Artificial Intelligence: Its Use in Medical Diagnosis

Robert Scott

The University of Maryland, Baltimore, Maryland

J Nucl Med 1993; 34:510–514

Much work has been done attempting to refine or make increasingly explicit the diagnostic process. Normally, diagnosis principally involves the processes of collection, analysis, recognition and classification of data. Patient data are obtained from interviews, examinations and tests. The physician, using his knowledge and experience, transforms the data into a diagnosis. If one were to view the physician as a system, the data collected would provide the inputs and the diagnoses the outputs. This system is shown schematically in Figure 1 in a black box representation with three parts: inputs, outputs, and a mapping or transformation F(I) from inputs into outputs taking place within the box. This simplistic representation of cognitive behavior is useful because the black box can represent many forms of internal processing of data.

The field of artificial intelligence (AI) has provided several approaches to defining systems that may be useful for medical diagnosis. There is a distinct dichotomy, however, between the approaches used by two of the most popular systems in what takes place in the black box described above. Expert systems (ESs) are rule-based to explicitly define the steps that take one from a set of inputs to outputs (*1*). The transformation appears as a progression through a number of IF-THEN type rules constructed with the help of domain experts such as physicians experienced in the diagnostic area of interest. If the domain of knowledge required for the diagnosis can be clearly defined by such rules, ESs may be successful (*2*). The rules must be so structured and implemented through the use of an "inference engine" that when a pattern of inputs is presented to the ES, a diagnosis or other form of response will be produced as output. This logically complete set of rules employed to perform the transformation is in contrast to artificial neural networks (ANNs). While processing data in an ES takes a serial or sequential progression through a number of IF-THEN type rules, the processing used by ANNs is a parallel form analogous to that of the brain, adaptive and not constrained by fixed rules.

Specialists in the fields of nuclear medicine, radiology and other disciplines in which data evaluation involves the interpretation of visual patterns may benefit from applications of ANN technology. Unlike conventional serial computers, the parallel processing of ANNs exhibits more brain-like behavior. If we contrast two types of human problem-solving, such as recognizing a familiar visual pattern and mentally solving a mathematical problem, the dichotomy of parallel versus serial processing is illustrated. A personal computer operating at a rate measured in nanoseconds per operation can outperform a human in solving a math problem by orders of magnitude, while the human has a tremendous speed advantage over the computer (in visual pattern recognition tasks). Neuronal processing speeds measured in milliseconds achieve remarkable systemic speeds because of the enormous number of highly interconnected processing elements that operate simultaneously in a distributed processing manner.

BASICS

An ANN is composed of simple processing elements (PEs) each of which is capable of communicating with a large number of similar PEs. This architecture is called parallel distributed processing (PDP), a term introduced by Rumelhart and McClelland (*3*). Parallel refers to the activity of a large number of PEs, in layers, simultaneously processing data. Distributed refers to the sharing of the learning task by many processors. Although PEs are often referred to as neurons, they are, in fact, gross simplifications of biological neurons. Whereas memory and adaptation in biological neurons take place in a complex interaction of neurotransmitters, synapses and dendrites, memory and adaptation in ANNs reside totally in the changing weights used to amplify the effects of afferent connections to each PE. Figure 2 shows the ultimate simplicity of the model with I representing the accumulated sum of all incoming signals from other PEs, each

Received Oct. 20, 1992; revision accepted Nov. 5, 1992.

For correspondence or reprints contact: Robert Scott, MD, Dept. of Information Systems Mgmt, University of Maryland Baltimore County Campus, Baltimore, MD 21228-5398.

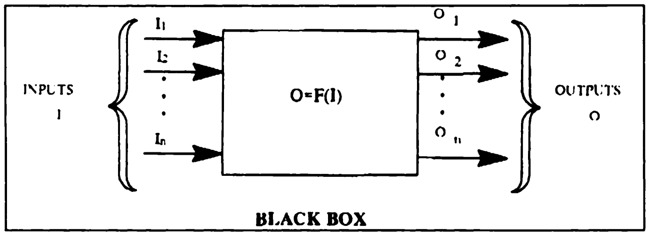

FIGURE 1. A schematic representation of a system mapping inputs into outputs.

multiplied by its weighted connection. This sum is compared with a bias obtained from a special input with a fixed excitation value of 1 which is also multiplied by its weight to produce a changeable threshold ($W_{bias} \times 1$). The PE fires or outputs a value of 1 only if (I-threshold) ≥ 0, otherwise the PE does not fire (output = 0). This conversion of I into a binary output is provided by a nonlinear transfer function called a step function. Other forms of transfer functions find utility in various network paradigms.

