Title

A Connectionist Encoding of Semantic Networks

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 9(0)

Author

Shastri, Lokendra

Publication Date

1987

Peer reviewed

A Connectionist Encoding of Semantic Networks

Lokendra Shastri
Computer and Information Science Department
University of Pennsylvania
Philadelphia, PA 19104

Abstract

Although the connectionist approach has lead to elegant solutions to a number of problems in cognitive science and artificial intelligence, its suitability for dealing with problems in knowledge representation and inference has often been questioned. This paper partially answers this criticism by demonstrating that effective solutions to certain problems in knowledge representation and limited inference can be found by adopting a connectionist approach. The paper presents a connectionist realization of semantic networks, i.e. it describes how knowledge about concepts, their properties, and the hierarchical relationship between them may be encoded as an *interpreter-free* massively parallel network of simple processing elements that can solve an interesting class of *inheritance* and *recognition* problems extremely fast - in time proportional to the depth of the conceptual hierarchy. The connectionist realization is based on an *evidential* formulation that leads to principled solutions to the problems of *exceptions, multiple inheritance*, and *conflicting information* during inheritance, and the *best match* or *partial match* computation during recognition.

1 Introduction

Connectionist networks are playing an increasingly important role in artificial intelligence (AI) and cognitive science and have been employed successfully to deal with a variety of problems in low and intermediate level vision, word perception, associative memory, word sense disambiguation, modeling of context effects in natural language understanding, speech production, and a wide range of issues related to learning (Cognitive Science 85; McClelland & Rumelhart, 1986; Rumelhart & McClelland, 1986). However, for connectionism to be considered a scientific language of choice for expressing solutions to problems in cognitive science and AI, it must be demonstrated that it can be used to represent highly *structured* knowledge and perform inferences based on such knowledge. A common criticism leveled against connectionism is that although it is appropriate for modeling "approximate" memory processes such as semantic priming associative recall, it is unsuitable for dealing with problems related to knowledge representation and reasoning.

The work described in this paper partially answers the criticism by demonstrating that the connectionist approach is extremely effective in solving certain problems in knowledge representation and inference. This paper presents a connectionist realization of semantic networks, i.e it describes how knowledge about concepts, their properties, and the hierarchical relationship between them, may be encoded as a connectionist network that can compute principled solutions to ***inheritance*** and ***recognition*** problems with extreme efficiency. Some salient features of the system are:

i) The connectionist semantic networks use *controlled* spreading activation to solve an interesting class of *inheritance* and *recognition* problems extremely fast - in time proportional to the depth of the conceptual hierarchy.

ii) The networks compute the solutions in accordance with an evidential formalization that derives from the principle of *maximum entropy*. This formalization leads to a principled treatment of *exceptions, multiple inheritance* and *conflicting information* during inheritance, and the *best match* or *partial match* computation during recognition.

iii) The networks operate without the intervention of a central controller and do not require a distinct interpreter. The knowledge as well as mechanisms for drawing *limited inferences* on it are encoded within the network.

iv) The networks can be constructed from a high-level specification of the knowledge to be encoded and the mapping between the knowledge level and the network level is precisely specified. Furthermore, the solution scales because the design is independent of the size of the semantic memory.

1.1 Representation and retrieval: An overview

The system's conceptual knowledge is encoded in a connectionist network referred to as the ***Memory network***. This network is capable of performing inheritance and recognition via controlled spreading activation. A problem is posed to the network by activating relevant nodes in it. Once activated, the network performs the required inferences *automatically* and at the end of a specified interval the answer is available implicitly as the levels of activation of a relevant set of nodes.

In keeping with the connectionist paradigm, the presentation of queries to the Memory network, and the subsequent answer extraction is also carried out by connectionist network fragments called ***routines***. Routines encode canned procedures for performing specific tasks and are represented as a sequence of nodes connected so that activation can serve to sequence through the routine. Routines pose queries to the Memory network by activating appropriate nodes in it. The Memory network in turn returns the answer to the routine by activating *response* nodes in the routine. The activation returned by a node in the Memory network is a measure of the evidential support for an answer. It is assumed that all queries originating in routines are posed with respect to an explicit set of answers and there is a response node for each possible answer. Response nodes compete with one another and the node receiving the maximum activation dominates and triggers the appropriate action. Thus, computing an answer amounts to choosing the answer that receives the highest evidence *relative* to a set of *potential answers*. The actual answer extraction mechanism explicitly allows for "don't know" as a possible answer. This may happen if there is insufficient evidence for all the choices or if there is no clear cut winner. This interaction between the Memory network and routines is depicted in Figure 1.

1.2 Semantic networks, inheritance, and recognition

The term "semantic networks" has been used in a very general sense in the AI literature. We will however, only focus on the central aspects of semantic networks namely, that concepts are represented in terms of their properties and that the subsumption relationship between concepts is captured by the IS-A hierarchy. This characterization is broad enough to capture the basic organizational principles underlying *frame*-based representation langauges such as KRL (Bobrow & Winograd, 1976) and KL-ONE (Brachman & Schmolze, 1985).

The organization and structuring of information in a semantic network leads to an efficient realization of two kinds of inferences which we will refer to as ***inheritance*** and ***recognition***. It can be argued that these two complementary forms of reasoning lie at the core of intelligent behavior and act as precursors to more complex and specialized reasoning processes.

Typically, inheritance refers to the form of reasoning that leads an agent to infer property values of a concept based on the property values of its ancestors. We define inheritance more generally to include looking up property values directly available at the concept - of course if such local information is not available then inheritance involves looking up properties attached to concepts higher up in the conceptual hierarchy. Many cognitive tasks may be shown to require inheritance as an intermediate step - word sense disambiguation, determination of case-fillers, and enforcement of selectional restrictions are some examples.

