EBL and SBL: A Neural Network Synthesis

Bruce F. Katz
The Beckman Institute for Advanced Science and Technology
University of Illinois

ABSTRACT

Previous efforts to integrate Explanation-Based Learning (EBL) and Similarity-Based Learning (SBL) have treated these two methods as distinct interactive processes. In contrast, the synthesis presented here views these techniques as emergent properties of a local associative learning rule operating within a neural network architecture. This architecture consists of an input layer, a layer buffering this input, but subject to descending influence from higher order units in the network, one or more hidden units encoding the previous knowledge of the network, and an output decision layer. SBL is accomplished in the normal manner by training the network with positive and negative examples. A single positive example only is required for EBL. Irrelevant features in the input are eliminated by the lack of top-down confirmation, and/or by descending inhibition. Associative learning then causes the strengthening of connections between relevant input features and activated hidden units, and the formation of "bypass" connections. On future presentations of the same (or a similar) example, the network will then reach a decision more quickly, emulating the chunking of knowledge that takes place in symbolic EBL systems. Unlike these programs, this integrated system can learn in the presence of an incomplete knowledge domain. A simulation program, ILϰ, provides partial verification of these claims.

INTRODUCTION

Learning is, and always has been, central to connectionist models of cognition. Numerous adaptive rules have been proposed that, in the context of their respective architectures, are able to improve the network's performance through observation of examples characteristic of a given domain. Although far removed in sophistication from Mill's (1843) system of induction, all such strategies are designed, like his, to extract the regularities by which similar causes are predictive of similar effects. Machine learning classifies such techniques, for obvious reasons, as Similarity-Based Learning. SBL continues to be of prime importance in both connectionist and "symbolic" models of intelligence.

Recently, however, non-connectionist learning research has placed equal emphasis on Explanation-Based Learning, and other more knowledge-intensive methods (DeJong & Mooney, 1986). In the classical formulation of the EBL problem (Mitchell, Kellar, & Kedar-Cabelli , 1986), one is given a set of domain rules, a training example, and a goal that can be inferred by the application of the domain knowledge to the example. An explanation structure is then constructed, with the input features at the leaves of this tree, and the goal node at the top. This structure may then be generalized using goal regression or other related techniques (Mooney & Bennet, 1987). The resulting structure may then be "flattened", so that a new rule is formed with the left hand side being the generalized example, and the right hand side the original goal. If the left-hand side is easily observable, then one will have a quick and easy way of predicting the goal concept given the appropriate inputs, without the need to produce what may be an extensive inference chain. To take a simple example, let us assume one knows that all professors are absent-minded, and that all absent-minded people misplace things. Suppose one sees Professor X misplacing his glasses. One forms the explanation of this event, and one emerges in the end with the general rule that professors will tend to misplace things. One may question the role of the example in the above, since, from a strictly logical point of view, it is unnecessary. The standard

response to this objection (Mitchell, Kellar, & Kedar-Cabelli, 1986) is that the example indicates which type of knowledge it may be profitable to chunk; the full deductive closure of one's current knowledge is not readily computable given spatial and temporal limitations.

EBL, then, differs primarily with SBL in that it is a knowledge intensive approach. It eliminates features irrelevant to the classification task not by noticing their joint occurrence in both positive and negative examples, as there is typically only one positive example, but by noting which features are necessary for the generalized explanation. E.g., in the above example, the fact that Professor X's specialty was medieval history was not part of the explanation structure, and was therefore deemed irrelevant.

Theoretical parsimony alone would suggest the desirability of unifying both EBL and SBL in a single system. However, there is another concern which is of equal importance. Classical EBL can only work when the domain under study is complete; i.e., there is always an unbroken chain of inference from the example to the goal (Rajamoney & DeJong, 1987). Such a restriction seems overly stringent, and is unlikely to be met in many common situations. It would be desirable to have SBL patch the missing links in a partially broken inference chain. In addition, it would also be highly advantageous to induce primarily over the end-products of an inferential system, rather than raw input features. Think of learning from written text -- clearly, very little learning is occurring at the pixel or letter level; most if not all learning is ideational.

