5-1-89

5

CONF-890634--5

DE93 002120

ELECTRIC POWER RESEARCH INSTITUTE

TUTORIAL

NEURAL NETWORKS AND THEIR POTENTIAL APPLICATION IN NUCLEAR POWER PLANTS

ROBERT E. UHRIG

Department of Nuclear Engineering
University of Tennessee, Knoxville
and
Instrumentation and Control Division
Oak Ridge National Laboratory

Prepared for

EPRI Conference
on
"Expert Systems Applications for the Electric Power Industry"

Orlando, Florida

June 5-8, 1989

ABSTRACT

A neural network is a data processing system consisting of a number of simple, highly interconnected processing elements in an architecture inspired by the structure of the cerebral cortex portion of the brain. Hence, neural networks are often capable of doing things which humans or animals do well but which conventional computers often do poorly. Neural networks have emerged in the past few years as an area of unusual opportunity for research, development and application to a variety of real world problems. Indeed, neural networks exhibit characteristics and capabilities not provided by any other technology. Examples include reading Japanese Kanji characters and human handwriting, reading a typewritten manuscript aloud, compensating for alignment errors in robots, interpreting very "noise" signals (e.g. electroencephalograms), modeling complex systems that cannot be modelled mathematically, and predicting whether proposed loans will be good or fail. This paper presents a brief tutorial on neural networks and describes research on the potential applications to nuclear power plants.

FG07-88ER12824

INTRODUCTION TO NEURAL NETWORKS

Neurons. The human brain is a complex computing device capable of thinking, remembering, and solving problems. There have been a number of attempts to emulate the brain functions with a computer model, and generally these have involved the simulation of a network of neurons, commonly called neural networks. The brain contains approximately 100 billion neurons that are densely interconnected with one thousand to ten thousand connections per neuron.

A neuron is the fundamental cellular unit of the brain's nervous system. It is a simple processing unit that receives and combines signals from other neurons through input paths called dendrites. (The basic components of a neuron are shown in Figures 1 and their schematic equivalents in Figure 2.) If the combined signal from all the dendrites is strong enough, the neuron "fires", producing an output signal along a path called the axon. The axon splits up and connects to hundreds or thousands of dendrites (input paths) of other neurons through synapses (junctions containing a neurotransmitter fluid that controls the flow of signals) located in the dendrites. Transmission of the signals across the synapses are electro-chemical in nature, and the magnitudes of the signals depend upon the synaptic strengths of the synaptic junctions. The strength or conductance (the inverse of resistance) of a synaptic junction is modified as the brain "learns". In other words, the synapses are the basic "memory units" of the brain.

Computer Simulation. The computer simulation of this brain function usually takes the form of artificial neural systems which consists of many artificial neurons, usually called processing elements or neurides. These processing elements are analogous to the neuron in that they have many inputs (dendrites) and combine (sum up) the values of the inputs. This sum is then subjected to a nonlinear filter usually called a transfer function, which is usually a threshold function or a bias in which output signals are generated only if the output exceeds the threshold value. Alternately, the output can be a continuous function (linear or nonlinear) of the combined input. Sometimes the outputs are "competitive" in which only one processing element has an output, or the processing elements may operate asynchronously in which outputs occur only when inputs occur simultaneously.

The output of a processing element (axon) branches out and becomes the input to many other processing elements. These signals pass through connection weights (synaptic junctions) that correspond to the synaptic strength of the neural connections. The input signals to a processing element are modified by the connection weights prior to being summed by the processing element. There is an analogy between a processing element and an operational amplifier in an analog computer in which many inputs are summed. The potentiometer settings on the amplifier inputs correspond to the connection weights and the output of the operational amplifier goes through some sort of nonlinear function generator.

