Explanation and Connectionist Systems

Joachim Diederich[1]

TR-89-016

April 3, 1989

Abstract

Explanation is an important function in symbolic artificial intelligence (AI). For example, explanation is used in machine learning and for the interpretation of prediction failures in case-based reasoning. Furthermore, the explanation of results of a reasoning process to a user who is not a domain expert must be a component of any inference system. Experience with expert systems has shown that the ability to generate explanations is absolutely crucial for the user-acceptance of AI systems (Davis, Buchanan & Shortliffe 1977). In contrast to symbolic systems, neural networks have no explicit, declarative knowledge representation and therefore have considerable difficulties in generating explanation structures. In neural networks, knowledge is encoded in numeric parameters (weights) and distributed all over the system.

It is the intention of this paper to discuss the ability of connectionist systems to generate explanations. It will be shown that connectionist systems benefit from the explicit encoding of relations and the use of highly structured networks in order to realize explanation and explanation components. Furthermore, structured connectionist systems using spreading activation have the advantage that any intermediate state in processing is semantically meaningful and can be used for explanation. The paper describes several successful applications of explanation components in connectionist systems which use highly structured networks, and discusses possible future realizations of explanation in neural networks.

[1] International Computer Science Institute, Berkeley, California.

Explanation and Connectionist Systems

Joachim Diederich

International Computer Science Institute
1947 Center Street, Suite 600
Berkeley, CA 94704

Abstract

Explanation is an important function in symbolic artificial intelligence (AI). For example, explanation is used in machine learning and for the interpretation of prediction failures in case-based reasoning. Furthermore, the explanation of results of a reasoning process to a user who is not a domain expert must be a component of any inference system. Experience with expert systems has shown that the ability to generate explanations is absolutely crucial for the user-acceptance of AI systems (Davis, Buchanan & Shortliffe 1977). In contrast to symbolic systems, neural networks have no explicit, declarative knowledge representation and therefore have considerable difficulties in generating explanation structures. In neural networks, knowledge is encoded in numeric parameters (weights) and distributed all over the system.

It is the intention of this paper to discuss the ability of connectionist systems to generate explanations. It will be shown that connectionist systems benefit from the explicit encoding of relations and the use of highly structured networks in order to realize explanation and explanation components. Furthermore, structured connectionist systems using spreading activation have the advantage that any intermediate state in processing is semantically meaningful and can be used for explanation. The paper describes several successful applications of explanation components in connectionist systems which use highly structured networks, and discusses possible future realizations of explanation in neural networks.

1. Introduction.

Explanation is a key function in artificial intelligence systems. Explanation is used to update knowledge structures in case-based reasoning when a prediction fails, i.e. for failure-driven learning. Explanation is also used to clarify the results of a reasoning process to a user. This user is not a domain expert in many cases but has the responsibility of accepting or rejecting a solution produced by an AI system. Furthermore, explanation can be used for knowledge-intensive learning whenever a complete and consistent domain theory is given.

Explanation is also a function which is difficult to realize in unstructured connectionist systems. Neural networks have no explicit, declarative knowledge structure which allows the representation of explanation structures such as reasoning paths, explanation of expectation failures etc. However, several proposals have been made for explanation components in feedforward perceptrons and relaxation-type neural networks.

This paper is organized as follows. First, a brief overview of explanation in conventional artificial intelligence is given. Next, various proposals for explanation in feedforward perceptrons and relaxation-type neural networks are discussed, and finally several successful realizations of explanation components in connectionist systems are described. It is shown that the introduction of structure to a network, e.g. the explicit representation of relations and modular network architectures, significantly facilitates the use of explanation in connectionist networks.

2. Explanation in symbolic artificial intelligence - a brief overview.

In conventional AI, the term explanation refers to an explicit structure which can internally be used for reasoning and learning, and externally for the explanation of results to a user. In rule-based systems, for example, explanation includes intermediate steps of the reasoning process, i.e. a trace of rule firings, a proof structure etc. This structure can be used to answer "Why" questions. For example, why was solution w produced by an inference system? Because conditions x and y where satisfied after the first data entry and have led to the conclusions w and z which satisfied the condition k, and so on.

Another possible form of explanation is to search for a similar case and to present it to the user. If a user does not accept or understand a solution, the system can select a similar case which generated the same or a similar response in the past. In contrast to the first form of explanation, providing a complete inference path, the second is not complete, i.e. the user must use his background knowledge to understand the similarity between the actual solution and the analogous case.

