Journal of International Information Management

Volume 2 | Issue 1 Article 7

1993

Neural networks: The natural road to artificial iiitelligence

Mohammad Dadashzadeh
The Wichita State University

Behnam Bahr
The Wichita State University

Neural networks: The natural road to artificial intelligence

Mohammad Dadashzadeh
Behnam Bahr
The Wichita State University

ABSTRACT

Despite the many advances of artificial intelligence (AI) technology, most notably in the area of expert systems, efforts to build intelligent systems that would approach the common-sense reasoning and sensory abilities of even a small child have not been rewarding. A small, but growing number of researches believe that the existing AI toolboxes of symbolic representation and heuristic search may not hold the answer, and that massively parallel networks of simple neuron-like processing elements may hold the key. In this overview article, we examine the use of neurally inspired concepts in the construction of intelligent machines, and address their practical applications, advantages, and limitations.

INTRODUCTION

Japan's announcement of the Sixth Generation project has marked the beginning of a new era. While the Japanese Fifth Generation project concentrated on artificial intelligence, the Sixth Generation emphasizes *natural intelligence* or brain-like computers (Soucek and Soucek, 1988). Brain-like computers (or neural networks) consist of a large number of simple processing units linked together in such a way that each unit receives signals from other units, perhaps tens of them, and sends signals to the same or other units. On its own, a processing unit is not intelligent. It merely multiplies each signal it receives by the weight associated with its connection to the transmitting unit, and performs some arithmetic on the results to decide whether or not to transmit a signal of its own. In this manner, a neural network seemingly imitates the brain, in which a signal from one neuron (brain cell) can trigger a cascade of thousands of other signals until a memory is evoked and a pattern is recognized. Also, the weights in a neural network are not fixed and may change. In fact, each network utilizes a *learning law*, an equation used by the processing units to adjust the weights of their input signals according to the success or failure of the network in its task.

Already, there have been many impressive demonstrations of neural network capabilities. Consider the following examples:

- NETtalk (Sejnowski and Rosenberg, 1987) is a neural network that converts *unrestricted* English text into speech. As a research project, it took one summer to develop and in 10 hours trained *itself* to the performance level of DECtalk, a commercial text-to-speech system that took several years to build and that uses a look-up table of about one million bits to store the phonetic transcription of words and uses phonological rules to produce the string of phonemes for words not found in its look-up table. In contrast, NETtalk used no look-up tables, and no phonological rules. It acquired its competence through 10 hours of practice.

- Denker et al. (1989) report on the construction of a system that recognizes hand-printed digits. The system utilizes a combination of classical pattern recognition techniques and neural network methods. It was trained and tested on real-world data derived from zip-codes seen on actual U.S. mail. On a data set of approximately 10,000 digits, the system rejected 14% of the images as unclassifiable, with only 1% of the remainder being misclassified. The results surpass other methods based on expert system technology.

- An automated signature verification system developed at Stanford University (Mighell et al., 1989) intended to reduce the costly verification of signatures when large sums of money and other issues may be involved, achieves a 2% true signature rejection with 2-4% false signature acceptance. The neural network employed is a feedforward network using a hidden layer and employs the backpropagation learning algorithm. The network is *simulated* in a program written in C running on an IBM PC/AT with binary images of the signature being input from a digitizing camera.

- ALVINN (Autonomous Land Vehicle In a Neural Network) is a neural network designed for road following. It takes images from a camera and a laser range finder as input and produces as output the direction in which the vehicle must travel in order to follow the road. After training on 1200 simulated road images, ALVINN learns to indicate the turn curvature within two units of the correct answer 90% of time on new, unfamiliar roads. It took ALVINN about one half hour of training to achieve comparable performance to the traditional vision-based autonomous navigation algorithms under the limited conditions tested (Pomerleau, 1989).

- Neural networks have been applied in robotics to solve the difficult inverse kinematics problem. Essentially, the problem is to convert the desired Cartesian coordinates of a robot's hand to required angular coordinates of the robot's joints. Present day models are not capable of adapting themselves to natural changes in the structure of the robot (e.g., due to wear) or to sudden changes in the dynamics of the robot that may occur. Neural network based systems, however, promise to be able to learn movement without an accurate model of the robot or complex parameter estimation. Mel (1989) reports on MURPHY, a vision-based controller that has learned a "mental model" of its camera-arm system and has used this model to reach for an object in the presence of obstacles.