The symbolic learning literature offers a few approaches to integrated learning. Among these are OCCAM (Pazzani & Flowers, 1987), Liebowitz's (1986) adaptation of UNIMEM to integrated learning, and Danyluk's (1987) interactive approach. All of these systems, however, are decomposable into separate EBL and SBL modules. The purpose of this work is to show that a neural network model can account for both types of learning as *emergent* properties of a local adaptive rule operating in a particular architecture. A simulation program, IL*, is presented which partially verifies this claim.

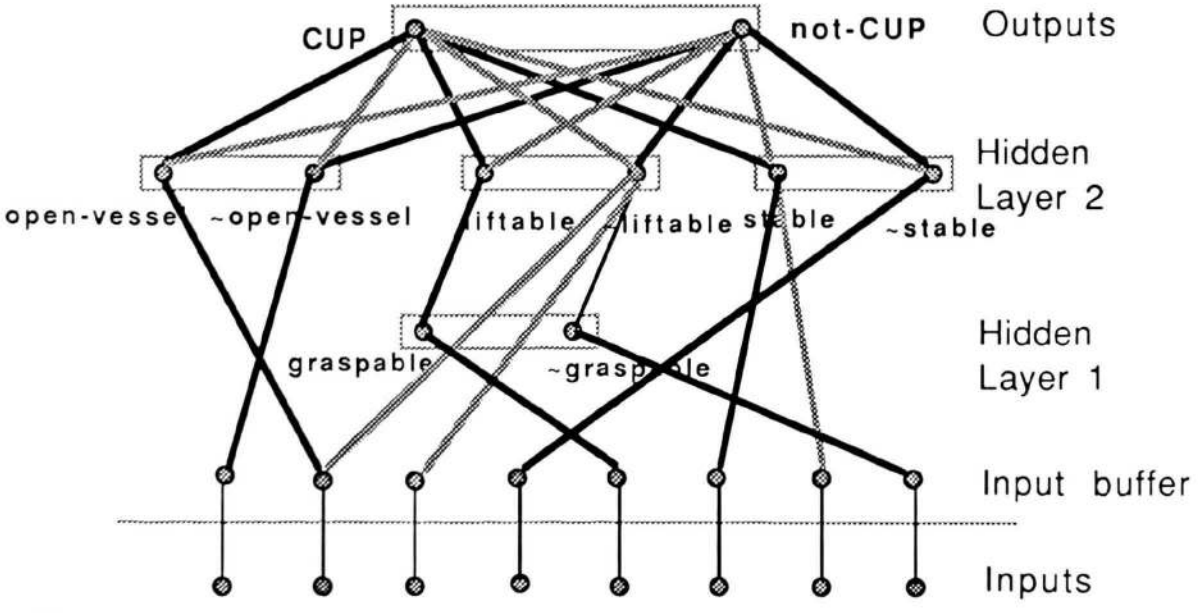

FIGURE 1. A SIMPLIFIED VIEW OF THE SYSTEM ARCHITECTURE

KATZ

ARCHITECTURE

A sample network in IL$\aleph$ (for an example which is discussed more fully later) is shown in figure 1. Input nodes are activated by features in the environment. These inputs are buffered by another layer, with one node for each corresponding node in the input layer. Unlike input nodes, which are clamped on or off by the environment, nodes in the input buffer may be affected by top-down control. This will prove important in the mechanism for EBL.

