5. SYMPOSIUM: HYBRIDIZATION OF NEURAL NETWORKS AND EXPERT SYSTEMS

Chaired by Frederick J. Bremner, *Trinity University*

Shades of Braitenberg's Vehicle 5

FREDERICK J. BREMNER and STEPHEN J. GOTTS
Trinity University, San Antonio, Texas

A model with a hybrid neural network-expert system brain was designed. This model proved to have better memory than that of Braitenberg's (1984) original simulated vehicles, on which it was based. The parallel nature of the neural network component provided more storage capacity with the same number of neurons, whereas the serial expert system component was able to use the increased memory to make fast, accurate decisions.

There is a general belief that computer-resident neural networks function like the nervous system (Bremner, Gotts, & Denham, 1994; Hinton & Anderson, 1981) and that expert systems use the logic of human cognitive processing. Although some authors (Lawrence, 1993, p. 32) have discussed neural networks and expert systems as opposing strategies, a hybridization of the two technologies is emerging (Fishman & Barr, 1991). The hybrids capitalize on the learning and pattern-recognizing ability of neural networks and the deductive logic of expert systems. It can be said that neural networks are rule free and that expert systems are rule bound. Neural networks do have rules but are autopoietic, in contrast to expert systems, for which the programmer generates the rules.

Expert systems can be contrasted with neural networks in a number of other ways as well. For example, expert systems use deductive reasoning, whereas neural networks use inductive reasoning (Lawrence, 1993, p. 3). On the other hand, expert systems can ask their questions (apply their rules) in any order (Negoita, 1985, p. 25), in contrast to neural networks, whose rules are elusive or even unknown (Bremner et al., 1994). Expert systems can also be contrasted with standard production systems, written in languages such as BASIC or FORTRAN, in which rules are numbered and must be executed in a particular order. The knowledge of an expert system is in its procedures (asking the right question), not in its database; therefore, adding new data does not necessitate reprogramming. Ordinarily, expert systems do not have the same capacity to learn and to generalize that learning as

neural networks do. In fact, the hallmark of neural networks is their dynamic capacity to generalize and make a best guess answer regarding new or partial data. Finally, one can query an expert system to justify its conclusion (Why are you going east? Because the sun is low in the west. There are no predators. Food need is low.) One cannot query a neural network to justify its conclusion (Fishman & Barr, 1991). At best, it would reply that its synaptic weights were in a particular state (Bremner et al., 1994).

We intend to capitalize on the strengths of both the neural network and the expert system to produce a hybrid system. We see the hybrid neural network/expert system as a model of living nervous systems, in which some behaviors are learned, and in which the nervous system dynamically changes to accommodate this learning. For instance, bluejays learn to avoid eating monarch butterflies (Gould, 1982, p. 264). On the other hand, some behaviors are hard-wired by the genetic code. For instance, the motor pattern used by *Tritonia* to escape predation is preprogrammed (Gould, 1982, p. 167). Therefore, a nervous system combining neural networks and expert systems seems to be realistic.

In this study, we capitalized on a kind of evolution in silicon similar to that presented by Braitenberg (1984). Braitenberg described a series of vehicles that he designed out of simple electronic components. One might consider these vehicles as an engineer's fantasy of how the brain evolved or as exercises in synthetic psychology (Braitenberg, 1984). All of the vehicles had two basic components: a photocell and a motor. Later models had several photo cells and at least two motors. The photocells acted as visual transducers, and the motors, as the muscle system.

Vehicle 5 (known as Logic) had two visual transducers, two reversible motors, and several memory regis-

Correspondence should be addressed to F. J. Bremner, Psychology Department, Trinity University, 715 Stadium Drive, San Antonio, TX 78212 (e-mail: fbremner@vm1.tucc.trinity.edu).

ters. Indeed, Vehicle 5 was logical and could do many amazing things, but it was memory limited. With 10 linear threshold devices (hidden neurons) it could store only a 10-digit binary number (Braitenberg, 1984, p. 24). Even more advanced Vehicles suffered from this linear mentality. Adding a connectionist, neural network would help to solve this limited memory problem without a significant increase in hardware. The memory of every digit would be distributed across all of the synapses of the 10 hidden neurons. If the receptor cells (cones) of the transducer are conceptualized as a 10×10 matrix retina (this is a throwback to Braitenberg's Vehicle 3c), then there are 1,000 connections with the 10 hidden neurons. Therefore, they can learn to store much more information than did Vehicle 5, without a significant increase in hardware (Brainmaker, 1990, pp. 10–13). We propose Vehicle 5n, which incorporates this newly evolved synaptic strategy.