With only two inputs, X and Y, a PE's output is mathematically an equation of a line that divides two-dimensional space into two parts. The line is called a linear discriminant. Any arbitrary line can be defined by assigning weights to the two input connections and the bias connection as shown in Figure 3. Here, the line that intercepts the x-axis at 8 and the y-axis at 6 is defined by the three connection weights shown. The output of the PE will take on a positive or negative sign depending on the position of a point on the surface (determined by inputs X and Y) with respect to that line. The function represented by the PE and its associated weights is shown as F(X, Y). Suppose, for example, that a patient's diagnosis with respect to either having a disease or not having a disease could be completely determined by two parameters that could be represented as numbers. Suppose the disease is hypertension, with a PE's positive output associated with the diagnosis hypertensive and a

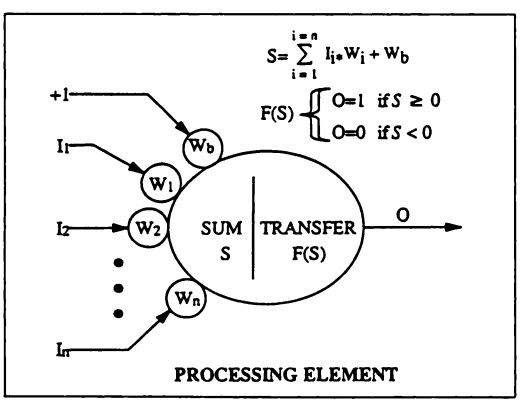

FIGURE 2. Elementary processing element in a neural network.

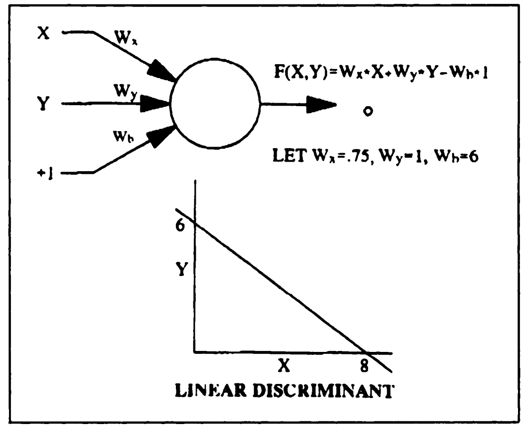

FIGURE 3. Example of a discriminant created by weights assigned to a processing element.

negative output associated with normality. The input parameter values are blood pressure, X for systolic and Y for diastolic. In actual practice, it may not be possible to find a line that will correctly separate all points in a space. For example, Figure 4 shows a set of sample points of men, X, and women, O, identified by height and weight. The task of a PE is to find a set of weights (linear discriminant) that will correctly associate an input pair (height, weight) as belonging to a man or a woman. The best choice of a linear discriminant is one that minimizes the number of incorrect classifications.

HISTORY

In 1943, McCulloch and Pitts (*4*) proved that simple threshold sensitive ANNs could be configured to perform any logical function. The implication of this was to show that networks of simple neural-like processors could be designed to operate logically as digital computers now do. It is believed that von Neumann was inspired by this

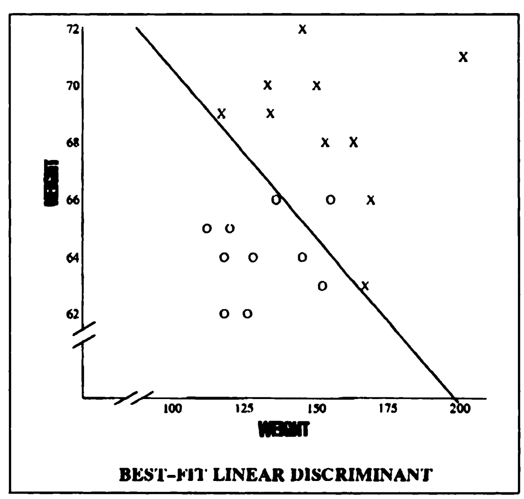

FIGURE 4. Best (least error) discriminant classifying males (x) and females (o) on the basis of height and weight.

early work to develop the first stored program digital computer. For the McCulloch-Pitts model to be biologically plausible, neurons must behave in a binary manner producing an output that is either all or none. In 1949, Hebb (5) advanced a postulate of neuronal learning based on cooperative synaptic reinforcement: *"When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased."* Subsequent research into ANNs used variations in this theory known as Hebb's rule to provide learning paradigms for networks. Since only the connection weights carry information, all learning algorithms must function in such a manner so that as each input pattern is presented, the connection weights to each PE will be adjusted to reduce the error or difference between the desired and actual output of the PE and eventually minimize the error. The reader will find excellent accounts of the early development of ANNs in the literature (6,7). The Perceptron, an ANN proposed by Rosenblatt (8) in 1958, applied this technology to recognize two-dimensional patterns, such as letters, using the retina as a model. Rosenblatt proved that a perceptron trained with his learning algorithm will converge or learn to properly classify input patterns if the patterns are linearly separable. This will be discussed later.