Recognition is the dual of the inheritance problem. The recognition problem may be described as follows: "Given a description consisting of a set of properties, find a concept that best matches this description". Note that during matching all the property values of a concept may not be available locally and may have to be determined via inheritance from its ancestors. For this reason, recognition may be viewed as a very general form of ***pattern matching***: one in which the target patterns, i.e. the set of patterns to which an input pattern is to be matched, are organized in a hierarchy, and where matching an input pattern A with a target pattern T_i involves matching properties that appear in A with properties local to T_i as well as to properties that T_i inherits from its ancestors.

A recognition step followed by an inheritance step amounts to an important sort of reasoning namely, *pattern completion*. Using recognition a process can determine the identity of an object based on its partial description, and having determined the object's identity the process may perform an inheritance step to determine the unknown properties values of the object.

In addition to their ubiquity, inheritance and recognition are also significant because in spite of operating with a large knowledge base, humans perform these inferences effortlessly and *extremely fast* - often in a few hundred milliseconds. This suggests that inheritance and recognition are perhaps *basic* and *unitary* components of symbolic reasoning - probably the smallest and simplest cognitive operations that i) produce *specific responses*, and ii) can be initiated, and have their results accessed by complex and higher-level symbolic reasoning processes. The speed with which these operations are performed also suggests that they are performed fairly automatically, and typically do not require any conscious and attentional control. Given the significance of inheritance and recognition, it appears reasonable to pursue a *computational account* of how these inferences may be drawn with the requisite efficiency.

In addition to offering computational effectiveness, the connectionist network computes solutions to the inheritance and recognition problems in accordance with a theory of *evidential reasoning* that derives from the principle of *maximum entropy*. Under the evidential formulation, inheritance and recognition are posed as problem whose answers involve choosing the *most likely* alternative from among a set of alternatives - the computation of likelihood being carried out with respect to the knowledge encoded in the conceptual hierarchy. This reformulation provides a principled way of handling *exceptions* and resolving *conflicting information* during multiple inheritance, and finding *best matches* based on *partial information* during recognition.

This paper is about the connectionist realization and a detailed discussion of the underlying evidential formulation and the motivation for adopting it, is beyond the scope of this paper. The evidential formulation, its relation to Bayes' rule, and its merits are discussed in (Shastri, 1987); a brief version that deals primarily with inheritance appears in (Shastri & Feldman, 1985). Therein, it is argued that although non-evidential treatments such as proposals based on Default Logic (Etherington & Reiter, 1983) or on the Principle of Inferential Distance Ordering (Touretzky, 1986) can handle exceptions, they do not deal with conflicting information adequately - they either make arbitrary choices or simply report an ambiguity. In contrast, the evidential approach provides a semantically justifiable way of combining all the relevant information (even though some of it may be conflicting) to obtain the most likely answer.

1.3 Related work on parallel encoding of semantic networks

Fahlman's NETL (Fahlman, 1979) was the first attempt at encoding semantic networks as a massively parallel network of simple processing elements. NETL elements communicated with one another under the control of a central controller by propagating discrete messages called *markers*. A network element could only detect the presence or absence of a marker in the input. This all or none nature of the system made it incapable of supporting "best match" or "partial match" operations. For example, in NETL recognition amounted to finding a concept that possessed *all* of a specified set of properties. Furthermore, NETL's solution to the inheritance problem was sensitive to race conditions in the presence of multiple hierarchies. These limitations of marker passing systems are discussed at length in (Fahlman, 1982; Fahlman et al., 1981; Brachman, 1985)[1]. Finally, NETL did not fully utilize the potential for parallelism because the inter-node communication critically depended on instructions issued by a central (serial) controller.

Hinton proposed a "distributed" encoding of semantic networks using parallel hardware (Hinton, 1981). The information encoded in the network was interpreted as a set of triples of the form: [relation, role1, role2]. The proposed system had several interesting properties: given two components of a triple, the network could determine the third tuple, the network could be programmed using the perceptron convergence rule, and it could perform simple property inheritance. The system however, lacked sufficient structure and control to handle general cases of inheritance and "partial matching" - especially if these occurred in a mutli-level semantic network that included multiple hierarchies and exceptional or conflicting information.

More recently, Derthik (Derthik, 1986) is implementing a variant of KL-ONE using the Boltzman machine formulation (Ackley et al., 1985). However, the representation language being implemented does not admit exceptions and conflicting information and is open to the same objections that apply to other non-evidential

[1]Subsequent work by Touretzky has remedied certain problems with inheritance. The use of discrete markers however, still precludes partial match and best match operations.

formalizations.

Recent work on Bayesian networks (Pearl, 1985) deals with evidential reasoning in a parallel network. Pearl's results however, apply only to singly-connected networks (networks in which there is only one underlying path between any pair of nodes). More complex networks have to be *conditioned* to render them singly-connected. This is in part due to the unstructured form of the underlying representation language employed by Pearl. The language does not make distinctions such as "concept", "property", "property-value" that we make (cf. section 2 below) and hence its ability to exploit parallellism is limited.

2 A restricted language for representing conceptual knowledge

The representation langauge may be viewed as an extension of inheritance hierarchies to include relative frequency information specifying how instances of some concepts are distributed with respect to some property values. A summary description of the language follows. The agent's knowledge consists of the quintuple:

$\Theta = \langle C, \Phi, \#, \delta, << \rangle$, where

C is the set of *concepts*, Φ is the set of *properties*, # is a mapping from C to the integers I, δ the *distribution* function is a partial mapping from C X Φ to the power set of C X I, and << is a partial ordering defined on C.

For each C $\in$ C, if C is a Token (instance) then #C = 1, and if C is a Type (generic concept) then #C = the number of instances of C *observed by the agent*. By extension #C[P,V] = the number of instances of C that are observed by the agent to have the value V for property P. For example, #APPLE[has-color, RED] equals the number of red apples observed by the agent. Finally, $\#C[P_1,V_1][P_2,V_2] \ldots [P_n,V_n]$ = the number of instances of C, observed to have the value V_1 for property P_1, value V_2 for property P_2, ... and value V_n for property P_n.