Activation flows from the input buffer to sets of nodes in one or more hidden layers. Solid lines represent excitatory connections, while shaded lines represent inhibitory connections. In addition, the dotted boxes in the figure are shorthand representations for sets of mutually inhibitory nodes at the same layer. Nodes in these layers also have excitatory connections to themselves. This sub-architecture has been shown to produce winner-take-all networks (Rumelhart & Zipser, 1986), that is, the node in the set receiving the most activation will reach maximum value, while all others will be driven to zero activation. Activation spreads in parallel in all directions until one unit in the output layer "wins" and becomes the decision. In this case, the network decides whether the input is a cup or some other object. The relaxation process is described more fully in the next section.

INFERENCE

Inference is accomplished by the spread of activation. The activation of a unit is a weighted sum of its inputs, as in (1). In this equation, a_i represents the net activation level of unit i, w_{ij} is the weight between units i and j, and o_j is the output of unit j.

$$a_i = \Sigma\, w_{ij}\, o_j \qquad (1)$$

Weights may be either positive (excitatory), or negative (inhibitory), and are unbounded. In contrast, the output of a unit is held between 0 and 1 by the sigmoidal function in equation (2). In this formula, T is a free parameter representing the "temperature" (cf., Hinton and Sejnowski, 1986) of the network, and θ is a constant threshold. Lower temperatures make it more likely for a unit to reach extremal values at relaxation, while the threshold controls the amount of activation a unit needs to fire.

$$o_i = 1 / (1 + \exp(-(a_i - \theta) / T)) \qquad (2)$$

In addition to bounding a unit's output, this non-linear function controls for noise at sub-threshold activation levels (Rumelhart, Hinton, & Williams, 1986).

Activation propagates throughout the network, until the network reaches a steady state. Hopfield (1985) has shown that networks with symmetric weights (which are currently used exclusively) are guaranteed to converge to a fixed point. In addition, if the temperature in (2) is sufficiently low, one node in a group of competing nodes will always "win", and the network will make a discrete decision.

THE LEARNING RULE

In this section, a learning rule is offered, which, in conjunction with the architecture in IL$\aleph$, performs both SBL and EBL. The starting point for the development of the learning rule is Hebb's (1949) observation that simultaneous activity of two units indicates that the weight between these units should be strengthened. This extensively used rule is shown in equation (3), where the change in weight between units is equal to the product of the outputs of the nodes at a given time multiplied by a learning rate constant, λ. The second term in (3) allows unlearning of connections and the development of inhibitory connections.

$$\Delta\, w_{ij} = \lambda\, o_i\, o_j - \delta\, |o_i - o_j| \qquad (3)$$

For reasons that will be made clearer in the next section, it is desirable that the network learn only after relaxation, as a means of controlling spurious correlations. One simple way to do this is to only apply (3) after the network relaxes. However, this would require a global "homunculus" watching the network that tells each unit when to learn. One local solution to this difficulty, and the one adopted here, is to divide the right hand side of (3) by the function

$$\mathfrak{D}(o_i, o_j) = \begin{array}{ll} 1 & \text{if } |d(o_i)/dt| + |d(o_j)/dt| < \varepsilon, \text{ and} \\ 1000 & \text{otherwise.} \end{array} \quad (4)$$

Thus, only when both units are no longer changing will the weight change be significant. It should be noted that (3) is capable of learning conjunctive concepts only; no disjuncts must appear in the target concept. Learning is not limited to classifying orthogonal input patterns, however, as is typical in Hebbian schemes (Jordan, 1986), because of the winner-take-all decision procedure.

EXPLANATION-BASED LEARNING

While Hebbian associative learning is a clear candidate for SBL, its performance on EBL tasks is less established. Figure 2 is a highly schematic view of how EBL is accomplished in IL$\aleph$ using the learning procedure outlined above. Panel **A** represents the state of the network before relaxation. Note that the input buffer is a veridical representation of the input vector. Panel **B** represents the network after relaxation. Descending inhibition has turned off the two rightmost units in the input buffer (the threshold in equation 2 can also be adjusted so that merely the lack of excitatory confirmation also results in a dampened unit.). The network has "decided" that these features were not crucial in the determination of the final decision, or in the final activated state of the intermediate units leading to this decision. It is suggested that the process of moving from **A** to **B** is equivalent, in effect, to forming a proof structure of the goal concept from the inputs in that previous knowledge is used to weed out irrelevant attributes in the data. Unlike symbolic techniques, where relevance is determined by the explicit computation of a proof structure, in IL$\aleph$ it is an emergent property of top-down attentional control. Like its symbolic counterpart, though, this method can profit by a single positive example, since large numbers of positive and negative examples are not needed to determine relevant features.