In 1960, Widrow and Hoff (9) implemented in hardware an adaptive processing element called Adaline. Variations of their learning algorithm, Widrow-Hoff, are used with current ANNs. This algorithm is also known as the delta rule or gradient descent method in weight space. Somewhat later, ANN research entered the "Dark Ages" (10) when in 1969, Minsky and Papert were critical of the future of ANN research (11) and pointed to the inability of the simple perceptron to handle the logical AND function. Nonlinear networks were needed to solve many problems. At that time, major funding for AI was divided between expert systems and neural networks. While AI research using digital computers continued, research in ANNs ebbed. A serious drawback for ANN was the lack of a learning algorithm that could reduce output error through weight adjustment in a multi-layered perceptron (the credit assignment problem). In 1974, Werbos (12) introduced a method to solve the credit assignment problem in dynamic modeling. Rumelhart et al. refined a multi-layer learning rule in 1986 (13). The 1980s brought a resurgence in ANN research due in large part to the availability of low cost, high speed computers on which to simulate ANNs. The algorithm known as "back-propagation" is the most widely used learning algorithm for multi-layered networks. As Lippmann (14) shows, most interesting problems can be solved by using three-layered networks.

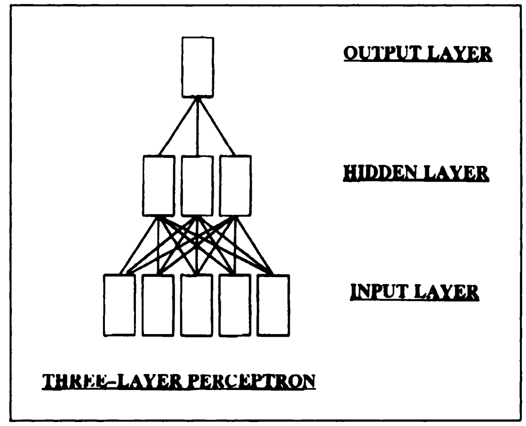

FIGURE 5. Example of a three-layer Perceptron neural network.

APPLYING NEURAL NETWORKS

Although an ANN may have any number of layers, the most common ANN, a three layered perceptron composed of an input layer, a hidden or intermediate layer and an output layer, is used as an example (Fig. 5). Input patterns must be paired with predetermined outputs or targets toward which the network's weights must be adjusted. The weights are adjusted according to rules based on the error between the desired outputs (those presented with the input data) and the actual outputs of the network before weights are adjusted. The training algorithm handles the credit assignment problem of determining the degree of error contributed by each PE in the network and makes appropriate incremental weight changes where needed to reduce the error.

There are two steps involved in this type of learning: first, an input stimulus is applied to the network and allowed to propagate through the network to the outputs; second, the network outputs are compared with the desired outputs and the error or difference between them is reflected back through changes to the network's connection weights. These two steps are executed for each input pattern and the process is repeated until the overall error has been reduced to an acceptable level.

Training a network has two goals: first, to reduce the network error to an acceptable level (convergence) and second, to produce a set of network weights that will extend the relationship between the input and output pairs established during training to produce correct outputs when the network is presented with input patterns not previously seen. This ability is called "generalization." If the second goal cannot be attained, the ANN is doing little more than finding unique relationships between trained inputs and outputs. This is similar to a "table look-up" procedure where the input represents an address and the output is the data found at that address. The multi-layered perceptron is the form of ANN most used for classification problems. The simple feed-forward type network moves signals from the input layer to the hidden layer and then to the output layer. Each PE's output is generally (fully) connected to all of the PEs in the layer immediately above. Except for the input layer, the bias PE, which always has an output value of one, is also connected to each PE. With a single hidden layer, the ANN transforms patterns or points in the input space into points in another space whose dimensionality is determined by the number of PEs in the hidden layer. The real power of the ANN lies in hidden layer's internal representation of the input pattern. Correctly sizing the number of PEs in the hidden layer is often critical to performance of the network. Too large a number may cause the network to do little more than provide a table look-up for the input patterns, while too few PEs may not be able to fully extract the critical input/output relationships in the training patterns. Either extreme may lead to poor generalization of new patterns once the network has been trained. In an autoassociative network, where the desired output is the same as the input and the number of PEs in the hidden layer is less than the number of PEs in the input layer, the reduced dimensionality of the hidden layer's output represents a compression of the data (*15*). The output layer provides the final transformation needed for the desired output. For example, if the network was to simply separate input patterns into two classes, those which are "normal" from those which are "abnormal," then a single PE in the output layer could represent normal patterns by producing a one out and the abnormal patterns with a zero out.

