Learning programs

Teaching computers to acquire knowledge

Daniel C. St. Clair

Computer programs that learn are called Machine Learning programs. These machine learning programs are being used to automatically create and maintain knowledge bases for such applications as: diagnosis, computer vision, speech understanding, autonomous robots, discovery learning systems, and intelligent tutoring systems. Some researchers are working on computer programs that learn to make philosophical assessments !

Using mathematics and logic, researchers have constructed programs capable of "learning." These programs develop concepts, infer new concepts from existing concepts and revise incorrect concepts. Similar to humans, machine learning usually focuses on one or two subjects at a time, and builds on knowledge already learned. For example, humans learning about the design of microcomputers must use their knowledge about Boolean algebra. In machine learning, the subject area being learned is called the domain.

The set of concepts created by a machine learning algorithm is referred to as a knowledge base. Knowledge bases for one domain may be combined with knowledge bases for other domains. This notion is similar to the way humans increase their knowledge. Knowledge bases can be used to answer questions and make conjectures about new situations. For example, a knowledge base containing concepts about a particular set of circuit boards might be used to diagnose defective boards. While conventional programs have been written for such purposes, they are unable to make conjectures about situations for which they have not been trained.

The actual learning of concepts may be supervised or unsupervised. In supervised learning a "teacher" guides the program through the learning process by providing information about the domain being studied. This guidance may be in the form of sets of examples which are then used to train the system. Some programs require that information be in the form of a model of the domain. Programs requiring only training examples are often referred to as empirical learning algorithms while those that require a model are called knowledge intensive algorithms.

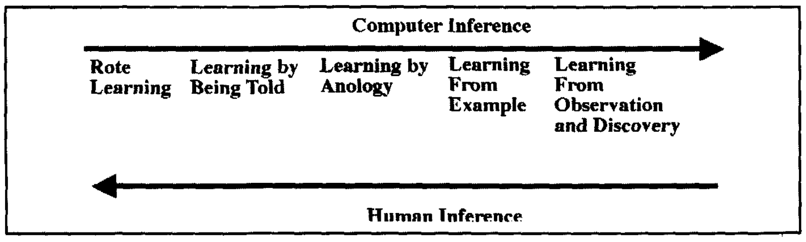


Fig. 1 Types of Inference

Unsupervised or discovery learning programs develop concepts by performing various analyses of the input they receive. Many of these programs depend on statistical techniques to help them discover patterns in input information. They may or may not rely on knowledge learned previously. This approach has been used to discover concepts completely unanticipated by researchers and domain experts.

Difficulties

One major problem in building these programs is that computers and humans are not good at the same types of learning tasks. For example, consider the several types of inference shown in Fig. 1. Inference is the process of developing concepts from specific inputs such as examples, facts, and descriptions. The arrow labeled "computer inference" indicates that role learning is easier for computers than learning concepts from observation and discovery. The arrow at the bottom indicates that humans, in general, are much better at learning from observation and discovery than they are at rote learning or at learning by being told. (Perhaps professors should be lecturing computers instead of students.) In any case, while scientists understand many things about the learning of concepts, they have been unable to completely define concept learning in an algorithmic way. If this were possible, computer programs could be written which would cause computers to behave in more "human like" ways. To better understand the complexity of machine learning, consider the set of objects shown in Fig. 2. The concept to be learned from these objects so simple it could easily be understood by a child of age five. However, the challenge lies not in the difficulty of the concept but in knowing which features of the figure are significant.