Explanation also plays an important role in text and story understanding and defines a method to "assign a motivation to a character based on his or her actions" (Charniak

1987). This form of explanation has been realized by spreading activation and marker propagation methods, i.e. a breadth first search in a semantic network to find connections among concepts, plus a component to evaluate relations among concepts. In marker propagation systems, objects in an "is-a" hierarchy are indexed by a spreading activation process and are used to build explanations. Charniak (1987) gives the following example:

> "So, upon seeing a sentence like "Jack got some milk" we might suggest explanations like "He will eat cereal" "He will drink the milk" etc. Presumably we know that milk is put over cereal, and that milk is a beverage, and beverages are typically used for drinking. Thus it seems reasonable to index activities by the objects that get used in them ... and then, given an action like Jack's getting milk, look at milk, and the things above milk in the is-a hierarchy for actions which are indexed there."

Explanation also plays a key role in machine learning. For example, it is a key function in case-based reasoning to explain failure of expectations. This is always the case when a situation does not conform to a prior case. The new situation has to be classified, and the discrepancy between the predictions and the actual event triggers learning (also see Slade 1988). Learning means updating one or more knowledge structures and requires explanation which itself is an *explicit* structure.

The term explanation-based generalization (EBG) is used by Mitchell et al. (1986) and refers in symbolic artificial intelligence to knowledge-intensive learning methods which require only a single example and use domain knowledge to constrain search for a possible generalization. This involves generating a new chunk of knowledge which describes a set of features including the properties of the training example. EBG is a two-step learning method: First, construct an explanation of why an example fits a particular *goal concept* and find those features of the training example which are relevant to satisfy this goal concept. Second, search for sufficient features of the example to build the general concept definition, the goal of the method. EBG, however, requires a correct, complete and consistent domain theory, a pre-condition which is unrealistic for real-world applications. Consequently, efforts were made to use weaker domain models (see Lewis 1988 for a discussion).

In addition, explanation is absolutely crucial for the user acceptance of an inference system. Experience with expert systems has shown that most users demand an explanation of a result produced by an expert system and do not accept a solution without explanation (Davis, Buchanan & Shortliffe 1977). Consequently, efforts were made in the expert systems area to allow explanations which are meaningful to a user **who is not a domain expert** in many cases.

3. Proposals for explanation components in neural networks.

First of all, is explanation really necessary in connectionist systems? An extreme position is to have full confidence in training, i.e. the learning process of the neural network, in particular when a huge set of training examples is available (a few hundred thousand instances for example) and the network is allowed to learn a "complete" domain theory. In this particular case, a very powerful learning technique would be sufficient and there would be no need for explanation at all. The network could easily recognize previously-seen patterns and generalization should be reliable, too. However, these assumptions are certainly unrealistic for real-world applications. Natural environments are continuously changing; the reasoner has to deal with incomplete and inconsistent information; learning may be interrupted; and training has to be finished in a reasonable period of time. Even a perfect training procedure does not affect the need for user explanation, which remains a key function for any inference system.

Explanation of the kind described in section 2 is not easy to realize in relaxation-type neural networks. Artificial neural systems have no explicit, symbolic and declarative knowledge representation. Instead, in connectionist systems knowledge is captured in a numeric weight matrix distributed over the entire system, a distribution of weights which in most cases was learned by training procedures such as backpropagation, Boltzmann machine learning etc.

This limits the range of possible explanation methods in neural networks significantly, e.g. when a solution has to be explained to a user. Even if the weight matrix is accessible to the user, it is close to meaningless, since the user has no way to "decompile" the weights.

Charniak (1987) has shown the lack of explanatory power in connectionist systems by describing the so-called "infer-everything" problem. Explanation in Charniak's terms means to generate a picture of a situation including the acting persons and their possible motivation. In order to guarantee that such a picture of a situation can be produced, the connectionist system must have either nodes which represent all possible combinations of facts (because it could be important for explanation) or the system must have the generative power to produce explanations. If the latter is done by pattern completion in a relaxation-type system, **all** features of a schema would be produced and not only those features which make up a good explanation.

Another problem is that there is no well-known way to allow meta-reasoning in neural networks. Meta-reasoning means inferencing about the reasoning capabilities of a system. Meta-reasoning can be important for the ability to give explanations since a system must know about the limits of its domain knowledge and inferencing capabilities in order to give a meaningful explanation of results.