The distribution function δ(C,P), specifies how instances of C are distributed with respect to the values of property P. Recall that a concept may have several values for the same property and hence, if C is a Type, then δ(C,P), corresponds to the summary information abstracted in C based on the instances of C. Using the # function, δ(C,P) may be expressed in terms of #C[P,V]'s. Thus, δ(APPLE, has-color) may be expressed as: { #APPLE[has-color, RED] = 60, #APPLE[has-color, GREEN] = 40}. Note that δ is only a partial mapping; an agent may not know δ(C,P) for many concept-property pairs. In general, for a given C and P, an agent knows δ(C,P) only if this information may prove useful in making inferences about C.

A salient feature of the language is that either a concept is an instance of (subtype of) another concept or it is not, and the << relation specifies this unequivocally. Exceptions only apply to property values. Furthermore, both *necessary* properties as well as *default* properties may be represented. This goes a long way in assigning a clean semantics to the representation language.

In terms of the above notation, the inheritance and recognition problem may be restated as follows:

Inheritance

Given: A concept C, a property P, and a set of property values, V-SET = $\{V_1, V_2, \ldots V_n\}$,

Find: V* $\in$ V-SET, such that among members of V-SET, V* is the *most likely value* of property P for concept C. In other words, find V* $\in$ V-SET such that, for any $V_i \in$ V-SET, the best estimate of #C[P,V*] $\geq$ the best estimate of $\#C[P,V_i]$'s.

For example,the inheritance problem where C = APPLE, P = has-color, V-SET = {RED, BLUE, GREEN}, may be paraphrased as: Is the color of an apple more likely to be red, green or blue?

Recognition

Given: a set of concepts, C-SET = $\{C_1, C_2, \ldots C_n\}$, and an appropriate description consisting of a set of property value pairs, i.e., a DESCR = $\{ [P_1,V_1], [P_2,V_2], \ldots [P_m,V_m] \}$.

Find: C* $\in$ C-SET such that *relative* to the concepts specified in C-SET, C* is the *most likely* concept described by DESCR.

If C-SET = {APPLE, GRAPE}, DISCR = {[has-color, RED], [has-taste, SWEET]} then the recognition problem may be paraphrased as: "Is something red in color and sweet in taste more likely to be an apple or a grape"?

The solutions to the two problems are based on the principle of maximum entropy [Jaynes 1979] and are described in [Shastri 1987].

3 Connectionist encoding

A connectionist network (Feldman & Ballard, 1982) consists of a large number of nodes connected via links. The nodes are computational entities defined by a small number (2 or 3) of states, a real-valued potential in the range [0,1], an output value also in the range [0,1], a vector of inputs $i_1, i_2, ... i_n$, together with functions P, V and Q that define the values of potential, state and output at time t+1, based on the values of potential, state and inputs at time t. Nodes receive inputs via *weighted* links. A node may have multiple input *sites*, and incoming links are connected to specific sites. Each site has an associated *site functions*. These functions carry out local computations based on the inputs incident at the site, and it is the result of this computation that is processed by the functions P, V and Q.

Connectionist networks offer a natural computational model for encoding evidential formalisms because of the natural correspondence between nodes and hypotheses, activation and evidential support, and potential functions and evidence combination rules. However, in order to solve the inheritance and recognition problems, the network must perform very specific computations and it must do so *without the intervention of a central controller*. The design involves introducing explicit "control nodes" (binder and relay nodes) throughout the network to mediate and control the spread of activation.

Before describing the encoding in detail, we consider an example. Figure 2 shows a network that encodes:

> "Dick is a Quaker and a Republican, most Quakers have pacifist beliefs, while most Republicans have non-pacifist beliefs"

It is assumed that one of the properties attached to persons is "has-belief", some of whose values are "pacifist" and "non-pacifist". The Figure only shows about half the connections. In particular, the connections from property values to concepts have been suppressed for better readability. The likelihoods of being pacifists and non-pacifists for Quakers and Republicans are encoded as weights of appropriate links (Cf. Section 3.1)

The question of Dick's beliefs on pacifism (or lack of it) can be posed to the network by activating the nodes DICK, has-belief, and BELIEF. The resulting potentials of the nodes PACIFIST and NON-PAC will determine whether Dick is more likely to be a pacifist or a non-pacifist. It can be shown that the potential of the node PACIFIST equals:

$$\frac{\#\text{QUAKER[has-bel,PACIFIST]} \times \#\text{REPUBLICAN[has-bel,PACIFIST]}}{\#\text{BELIEF} \times \#\text{PERSON[has-bel,PACIFIST]}}$$

The potential of the node NON-PAC is given by an analogous expression in with NON-PAC replaces PACIFIST.

Ignoring the common factor, #BELIEF, in the above expression, the potential of PACIFIST (NON-PAC) corresponds to the best estimate of the number of persons that are both quakers and republicans and believe in pacifism (non-pacifism). Hence, a comparison of the two potentials will give the most likely answer to the question: Is Dick a pacifist or a non-pacifist.

3.1 Encoding the conceptual structure

The encoding employs five distinct unit types. These are the *concept* nodes (ξ-nodes), *property* nodes (ϕ-nodes), *binder* nodes, *relay* nodes and *enable* nodes. With reference to Figure 2, all solid boxes denote ξ-nodes, all triangular nodes denote binder nodes, and the single dashed box denotes a ϕ-node. Relay nodes are used to control directionality of spreading activation along the conceptual hierarchy, while enable nodes are used to specify the type of query (inheritance or categorization). Relay and enable nodes are not shown in Figure 2.

Each concept is represented by a ξ-node. These nodes have six sites: **QUERY, RELAY, CP, HCP, PV** and **INV**. With reference to the partial ordering <<, if concept B is a parent of concept A then there is a $\uparrow$ (bottom up) link from A to B and a $\downarrow$ (top down) link from B to A. The weight on both these links equal # A/ #B and they are incident at the site **RELAY**. As the ↑ and ↓ links always occur in pairs, they will often be represented by a single undirected arc. Arcs between DICK and QUAKER, and QUAKER and PERSON, are examples of such interconnections.

