Machine Learning, 9, 95-98 (1992)


# *Book Review*

## Neural Network Design and the Complexity of Learning,
by J. Stephen Judd. Cambridge, MA: MIT Press, 1990.

VASANT HONAVAR
*Department of Computer Science, Iowa State University, 226 Atanasoff Hall, Ames, IA 50011*

**Editor:** Alberto Segre

*Neural networks* or *connectionist networks* (CN) (networks of relatively simple computing elements) offer an attractive and versatile framework for exploring machine learning for a number of reasons (such as, massive parallelism of computation, fault and noise tolerance, etc.). One frequent application of such networks is to learn complex *mappings* from a set of input patterns to a set of output patterns using supervised learning. Typically, such learning involves modifying the *weights* on the links within an a-priori fixed topology (Hinton, 1989). The discovery of *generalized delta rule*, popularly known as the *backpropagation* algorithm (Werbos, 1974; Parker, 1985; Le Cun, 1985; Rumelhart, Hinton & Williams, 1986) for weight modification learning in multi-layer feed-forward CN is at least partially responsible for the current interest in this approach to machine learning.

Although several weight modification learning algorithms have been proposed and empirically studied over the past several years, many questions that concern the time complexity of learning, and the choice of appropriate network topologies for particular tasks remain to be answered. Stephen Judd's book is a welcome addition to the growing family of books and monographs on CN. It represents one of the very first attempts to formalize, in computational terms, *learning* in multi-layer feed-forward CN.

The learning task studied by Judd—the *loading problem*—simply stated, is the task of memorization of some given data by a given feed-forward CN with a fixed architecture (i.e., the number of nodes and the connectivity among them is not allowed to change during learning). The first three chapters of the book introduce the *loading model* of learning and compare it with two other formal models of learning, viz., the ones proposed by Gold (1967) and by Valiant (1984). Some of the early complexity results for the *linear Perceptron learning model* (Rosenblatt, 1961) derived by Minsky & Papert (1969) and others are reviewed as well.

The loading model is inspired by the computational task underlying many popular CN approaches to learning. Valiant's model is concerned with what is feasibly learnable. Judd's loading model is concerned with what is feasibly learnable in a CN with a certain a-priori fixed structure. The question asked is whether a *particular* network can be made to represent the given data (assuming of course, that such a representation exists, given the particular network topology and node functions—it is meaningless to expect a CN to learn a function that is incapable of representing) in time $poly(|A| + |T|)$ (where $|A|$ is the size of the network; $|T|$ is proportional to the number of items in the data set to be loaded and the number of bits used to encode each item of data; and $poly(X)$ is a polynomial function of $X$); not whether it is possible to find *some* network to represent the same data. This assumption of fixed architecture has important theoretical as well as practical implications (see below).

Judd shows that the loading problem is NP-complete in the worst case for a given architecture by reducing the loading problem from the well-known 3SAT problem (Garey & Johnson, 1979). A different proof of a similar result has been given by Blum and Rivest (1988). Judd proves that the loading problem remains NP-complete even when: the network depth is at most 2 and the fan-in of individual nodes is at most 3; only 67% of the data items are to be retrieved correctly; or when the data items are at most 2-bits long; or when the number of data items to be loaded is no more than 3. He further shows that loading *shallow networks*, i.e., networks of bounded depth but unbounded width (and some special cases of such networks too technical to discuss here) is NP-complete. He then goes on to show that certain restricted classes of shallow networks, the so-called *columnar lines*—with a fish-net pattern of connectivity—can be loaded in time bounded by $poly(|A| + |T|)$. Judd further points out that his results are valid for a variety of node functions (the proof in the case of the frequently used *logistic* function is given in an appendix).

Generalization—loosely speaking, the ability to produce correct outputs in response to previously unseen data items—is an important consideration in evaluating different learning algorithms. Judd argues that intractability of loading implies intractability of generalization as well. The line of argument used here is as follows: Good generalization performance presupposes efficient and reasonably accurate memorization. Thus a network which fails to memorize the items in its data set adequately cannot possibly generalize adequately.

Judd obtains his intractability results on the loading problem as a function of the size of the a-priori chosen network architecture and the number of data items in the data set to be loaded. It is debatable whether this is in fact an appropriate measure of the complexity of the underlying task because it does not reflect the properties (e.g., inherent regularities) of the data items being loaded; nor can it capture possible effects of choosing a particular network topology for loading a given set of data items. Experiments with many popular CN learning algorithms suggest that the choice of network architecture can have a significant influence on both the learning time as well as the generalization performance. It would be interesting to see extensions of the loading model that can factor into the time-complexity estimates of learning, such interactions between the network architecture and intrinsic structure of the data set to be loaded.