However, several suggestions have been made about how to allow explanation in relaxation-type connectionist systems despite their severe limitations. One idea is based on an assumed analogy between the search of rule-based systems in problem-space and the search of a relaxation-type connectionist system in state-space. Both problem-solving processes are based on a specified starting point (the input) and an exit situation which is the satisfaction of a goal criterion in the first case and a local or global minimum in the second case. One of the important differences is that it is easy to keep a trace of the path of a rule-based system through problem space, but it is extremely difficult to keep track of all the changes in a relaxation-type connectionist system.

The idea of an analogy between problem-space and state-space must be rejected. In general, a connectionist system can settle in various local minima while it is unknown in most cases how many local minima the system has. Even if the number of minima is known, how can one **explain** why a particular solution was produced and not a similar one? Furthermore, it is not guaranteed that a user can realize similarities among solutions produced by a relaxation-type neural network. It might be that the user has no way to understand the similarity based on the state of his domain and background knowledge.

Simulated annealing (for example in non-deterministic Boltzmann machines) makes explanation even more difficult. The behavior of the systems is governed by the statistical properties of the optimization method plus random factors. The system is performing steepest descent including the possible crossing of energy barriers. In other words, the system changes its states and settles into a stable solution (eventually), but this might include movements "uphill" the energy landscape and apparently random behavior. Explanation must therefore be based on an "idealized" behavior of the system and not on a particular path. This idealized behavior could be the energy landscape itself. Key points where energy significantly changes could be used for explanation (J. Feldman 1989, personnel communication).

The situation is even worse in feedforward-perceptrons such as simple backpropagation networks (i.e. non-recurrent networks). Classification is done in a simple forward pass of activation, which is extremely efficient on the one side but leaves no room for explanation on the other side. There are no intermediate steps in classification which can be used for explanation. Instead, classification means to present an input-vector to the systems (e.g. a set of features) and to get the immediate response. The user has no choice but to trust the systems engineer and to believe that learning was done correctly with a sufficient set of training examples and that the system is reliable.

The situation is different when relearning is done in feedforward perceptrons (G. Hinton 1988, personnel communication). Consider a multi-layer network which has learned domain knowledge and new facts are presented for learning. According to Hinton (1988) this will lead to major changes in weight matrix of the network where the new facts differ from the previously learned knowledge. These changes can be monitored by an outside component and can be used for explanation. Again, this requires full confidence in training and only makes sense in those cases where the network is forced to learn local representations; the localization of changes would be almost impossible otherwise. Furthermore, it is an open question as to which kind of explanation can be supported by monitoring changes in weight space.

The second possible way to give an explanation, presenting a similar case, can be much easier for a connectionist system. This can be possible in principle, since the representation of schemas and events in distributed systems is similarity-based by its nature.

We conclude from the discussion above that explanation is difficult to realize in relaxation-type connectionist systems and simple feedforward-perceptrons. The next sections discuss explanation and learning in connectionist systems with a special emphasis on straight-forward spreading activation networks and structured connectionist systems.

4. Explanation in structured connectionist systems.

This section discusses the suitability of connectionist semantic networks (CSNs) for explanation and gives a brief summary of connectionist systems which have realized various types of explanation. It should be noted that spreading activation and marker propagation systems have been used very successfully as part of explanation components (I gave an example from Charniak (1987) in section 2). Marker propagation models are massively parallel reasoning systems which use a heuristic reasoning component (the "path evaluator") for the analysis of paths of activated concepts. The following discussion excludes these kind of models and emphasizes interpreter-free connectionist models with weighted connections and bit- or value-passing.

4.1. Explanation and Connectionist Semantic Networks.

CSNs are better suited for explanation than plain relaxation-type networks because of their explicit structure, e.g. the inheritance hierarchy. Straight-forward spreading activation systems have the advantage that *each state in processing is meaningful and intermediate states can be used for processing*. The use of explicit structures in connectionist networks is crucial in this context.

The kind of explanation which can be given by a connectionist semantic network is determined by the type of relations encoded. For example, in a standard CSN using subsumption relations of concepts, the activation of attributes and values is completely determined by the architecture of the concept hierarchy. This means that only the subclass relation can be used for explanation, and not arbitrary relations as in logic-based systems (there are only a few connectionist systems which allow the representation of n-ary predicates and variable binding, e.g. Shastri, L. & Ajjanagadde 1989).