As it can be seen from the above examples, commercial applications of neural network technology will be in areas that have resisted traditional AI approaches. These areas include continuous speech recognition, diverse pattern recognition problems, computer vision, and adaptive robotics. Projections of neural network capabilities include intelligent systems that will be able to make sense of normal conversation and teach themselves how to speak by listening; robots that will be able to pick up a part, orient it properly, insert it, and be able to identify and adapt to any malfunction in the assembly process; financial analysis systems that will be able to look at seemingly unrelated masses of data and identify patterns in it;

and user interfaces that routinely get past misspellings or incomplete input by recognizing what the user intended (Soucek and Soucek, 1988). In short, in the same manner that research in artificial intelligence has advanced robots from simple programmed mechanical manipulators to intelligent machines capable of expert-like planning and fault diagnosis, research in neural computing promises to add further dimensions of noise-tolerant sensory capabilities and graceful movement to allow robots to behave more like living organisms. In the remainder of this article, we briefly examine the structure of neural networks, and address their practical applications, advantages, and limitations.

THE ANATOMY OF A NEURAL NETWORK

Apart from the fact that it processes information, a digital computer bears little resemblance to the human brain. It works in serial fashion, under the control of a central processing unit, according to explicit instructions which are stored along with the symbols being manipulated in a location-addressable memory. The brain is altogether another kind of device. It contains some one hundred billions of processing units, called neurons, each connected to anywhere from hundreds to thousands of others. Compared to the nanosecond speeds of today's digital computers, the millisecond transmission speeds of nerve impulses seem incredibly slow. However, the brain achieves its astonishing computation speed through parallel processing by recruiting its entire force of processors (neurons) to work on a given problem all at once. Also, the brain appears to utilize a form of content-addressable memory manifested in its abilities to store a huge amount of knowledge and to find relevant items in this knowledge-base very quickly and to retrieve a data item from only a fragment of it or its association with another item.

By the early 1940s, a great deal of work in neuroanatomy had produced a rough "circuit diagram" of prototypical brain tissue. The individual neurons were known to receive electrical signals from other neurons through their multi-branched arrays of input wires (called dendrites). If the average of such signals over a short time interval is sufficiently large, a neuron becomes electrically active (or fires). Upon firing, a neuron emits electrical signals that travel over an array of output wires (called axons) connected to the dendrites of other neurons (via connections called synapses), and which either enhance or impede the firing of these other neurons whenever the first neuron fires.

Along with the progress in neuroanatomy, psychologists were advancing models of human learning. One such model proposed by D. O. Hebb in 1949 answered the perplexing question of how learning could take place without a teacher. Basically, Hebb proposed that a synapse connecting two neurons is strengthened whenever both of these neurons fire. Although Hebb's learning law has a few drawbacks including the fact that it does not deal with negative or inhibitory connections between neurons, it became the starting point for artificial neural network training algorithms.

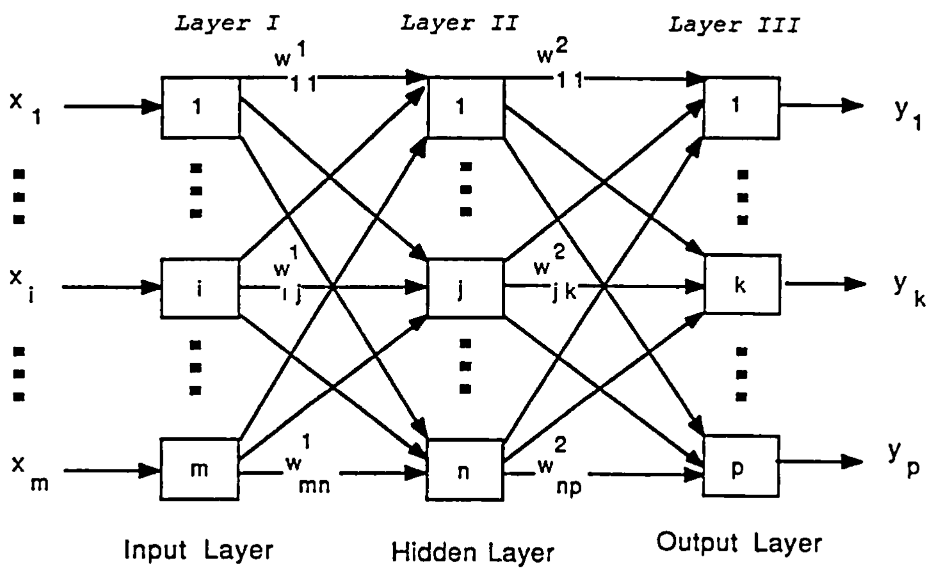
In June 1960, Frank Rosenblatt of Cornell University unveiled the Perceptron. It had an eye made of photoelectric cells that scrutinized letters inscribed on cards and passed signals on to randomly interconnected input units which in turn sent signals to a set of output units indicating what letter was on the card. The Perceptron used a *supervised* training algorithm where after each trial each unt's interconnection weights were adjusted to reduce the error. The Perceptron learned to recognize letters on its own, and although it had many shortcomings—for example, it could not identify the letter if it was partly hidden or printed in an unfamiliar typeface—it created a sensation.