Vehicle 5n employs a back propagation neural network that serves as a visual transducer complete with sensory memory (Bremner & Gotts, 1993; Gotts, 1994). The output of the neural network acts as part of the knowledge base for the expert system. In short, the nervous system of Vehicle 5n uses two strategies: a neural network for stimulus discrimination, and an expert system for approach/avoidance decisions.

METHOD

Subjects

The subject for this study is a simulated vehicle similar in design to Vehicle 5 described by Braitenberg (1984). Our vehicle (Vehicle 5n) differs from Vehicle 5 in that its visual transducer is a back propagation neural network that can discriminate sun position. It also differs in that its nervous system beyond the transducer is like an expert system. The biological models of Vehicle 5n are the gecko and the bluejay.

Apparatus

The computer used for the simulations was a Dell 386DX with 4 Mb of memory and a math coprocessor chip (63779 SX). A number of software packages were used: the simulation of the visual transducer was done with *Brainmaker*, Version 2.0 (1990); the simulation of the expert system was done with *MicroSaint*, Version 3.2 (1987); and the interface between *Brainmaker* and *MicroSaint* was done with *CompuAdd Windows*, Version 3.1 (1990). In addition, a combination neural network/expert system program was written in the C computer language (*Microsoft C/C++* compiler, Version 7.0, 1991).

Procedure

Two vehicles were designed: one using commercially available programs and one using custom software. Although the flow chart and functional strategy of both vehicles were identical, the custom software version (written in C) more easily implemented the hybridization. This custom interface between the back propagation network and the expert system provided a freer exchange of information.

We will present only the version using the commercial programs in detail in this paper. For more information on the custom software hybrid program see Gotts and Bremner (1995). Each of these simulated vehicles, regardless of how it was programmed, had to solve two problems. The simpler problem was for Vehicle 5n gecko to

face east when the sun was in the west. For a poikilotherm, turning away from the warmth of the sun is a counterintuitive response; it was nicely mastered by Vehicle 5n gecko. The more complex problem was for Vehicle 5n bluejay to discriminate a monarch butterfly from a viceroy butterfly. In the real world, monarchs are poisonous to bluejays but viceroys are not. The vehicle was to approach the viceroy and avoid the monarch. We chose this task because discriminating the wing pattern of the two butterflies was a difficult challenge for the neural network.

The neural network used as a transducer in all cases was a three-layer network of the back propagation type. Optical patterns of the sun position or butterfly wing patterns were mapped onto the matrix of transducer elements. The task of the neural network was to classify these patterns. The patterns were displayed on an 8×20 matrix of transducer elements (cones) that was connected to 10 hidden neurons, giving the network 1,600 input synapses (see Figure 1). The hidden layer was connected to 4 output neurons (sun west, sun overhead, dark, sun east) for Vehicle 5n gecko. The 4 output neurons allowed us to use fuzzy sets to discriminate the sun position. The back propagation network assigned a membership value to each sun position in each of the four fuzzy output sets. For example, if it was noon, the output pattern might be sun east .3, sun west .4, sun over head .8, and dark .01. There were only 2 output neurons for Vehicle 5n bluejay (monarch, viceroy); otherwise it was the same as gecko. In this version of the model, some patterns were not discriminated (monarch .5 and viceroy .459) which caused the expert system to avoid the monarch (if monarch < .6 & viceroy > .45, then avoid).

The expert system was of the following form: (1) if information from the neural network answers question positive, then act; (2) if information from the neural network is not positive, then look again. Node 2, shown in the lower half of Figure 1, contained two procedural knowledge statements. The first statement was "If the membership value of the sun position in fuzzy set west was larger than .95, then call node goEAST." If the sun position was less than .94, Node 2 called node "look," initiating a look again loop. In addition to the information from the sun transducer, there were knowledge bases for potential predator attack and potential food source. The internal model was a series of questions posed by the expert system which compared sun position with potential attack and food availability. This represents procedural knowledge as opposed to a "lookup table" approach (Negota, 1985).

RESULTS AND DISCUSSION

Figure 2 shows the output of the neural network. Notice the fuzzy set membership values assigned by each of the four output neurons. Remember that the total membership value of an element belonging to several fuzzy sets does not have to equal one (Zadeh, 1965).