Suppose the learner is told that rectangles A, B, C, and D are examples of the desired concept and rectangles E, F, G, and H are counter examples of the concept. While this information influences one's thinking about the problem, the concept to be learned is still not evident. Next, suppose the learner is told that the concept of interest is associated with square #3 in each rectangle. The problem is now trivial since examination of these squares reveals that the rightmost triangle in square #3 of rectangles A, B, C, and D is oriented in exactly the same direction! Those of rectangles E, F, G, and H are not. This simple example illustrates a very difficult problem in developing computer programs that learn. In order to

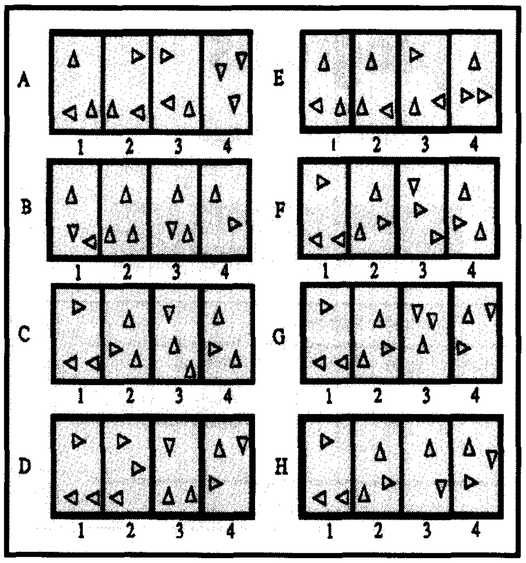


Fig. 2 Example of Simple Concept Learning

class	outlook	temp	humidity	windy
neg	sunny	hot	high	false
neg	sunny	hot	high	true
pos	overcast	hot	high	false
pos	rain	mild	high	false
pos	rain	cool	normal	false
neg	rain	cool	normal	true
pos	overcast	cool	normal	true
neg	sunny	mild	high	false
pos	sunny	cool	normal	false
pos	rain	mild	normal	false
pos	sunny	mild	normal	true
pos	overcast	mild	high	true
pos	overcast	hot	normal	false
neg	rain	mild	high	true

Fig. 3 Physical Data (Source: Quinlan, J. R., "Induction of Decision Trees," Machine Learning, Kluwer Academic Publishers , Vol. 1, 1986, pp. 81 - 106.)

be able to learn concepts, humans as well as computers must be able to decide what information is important and what information can be discarded.

Learning concepts in the real world is further complicated by factors such as the quality and quantity of inputs the learner receives. Too many inputs adds unnecessary complexity to the learning task. Too few inputs or noisy inputs result in the formulation of incomplete and incorrect concepts. For example, in many machine learning applications where real data is used for training, it is often difficult to separate noisy data from "sparse but accurate" data. Sparse but accurate data occurs when you have only a few correct training examples of a particular concept. The training examples are so few that statistical noise reduction techniques often label these training examples as noise and tag them for removal from the training data. This situation occurs in the diagnosis of aircraft avionics systems where test programs are used to identify most diagnostic situations. Current avionics test programs do an excellent job of circuit diagnosis; however, rare, but valid situations may occur which are not in the current diagnostic knowledge base. Concepts surrounding these situations must be learned if avionics diagnostics are to be performed in a timely manner. Machine learning algorithms must be able to distinguish sparse but accurate inputs from noisy inputs.

Some current approaches to machine learning

Knowledge created by machine learning algorithms takes one of two basic forms: symbolic or analog. Symbolic machine learning algorithms create knowledge bases whose contents can be thought of as rules and facts. For example, a knowledge base might contain the rule:**If the switch is on, and the car won't start,then check the battery.** If it is known that the hypotheses are true; i.e.. the switch is on, and the car wont start, then the resulting action is to check the battery.

One well-known symbolic algorithm, ID3, was developed by Dr. Ross Quinlan in 1979. The ID3 algorithm uses training examples as input. From these examples, ID3 constructs a decision tree. Each path in the decision tree corresponds to a rule.

The set of training data shown in Fig. 3 can be used to illustrate ID3. Associated with each set of features or attributes; outlook, temperature, humidity, and windy is a classification of pos (positive) or neg (negative). The objective is to develop rules which use one or more of the attributes; outlook, temp, humidity, and windy to predict classification.