Each property is also encoded as a node. These nodes are called ϕ-nodes, and they has one input site: **QUERY**.

If $\delta(A,P)$ is known, then for every value V_i of P there exists a pair of binder nodes [A,P -> V_i] and [P,V_i -> A] that are connected to A, P and V_i as shown in Figures 3 and 4 respectively. A binder node such as [A,P -> V_i] is called a i-binder node and has two sites: **ENABLE** and **EC**. The node [A,P-> V_i] receives one input from node A and another from node P. Both these inputs are incident at site **ENABLE**, and the weight on these links is 1.0. The link from [A,P -> V_i] to V_i is incident at site **CP** and the weight on this link is given by #A[P,V_i] / #V_i. A binder node such as the node [P,V_i -> A] is called a r-binder node and has one site **ENABLE** where it receives inputs from nodes P and V_i; the weights on these links are 1.0. The output from [P,V_i -> A] is incident at the site **PV** of A, and the weight on this link is given by #A[P,V_i] / #A.

If B is a parent of A such that $\delta(B,P)$ is known, and there is no concept C between A and B for which $\delta(C,P)$ is known, then there is a link from [A,P -> V_i] to [B,P -> V_i], incident at site **INV** with a weight of # A[P,V_i]/ #B[A,V_i] (refer to Figure 5). Similarly, there is a link from [P,V_i -> B] to A incident at site **INV** with a weight of #B[P,V_i] / #B (refer to Figure 6). Finally, if B is such that it is the highest node for which $\delta(B,P)$ is known, then the link from [B,P -> V_i] to V_i is incident at site **HCP**, instead of site **CP**.

Besides the interconnections described above, all nodes representing concepts, properties, and values (ξ-nodes and ϕ-nodes) have an external input incident at the site **QUERY**, with a weight of 1.0. In addition to the unit types described above, there are two other enable units: INHERIT and RECOGNIZE.These units have one input site: **QUERY**, at which they receive an external input. Each i-binder node receives an input from the node INHERIT at the site **ENABLE** while each r-binder node receives an input from the node RECOGNIZE also at the site **ENABLE**.

3.2 Description of network behavior

Each unit in the network can be in one of two states: **active** or **inert**. The quiescent state of each unit is **inert**. A unit switches to an **active** state under conditions specified below, and in this state the unit transmits an output equal to its potential. The computational characteristics of various unit types are described below:

ξ-nodes:
State: Node is in **active** state if it receives one or more inputs.
Potential: If no inputs at site **HCP** then
potential = the product of inputs at sites **QUERY, RELAY, CP,** and **PV** divided by the product of inputs at site **INV**.
else potential = the product of inputs at sites **QUERY, RELAY, HCP**

i-binder nodes:
State: Node is in **active** state if and only if it receives all the three inputs at site **ENABLE**.
Potential: If state = **active** then
potential = 1.0 * the product of inputs at sites **EC**
else potential = NIL

r-binder nodes:
State: Node is in **active** state if and only if it receives all three inputs at site **ENABLE**.
Potential: If state = **active** then potential = 1.0 else potential = NIL

ϕ-nodes, INHERIT node, and RECOGNIZE node switch to **active** state if they receive input at site **QUERY**, and in this state their potential always equals 1.0.

The networks have the additional property that unlike other links that always transmit the output of their source node, the ↑ and ↓ normally remain disabled, and transmit activity only when they are enabled. This control is affected via *relay* nodes that are associated with ϕ-nodes. The details of this mechanism are beyond the scope of this paper.

3.3 Posing queries and computing solutions

In the context of the network encoding the inheritance and categorization are posed as follows:

Inheritance

Given: A concept C, a property P, a set of possible answers, V-SET = $\{V_1, V_2, \ldots V_n\}$, and a concept REF where REF is an ancestor of every memebr of V-SET. (Typically, REF is a parent of V_i's. For example, if V_i's are RED, GREEN, BLUE ... then REF could be COLOR).

Find: $V^* \in$ V-SET such that relative to the values specified in V-SET, V^* is the most likely value of property P for concept C.

The inheritance query is posed by setting the external inputs, i.e. the inputs to the site **QUERY**, of nodes C, P and INHERIT to 1.0. If one or more members of V-SET reach an **active** state within three time steps, the external input to REF gets set to 1.0, and the ↓ links leaving REF are enabled. If none of the members of V-SET receive any activation, the external input to REF is set to 1.0, and the ↓ links leaving REF as well as the ↑ links leaving C are enabled. After d+3 time steps - where d is the longest path in the ordering graph defined by C and <<, the potentials of nodes will be such that for any two nodes V_i and $V_j \in$ V-SET, the following holds:

$$\frac{\text{potential of } V_i}{\text{potential of } V_j} = \frac{\#C[P,V_i]}{\#C[P,V_j]}$$

It follows that the node $V^* \in$ V-SET with the highest potential will correspond to the value that is the solution to the inheritance problem.

Recognition

Given: a set of concepts C-SET = $\{ C_1, C_2, \ldots C_n)$, a reference concept REF, such that REF is an ancestor of all concepts in C-SET, and a description consisting of a set of property value pairs, i.e. a set DESCR = $\{ [P_1,V_1], [P_2,V_2], \ldots [P_m,V_m] \}$

Find: $C^* \in$ C-SET such that relative to the concepts specified in C-SET, C^* is the most likely concept described by DESCR.

The solution to the above problem may be computed as follows: For each $[P_j,V_j] \in$ DESCR, set the inputs to the site **QUERY** of nodes P_j and V_j to 1.0. At the same time, set the input to the site **QUERY** of RECOGNIZE and REF to 1.0, and enable the ↓ links emanating from REF. Wait d + 3 time steps, where d is the longest path in the ordering graph defined by **C** and <<. At the end of this interval, the potential of the nodes will be such that for any two nodes C_i and $C_j \in$ C-SET, the following holds:

$$\frac{\text{potential of } C_i}{\text{potential of } C_j}$$

equals the best estimate of $\#C_i[P_1,V_1][P_2,V_2] \ldots [P_m,V_m]$ divided by the best estimate of $\#C_j[P_1,V_1][P_2,V_2] \ldots [P_m,V_m]$. It follows that the node $C^* \in$ C-SET with the highest potential corresponds to the solution of the recognition problem.

