| S | I' | A |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 0.1 | 0.1 |
| 0 | 0.2 | 0.2 |
| ⋮ | ⋮ | ⋮ |
| 0.1 | 0 | 0.1 |
| 0.2 | 0 | 0.2 |

Fig. 2

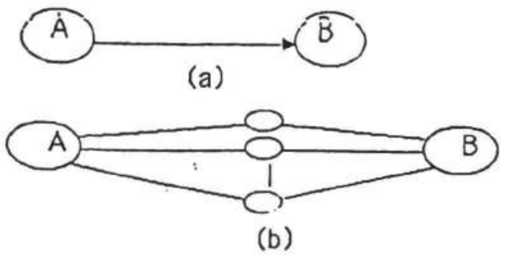


Fig. 4

| state | I1 | I2 | A |
|---|---|---|---|
| don't care | 0 | 0 | 0 |
| " | 0 | 1 | 0 |
| " | 1 | 0 | 0 |
| " | 1 | 1 | 1 |

Fig. 6

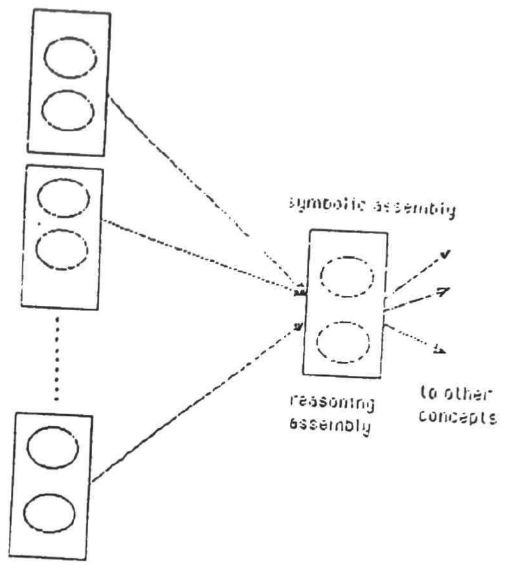


Fig. 7

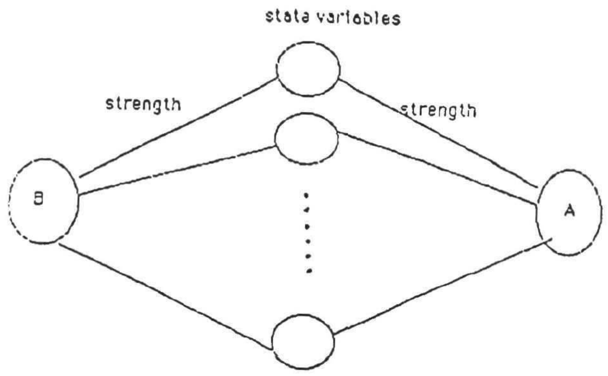
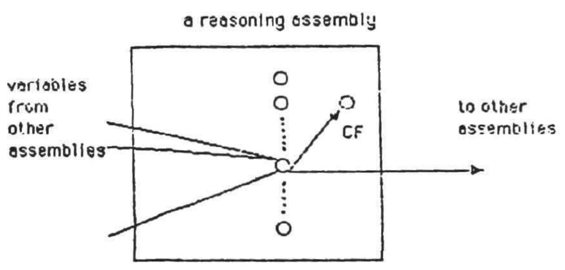


Fig. 5



ith variable cell in a reasoning assembly

Fig. 8

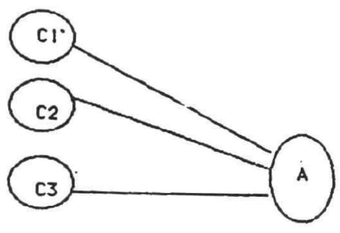


Fig. 9