DOCUMENT RESUME

ED 405 832 IR 018 272

AUTHOR	Snyder, Robin M.
TITLE	Neural Networks for the Beginner.
PUB DATE	96
NOTE	8p.; In: Association of Small Computer Users in Education (ASCUE) Summer Conference Proceedings (29th, North Myrtle Beach, SC, June 9-13, 1996); see IR 018 247.
PUB TYPE	Reports - Descriptive (141) -- Speeches/Conference Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	*Artificial Intelligence; Computer Oriented Programs; *Computer System Design; Expert Systems; *Information Networks; Information Processing; *Information Technology; *Pattern Recognition; Problem Solving
IDENTIFIERS	*Neural Networks

ABSTRACT

Motivated by the brain, neural networks are a right-brained approach to artificial intelligence that is used to recognize patterns based on previous training. In practice, one would not program an expert system to recognize a pattern and one would not train a neural network to make decisions from rules; but one could combine the best features of each in solving a problem. One of the simplest and most common neural network models is the fully connected three layer model that consists of the input layer, the hidden layer, and the output layer. Instead of programming a neural network, the neural network is trained by presenting a history of inputs and outputs to the network. Training can be difficult and time-consuming, but after training, the neural network can quickly recognize patterns. One of the easiest places in which to integrate neural networks into the curriculum is a follow-on to the study of regression. The goal of regression is to determine a functional relationship between a dependent variable and one or more independent variables. Neural network software is becoming more available and more affordable. An annotated bibliography of selected literature on neural networks and a brief history of neural networks follows the discussion in this paper. (Contains 11 references.) (AEF)

Neural Networks for the Beginner

Robin M. Snyder
Associate Professor of Computer Science
Byrd School of Business
Shenandoah University
Winchester, VA
(540) 665-4938
rsnyder@su.edu

Abstract

Artificial neural networks are an increasingly important problem solving tool that have been applied to practical problems in many disciplines. This talk will provide a nontechnical introduction to neural networks: the problems for which they can be used with success, the types of problems for which they are ill-suited, and how to get started in neural networks. And, since neural networks are trained and not programmed, it is possible for the non- programmer to use neural network technology to advantage. Guidelines for integrating the topic into the curriculum and pointers to the literature will be provided.

Artificial Intelligence

AI (artificial intelligence) is the search for ways to make machines, more precisely, computers, more intelligent. A better name might have been machine intelligence. The paradox of artificial intelligence is that once something (that appears intelligent) is understood, it is no longer considered intelligent. Some areas of artificial intelligence include (traditional rule-based) expert systems, fuzzy systems, neural networks, and combinations of these such as fuzzy-neural-expert systems.

Expert Systems

Expert systems are a left-brained approach to AI that makes use of facts and rules in a systematic logical manner in order to arrive at conclusions that mimic experts in a field, allowing novices to perform more like experts in handling discipline- specific problems (i.e., the experts are still needed). Practical applications include tax preparation, grammar checkers, legal advice, help desks, and medical diagnosis.

Expert systems require a precise set of rules. Forget a rule, and your expert system may bomb. One of the early expert systems for providing user help responded to a user question

"How do I get more disk space?"

with the answer

"Use the command erase *.*.".

Oops. A rule was forgotten. In practice, expert systems are developed until they prove sufficiently useful.

Fuzzy Systems

Fuzzy systems, derived from fuzzy, or multivalent, logic, as opposed to crisp, or two-valued logic, are used primarily for process control applications. As a simple example, consider maintaining the water from a faucet at a constant temperature. If it gets a litte too hot, turn the control knob a little to make it cooler. If it gets a little too cold, turn the control knob a little to make it warmer. If it is very hot, turn the control knob more than if it is just a little hot. And so on. Fuzzy rules provide a convenient way to make decisions in the face of many potentially conflicting sources of information. Applications of fuzzy logic include automatic camera focusing, driving high speed rail systems, smart washing machines and vacuum cleaners, and fuzzy database queries.