THE PROMISE

ANNs are being applied to many areas from weather forecasting to stock market predictions and vary greatly in form and function. The four learning paradigms listed by Rummelhart and Zipser (*16*) are:

1. Auto-associator, which maps a pattern onto itself (A $\rightarrow$ A).
2. Pattern Associator, which maps a set of patterns into a different set of patterns (A $\rightarrow$ B).
3. Classification Paradigm, which maps all variations of a set of input patterns into a fixed set of categories.
4. Regularity Detector, which discovers input pattern dissimilarities and develops its own appropriate output classes rather than using previously assigned categories.

The focus here will only be on ANN research that offers the greatest promise in image classification, recognition and enhancement.

First a note of caution. The black box approach described earlier limits learning to data provided by the user. The goodness of the output when tested on trained patterns as well as new patterns depends entirely on the statistics of the patterns used in training that make up the data base. The old computer adage of "garbage in, garbage out" certainly applies here. For example, suppose a medical specialist is skilled in one form of diagnostic imaging. Let I_p be an image obtained from Patient p using this familiar imaging system, I. Also, let the specialist's evaluation of this image be $D(I_p)$. If the specialist is introduced to a new form of potentially valuable imaging, I′, initially he may be unable to interpret the new images. If images using both systems are made of Patient p, then if the system I′ is to perform at least as well as system I on data from Patient p, the specialist would expect $D(I_p) = D(I_p')$. The ANN can then be trained on data obtained from the new system, I′, using correct diagnoses $D(I_p)$ as the output on which to be trained. If the training of the ANN is successful, the network should be able to generalize on the transformation learned during training and apply it correctly to images, I′, that are not part of the training data. The supervised training provided the ANN is really a form of conceptualization where the ANN, after repeated exposure to data that has already been correctly classified, discovers the invariance of certain data items that distinguish class membership. Concepts correctly learned can then be extended or generalized to input patterns not previously seen.

Most of today's research on ANN applications is done on conventional computers using software simulations with hardware accelerator boards offering additional processing speed. While use of a software simulator may seem to defeat the advantage of parallel processing speeds, the flexibility of design changes offered by simulation compensates for the speed disadvantage. Once an ANN has been trained, the connections and their weights completely specify the mapping of inputs to outputs. Analog and digital integrated ANN circuits are currently being developed to implement various algorithms in hardware.

As stated earlier, ANNs learn by example, not by rules. There really is no mystery, however, as to what is happening during the learning process for most learning paradigms are deterministic. Training is terminated once the synaptic weights have adjusted or converged to correctly map the training patterns of input/output pairs. The adequacy of the weights, once trained, is totally dependent on the proper selection of training data. We would like the network, once trained, not only to map the training set of inputs into correct outputs, but also to be able to generalize from the concept learned to correctly map new inputs (not part of the training set) into correct and useful outputs.

It is fortunate that technology has favored the use of digital images. Gamma camera images, computed tomography (CT), ultrasound and magnetic resonance (MRI) all create digital images easily manipulated by computers and ANNs. There are a number of researchers reporting favorable results in using ANNs for image classification.

TABLE 1
Comparison of ANN and ES

Approach	Design	Execution	Analysis
ANN	Easy	Rule-following	Difficult
ES	Difficult	Rule-based	Easy

Dawson et al. (*17*) reported good agreement between neural network and human classification of breast carcinoma using nuclear graded images. Neonatal chest x-rays (*18*) were used to train an ANN to choose one or more diagnoses from a list of 12 possible diagnoses. After training, the researchers found good diagnostic agreement between the ANN and participating radiologists. A backpropagation ANN was applied to the problem of segmentation of the aorta from MRI images (*19*) with limited success. A screen for abnormal cells from slide images using an ANN has been shown to be a useful diagnostic tool (*20*) in hemotology.