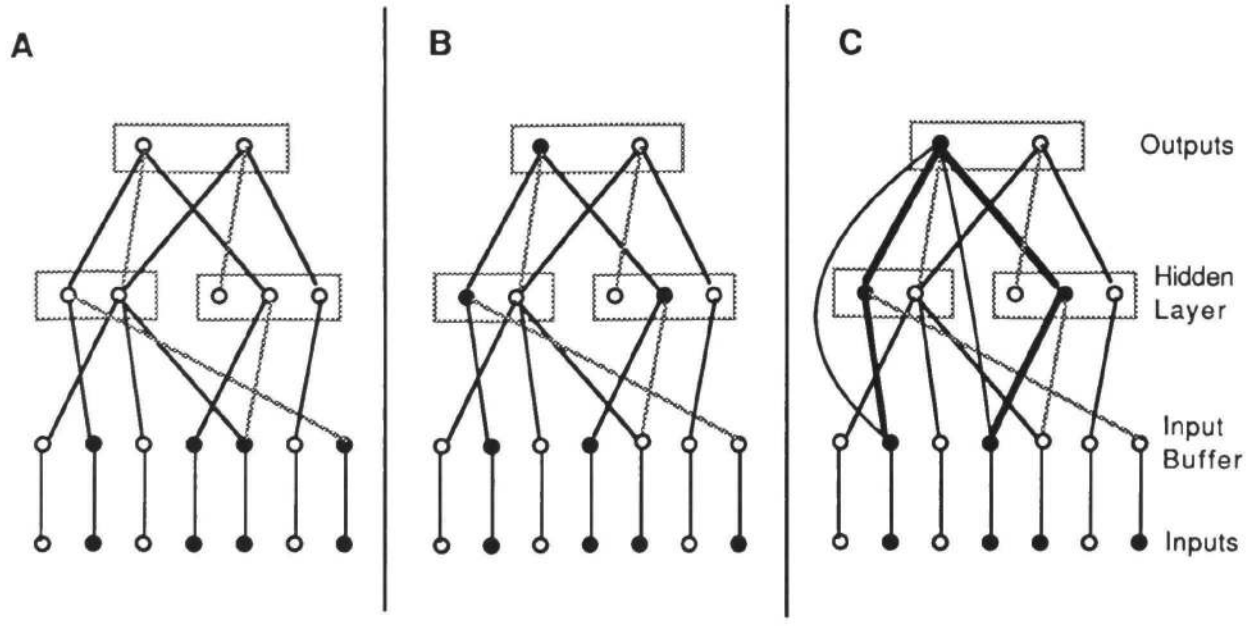

FIGURE 2. EBL IN IL$\aleph$

Panel C represents the state of the network after learning. Recall that the learning rule constrains the network to learn primarily after relaxation. Thus no correlations are made between data that was originally present in the input buffer, but turned off during relaxation (no direct learning is permitted from the input layer to other layers). Existing connections between units active at relaxation are strengthened, and new connections may also form. These "bypass" connections can be seen in C as new lines between the input buffer and the activated output node. These new connections, along with strengthened old ones, cause the network to relax at a much faster rate given a similar input pattern. This occurs because the competition time between sets of mutually inhibitory nodes (those in the dotted boxes) is proportional to the difference in activation values of these nodes, and the strengthened and bypass connections increases this difference.