The hybrid Vehicle 5n successfully solved both problems presented to it and demonstrated the efficacy of a duplex nervous system, part of which is dynamic and part of which is more rigidly fixed. The sensory memory of the back propagation network, with its 160-cone transducer and its 10 hidden neurons, had 1,600 connections to store hundreds of sun angles or butterfly wing patterns. This allowed for accurate visual discrimination. Although the system built from the commercially available programs (*Brainmaker*, *MicroSaint*, *Windows*) worked well, the interface among programs was somewhat cumbersome, with the operator having to move some files between programs. The custom program writ-

Figure 1. Flow chart of the hybrid neural network/expert system. Top half shows the connections to the 10 hidden neurons (only 3 of which are illustrated) of the back propagation network. Bottom half shows the arrangement of the three parallel subroutines of the expert system.

ten in the C language did its own file transfer and interacted with the operator only to establish the scenario.

Building hybrid neural network-expert system models with custom software has a major advantage over existing commercial programs: It is possible to use information from the expert system to retrain the neural network component, creating a dynamic, interactive, self-learning hybrid system. Nevertheless, there are advantages to using commercial programs. One is the shorter time needed for building the desired simulation; the other is the reliability of commercial programs that have been in the field for several years. (There are commercial expert system programs that have a neural network interface

EXSYS, Version 4.0, 1994). If the power of the neural network-expert system hybrid built with commercial programs is great enough, some manual-operator-initiated file transfer is worthwhile.

The goal of this paper has been to demonstrate the power of combining two fundamentally different strategies in problem solving—namely, neural networks and expert systems—into a hybrid. This approach allows the cognitive scientist to construct the expert system part of the simulation on the basis of human expert opinion, but it does not prevent modeling the part for which no human expert exists, because a neural network learns to be its own expert.

Figure 2. A representation of the 8×20 cone matrix containing one input pattern and the fuzzy set classification of that pattern.

REFERENCES

- BRAINMAKER PROFESSIONAL VERSION 2.0 [Computer program] (1990). Grass Valley, CA: California Scientific Software.
- BRAITENBERG, V. (1984). *Vehicles: Experiments in synthetic psychology*. Cambridge, MA: MIT Press.
- BREMNER, F. J., & GOTTS, S. [J.] (1993). Comparing two methods of visual abstraction to human data. *Behavior Research Methods, Instruments, & Computers*, **25**, 212-214.
- BREMNER, F. J., GOTTS, S. J., & DENHAM, D. L. (1994). Hinton diagrams: Viewing connection strengths in neural networks. *Behavior Research Methods, Instruments, & Computers*, **26**, 215-218.
- COMPUADD WINDOWS VERSION 3.0 [Computer program] (1990). Austin, TX: CompuAdd.
- EXSYS PROFESSIONAL, VERSION 4.0 [Computer program] (1994). Albuquerque, NM: EXSYS.
- FISHMAN, M. B., & BARR, D. S. (1991, August). A hybrid system for market timing. *Technical Analysis of Stock & Commodities*, **9**(8), 26-33.
- GOTTS, S. J. (1994). *Neural network models of relative depth perception employing stereoscopic disparity*. Unpublished honor's thesis, Trinity University, San Antonio, TX.
- GOTTS, S. J., & BREMNER, F. J. (1995). Neural network-expert system models of relative-depth perception. *Behavior Research Methods, Instruments, & Computers*, **27**, 173-177.
- GOULD, J. L. (1982). *Ethology: The mechanisms and evolution of behavior*. New York: W. W. Norton.
- HINTON, G. E., & ANDERSON, J. A. (Eds.). (1981). *Parallel models of associative memory*. Hillsdale, NJ: Erlbaum.
- LAWRENCE, J. (1993). *Neural networks: Design, theory, and applications*. Nevada City, CA: California Scientific Software.
- MICROSAINT [Computer program] (1987). Boulder, CO: Micro Analysis & Design, Inc.
- MICROSOFT C/C++ VERSION 7.0 [Computer program]. (1991). Redmond, WA: Microsoft Corporation.
- NEGOITA, C. V. (1985). *Expert systems and fuzzy systems*. Menlo Park, CA: Benjamin/Cummings Publishing.
- ZADEH, L. A. (1965). Fuzzy sets. *Information & Control*, **8**, 338-353.

(Manuscript received November 18, 1994;
revision accepted for publication January 20, 1995.)