NEURAL NETWORKS

Neural Networks. A neural network consists of many processing elements joined together to form an appropriate network with weighting functions for each input. These processing elements are usually organized into a sequence of layers with full or random connections between layers. Typically, there are three or more layers: an input layer where data are presented to the network through an input buffer, an output layer with a buffer that holds the output response to a given input, and one or more intermediate or "hidden" layers. A typical neural network arrangement is shown in Figure 3.

The operation of an artificial neural network involves two processes: learning and recall. Learning is the process of adapting the connection weights in response to stimuli presented at the input buffer. The network "learns" in accordance with a learning rule which governs how the connection weights are adjusted in response to a learning example applied at the input buffers. Recall is the process of accepting an input and producing a response determined by the learning of the network.

Learning. There are several different kinds of learning commonly used with neural networks. Perhaps the most common is the so-called supervised learning in which a stimulus is presented at the input buffer of the network and the output from the output buffer is sent to a system that compares it with a desired output and then uses a corrective or learning algorithm to convert the difference (error signal) into an adjustment of the weighting coefficients (connection weights) that control the inputs to the various processing elements. A typical supervised learning system is shown in Figure 4. In a typical situation, the initial weighting functions are set randomly and then subjected to incremental changes determined by the learning algorithm. When an input is again applied to the input buffer, it produces an output which again is compared with the desired output to produce a second error signal. This iterative process continues until the output of the artificial neural network is substantially equal to the desired output. At that point, the network is said to be "trained". Through the various learning algorithms, the network gradually configured itself to achieve the desired input-output relationship or "mapping".

There are several other kinds of learning that are commonly used. For instance, in unsupervised learning, only the input stimuli are applied to the input buffers of the network. The network then organizes itself internally so that each hidden processing element responds strongly to a different set of input stimuli. These sets of input stimuli represent clusters in the input space (which often represent distinct real-world concepts). There is also "random learning" in which random incremental changes are introduced into the weighting functions, and then either retained or dropped depending upon whether the output is improved or not (based on whatever criteria the user wants to apply). A fourth type of learning is "graded" learning in which the output is graded on some numerical scale or perhaps simply classified as "good" or "bad" and then the connection weights are adjusted in accordance with the grade assigned to the output.

The common learning algorithms are: 1) Hebbian learning where a connection weight on an input path to a processing element is incremented if both the input is high (large) and the desired output is high. This is analogous to the biological process in which a neural pathway is strengthened each time it is used, 2) Delta-rule learning in which the error signal (difference between the desired output response and the actual output response) is minimized using a least-squares process, and 3) competitive learning in which the processing elements compete among each other and only the one that yields the strongest response to a given input modifies itself to become more like the input. In all cases, the final values of the weighting functions constitutes the "memory" of the neural network.

In the recall process, a neural network accepts a signal presented at the input buffer and then produces a response at the output buffer that has been determined by the "training" of the network. The simplest form of recall occurs when there are no feedback connections from one layer to another or within a layer (i.e., the signals flow from the input buffer to the output buffer in what is called a "feed forward" manner). In this type of network the response is produced in one cycle of the computer. When the neural networks have feedback connections, the signal reverberates around the network, across the layers or within layers, until some convergence criteria is met and a steady-state signal is presented to the output buffers.

Characteristics of Neural Networks. The characteristics that make neural network systems different from traditional computing and artificial intelligence are 1) learning by example 2) distributed associative memory 3) fault tolerance and 4) pattern recognition.

The memory of a neural network is both distributive and associative. Distributed means that the storage of a unit of knowledge is distributed across all memory units (connection weights) in the network. This knowledge shares these memory units with all other items of knowledge stored in the network. Associative means that when the trained network is presented with a partial input, the network will choose the closest match to that input in its memory and generate an output that corresponds to the full output.