Applying ID3 to this data produces the decision tree shown in Fig. 4. **The dashed path illustrates the rule: If outlook is sunny, and humidity is normal, then class is pos.** In developing the tree, ID3 uses a special measure called entropy to select the attribute to be used at each node. The attribute is chosen which minimizes the depth of the resulting tree, since attributes are chosen locally In general, the smaller the depth of the tree , the more general and more powerful the set of rules represented by the tree.

The ID3 algorithm is popular because it is relatively easy to program and to use. Other machine learning algorithms, such as Michalski's AQ15 or 16, use training examples to form a set of rules. These rules constitute the knowledge base. However, AQ 15 requires the user to initialize numerous parameters while ID3 does not. Still other symbolic algorithms are called "knowledge intensive" because they require a prior knowledge about the domain being modeled.

Explanation Based Learning systems, EBL, are examples of knowledge intensive machine learning algorithms. These algorithms use prior domain knowledge consisting of facts and rules to form a generalization of each training example. These generalizations are not only used to extend the current knowledge base, but can be used to provide explanations of responses given by the system.

Genetic algorithms are another machine learning paradigm which generate rules. They are interesting because their basic philosophy is patterned after natural genetics. Genetic algorithms represent rules as vectors containing bits. Each bit in the string corresponds to one value of one of the attributes. A string of bits is used to represent a rule. These bit strings are called chromosomes. Operations on chromosomes include mutation, crossover, and reproduction. Genetic algorithms have been successful in a number of applications.

Analog machine learning algorithms represent knowledge in the form of weight parameters and functions. These weights and functions are chosen so as to represent the concepts being learned. Learning is effected by adjusting weight parameters and function definitions until the desired responses are achieved. This type of learning algorithm tends to be very mathematical in nature. Because of this, the knowledge created by analog machine learning algorithms is usually more difficult to understand because function and weight values are much less intuitive than sets of rules.

Artificial Neural Networks (ANN) depict a broad class of analog learning algorithms. One type of ANN, Backpropagation (BP), represents knowledge as two or more layers of nodes. Each node is connected by one or more arcs. Figure 5 illustrates a BP network with four layers; an input layer (I), an output layer (O), and two hidden layers (H1 and H2). The number of nodes in the input layer, I, corresponds to the number of input attributes. The number of output nodes, O, correspond to the number of classifications. Determining the number of nodes in each hidden layer as well as the number of hidden layers in a network are currently active areas of ANN research.

Figure 6 illustrates the basic structure of a node. Associated with each arc is a weight, w_i and associated with each node is the threshold function, f(z). The first process performed at the node is to sum the weighted incoming signals,

$$z = \sum_i w_i x_i$$

where x_i denotes the output of node i from the previous layer. The ouput:

$$y=f(z)$$

is used as an input to nodes in the next level. There are several choices of activation functions. One function that works well in practice is the hyperbolic tangent function, viz.

$$f(z) = (1 - e^{-2z}) / (1+e^{-2z})$$

which produces values for f(z) in [-1, 1].

Training examples are introduced at the input layer, I. The values are then propagated through the network. For instance, output node with the highest value represents the network's result. Training consists of using the training examples to adjust the values of appropriate weights. Mathematical programming algorithms such as steepest descent or conjugate gradient are typically used to determine weight adjustments. Training can occur after each training instance is processed or after each Epoch. An Epoch represents one complete pass through the training set. This procedure is repeated until an acceptable number of training examples have been correctly classified. At this point, the network can be used to classify unseen inputs.