Judd's results, as they stand, do not appear to rule out tractable solutions to the memorization task for particular choices of data sets and architectures. Of particular interest are data sets in which *similar* data items are to be mapped to *similar* network outputs, and data sets that exhibit intrinsic spatial or temporal or spatio-temporal structure (e.g., 2-dimensional visual patterns that are *compact* and *connected* as opposed to *random* dot patterns). The investigation of architectural constraints and the corresponding representational and inductive biases suitable for particular domains appears to be a promising area for further research.

If one were to relax the fixed network architecture assumption, any given memorization task can be trivially solved by constructing a look-up table. As Judd correctly points out, this is not particularly interesting. Furthermore, such an approach cannot yield good generalization. However, as has been proposed by Baum (1989), it might be possible to design efficient learning algorithms that adaptively alter the necessary network connectivity in non-trivial ways while searching for a suitable set of connection weights. Indeed, several such *generative* or *constructive* learning algorithms for CN are currently being investigated (Honavar & Uhr, 1988; Diederich, 1988; Ash, 1989; Nadal, 1989; Rujan & Marchand, 1989; Fahlman & Lebiere, 1990; Hanson, 1990; Gallant 1990). Since Judd assumes an a-priori fixed architecture, it is not at all clear as to what his results would mean in the context of such constructive learning algorithms. It would be interesting to see extensions of Judd's line of work to learning models in which the network adaptively assumes the necessary architecture as well as an appropriate setting of weights.

In summary, Judd's book is a valuable contribution to the theory of learning in neural networks. One hopes to see this work extended in a variety of contexts including those of non-feed-forward networks and constructive learning algorithms. The book is very well written and the exposition is at a level that should be accessible to beginning graduate students in computer science. Thought-provoking quotations interspersed throughout the book both inform as well as entertain the reader. The book should be of interest to a wide audience of researchers in complexity theory, computational learning theory, neural networks, and machine learning.

## References


Ash, T. (1989). Dynamic node creation in backpropagation networks. *Connection Science—Journal of Neural Computing, Artificial Intelligence and Cognitive Research*, 1, 365–375.

Baum, E.B. (1989). A proposal for more powerful learning algorithms. *Neural Computation*, 1, 201–207.

Blum, A., & Rivest, R. (1988). Training a 3-node neural network is NP-complete. In D.S. Touretzky (Ed.), *Advances in neural information processing systems (Vol. 1)*. San Mateo, CA: Morgan Kaufmann.

Diederich, J. (1988). *Knowledge-intensive recruitment learning*. (Technical report TR-88-010). Berkeley, CA: International Computer Science Institute.

Fahlman, S.E. & Lebiere, C. (1990). The cascade-correlation learning architecture. In D.S. Touretzky (Ed.), *Advances in neural information processing systems (Vol. 2)*. San Mateo, CA: Morgan Kaufmann.

Gallant, S.I. (1990). Perceptron-based learning algorithms. *IEEE Transactions on Neural Networks*, 1, 179–???.

Garey, M.R. & Johnson, D.S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. San Francisco, CA: W.H. Freeman.

Gold, E. (1967). Language identification in the limit. *Information and Control*, 10, 447–474.

Hanson, S.J. (1990) Meiosis networks. In D.S. Touretzky (Ed.), *Advances in neural information processing systems (Vol. 2)*. San Mateo, CA: Morgan Kaufmann.

Hinton, G.E. (1989). Connectionist learning procedures. *Artificial Intelligence*, 8, 232–264.

Honavar, V., & Uhr, L. (1988). A network of neuron-like units that learns to perceive by generation as well as reweighting of its links. In G.E. Hinton, T. J. Sejnowski, & D.S. Touretzky (Eds.), *Proceedings of the 1988 Connectionist Models Summer School*. San Mateo, CA: Morgan Kaufmann.

Le Cun, Y. (1985). Une procedure d'apprentissage pour reseau a sequil assymetrique. In *Proceedings of Cognitiva*.

Minsky, M., & Papert, S. (1969). *Perceptrons: An introduction to computational geometry*. Cambridge, MA: MIT Press.

Nadal, J.P. (1989). Study of a growth algorithm for a feedforward network. *International Journal of Neural Systems*, 1. 55.

Parker, D.B. *Learning logic* (Technical report TR-47). Cambridge, MA: Massachusetts Institute of Technology.

Rosenblatt, F. (1961). *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. Washington, D.C.: Spartan Books.

Rujan, P., & Marchand, M. (1989). Learning by activating neurons: A new approach to learning in neural networks. *Complex Systems*, 3, 229–???.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning internal representations by error propagation. In *Parallel distributed processing: explorations into the microstructure of cognition (Vol. 1: Foundations)*. Cambridge, MA: MIT Press.

Valiant, L.G. (1984). A theory of the learnable. *Communications of the ACM*, 1134–1142.

Werbos, P.J. (1974). *Beyond regression: New tools for prediction and analysis in behavioral sciences*. Ph.D. Thesis, Harvard University.