Traditional computer systems are rendered useless by any damage to its memory. However, neural-computing systems are fault tolerant in that if some processing elements are destroyed or disabled or have their connections altered incorrectly, the behavior of the network is changed only slightly. As more processing elements are destroyed, performance degrades gradually, i.e., the network performance suffers but the system does not fail catastrophically. This is because the information is not contained in any single memory unit, but rather is distributed among all the connection weights of the network. Such arrangements are well-suited for systems where failure may be unacceptable or introduce difficult problems (e.g., in nuclear power plants, missile guidance, and space probes.

Pattern recognition is the ability to match large amounts of input information simultaneously and generate a categorical or generalized output. It requires that the network provide a reasonable response to noisy or incomplete inputs. Experience shows that neural networks are very good pattern recognizers which also have the ability to learn and build unique structures for a particular problem.

NEURAL COMPUTING AND APPLICATIONS

Neural-computing networks consists of interconnected units that act on data instantly in a massively parallel manner. This provides an approach that is closer to human perception and recognition than conventional computers and can produce reasonable results with noisy or incomplete inputs. Neural computing is at an early stage of development. The results to date have been impressive, and they appear to complement expert systems. Future applications appear unlimited, but much development work remains to be done. A few of the recent applications of neural networks are given below to illustrate the wide spectrum of applications to which neural networks have been applied.

1. Complex system modeling. A system with multiple inputs and outputs can be modeled using a neural network by applying the system inputs to the network and using the system outputs as the desired outputs of the neural network. After an appropriate number of iterative learning cycles the neural network then constitutes a nonstructured non-algorithmic model of the process involved. Such modeling can be used on physical systems, business and financial systems, or

social systems. Current applications include the use of a neural network to determine whether loan applications should be approved using the previous five years experience of that bank as the input training data.

2. Image (data) compression involves the transforming of image data to a different representation that requires less memory. Then the image must be reconstructed from this new representation in such a way that there is an imperceptible difference from the original. Compression ratios of several hundred to one have been achieved in some cases.

3. Character recognition, a special case of pattern recognition, is the process of visually interpreting and classifying symbols. Neural networks were the first systems to efficiently read Japanese Kanji characters. This it effectively broke the input barrier for computers used in Japan.

4. Handwriting recognition involves a neural computing system that accepts handwriting on a digitized pad as a computer input and is trained by interpreting a set of handwriting types. The system can then interpret a type of handwriting it has never seen before and can make a "best guess" when confronted with a confusing character. Accuracy improves when the training is on the type of writing being read (e.g., on one individual's handwriting). A recent advance in Japan uses a process that simulates the way visual information feeds forward in the brain and has the advantage that it can recognize patterns regardless of orientation or distortion.

5. Target classification. Neural networks have been used to classify sonar targets by distinguishing between large metal cylinders and rocks of a similar size. The neural networks integrates 60 spectral energy values produced from 60 frequency bands. Its performance was comparable to the best trained human operators on the same data and significantly better than normal operators or other computer-based classifiers.

6. Noise filtering. Neural networks are able to filter noisy data and preserve a greater depth of structure and detail than any of the traditional filters while still removing the noise. Applications include removal of background noise from voice communications and separation of the fetal heart beat from a mother's heart beat.

7. Servo-control systems. Complex mechanical servo-systems, such as those used in robots, must compensate for physical variations in the system introduced by misalignments in the axes, or deviation in members due to bending and stretching induced by loads. These quantities are extremely difficult to describe analytically. A neural network can be trained to predict and respond to these errors in the final position of a robot member. This information is then combined with the desired position to provide an adaptive position correction and improve the accuracy of the member's position.

8. Text-to-speech conversion. In this application the printed symbols or letters in a text were converted into the spoken language using a neural network that taught itself to translate written text into speech in the same way that a human child learns to read. The printed transcript is broken down into the fundamental components of speech called "phonemes" which became the desired output of the neural network when the input was the corresponding text. After training, the phonemes become the input to a voice synthesizer which provides the verbal output.