4 Some examples

In order to explicate the behavior of networks and demonstrate the nature of inferences drawn by them, several examples that are often cited in the knowledge representation literature as being problematic have been simulated. The first example is an extension of the "quaker example" discussed in section 3. It demonstrates how the network performs inheritance in the presence of conflicting information arising due to "multiple inheritance". Figure 7 depicts the underlying information. There are two properties has-bel (has-belief) with values PAC (pacifist) and NON-PAC (non-pacifist), and has-eth-org (ethnic-origin) with values AFRIC (african) and EURO (european). In broad terms, the information encoded is as follows:

Most persons are non-pacifists, most quakers are pacifists, most republications are non-pacifists, most persons are of european descent, most republicans are of european descent, and most persons of african decent are democrats.

Such information is specified to a network compiler in terms of: i) the set of concepts, ii) the set of properties and their associated values, iii) a list specifying the partial ordering together with the ratios #A/#B (for all pairs of concepts A and B such that B is a parent of A), and iv) a partial mapping $\delta(C,P)$ in terms of #C[P,V]'s. The specification does not refer to any network level detail and the compiler directly translates such a specification into a connectionist network.

As our first example of inheritance, consider the query: "Is Dick a pacifist or a non-pacifist?" The normalized potentials of PAC and NON-PAC as a result of this query are 1.00 and 0.66 respectively. Thus, on the basis of the available information, Dick who is a republican and a quaker is more likely to be a pacifist than a non-pacifist, the ratio of likelihoods being 1.00 : 0.66, i.e., about 3:2. Similar simulations for RICK, PAT, and SUSAN lead to the following results: Rick who is a mormom republican is more likely to be a non-pacifist. The ratio of pacifist v/s non-pacifist for Rick being 0.39 v/s 1.00. Pat who is mormon democrat is also more likely to be a non-pacifist, but only marginally so (0.89 v/s 1.00). Finally, Susan who is a quaker democrat is likely to be a pacifist with a very high probability (1.00 v/s 0.29).

As an example of recognition, consider the query: "among Dick, Rick, Susan, and Pat, who is more likely to be a pacifist of african descent?" The resulting normalized potentials are SUSAN 1.00, PAT 0.57, DICK 0.11, and RICK 0.05. As would be expected, Susan who is a democrat and a quaker, best matches the description "person of african descent with pacifist beliefs". The least likely person turns out to be Rick (notice that Rick is neither a democrat who correlate with african origin nor is he a quaker who correlate with pacifism).

In order to illustrate how exceptions are handled, the information given in Figure 8 was encoded in a network. The information captures the following aspect of the domain: "Most Molluscs are shell-bearers, Cephalopods are Molluscs, but most Cephalopods are not shell-bearers, Nautili are Cephalopods, and all Nautili are shell-bearers".

The normalized potentials of SHELL and SKIN as a result of the inheritance of the property epidermis-type of MOLLUSC, CEPHAL, and NAUTILUS are as follows: (the potentials of FUR and FEATHER were consistently 0.0):

VALUE	MOLLUSC	CEPHAL	NAUTILUS
SHELL	1.00	0.25	1.00
SKIN	0.43	1.00	0.00

Thus, a Mollusc is more likely to be a shell-bearer. A Cephalopod is not likely to be a shell-bearer. Finally, a Nautilus is *definitely* a shell-bearer (note that the likelihood of a Nautilus having an epidermis-type other than shell computes to 0.00, this is because ALL Nautilus are shell-bearers).

5 Conclusions

This effort has lead to the design of a connectionist network that provides a computational account of how an interesting class of inheritance and recognition problems may be solved extremely fast. The networks also have a *provable behavior*; they compute solutions to the inheritance and recognition problems in accordance with a theory of evidential reasoning. The use of evidential reasoning redefines these problems so that conflicting information can be interpreted in a semantically consistent manner. The work also identifies specific constraints that must be satisfied by the conceptual structure in order to achieve an efficient connectionist realization. These are discussed at length in (Shastri 87).

Besides offering a natural way of describing the evidential interactions between pieces of knowledge, the network encoding *suggests* how a physical system may extract from its environment the information required to solve inheritance and recognition problems. An examination of the weights on the links reveals that in most cases the weights are directly related to Hebb's interpretation of synaptic weights (Hebb, 1949). The weight on these links is equal to the ratio: *"how often when the destination node was active, was the source node also active"*.

A discussion of a connectionist system often leads to the question of its biological plausibility. It may be felt that the computational characteristics of nodes described in section 3.1 are too complex to be biologically plausible. The proposed encoding is certainly not intended to be a blueprint for building "wetware". Yet it does satisfy nearly all the constraints proposed in (Feldman & Ballard, 1982). The only serious violation of biological plausibiltiy is the requirement that nodes perform high precision multiplication. One may interpret the connectionist system described here as an ideal realization of a formal model of evidential reasoning. One can try and identify more plausible "approximations" of the ideal system and study the manner in which their response deviates from the prescribed behavior. Such an exercise may be rewarding and point out further constraints that govern the organization of conceptual structure.

6 Acknowledgement

I am grateful Jerry Feldman for his guidance, insights, and support. This research was supported in part by the National Science Foundation under Grants MCS-8209971M IST-8208571, and DCR-8405720, DCR-86-07156, and U.S. Army Research Office grant ARO-DAAG29-84-K-0061.

7 References

Ackley, D.H., Hinton, G.E. & Sejnowski T.J. (1985). A learning algorithm for Boltzmann Machines, *Cognitive Science*, 9, (1), Jan. - March 1985.

Bobrow, D.G. & Winograd, T. (1976). An overview of KRL: A Knowledge Representation Language. CSL-76-4. Xerox Palo Alto Research Centre.