The great interest in Perceptron learning devices was given a death blow with the publication of *Perceptrons* (Minsky and Papert, 1969). They proved that the two layer networks (i.e., consisting of only input and output units) were theoretically incapable of solving many simple problems which can be characterized formally as *not linearly separable,* such as telling apart inputs that differed in only one bit. Moreover, they conjectured that such limitations would prove true for more complicated networks.

Their conjecture was disproved only recently by Robert Hecht-Nielsen who showed that by adding an intermediate layer of processing units (often called the hidden layer) a neural network will be theoretically capable of solving all those nonlinearly separable problems (Caudill, 1988).

Today's neural networks often adopt a hierarchically layered architecture which includes one or more hidden layers. Figure 1 depicts a three-layer, feedforward neural network which is fully interconnected.

Figure 1. A Fully Interconnected Three-Layer Neural Network



x_i and y_k denote respectively the inputs of the network, while w^r_{st} denotes the weight associated with connection from node s at layer r to node t in the next layer.

The activation function associated with a processing unit (neuron) is responsible for producing the unit's output signal. The activation function may be a simple linear function that produces an output of 1 only if the weighted sum of the input signals is greater than a constant threshold value, or it may be a nonlinear function returning values between zero and one. It has been shown that linear activation functions severely limit the ability of neural networks, and thus most sophisticated neural networks employ nonlinear activation functions. Also, most complex networks do contain feedback connections which allows their outputs be determined not only by the current inputs and the weights but also by a history of their previous inputs.

The last component of neural networks is their learning algorithm. In supervised training, a neural network learns by being presented with a training pair, i.e., an input vector (X in Figure 1) along with the desired output vector (i.e., Y in Figure 1). The learning algorithm calculates the network's output vector and compares it with the desired output and uses the difference (error) to adjust the weights (i.e.,W_{ij} in Figure 1). Using a training set consisting of *thousands* of training pairs, the learning algorithm converges on a set of weights that optimizes response correctness over all sampled input vectors. When this happens, the weights are frozen and the training is considered to be complete.

Despite the many successes of supervised training, unsupervised training is considered to be a far more plausible model of learning in biological systems (Wasserman, 1989). In unsupervised training, the training set consists only of input vectors. The learning algorithm tries to adjust the weights in such a way that input vectors that are sufficiently similar produce the same output vector. In a very real sense, the neural network learns to create categories and extract the essential statistical properties of the training set.

Only ten years ago there were no learning algorithms capable of dealing with neural networks with multiple hidden layers. Today, there are a tremendous variety of learning algorithms in use. Table 1 lists some of the most prominent ones leaving others and the treatment of their specific advantages and limitations to elsewhere in literature (Butler and Caudill, 1989; Caudill, 1987-1989; Hecht-Nielsen, 1988; and Wasserman, 1989).

Table 1. Few Prominent Neural Network Learning Algorithms

ALGORITHM	FEEDBACK	TRAINING
Adaptive Resonance Theory	Yes	Unsupervised
Backpropagation	No	Supervised
Boltzmann	No	Unsupervised
Hebb	No	Unsupervised
Hopfield	Yes	Unsupervised
Kohonen	No	Unsupervised
Perceptron	No	Supervised

NEURAL COMPUTING VERUS TRADITIONAL COMPUTING

Neural computing is a fundamentally different information processing paradigm than algorithmic programming (Hecht-Nielsen, 1988). The salient characteristics of neural networks that are considered especially attractive are listed in Table 2. The hallmark of neural networks is their distributed, superimposed memory. A fact, a concept, an item of knowledge is not stored in a specific place as in the traditional systems. Instead, information is stored in the specific configuration of synaptic connection strengths, as shaped by past learning and is "released" by the whole pattern of activity that comes to life as different processing units turn on and off and send signals to one another along their weighted connections. Moreover, not only is an item of knowledge spread out across a region of the network, it is also superimposed on other items, so that a single processing unit is involved in encoding many items, and many processing units take part in the representation of a single item. Although superimposing memory might at first be thought to lead to confusion, the system, in fact, works in such a way that items that are sufficiently different are kept separate, but items that are sufficiently similar interact.