The following is a brief description of a CSN. The network used in Diederich (1989a,b) has five modules; each module is an n-layer network with mutual excitatory connections between layers (in both directions) and inhibitory connections within layers. This architecture is similar to the interactive activation model in McClelland & Rumelhart (1981). Each layer is a specialization of a "winner take all" (WTA) network. Competition among units in a layer results in the strong activation of a winner unit, but several units might be active simultaneously if they receive strong outside excitation. Initially, there are small random weights in this network: weights within a layer are inhibitory [0, -10] (weights are integers [-1000, 1000], in general); weights to the layer above are initially excitatory [0, 10], as are weights to the layer below [0, 100]. Recruitment learning can change random initial connections between layers to strong negative or positive weights.

The total network has fives "spaces," i.e. four network modules with the architecture described above, and an additional single space containing a set of units without internal organization (the instance space). In detail, there is a space for the representation of structured *objects*, a space for the representation of *attributes*, a space for the representation of *values* of attributes, the *instance* space mentioned above and a *context* space for the representation of episodic information (not used in the learning by instruction process). All representations are localist; there is a single unit for the representation of each structured object, attribute, value, instance and context.

Units are not only connected to units in their own space, but also to units in other spaces. Object, attribute and value units are connected by *binder units*. A binder unit has three sites where input-lines come in, and it has an activation function which requires input from at least two sites for a positive output. There is one single binder unit for each connection between an object, attribute and value.

Units with an assigned meaning build a hierarchy, i.e. object, attribute, value and context units are embedded in a multi-layer network and form a straight-forward spreading activation network. This spreading activation network is used for reasoning. Inheritance, for instance, can be described by the following example: the object unit A is activated by some input. A will activate all units in the attribute and value space associated with A and the more general unit B. B itself will also activate its properties in the attribute and value space; therefore additional feature units become "on."

This process corresponds to simple inheritance in semantic networks. While the spreading activation process continues, more and more general object units are activated and more and more relevant feature units are turned on. Exceptions which would realize a non-monotonic style of reasoning are built in by additional inhibitory links which modify the flow of activation from more general concept units to units in the attribute space. These links modify the inheritance process. More specific units can modify the effect that links from more general object units have on units in the attribute space. So the connections between more specific objects and their attributes become dominant. This realizes a basic assumption of semantic networks: the more specific should dominate the more general.

This brief description demonstrates why each intermediate state in a spreading activation CSN is meaningful. The input causes a flow of activation along the links of the inheritance hierarchy which corresponds to a sequential application of the superclass/subclass relation plus property retrieval. Each intermediate state is the result of the use of this relation. Furthermore, it is possible to record and replay intermediate states during processing by use of recruitment learning, i.e. patterns of activation are "frozen" into the weight of a single unit representing a particular time step (the weight is set to the input value at the corresponding time step).

Recruitment learning in the context of sequential processing in connectionist networks was first described by Fanty (1988).

These methods are probably not sufficient to realize explanation in CSN using simple inheritance (i.e. no cancellation links), but two important conditions for explanation are given and can be used for future explanation components: semantic meaningful intermediate states and a possible replay of inferencing (or parts of the reasoning process) with various degrees of granularity.

4.2. Explanation in connectionist diagnostic problem solving.

Peng & Reggia (1989) and Wald, Farach, Tagamets & Reggia (1989) describe a connectionist model for diagnostic problem solving including abductive reasoning. The method is competition-based, using a "multiple winner take all" approach which allows several "winner units" as the result of the competition among units with mutual inhibitory connections. A similar approach, based on a Hopfield-type network, can be found in Goel, Ramanujam and Sadayappan (1988).

Abduction denotes to the task of infering a hypothesis that best explains data (Goel, Ramanujam and Sadayappan 1988). This inference class is computationally difficult in that multiple disorders may occur simultaneously and a global minimum in the space exponential to the total number of possible disorders is sought as a solution (Peng & Reggia 1989), i.e. the diagnostic problem is viewed as a non-linear optimization problem. Peng & Reggia's (1989) approach is conceptually based on the *parsimonious covering theory*. In this theory, there is a set of disorders D, a set of manifestations M, and a relation $D \times M \supseteq C$ representing the causal association between a set of disorders and manifestations. A pair $<d_i, m_j>$ is in C iff "disorder d_i may cause manifestation m_j."

Based on the relation C, two sets, "effects" and "causes," are defined for each $d_i \in D$ and each $m_j \in M$. A hypothesis D_i represents a potential explanation for a set of manifestations M^+ (i.e. all present manifestations) in that it is a set of disorders, which when present, could cause or account for M^+ (after Peng & Reggia 1989).