APPLICATIONS TO NUCLEAR POWER PLANTS

When a complex system plant is operating safely, the outputs of hundreds, or even thousands, of sensors or control room instruments form a pattern (or unique set) of readings that represent a "safe" state of the plant. When a disturbance occurs, the sensor outputs or instrument readings form a different pattern that represents a different state of the plant. This latter state may be safe or unsafe, depending upon the nature of the disturbance. The fact that the pattern of sensor outputs or instrument readings is different for different conditions is sufficient to provide a basis for identifying the state of the plant at any given time. To implement a diagnostic tool based on this principle, that is useful in the operation of complex systems, requires a rapid (real-time), efficient method of "pattern recognition." Neural networks offer such a method.

Useful Features of Neural Networks. Neural networks may be designed so as to classify an input pattern as one of several predefined types of faults or transients (e.g., the various fault or transient states of a power plant) or to create, as needed, categories or classes of system states which can be interpreted by a human operator. Neural networks have demonstrated high performance even when presented with noisy, sparse and incomplete data.

A second desirable feature of neural networks is their ability to respond in real-time to the changing system state descriptions provided by continuous sensor input. For complex systems involving many sensors and possible fault types (such as nuclear power plants), real-time response is a difficult challenge to both human operators and expert systems. However, once a neural network has been trained to recognize the various conditions or states of a complex system, it only takes one cycle to detect a specific condition or state. Because neural networks can be trained to recognize the patterns of different sensor outputs or instrument readings that give rise to different system states or faults, they are ideally suited for real-time diagnostics.

Neural networks have the ability to recognize patterns, even when the information comprising these patterns is noisy or incomplete. Unlike most computer programs, neural network implementations in hardware are very fault tolerant; i.e. neural network systems can operate even when some individual nodes in the network are damaged. The reduction in system performance is about proportional to the amount of the network that is damaged. Thus, systems of artificial neural networks have high promise for use in environments in which robust, fault-tolerant pattern recognition is necessary in a real-time mode, and in which the incoming data may be distorted or noisy. This makes artificial neural networks ideally suited as a candidate for fault monitoring and diagnosis, control, and risk evaluation in complex systems, such as nuclear power plants.

NEURAL NETWORK PROJECTS AT THE UNIVERSITY OF TENNESSEE

In October 1988, a three-year Department of Energy contract entitled "Enhancing the Operation of Nuclear Power Plants through the Use of Artificial Intelligence" was initiated at the University of Tennessee at a funding level of about $250,000 per year. This program calls for a number of investigations of how both expert systems and neural networks can be applied (both on-line and off-line) to enhance the overall operation of nuclear power plants. Most of these projects involve the use of computers to carry out the work and to simulate a nuclear power plant or certain plant systems in order to demonstrate the feasibility of the process being investigated. The investigations currently in progress under this program include:

1. <u>Multi-sensor fusion</u>. This project has as its goal the integration of diverse types of signals (temperature, pressure, neutron flux, etc.) at many locations (in the core, the primary coolant loop, the steam generators, etc.) in order to provide a reasonable representation of the processes involved in a nuclear power plant. The method involves the adaption of neural network technology from "scene representation." This project is being carried out at the University of Tennessee Space Institute in Tullahoma, Tennessee under the direction of Dr. Alianna J. Maren.

2. <u>Signal validation</u>. This project is investigating the feasibility of using neural networks to validate the signals coming from many sensors of the same or similar type located in a subsystem of a plant. The acceleration of the backpropagation network training and minimization of signal prediction error are being studied.[1] This project is being carried out under the direction of Dr. Belle Upadhyaya in the Department of Nuclear Engineering at University of Tennessee, Knoxville.

3. <u>Nuclear fuel management.</u> This project is investigating the feasibility of using neural networks to significantly reduce the computation involved in establishing core reload configurations in nuclear power plants. A separate investigation involves the application of the optimization methods of the "traveling salesman problem" to optimizing the fuel element pattern for some specific parameter (e.g., economy, minimum leakage of neutrons from the core, minimum uranium consumption, lowest peaking factor, etc.). This project is being carried out by Dr. Laurence Miller in the Department of Nuclear Engineering at the University of Tennessee, Knoxville.