Table 2. Salient Characteristics of Neural Networks

Distributed, Superimposed Memory
Adaptive Learning
Associative Recall
Generalization
Abstraction
Inherent Fault Tolerance

From a practical point of view, this distributed, superimposed memory ensures durability and an inherent fault-tolerance, especially valuable in building massively parallel systems. The loss of a few components, even quite a few, has a negligible effect on the character of the overall transformation performed by the surviving network. As pointed out by Fahlman and Hinton (1987), it is much easier to build a billion-transistor system in which only 95 percent of components have to work, than building a million-transistor system that has to be perfect.

The adaptive nature of neural networks, more than any other characteristic, is responsible for the interest they have received. Neural networks learn to modify their behavior in response to their environment. Instead of containing formal descriptions of an object, a neural network learns to recognize the object through practice.

Neural networks are also at home with imperfect information. A network responds to a new or defective item by generalizing and finding a closest match. The ability to see through noise and distortion is an inherent property of neural networks, and not the result of human intelligence embedded in clever computer programs. However, neural networks do not promise to be exact. They excel in those situations where close is a good enough answer.

Finally, an emerging property of neural networks is the ability to abstract the essence of a set of inputs to form categories and create prototypes. Campbell (1989) reports of a network devised by James McClelland that was given descriptions of members of two New York street gangs, the Sharks and the Jets. The descriptions included name, age, occupation, marital status, and education of the members. Given these inputs, the network was able to describe a *typical* Jet as being single, in his twenties, with a junior-high-school education. "In fact, however, the typical Jet did not exist. There was no member of the gang who matched that description exactly." (Campbell, 1989, p. 177)

LIMITATIONS OF NEURAL NETWORKS

Neural networks are not a panacea. In general, they do well in the kinds of problems people can solve easily and fare poorly in the kinds of problems traditional computers do well in but people solve slowly and inefficiently. Neural networks are unsuited for tasks requiring precise, numerical computations. However, they may become the preferred approach in a large class of pattern recognition tasks (including continuous speech recognition, computer vision, and autonomous vehicles) that have defied traditional AI approaches.

In Table 3 we list several consequential limitations of neural networks. First, a significant question mark for neural computing in its present state is that the amount of training needed to make a network intelligent may be exorbitant. The idea that if you can train a network with a few hundred connections, then there is no reason why a network with millions of modifiable connections cannot be trained ignores the combinatorial explosion problem—that the number of training runs required may be unmanageably large. A crucial challenge facing researchers is to develop faster learning schemes that can be scaled up to massive networks.

Table 3. Consequential Limitations of Neural Networks

Combinatorial explosion problem of training
Inability to explain their reasoning
Unpredictability and the accompanying unrealiability
Potential inaccuracies and oversimplifications in the theoretical foundation

Another question mark over the theoretical foundation of neural computing is that the codes of the brain may be more elaborate than was once believed. As pointed out by Campbell (1989), the visual system, for example, seems to use a form of multiplexing in transmitting signals, and if the brain as a whole turns out to use such intricate codes, then the doctrine that the processing units need only send very simple signals to one another would be challenged.

Another limitation of neural networks is their inability to *explain* how they solve problems. This is in stark contrast to expert systems that can trace their reasoning process so that a human can check it for reasonableness.

A final limitation of neural networks is their unpredictability. Unless every possible input has been tested, there is no way to predict the output. Such exhaustive testing is impractical for all but the most trivial networks, and thus achieving a desired statistical estimate of performance must suffice. Therefore, neural networks will sometimes make errors (e.g., accepting a forged signature) even when they are functioning correctly. However, for most people this translates into unreliability—a characteristic that we have found unacceptable in our machines. As pointed out by Wasserman (1989), our attitude is not changed by the fact that a human in the same situation might also make mistakes, and questions regarding the reliability of neural networks must be resolved before they are put to use in applications where human life or valuable assets are at stake.