Neural Networks

Motivated by the brain, neural networks are a right-brained approach to artificial intelligence that is used to recognize patterns based on previous training. Practical applications include adaptive noise canceling, mortgage risk evaluation, bomb sniffing, word recognition, forecasting, and handwriting analysis.

The field of neural networks is sometimes called artificial neural networks to separate it from human brain neural networks. For the purposes of the present discussion, neural networks will mean artificial neural networks that are used to solve pattern recognition problems.

A neuron is the simplest processing element of a neural network. Neurons are connected (and communicate) by means of synapses. When a neuron receives sufficient stimuli (e.g., from other neurons), it fires. That is, the neuron either fires or does not fire, which can cause other neurons to fire or not fire. And so on. Neurons and synapses can be combined into neural network models that can be used to solve pattern recognition problems.

Neural networks are not programmed. They are trained. Of course, good judgment is needed to properly set up and perform the training exercises. Consider the training of a vision system to recognize tanks. It has been said that the army took pictures of a tank from various positions, set up the neural network and trained it. During evaluation, the neural network performed poorly. It was eventually discovered that the pictures were of the same tank and that a beer can was sitting on the front side of the tank. The neural network trained itself to identify a tank by the beer can sitting on the tank. Of course, during performance evaluation, the tanks did not have the beer can in the right place and were, appropriately not recognized as tanks. In this case, the neural network was evidencing intelligent behavior, but was improperly trained.

In a classroom setting, students sometimes key in on the wrong ideas, usually based on an incomplete representative sample, and, come test time, make mistakes based on that training.

In practice, one would not program an expert system to recognize a pattern and one would not train a neural network to make decisions from rules. But one could combine the best features of each in solving a problem.

Three Layer Model

One of the simplest and most common neural network model is the fully connected three layer model that is trained using the backpropagation algorithm. The layers consist of the input layer (sometimes not counted as a layer, in which case this is the two layer model), the hidden layer, and the output layer.

- The input layer consists of a number of neurons that take input from the outside world. It is best to process and/or otherwise scale this information into a form that is easy for the computer to handle.
- Each neuron in the hidden layer (there can be more than one hidden layer) has a connection from each neuron in the input layer. By a suitable combination of these inputs, each neuron in the hidden layer either fires or does not fire. The results are sent to the output layer.
- Each neuron in the output layer has a connection from each neuron in the hidden layer. By a suitable combination of these inputs, each neuron in the output layer either fires or does not fire. The results are processed into an appropriate form that is easy for a human to handle.

Instead of programming a neural network, the neural network is trained by presenting a history of inputs and outputs to the network. The neural network adapts by adjusting errors that determine when individual neurons fire. Thus, after a neural network is suitably trained, the test comes when the neural network is presented with a pattern that may or may not be in the training set. The neural network will then attempt to correctly identify the pattern, even in the presence of missing or incomplete data.

The backpropagation algorithm uses a gradient descent method, which means that a representative error term is minimized in the direction that minimizes the error. Somewhat like trying to minimize your elevation by walking downhill. The problem is that one may get stuck in a local minimum and not find the desired global minimum. Methods have been developed to attempt to get around this problem.

Training can be difficult and time-consuming. But, after training, the neural network can quickly recognize patterns.

There are many other methods that have been developed for special cases of learning. For example, the backpropagation method is an example of supervised learning. Other methods are adaptive in that they work (for special cases) for unsupervised learning.

Although the concept of neural networks seems simple, in practice, subtle nuances can affect the ability to train and use a neural network. However, such nuances are beyond the scope of the paper.

Integration into the Curriculum

One of the easiest places in which to integrate neural networks into the curriculum is as a follow-on to the study of regression.

The goal of regression is to determine a functional relationship between a dependent variable and one or more independent variables. This is done by hypothesizing a functional relationship and then using a regression software package to determine the best fit for that particular regression model.

A typical regression problem might be, "What is the relationship between TV screen size (the independent variable), measured diagonally, and TV cost (the dependent variable)?". A more involved regression problem might be, "How is the price (the dependent variable) of a computer dependent on the amount of RAM, processor speed, disk drive size, etc. (the independent variables) of a computer.