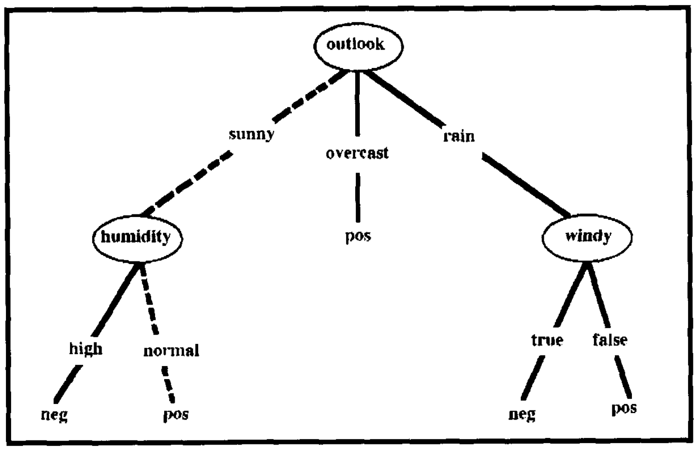


Fig. 4 ID3 Decision-Tree From Physical Data

Fig. 5 Backpropagation Neural Network

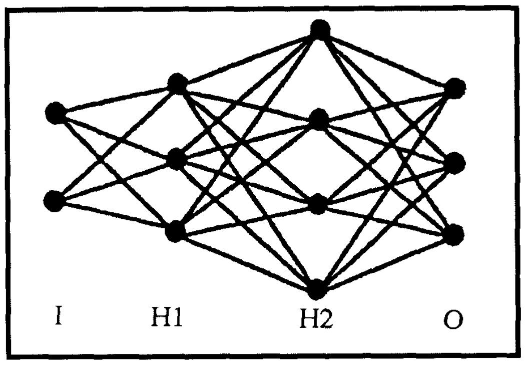
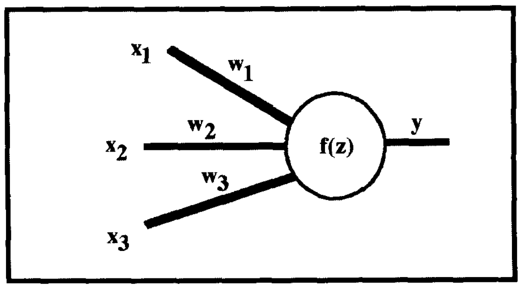


Fig. 6 Node Detail



NonDestructive Evaluation (NDE)

One application of machine learning currently under investigation at McDon-

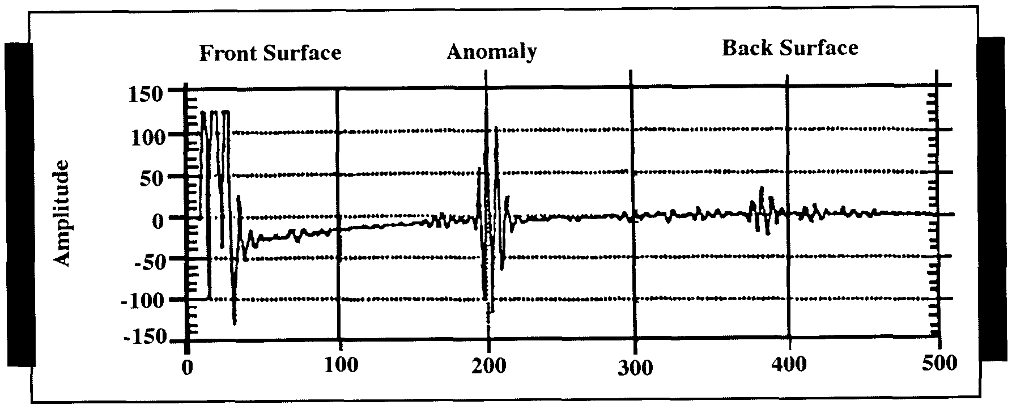


Fig. 8 Typical A-Scan Representation of a Significant Anomaly

nell Douglas Research Laboratories in St. Louis, MO is the automation of NonDestructive Evaluation (NDE) techniques. The statement of the problem is simple: detect subsurface anomalies in aircraft structures. The identification of such anomalies is crucial to aircraft safety. Technicians often use ultrasonic waves to find and identify these anomalies. During the tests, a probe is placed at various locations on a structure and the resulting waveform outputs observed on a CRT. Waveform features indicate the location and type of any anomalies. Ultrasonic NDE testing and evaluation is presently carried out manually. Since each aircraft has several thousand points which have to be tested, the process is extremely time-consuming. In addition, the results require careful interpretation.