4. <u>Modeling and Diagnostics in Nuclear Power Plants</u>. This project has as its goal the development of diagnostic methods using neural networks to detect faults and transients in nuclear power plants. In some cases, this will involve the development of non-structured, non-algorithmic models of process or major components using neural network techniques. In other cases, this model will be part of the knowledge base of an expert system. This work is being carried out under the direction of Dr. Robert E. Uhrig in the Department of Nuclear Engineering at the University of Tennessee.

5. <u>Prediction of Energy Needs of the United States using Neural Networks</u>. This project has as its goal the development of multiple neural networks to predict the future energy needs in the United States. The underlying goal of this project is <u>not</u> just the prediction of energy needs, but rather, the development of a methodology using multiple neural networks which may also have application to the diagnostic processes. This work is being carried out under the direction of Dr. Robert E. Uhrig in the Department of Nuclear Engineering at the University of Tennessee.

6. <u>Neural Network Algorithms that "Learn" in a Single Cycle</u>. A new approach to learning in neural networks has been developed that allows the network to learn to recognize a pattern in a single cycle. It uses competitive learning in the "hidden" layer and can learn as many patterns as there are processing elements in this layer. Both uni-directional and bi-directional versions of this algorithm have been demonstrated. This work is being carried out under the direction of Dr. Robert E. Uhrig in the Department of Nuclear Engineering at the University of Tennessee, Knoxville.

Some of these projects involve only simulation on computers to demonstrate proof of principle. Many of them can benefit significantly by utility involvement to demonstrate their usefulness in nuclear power plants. Some of them will require extensive testing in a training or engineering simulator of a nuclear power plant prior to implementation in nuclear power plants. For instance, the use of neural networks to identify abnormal conditions or transients in nuclear power plants would require extensive use of a sophisticated nuclear power plant simulator, preferably a full-fidelity simulator of a particular nuclear power plant.

IDENTIFICATION OF SYSTEM TRANSIENTS

Let us look at a typical application of neural networks, the identification of system transients in a nuclear power plant. The goal is the demonstration of the feasibility of using a neural network to diagnose different fault conditions or transients in a nuclear power plant. The initial task was the simulation of transients of a steam generator of a pressurized water reactor. For demonstration purposes, a set of data from a simulation of an isolated U-tube steam generator (UTSG) is used as the training data for the neural network.[2,3]

For this demonstration, only six step-change perturbations were introduced into the system:

1. Positive and negative step changes in primary inlet temp.,
2. Positive and negative step changes in feedwater temp., and
3. Positive and negative step changes in the valve coefficient (percentage of the full-open position).

Four variables (i.e., sensor outputs or instrument readings) were chosen as the system responses to represent the system behavior for each perturbation. They were: the primary inlet temperature, the primary outlet temperature, the downcomer water level, and the steam pressure. The time-records of these four quantities show that the patterns are quite different for each of the six perturbations listed above. Since the back-propagation network can be trained to distinguish between the different patterns, it can distinguish between the different kinds of steam generator perturbations.

The time-records of these six perturbations (at the 10°F and 10% perturbation levels) constitute the input pattern for training the back-propagation neural network. Ten samples were taken at ten second intervals for each of the four variables from the response curves (a total of 40 values) for each of the six simulations, digitized, and used as inputs to the neural network. The desired outputs for training of the network were defined by a 3-bit binary quantity, representing the output states of the six perturbations.

In this study, a three layer back-propagation network was set up with 40 processing elements (PEs) in the input layer and three PEs in output layer to match the dimensions of the training input and output vectors. The middle layer contained 12 PEs because this size represented a reasonable compromise between ease of training and precision for the number of inputs and outputs.