Existing EBL algorithms include a step in which the proof structure is generalized. In the current model, this type of generalization is a side-effect of the activation of higher-order nodes. For example, in an EBL task involving Clyde the elephant, if the mammal unit receives top-down confirmation and fires, its connections to other units would change in a manner similar to the connections emanating from the elephant unit. The system would then reap the rewards of the earlier training with Clyde in a similar context involving Sam the giraffe (also a mammal).

EXPERIMENTAL RESULTS

In the following experiment, eight examples of cups and non-cups (common household objects) were used. Horn-clause rules for cup recognition (as found in the EBL literature) were translated directly into the network in Figure 1. The connections were hardwired such that the network gave the correct response on each example. The examples were presented to the network in random order, and the number of synchronous cycles until network relaxation was measured for each example. The graph in figure 3 summarizes these results. Initially, the network took 19 cycles to relax; after the presentation of 200 examples, this figure was reduced to 7 cycles. An effect similar to rule compilation in EBL was achieved by the formation of bypass connections (and strengthened connections) in a neural network. Irrelevant features in the input pattern did not enter into learning, as they were in low states of activation at relaxation due to the lack of descending confirmation.

The second experiment focused on the relation between recognition errors and the completeness of the knowledge domain. Two cases were studied: the cup domain discussed above, and a randomly generated boolean formula with three disjunctive terms. They were examined under three conditions: full prior knowledge, partial prior knowledge, and no knowledge prior to learning. Table 1 summarizes these results. Naturally, in both cases, the complete domain yielded no

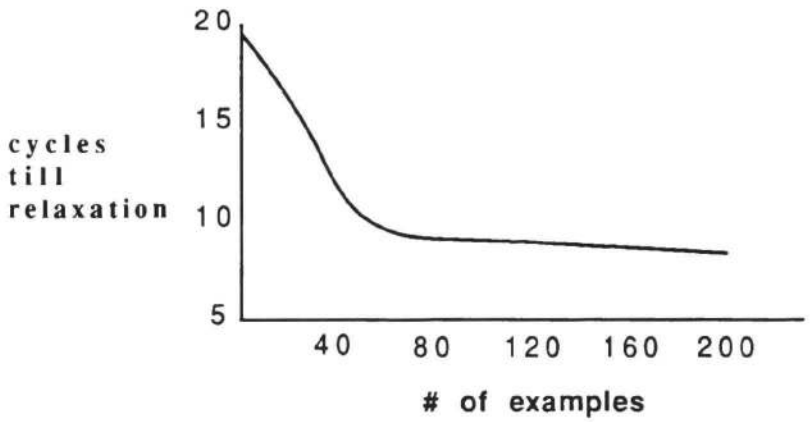

FIGURE 3. RECOGNITION TIME DECREASE IN A COMPLETE DOMAIN

recognition errors, and only one sweep through all the examples were needed to verify this. In the no knowledge case, no prior connections between the input buffer and the output layer existed, and no hidden units were used in the cup case. For the boolean formula, five hidden units were in place, and a modified reinforcement learning procedure, similar to Barto and Anandan's (1965) associative reward-penalty algorithm, that is capable of acquiring disjunctive concepts, was used. In the partial knowledge case, connections between the input buffer and hidden units were in place, as in figure 1, but the connections from the hidden units to the output units were removed. Note that this represents one of the traditionally difficult cases for EBL, that of an incomplete domain. The partial knowledge helped the network outperform, to a small extent, a network with no knowledge in the cup domain. More dramatic increases in performance were seen in the boolean case, as expected, with the hidden units doing the hard work of encoding the relevant disjuncts (cf. Rivest, 1984).