Figure 7 illustrates an example test structure. Placing the test probe at position A gives a no-anomaly reading while placing it at position B gives an anomaly reading. Figure 8 shows the waveform taken from position B. The waveform is produced by reflections of the ultrasonic waves. The high front and back surface amplitudes are produced when the ultrasonic waves reflect from the top and bottom of the structure. Note the additional high amplitude produced at the Anomaly position. The corresponding no-anomaly waveform is similar except there are no extreme amplitude changes between the Front and Back Surfaces. In addition, the amplitude at the Back Surface is about the same as it is at the Front Surface. All anomalies are not as easy to identify. For example, anomalies close to the Front /Back Surfaces are not as pronounced. The structure geometry also affects the ease with which anomalies can be detected.

McDonnell Douglas Researchers have applied both symbolic and analog machine leaning techniques to NDE waveforms. These algorithms produce knowledge bases which can be used to quickly and accurately evaluate waveforms from NDE tests. Expert NDE test technicians have carefully supervised testing and have reviewed results. They indicate that the knowledge bases produced perform at the "expert" level on the structures evaluated .

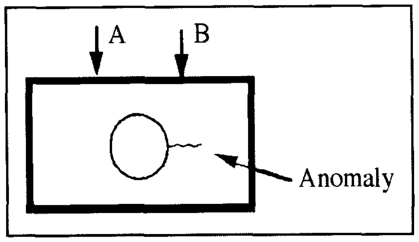


Fig. 7 Example Test Structure

Read more about it

• Bond, W.E., St. Clair, D.C., Amirfathi, M.M., Merz, C.J., and Aylward, S., *Neural Network Anaysis of Nondestructive Evaluation Patterns,* Proceedings 1992 ACM/SIGAPP Symposium on Applied Computing, Vol. II, ACM Press, March 1992.

• DeJong, K, *Learning with Genetic Algorithms: An Overview,* Machine Learning, Kluwer Academic Publishers, Vol. 3,1988, pp. 121-138.

• Ellman, T., *Explanation-Based Learning: A Survey of Programs and Perspectives,* ACM Computing Surveys, Vol. 21, No. 2, June 1989, pp. 163-221.

• Quinlan, J. R., *Induction of Decision Trees,* Machine Learning, Kluwer Academic Publishers, Vol. 1 , 1986, pp. 81 - 106.

• Kodratoff, Y., Michalski, R., Machine Learning: An Artificial Intelligence Approach, Vol. III, Morgan Kaufmann, 1990.

• Lippmann, R., *An Introduction to Computing with Neural Nets,* Artificial Neural Networks: Theoretical Concepts, Ed. V. Vemuri, IEEE Computer Society, 1988, pp. 3654.

• *Machine Learning* is published five times annually by Kluwer Academic Publishers.

• *Proceedings of the International Machine Learning Conference/Workshop* which is held annually.

Special thanks

The author wishes to especially thank Dr.s W.E. Bond and B.B. Flachsbart of McDonnell Douglas Research Laboratories for their comments and suggestions during the preparation of this manuscript.

About the author

Dan St. Clair is Professor of Computer Science at The University of Missouri-Rolla's Engineering Education Center in St. Louis. He also holds the position of Visiting Principle Scientist at McDonnell Douglas Research Laboratories in St. Louis. There his research focuses on the application of machine learning algorithms to the construction and maintenance of diagnostic systems for aircraft. ■

Work supported by the McDonnell Douglas Independent Research and Development program and Statement of Work WS-MDRL-4062 with the University of Missouri-Rolla Engineering Education Center in St. Louis.