Table 1 contrasts expert systems and ANNs with respect to three stages of design, execution and analysis. Expert systems are driven by a set of rules provided by domain experts. Acquiring the knowledge base for the expert system is the most difficult process with the effectiveness of the system being totally dependent upon the completeness and accuracy of the rules upon which the system is based. The analysis, however, of the outputs of an expert system is easy since the line of reasoning is simply the progression through the set of rules. On the other hand, the rules in an ANN are implicit in the training data provided to the network. Expertise is still needed, however, to insure the correctness of the training data. While the expert system's outputs are easy to analyze, the ANNs' outputs, based on distributed processing of the inputs and dependent solely on the weighted connection set arrived at through training, are difficult to analyze.

SUMMARY

In nuclear medicine, analysis, interpretation and diagnosis may each be appropriate as applications for ANNs. Image processing and pattern recognition are two application areas of ANN technology that appear promising. An ANN (*21*) was used to classify normal and abnormal FDG-PET scans and performed better than discriminant analysis. Favorable results have been obtained from an ANN (*22*) in the interpretation of data recorded by experienced observers using various standard V/Q scans to determine the likelihood of pulmonary embolism. Carver Mead's book (*23*) on the development of hardware neural systems defines new approaches to ANN fabrication, including an electronic 100,000 transistor retina. As hardware is developed to permit the design of very large networks, we may expect many useful imaging applications to emerge for ANNs in the medical field.

REFERENCES

1. Lavesque HJ. Knowledge representation and reasoning. *Ann Rev Comput Sci* 1986;1:255–287.
2. Parsaye K, Chignell M. *Expert systems for experts*. New York: John Wiley; 1988.
3. Rumelhart DE, McClelland JL. *Parallel distributed processing, explorations in the microstructure of cognition, volume 1*. Cambridge, MA: MIT Press; 1986.
4. McCulloch WS. *Embodiments of mind*. Cambridge, MA: MIT Press; 1988:19–39.
5. Hebb, DO. *The organization of behavior*. New York: Wiley; 1949:62.
6. Eberhart RR, Dobbins RW. Early neural network development history: the age of Camelot. *IEEE Engineering Med Biol* 1990;3:15–18.
7. Johnson RC, Brown C. *Cognizers: neural networks and machines that think*. New York: Wiley; 1988.
8. Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psych Rev* 1958;65:386–408.
9. Widrow B, Hoff ME. Adaptive switching circuits. *WESCON Convention Record, Part 4, IRE* 1960:96–104.
10. Eberhart RC, Dobbins RW. Background and history. In: Eberhart RC, Dobbins RW, eds. *Neural network PC tools*. San Diego, CA: Academic Press; 1990:9–34.
11. Minsky ML, Papert SA. *Perceptrons*. Cambridge, MA: MIT Press; 1969.
12. Werbos PJ. Beyond regression: new tools for prediction and analysis in the behavioral sciences. PhD thesis. Harvard, Cambridge, MA; 1974.
13. Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation. In: Rumelhart DE, McClelland JL, eds. *Parallel distributed processing, explorations in the microstructure of cognition. volume 1*. Cambridge, MA: MIT Press; 1986:318–362.
14. Lippmann RP. An introduction to computing with neural nets. *IEEE ASSP Magazine* 1987:4–22.
15. Elman JL, Zipser D. Learning the hidden structure of speech. *Institute of cognitive science report no. 8701*. La Jolla, CA: University of California; February, 1987:2.
16. Rumelhart DE, Zipser. Feature discovery by competitive learning. *Cogn Sci* 1985;9:75–112.
17. Dawson AE, Austin RE, Weinberg DS. Nuclear grading of breast carcinoma by image analysis. Classification by multivariate and neural network analysis. *Am J Clin Pathol* 1991;95:29–37.
18. Gross GW, Boone JM, Greco-Hunt V, Greenberg B. Neural networks in radiologic diagnosis. II. Interpretation of neonatal chest radiographs. *Invest Radiol* 1990;25:1017–1023.
19. Katz WT, Merickel MB. Translation-invariant aorta segmentation from magnetic resonance images. *International Joint Conference on Neural Networks, Volume 1*. 1989:327–333.
20. Moallemi C. Classifying cells for cancer diagnosis using neural networks. *IEEE Expert* 1991;6:8–12.
21. Kippenham JS, Barker WW, Pascal S, Nagel J, Duara R. Evaluation of a neural-network classifier for PET scans of normal and Alzheimer's disease subjects. *J Nucl Med* 1992;33:1459–1467.
22. Scott JA, Palmer EL. Neural network analysis of ventilation-perfusion lung scans. *Radiology* 1993:in press.
23. Mead C. *Analog VLSI and Neural Systems*. Reading, MA: Addison Wesley; 1989.