The key point is that in regression models, the user must hypothesize a specific model and then run the software package on the data. And it can be difficult to handle incomplete or missing data points.

Once a functional relationship is determined, the forecast can be made by supplying independent variables that are used to determine the prediction for the dependent variable.

By contrast, a (simple backpropagation) neural network model is presented with the (scaled) independent data as input and the (scaled) dependent data as output, and trained until it can sufficiently predict the output data given the input data. The network is then presented with input data that it has not seen and then calculates a predicted value as an output value. Only a few decisions need to be made, such as how many hidden levels to use, how many nodes to use at each level to use, cutoff points, momentum factors, etc.

Software

Neural network software is becoming more available and more affordable. High performance neural network software is still quite expensive. The author has written a neural network software package that uses the backpropagation method. It is text-based in that the use writes a program as text that specifies the layers, connections, training data, etc.

Annotated Bibliography

As there is so much literature available on neural networks, and the amount of available information is increasing at an even faster rate, this section provides a short annotated bibliography of some references that the author has found useful for one purpose or another in understanding and/or implementing neural networks.

- Naturally intelligent systems (Caudill, 1990, 304 pages) is a very well written, readable, and entertaining account of neural network history and technology, without equations, derivations, or involved proofs. Many examples are presented in order to give intuitive meaning to concepts in the field of neural networks.

- Building neural networks (Skapura, 1996, 286 pages) is an invaluable resource for learning how to apply neural networks in practice. Many other books cover derivations and proofs that are not particularly useful to the person who just wants to apply neural techniques using existing software. A sampling of chapter topics include application design, business and existing software. A sampling of chapter topics include application design, business and financial applications, pattern classification, image processing, process control and robotics, and fuzzy neural systems.

- A practical guide to neural nets (Nelson, 1991, 344 pages) is a good nontechnical introduction to neural networks. Many practical applications are discussed.

- Neural computing: theory and practice (Wasserman, 1989, 230 pages), although somewhat dated, provides a good technical background to standard methods in the field without too much in the way of derivations and proofs. A computer scientist or software engineer should be able to implement many of the algorithms from the descriptions provided.

- Neural networks in computer intelligence (Fu, 1994, 460 pages) is a very thorough and readable overview, although grammatically somewhat boring, of the entire field of neural networks with many examples and variations of the neural network model. A sampling of chapter topics include knowledge-based neural networks, incremental learning, mathematical modeling, complex domains, discovery, structures and sequences, learning spatiotemporal patterns, expert systems, casual learning and modeling, validation and verification, rule generation from neural networks, learning grammars, and case studies that include genetic pattern recognition, drug discovery, and flow cytometric analysis of leukemia. More importantly, there are extensive references into the literature for each of these areas.

- Neural networks and fuzzy systems (Kosko, 1992, 449 pages) is a book written by an engineer for engineers. It is very technical and complete, and not very accessible to the layperson. However, Kosko does intersperse the technical details with discussions that involve pertinent philosophical issues of fuzzy and neural systems. For example, citing references in the literature, he concludes that there is nothing new about the backpropagation algorithm. *The algorithm did not offer a new kind of learning. Instead it offered only a new and computationally efficient way to implement an estimation method that statisticians had long explored* (Kosko, 1992, p. 198), stochastic approximation as used by statisticians in the 1950's. The author found his neural, fuzzy, and fuzzy-neural truck backer-upper systems interesting.

- Neural network computing (Bharath, 1994, 187 pages) is a somewhat simplistic, but yet useful introduction to neural networks. In particular, the comparison of neural networks with multivariate statistical techniques (eigenvalues, principal components, etc.) is very interesting.

- Introduction to the theory of neural computation (Hertz, 1991, 327 pages) is an excellent introduction to the theory of neural computation. Equations, derivations, and proofs are provided, as are many references into the literature. Comparisons are made with multivariate statistical techniques. Although thorough and well written, the technical presentation may be difficult for the casual reader.