Brachman, R. J. (1985). I lied about the trees. *The AI Magazine*, vol.6, no. 3, Fall 1985, pp. 80-93.

Brachman R.J. and Schmolze, J. (1985). An overview of KL-ONE Knowledge Representation System. *Cognitive Science* 9(2), April, 1985.

Cognitive Science, (1985) Special Issue on Connectionism. *Cognitive Science* 9 (1), Jan. - March, 1985.

Derthik, M. (1986) A connectionist Knowledge Representation System. Thesis Proposal, CMU, June 1986.

Etherington, D. W. & Reiter, R. (1983). On inheritance hierarchies with exceptions. *Proc. AAAI-83*, Washington D.C.

Fahlman, S. E. (1982). Three flavors of parallelism. *Proc. CS-CSI-82*. Canada 1982.

Fahlman, S. E., Touretzky, D.S., and van Roggen Walter. (1981). Cancellation in a parallel semantic network. *Proc. IJCAI-81*. Vancouver, B.C. 1981.

Fahlman, S.E. (1979) *NETL: A System for Representing and Using Real-World Knowledge*. The MIT Press, 1979.

Feldman, J. A. and Ballard, D. H. (1982). Connectionist models and their properties. *Cognitive Science*, 1982, 6, pp. 205-254.

Hebb, D.O. (1949) *The Organization of Behavior*, Wiley, New York, 1949.

Hinton, G.E. (1981) Implementing Semantic Networks in Parallel Hardware. In *Parallel Models of Associative Memory*. Hinton G.E., Anderson J.A. (Eds). Lawrence Erlbaum Associates, 1981.

Jaynes, E.T. (1979) Where Do We Stand on Maximum Entropy. In *The maximum entropy formalism*, R.D. Levine and M. Tribus (Eds.) MIT Press, Cambridge Massachussets. 1979.

McClelland, J.L. & Rumelhart, D.E. (1986). (Eds) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Vol II. Bradford Books/MIT Press, Cambridge, MA, 1986.

Pearl, J. (1985). Bayesian Networks: A model of self-activated memory for evidential reasoning. *Proc. 7th. Cognitive Science Conference*, Irvine, CA 1985.

Rumelhart, D.E. & McClelland, J.L. (1986). (Eds.). *Parallel Distributed Processing: Explorations in the*

Microstructure of Cognition. Vol I. Bradford Books/MIT Press, Cambridge, MA, 1986.

Shastri, L. (1987) *Semantic Networks: An Evidential Formalization and its connectionist realization*. To appear Morgan Kaufman, Los Altos, CA, and Pitman, London. 1987.

Shastri L. & Feldman J.A. (1985). Evidential reasoning in semantic networks: a formal theory. In *Proc. IJCAI-85*. Los Angeles CA.

Touretzky, D. S. (1986). *The Mathematics of Inheritance Systems*. Morgan Kaufman, Los Altos, CA, Pitman London, 1986.

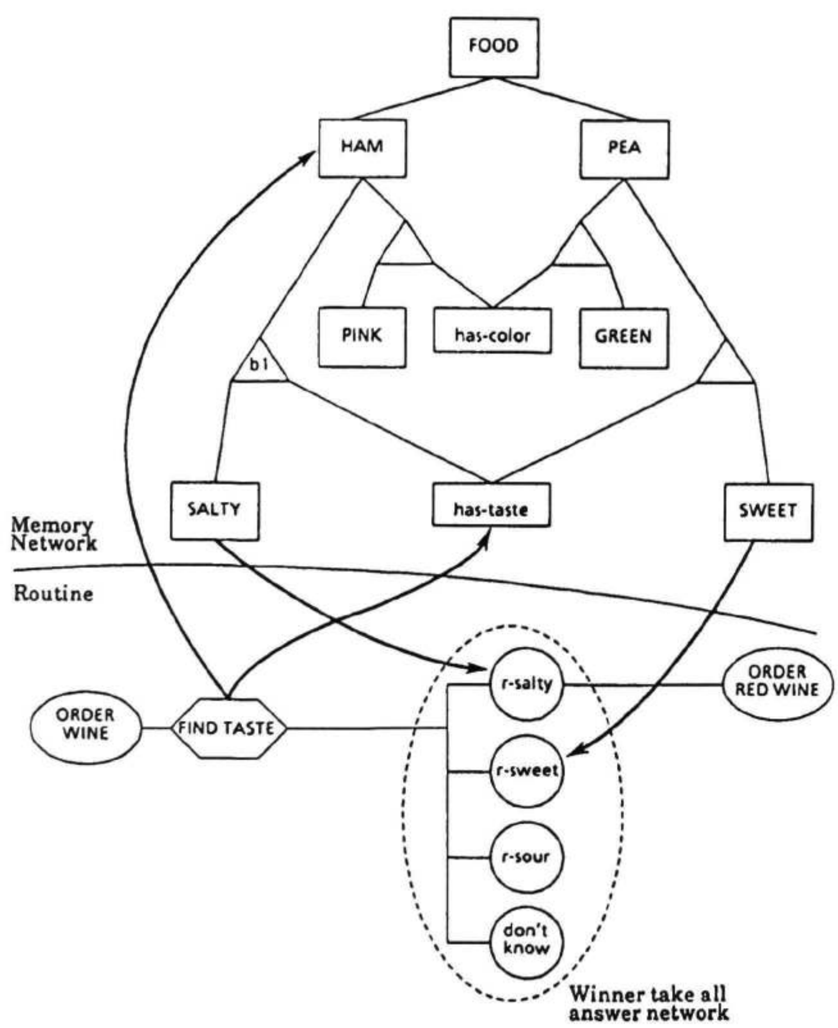
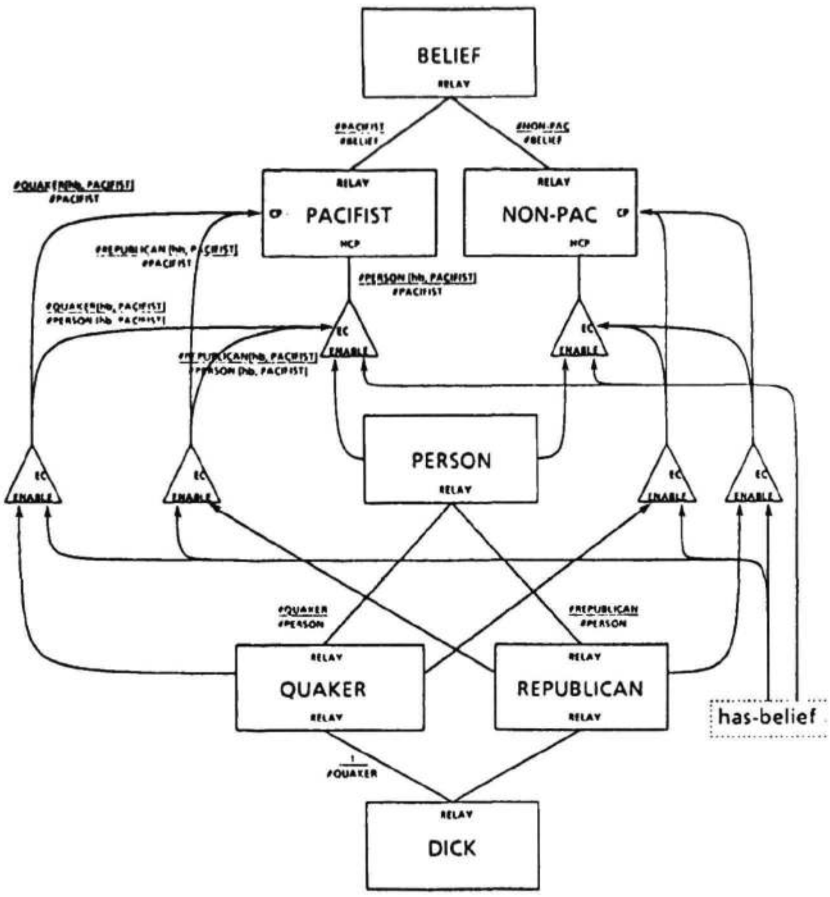