The digitized time-record for the perturbations of ± 10°F in primary inlet temperature, ± 10°F in feedwater water temperature, and ± 10% in valve opening coefficient were used as six sets of training data for the network. The data were normalized before putting them into the neural network to satisfy the requirement of the input form of the neural network used. After 500 data training cycles using the back-propagation learning algorithm (which may require anywhere from a few seconds to a few hours, depending on the computer hardware and software used), the network readily distinguished between the six different perturbations. Random fluctuations were introduced into the trained network, and the recall results showed that the network could still identify each of the UTSG transients correctly, even when the amplitude of the noise was equal to 90% of the amplitude of the signals.

Acknowledgement. Some of the work described here was supported by the U. S. Department of Energy under contract # DOE DEFG-88ER12824, entitled "Enhancing the Operation of Nuclear Power Plants Through the Use of Artificial Intelligence." This support is gratefully acknowledged.

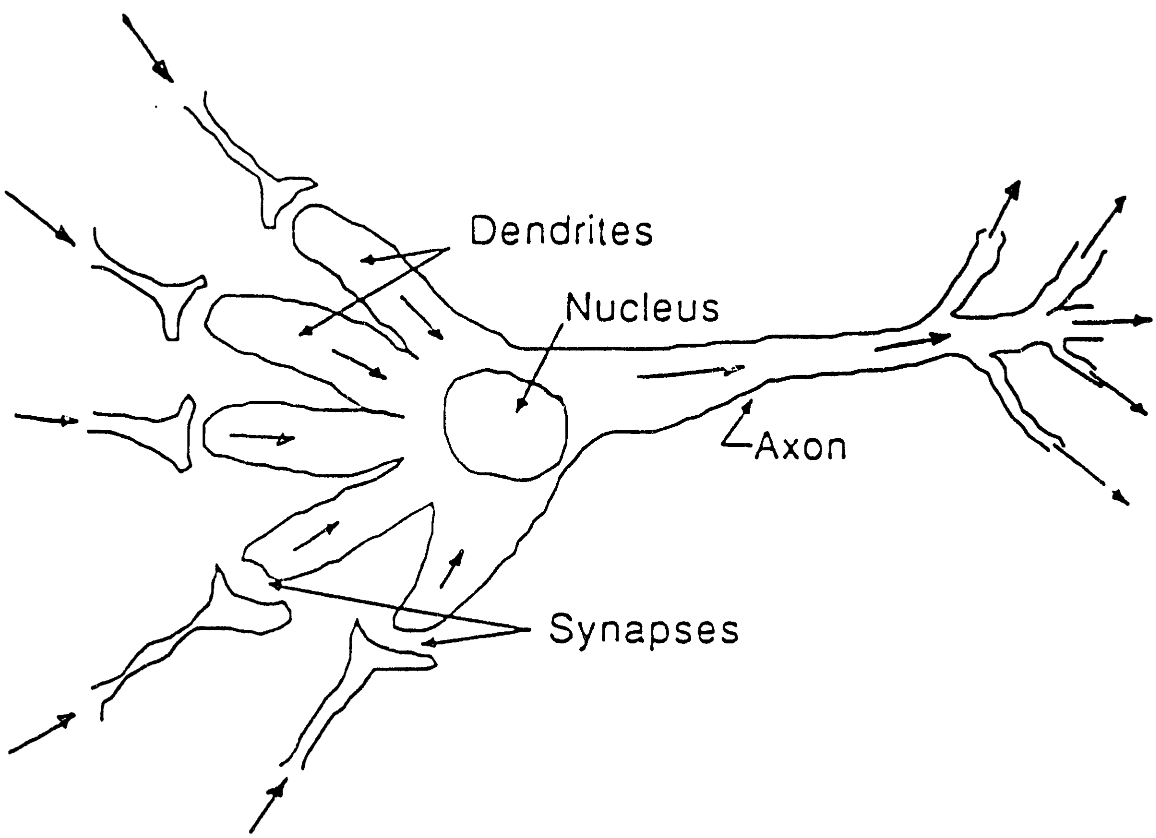

FIGURE 1. SKETCH OF A NEURON SHOWING COMPONENTS

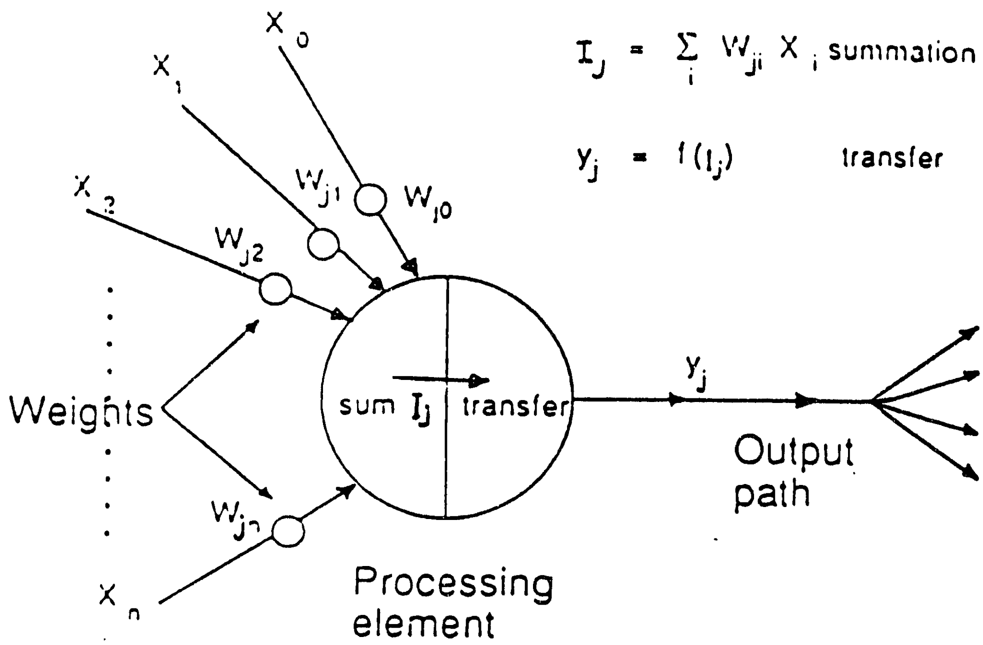

FIGURE 2. SCHEMATIC REPRESENTATION OF A NEURON

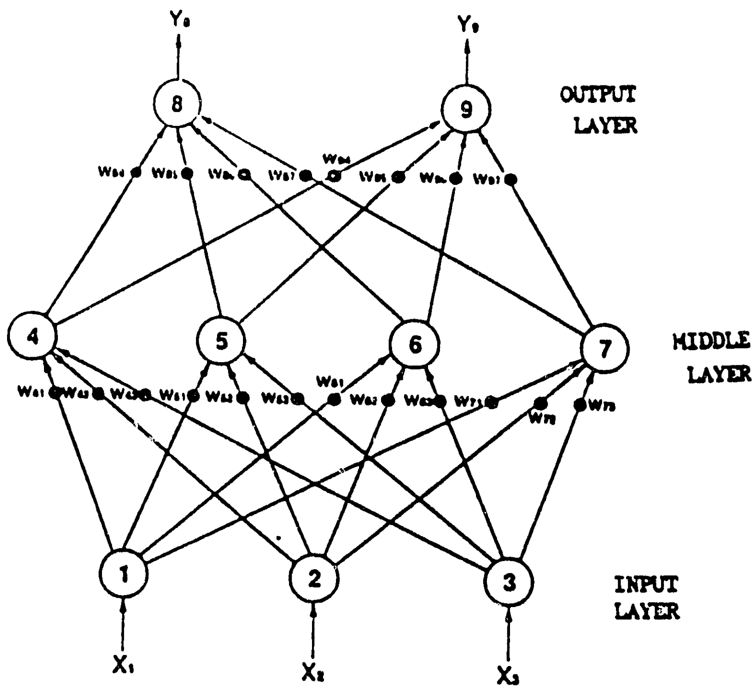

FIGURE 3. A SIMPLE 3-LAYER NEURAL NETWORK