This model can be used as a two-layer connectionist network: D and M are two sets of nodes and C is the set of connecting links. Each link $<d_i, m_j> \in C$ has a constant weight representing the causal strength. The model converges to a set of winners when relaxation reaches equilibrium at time t_e. In other words, for each d_i, $d_i(t_e)$ approximates either 1 or 0, then the set of disorders $D_S = \{d_i | \ d_i(t_e) \approx 1\}$ is taken to be the connectionist's model problem solution.

The performance of this network, i.e. both accuracy and efficiency, depends on the activation rule chosen. Wald, Farach, Tagamets & Reggia (1989) used a slightly different approach based on "simulated annealing" in order to guarantee globally optimal solutions. The important point for the discussion here is the ability of Peng & Reggia's (1989) network to *generate* explanations. An explanation is a set of "disorder" which account of a set of observed manifestations (symptoms). The causal strengths between manifestion and disorders, however, were not learned but randomly generated in Peng & Reggia (1989) and provided by physicians in Wald, Farach, Tagamets & Reggia (1989).

Peng & Reggia's (1989) approach is important for the discussion here because it allows the system to respond to "Why" questions (in principle). Given a number of manifestations, the system responds with a number of disorders explaining "why" these manifestations were produced. Unfortunately, the activation levels of winner "disorder" units approximates 1 (the maximum value) and it is not possible to treat the output as probabilities. A possible explanation of results would be much more detailed and meaningful if it were possible to explain to which degree manifestations have contributed to the result, i.e. generated disorders. This is not yet possible in Peng & Reggia's approach.

4.3. "Explanatory Coherence" in connectionist networks.

The next example is a structured connectionist system for the modelling of "explanatory coherence" (Thagard 1988), i.e. the selection of a set of hypotheses that is best explained by evidential data, direct observation and other hypothesis. The generation of internal coherence among these propositions can be modelled by a connectionist system.

Coming from the theory of scientific explanation, Paul Thagard (1988) describes a computational theory of "explanatory coherence" that applies to the acceptance or rejection of sets of scientific hypothesis (or "propositions"). Propositions compete to build stable coalitions, i.e. sets of consistent and non-contradictory hypothesis which are best explained by evidential data, direct observation and other hypothesis. The theory consists of seven principles which establish relations of local coherence among propositions. According to Thagard (1988) "a hypothesis coheres with propositions that it explains, or that explain it, or that participate with it in explaining other propositions, or that offer analogous explanation." Propositions that describe direct observation have acceptability in their own. An explanatory hypothesis is accepted if it coheres better overall than competing propositions.

Thagard's (1988) theory consists essentially of seven relations which establish coherence among propositions. They are symmetry, explanation, analogy, data priority, contradiction, acceptability and system coherence. We take two of these relations, explanation and analogy, in order to give a brief example for coherence relations. See Thagard (1988) for a complete description.

The explanation principle is described by Thagard (1988, p.3) in the following way:

If P_1 ... P_m explain Q, then:

(a) For each P_i in P_1 ... P_m, P_i and Q cohere.
(b) For each P_i and P_j in P_1 ... P_m, P_i and P_j cohere.
(c) In (a) and (b) the degree of coherence is inversely proportional to the number of propositions P_1 ... P_m.

The analogy principle is explained this way:

(a) If P_1 explains Q_1, P_2 explains Q_2, P_1 is analogous to P_2, and Q_1 is analogous to Q_2, then P_1 and P_2 cohere, and Q_1 and Q_2 cohere.
(b) If P_1 explains Q_1, P_2 explains Q_2, Q_1 is analogous to Q_2, but P_1 is disanalogous to P_2, then P_1 and P_2 incohere.

How are relations of the kind described above translated to a connectionist model and how does it work? First of all, there is a primitive high-level description language which allows expression such as (EXPLAIN (H1 H2) E1) and (CONTRADICT (H1 H2)) which means that hypothesis H1 and H2 both explain evidence E1, but H1 and H2 contradict eachother. These expressions are compiled into a connectionist network where each proposition is represented by a single unit. If two proposition cohere, then there is an excitatory, symmetric link between them with positive weight. Consequently there is an inhibitory link if two propositions incohere. Data priority is implemented by an excitatory link from a special data unit.

Activation values greater than 0 signify acceptance of a proposition (Thagard 1988, p.13) and the coherence of a whole system of propositions at time t is validated by a function which is the inverse to "energy" or "harmony" function in neural networks

$$H(t)=\sum_i\sum_j w_{ij}a(t)a_j(t)$$

where w_{ij} is the weight from unit i to unit j, and $a_i(t)$ is the activation of unit i at time t.