FIGURE 1: Connectionist Retrieval System



All inputs incident at site enable of δ-node have a weight of 1.0.
Not all sites and weights have been shown.
hb = has-belief

FIGURE 2: An example network

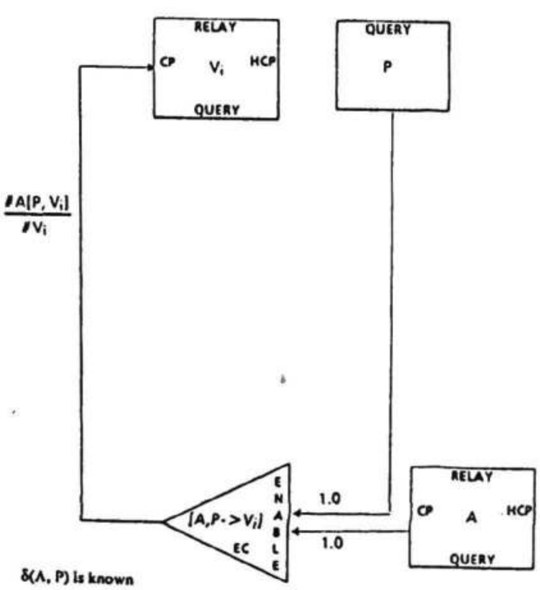


FIGURE 3: Parallel encoding for inheritance - I

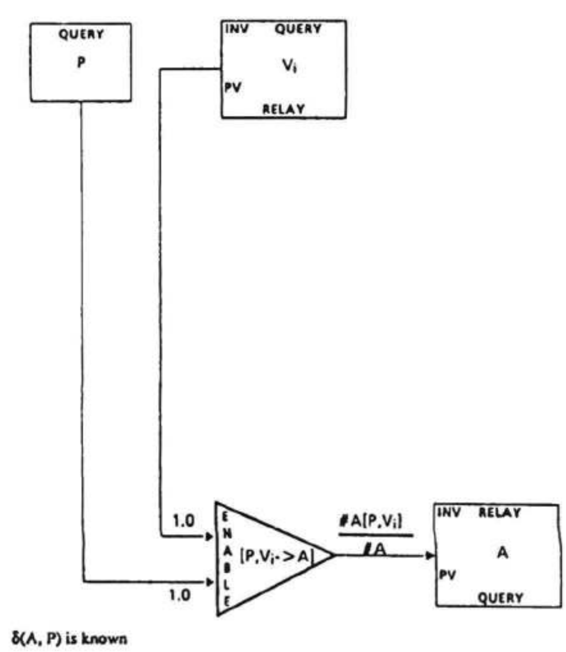
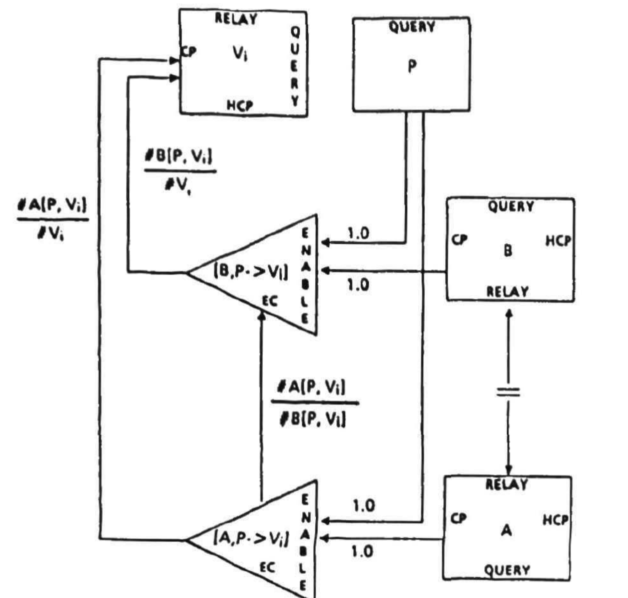
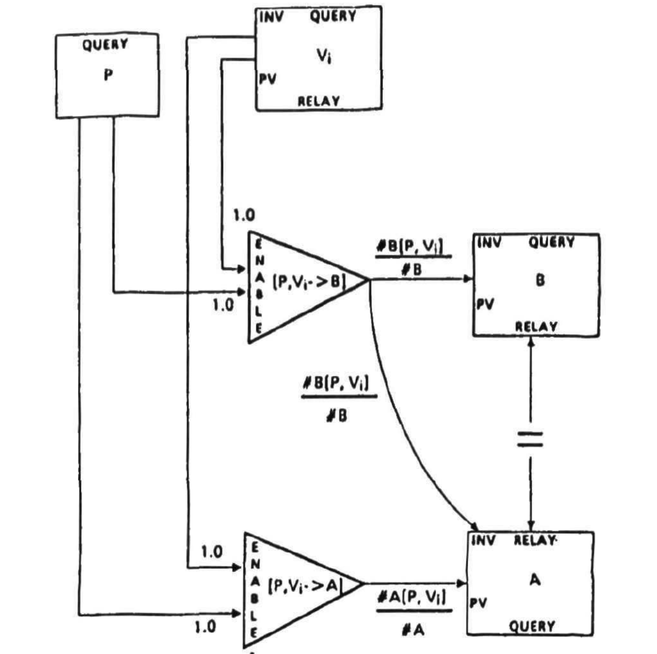