- Neurocomputing: foundations of research (Anderson, 1988, 729 pages) provides a collection of reprints of many of the important research papers in neural computing, from the 1890 article "Association" by William James, and the classic 1943 article by McCulloch and Pitts, through the mid 1980's. Each article is preceded by an introduction putting the reprinted article in perspective.

- The artificial intelligence debate: false starts, real foundations (Graubard, 1988, 311 pages) provides a number of well-written and very readable articles that discuss and debate the various viewpoints of artificial intelligence, primarily in terms of connectionist models and neural networks.

This collection of references is not complete by any sense of the imagination. There are many other good references that the author is either not aware of or has not had the opportunity to review.

History

The history of neural networks is somewhat interesting. Research on neural networks began in 1943 (Anderson, 1988) and continued until the late 1960's. In 1969, publication of the book Perceptrons: an introduction to computational geometry .by Minsky and Papert (Minsky, 1969), proved that there were certain problems that perceptrons, a simple version of a neural network, could not solve. This led to limited funding of the area, and thus restricted research in the area of neural networks. In the late 1980's, the backpropagation algorithm was discovered and became popular in that it avoided the restrictions of the earlier perceptron networks. Increased attention to the area eventually revealed that the roots of backpropagation could be traced to the early 1980's, then to the early 1970's, and, finally, to certain statistical techniques of the 1950's (Kosko, 1992, pp. 196-199).

Kolmogorov, a Soviet mathematician, proved a theorem in information theory in the 1950's that has consequences in neural networks (Caudill, 1990, p. 174-177). In simple terms, Kolmogorov's theorem (as shown by Hecht) shows that any function mapping of m input numbers, each a real number between 0.0 and 1.0, that is to produce n output numbers, each a real number, can be exactly implemented by a neural network with three layers where the input layer has m input neurons, the hidden layer has 2*m+1 neurons, the output layer has n output neurons, and each layer is fully interconnected to the adjacent layer. That is the good news. The bad news is that the proof is not constructive. It only says that it can be done. It gives no indication of how to do it, or whether a simpler network could solve the same problem. However, it does show that a solution to a given problem may be within the realm of possibility.

And, there continues to be feuds between groups who claim that neural networks should model the brain and groups who claim that even though neural networks are not a strict model of the brain, they can be used to solve problems of interest. Both viewpoints have merit.

But, from a problem solving point of view, however, neural networks offer one more tool with which to solve problems.

Summary

In summary, if there is no obvious solution to a problem, then various intelligent systems approaches can be used. If the problem involves some kind of pattern recognition, neural networks may provide a possible solution. If the problem involves some kind of process control, a fuzzy system may provide a possible solution. If there are facts and rules that can adequately describe the system, a traditional expert system may provide a possible solution. And, if the problem involves all of these and more, perhaps a fuzzy-neural-expert system may provide a possible solution.

References

Anderson, J., & Rosenfeld, E. (eds.). (1988). Neurocomputing: foundations of research. Cambridge, MA: The MIT Press.

Bharath, R., & Drosen, J. (1994). Neural network computing. New York: Windcrest/McGraw-Hill.

Caudill, M., & Butler, C. (1991). Naturally intelligent systems. Cambridge, MA: The MIT Press.

Fu, L. (1994). Neural networks in computer intelligence. New York: McGraw-Hill.

Graubard, S. (ed.). (1988). The artificial intelligence debate: false starts, real foundations. Cambridge, MA: The MIT Press.

Hertz, J., Anders, K., & Palmer, R. (1991). Introduction to the theory of neural computation. Reading, MA: Addison-Wesley.

Kosko, B. (1992). Neural networks and fuzzy systems. Englewood Cliffs, NJ: Prentice-Hall.

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

Nelson, M., Illingworth, W. (1991). A practical guide to neural nets. Reading, MA: Addison-Wesley.

Skapura, D. (1996). Building neural networks. Reading, MA: Addison-Wesley.

Wasserman, P. (1989). Neural computing: theory and practice. New York: Van Nostrand Reinhold.