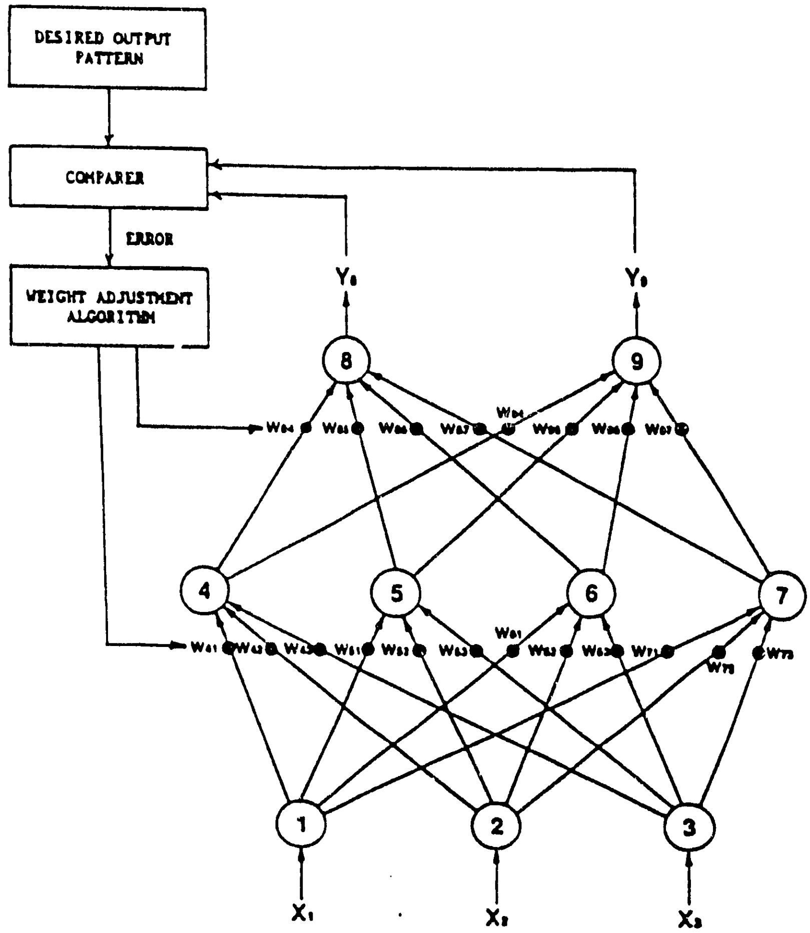

FIGURE 4. A TYPICAL SUPERVISED LEARNING SYSTEM

References

1. B. R. Upadhyaya, E. Eryurek, G. Mathai, "Application of Neural Computing Paradigms for Signal Validation," Seventh Power Plant Dynamics, Control and Testing symposium, Knoxville, TN, May 1989.

2. R. E. Uhrig, Z. Guo, "Use of Neural Networks in Diagnoses of Transient Operating Conditions Using a Back-Propagation Neural Network," Proceedings of SPIE-the International Society for Optical Engineering VII, "Application of Artificial Intelligence," Orlando, FL, March 28-30, 1989.

3. T. W. Kerlin, E. M. Katz, "Pressurized Water Reactor Modeling for Long-Term Power system Dynamics," EPRI EL-3087, Volumes 1, 2, May 1983.

DATE
FILMED
01/20/93