FIGURE 4: Parallel encoding for recognition - I



δ(A, P), δ(B, P) are known, but there does not exist a concept C such that A << C << B and δ(C, P) is known

FIGURE 5: Parallel encoding for inheritance - II



δ(A, P), δ(B, P) are known, but there does not exist a concept C such that A << C << B and δ(C, P) is known

FIGURE 6: Parallel encoding for recognition - II

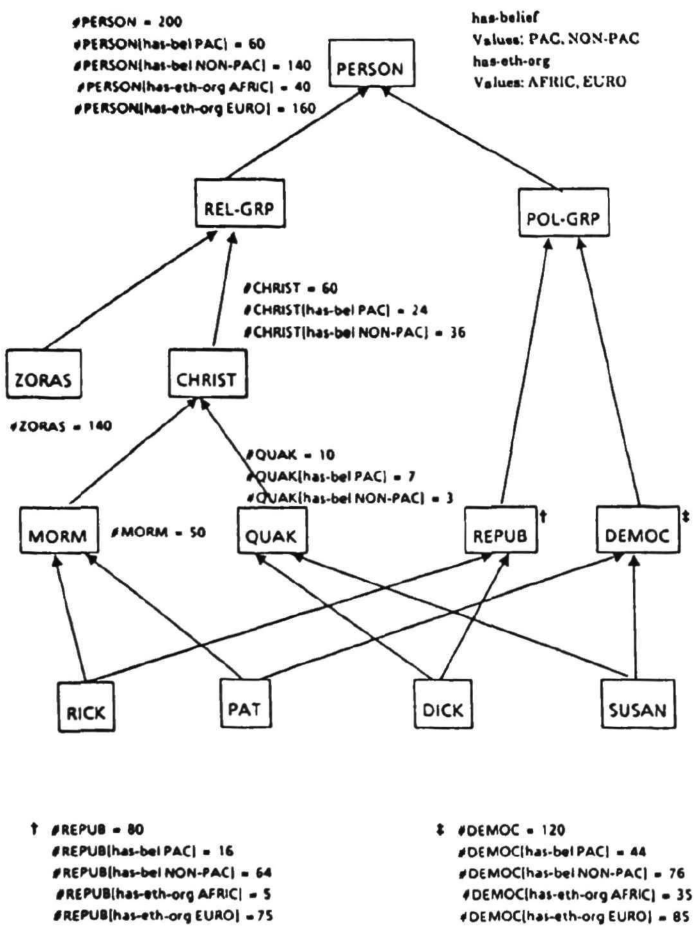


FIGURE 7: The Quaker example

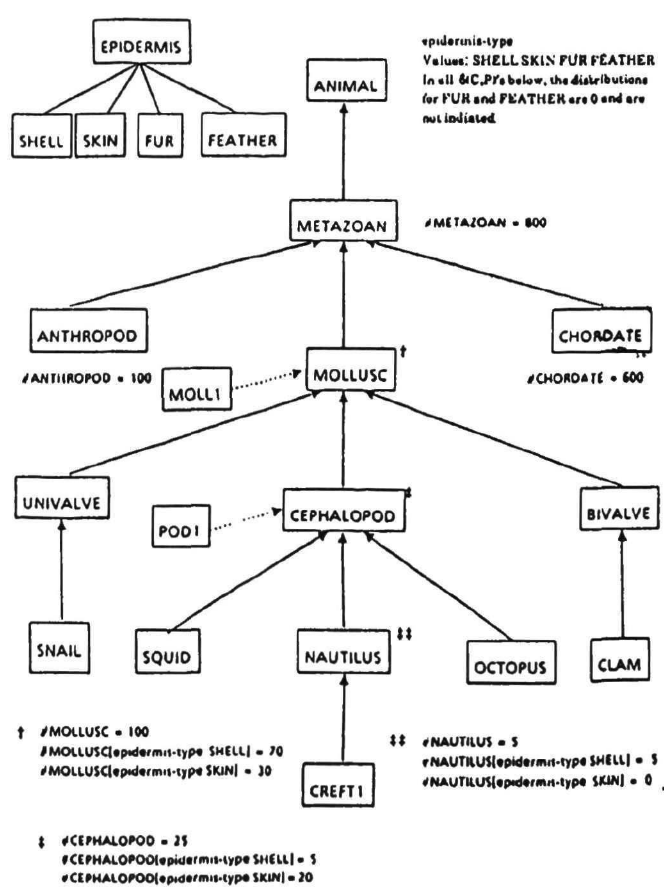


FIGURE 8: The Mollusc example