TABLE 1. MEAN SWEEPS UNTIL PERFECT RECOGNITION AS A FUNCTION OF PRIOR KNOWLEDGE

	complete knowledge	partial knowledge	no prior knowledge	partial/ none
CUP	1	3.6	5.2	69%
BOOLEAN	1	11.0	34.2	32%

DISCUSSION

A neural architecture has been outlined that provides seamless integration of Similarity and Explanation-Based Learning. Not fully treated in this paper are the following issues:
a) The acquisition of disjunctive concepts (as in backpropagation, e.g.), and the relation between disjunctive concept learning and EBL.
b) No unification is performed in the current model (as in symbolic EBL systems), yielding the binding problem (see Touretzky & Hinton, 1988 for a partial solution to this problem).
c) The relation between EBL in the above model and automaticity (Schneider, 1984) needs to be further explored.
d) The relation between sequential processing in a parallel network and EBL (extended chains of inference can currently be handled only by adding a new layer to the network for each link in the chain). Ultimately, one would like to show that the mind can convert lengthy sequential procedures into easily computable boolean functions by observing its own input-output relations. This research is the first step toward suggesting that this may be possible using a purely local algorithm.

ACKNOWLEDGEMENTS

I would like to thank Bob Stepp for his patient discussion of these issues and Marcy Dorfman for her suggestions.

REFERENCES

Barto, A.G., & Anandan, P. (1985). Pattern recognizing stochastic learning automata. *IEEE Transactions on Systems, Man, and Cybernetics, 15*.

Danyluk, A.P.(1987). The use of explanations for similarity-based learning. *Proceedings of the International Joint Conference on Artificial Intelligence*. Milan, Italy.

DeJong, G., & Mooney, R. (1986). Explanation-Based Learning: An alternative view. *Machine Learning 2*.

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

Hinton, G.E., & Sejnowski, T.J. (1986). Learning and relearning in Boltzmann machines. In Rumelhart, et. al. (Eds.), *Parallel Distributed Processing, Vol. I*. MIT Press.

Hopfield, J.J., & Tank, D.W. (1985). "Neural" computation of decisions in optimization problems. *Biological Cybernetics 52*, pp. 141-152.

Jordan, M.I. (1986). An introduction to linear algebra in parallel distributed processing. In Rumelhart, et. al. (Eds.), *Parallel Distributed Processing, Vol. I*. MIT Press.

Lebowitz, M. (1986). Integrated learning: Controlling Explanation. *Cognitive Science 10*, pp. 219-240.

Mitchell, T.M., Keller, R.M., & Kedar-Cabelli, S.T. (1986). Explanation-based generalization: A unifying view. *Machine Learning 1*.

Mill, J.S. (1843). *A System of Logic, Book III*. London.

Mooney,R. & Bennet,S. (1986). A Domain independent explanation-based generalizer. *Proceedings of AAAI*.

Pazzani, M., Dyer, M., & Flowers, M. (1987). Using prior learning to facilitate the learning of new causal theories. *Proceedings of the International Joint Conference on Artificial Intelligence*. Milan, Italy.

Rajamoney, S.A., & DeJong, G.F. (1987). The classification, detection, and handling of imperfect theory problems. *Proceedings of the International Joint Conference on Artificial Intelligence*. Milan, Italy.

Rivest, R.L. & Sloan (1988). Learning complicated concepts reliably and usefully. *Proceedings of the First Workshop on Computational Learning*.

Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). Learning internal representations by error propagation. In Rumelhart, et. al. (Eds.), *Parallel Distributed Processing, Vol. I*. MIT

Rumelhart, D.E., and Zipser, D. (1986). Feature Discovery by competitive learning. In Rumelhart, et. al. (Eds.), *Parallel Distributed Processing, Vol. I*. MIT Press.

Schneider, W., Dumais S.T., and Shiffrin R.M. (1984). Automatic and control processing and attention. In Raja and Davies (Eds.), *Varieties of Attention*, Academic Press.

Touretzky, D. S., and Hinton, G.E. (1988). A distributed connectionist production system. *Cognitive Science, Vol. 12*.