SUMMARY AND CONCLUSION

Attempts to produce intelligent machines have met during the past 35 years with a curious mix of progress and failure. Despite the many advances of AI technology, most notably in the area of expert systems, efforts to develop intelligent systems that would approach the commonsense reasoning and sensory abilities of even a small child have not been rewarding. A small, but growing number of researchers believe that classical AI is unlikely to yield intelligent machines, but systems that mimic the brain might. In this article, we have examined the use of neurally inspired concepts in the construction of intelligent machines, and have addressed their practical applications, advantages, and limitations.

There are several points that are becoming more apparent. First, most experts agree that the role of neural networks is that of a partner to traditional systems and not a replacement for them. As pointed out by Wasserman (1989), one can envision an intelligent system that manifests such a partnership as follows. The neural network component would produce an appropriate response to its environment under most circumstances, and because it can associate a confidence level to its decisions it would be able to refer those cases for which its confidence level is low to the expert system component for resolution. "The combination of the two systems would be more robust than either acting alone, and it would follow the highly successful model provided by biological evolution." (Wasserman, 1989, p. 9).

Second, as some have argued, neural networks can so far do nothing that *cannot* be done by other means (Kinoshita and Palevsky, 1987). Although there have been many impressive demonstrations of neural network capabilities, neural networks have yet to prove themselves in the marketplace. As noted by Wasserman (1989), there exists a danger that neural networks be oversold before their time as today's theoretical foundations are inadequate to support some of the projections made.

Finally, as pointed out by Arbib (1989, p. 11), our lack of knowledge of how the neuronal structure of each region of the brain fits it for its role in the overall functional structure of the brain may direct the future of neural computing in abandoning attempts at developing huge homogeneous networks utilizing some grand unified learning rule, in favor of "an understanding of a mode of cooperative computation which integrates the activity of diverse subsystems, many of which are quite specialized in nature."

The brain's massive parallelism, analog processing, and self-adaptive architecture provide it with superior computational advantages in solving such complex tasks that confront living organisms as recognizing a predator's outline in a noisy environment, recalling instantly how to flee its approach, navigating in a constantly changing physical and social environment, and so on. But, despite the many unknowns still remaining about the functioning of the brain, it need not be the only physical system capable of doing so. Neural networks which also exploit massive parallelism, analog processing, and feedback present a promising vista by attempting to adopt the nature's approach to creating intelligent systems.

REFERENCES

Arbib, M. A. & Amari, S., eds. (1989). *Dynamic Interactions in Neural Networks: Models and Data*. New York, NY: Springer-Verlag.

Butler, C. & Caudill, M. (1989). *Naturally Intelligent Systems*. Hillsdale, NH: Lawrence Erlbaum.

Campbell, J. (1989). *The Improbable Machine.* New York, NY: Simon and Schuster.

Caudill, M. (1987-1989). Neural networks primer, Parts I-VIII. *AI Expert*. Part I, December 1987, pp. 46-52.

Denker, J. S., et al. (1989). Neural network recognizer for hand-written zip code digits. In *Advances in Neural Information Processing I.* Edited by D. S. Touretzky. San Mateo, CA: Morgan Kaufmann.

Fahlman, S. E. & Hinton, G. E. (1987, January). Connectionist architectures for artificial intelligence. *IEEE Computer*, pp. 100-109.

Hecht-Nielsen, R. (1988, March). Neurocomputing: Picking the human brain. *IEEE Spectrum*, pp. 36-41.

Kinoshita, J. & Palevsky, H. G. (1987, May). Computing with neural networks. *High Technology*, pp. 24-31.

Mel, B. W. (1989). Further explorations in visually-guided reaching: Making MURPHY smarter. In *Advances in Neural Information Processing I.* Edited by D. S. Touretzky. San Mateo, CA: Morgan Kaufmann.

Mighell, D. A., et al. (1989). Backpropagation and its application to handwritten signature verification. In *Advances in Neural Information Processing I.* Edited by D. S. Touretzky. San Mateo, CA: Morgan Kaufmann.

Minsky, M. & Papert, S. (1969). *Perceptrons*. Cambridge, MA: MIT Press.

Pomerleau, D. A. (1989). ALVINN: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing I.* Edited by D. S. Touretzky. San Mateo, CA: Morgan Kaufmann.

Sejnowski, T. J. & Rosenberg, C. R. (1987). Parallel networks that learn to pronounce English text. *Journal of Complex Systems, 1,* 145-168.

Soucek, B. & Soucek, M. (1988). *Neural and Massively Parallel Computers.* New York, NY: John Wiley & Sons.

Wasserman, P. D. (1989). *Neural Computing: Theory and Practice*. New York, NY: Van Nostrand Reinhold.