Running the network means generating a coherent set of propositions (if this set is available) represented by a stable pattern of activation over propositional units. This stable coalition dominates other possible sets of hypothesis with less explanatory coherence. Thagard (1988) used this method for a number of simulations in the area of scientific explanation and reasoning, for example Lavoisier's argument for oxygen against phlogiston theory, Darwin's argument for evolution and against creationism and a number of cases of legal reasoning. The most complex application so far (about Copernicus' heliocentric theory) involved 150 units 210 cycles for a stable result.

Thargard (1988) demonstrates very nicely how competition can be used to built coherent sets of explanations, but Thagard does not explain how a single explanation can be produced by a connectionist network.

5. Conclusion.

We gave a brief overview of "explanation" in symbolic AI and discussed several proposals for the realization of explanation components in connectionist systems. It has been shown that connectionist systems benefit from the explicit encoding of relations and the use of highly structured networks. Structured connectionist systems using spreading activation have the advantage that any intermediate state in processing is semantically meaningful and can be used for explanation, in principle. It was also shown that connectionist systems contribute to explanation approaches which require multiple constraint satisfaction in problem solving.

6. Acknowledgement.

Research described in this paper has benefited from discussions with Larry Birnbaum, Joachim Buhmann, Jerry Feldman, Jim Hendler, Geoff Hinton, Christoph Lischka, Charles Rosenberg, Andreas Stolcke, and many others.

References

Charniak, E. (1987): *Connectionism and Explanation.* Paper presented at TINLAP-3, New Mexico State University, Las Cruses, January 1987.

Davis, R., Buchanan, B.G. & Shortliffe, E. (1977): *Production Rules as a Representation for a Knowledge-Based Consultation Program.* Artificial Intelligence 8 (1), 15-45.

Diederich, J. (1988a): *Connectionist Recruitment Learning.* ECAI-88, Proc. of the 8th European Conference on Artificial Intelligence, Munich.

Diederich, J. (1988b). *Steps toward knowledge-intensive connectionist learning.* To appear in: Pollack, J. & Barnden, J. (Eds.): Advances in Connectionist and Neural Computation Theory. Ablex Publ.

Diederich, J. (1988c): *Knowledge-Intensive Recruitment Learning.* ICSI-TR-88-010, International Computer Science Institute, Berkeley, November 1988.

Fanty, M. (1988): *Learning in Structured Connectionist Networks.* Ph.D Thesis, CS Department, University of Rochester.

Goel, A., Rmanujam, J. & Sadayappan, P. (1988). *Towards a neural architecture for abductive reasoning.* Proceedings of the 2nd IEEE International Conference on Neural Networks. Vol. 1, 681-688.

Lewis, C. (1988): *Why and How to Learn Why: Analysis-based Generalization of Procedures.* Cognitive Science, 12, 211-256.

Mitchell, T.M., Keller, R.M. & Kedar-Cabelli, S.T. (1986): *Explanation-Based Generalization: A Unifying View.* Machine Learning 1, 47-80.

Peng, Y. & Reggia, J.A. (1989). *A Connectionist Model for Diagnostic Problem Solving.* IEEE Transactions on Systems, Man and Cybernetics. In press.

Pitt, J.C. (Ed.; 1988): *Theories of explanation.* New York : Oxford University Press.

Slade, S. (1988): *Case-based reasoning: A Research Paradigm.* YALEU/CSD/RR #644, Yale University, New Haven, Conn., August 1988.

Shastri, L. (1988): *Semantic Networks: An Evidential Formalization and its Connectionist Realization.* Morgan Kaufman Publ., San Mateo, California 1988.

Shastri, L. & Ajjanagadde, V. (1989): *A Connectionist System for Rule Based Reasoning with Multi-Place Predicates and Variables.* MS-CIS-8905 & LINC LAB 141, Computer and Information Science Department, University of Pennsylvania, January 1989

Thagard, P. (1988): *Explanatory coherence.* Final draft for commentators. Behavioral and Brain Sciences. Cambridge University Press. December 1988.

Wald, J., Farach, M., Tagamets, M. & Reggia, J.A. (1989): *Generating Plausible Diagnostic Hypotheses with Self-Processing Causal Networks.* Journal of Experimental and Theoretical Artificial Intelligence. in press.