

A hybrid model using neural networks and ACT-R

J. GREGORY TRAFTON

NTI, Inc., Dayton, Ohio

Brooks Air Force Base, San Antonio, Texas

How do people make decisions given contradictory information? This paper presents a model of how expert DSOs (defensive system operators) on a B1 bomber examine a complex series of signals, categorize whether those signals are dangerous or not, and then make a decision on the basis of those signals. This decision is made more difficult because an automatic on-board computer sometimes identifies the signal incorrectly. Therefore, the DSO must compare the actual signal to the system ID "guess." The proposed model is a hybrid model, combining a standard neural network and ACT-R, a production system, which achieves a high degree of success.

How do people make decisions, given contradictory information? At the least, they must determine which cues are valid, decide what to do with the contradictory information, and then integrate the information they have into some coherent whole. This paper proposes a computational model of how DSOs (defensive system operators) make complex decisions in a limited amount of time, given contradictory cues. The proposed model represents only a portion of the DSO's overall task.

A DSO is one of a four-man crew on a B1 Bomber. The DSO is required to examine a radar display and determine which signals on the radar screen are real threats and which are nonthreats. For each threat, the DSO must take a particular defensive action (e.g., avoidance). Each signal appearing on the radar display can vary within a given range; this makes it more difficult to uniquely determine its identification. In addition, the task of the DSO is further complicated by the immense number of signals that are very similar.

The DSO's job is facilitated by an on-board computer, the ALQ-161. This computer system analyzes each signal and displays information on the DSO console representing the computer match for that signal. Because of the large number of signals and the great variability among the signals, the ALQ-161 may display information that does not correctly correspond to the radar signal.

Therefore, the DSO's responsibility is to examine each signal, verify that the computer information is correct, and take the appropriate defensive action. If the system has identified the signal incorrectly, the operator must recall from memory which threat corresponds to the signal,

correct the misidentified signal information, and finally take appropriate action. The ALQ-161 needs the correct signal identification in order to deploy an effective defense. This process has several critical phases: The DSO must (1) recognize the signal, (2) compare the actual signal with the system's identification of the signal, (3) change the system identification if necessary, (4) categorize the signal as dangerous or not dangerous, and then (5) initiate appropriate defense if necessary. These tasks must be performed rapidly: the plane is flying a predetermined route that demands arrival over specific geographic areas, or way points, at defined times. Also, the threats are often airborne and move to intercept the bomber.

The primary goal of the present research is to build a cognitively realistic computational model of how DSOs categorize and respond to signals.¹ Building a computational model of how a DSO performs his job was found to be difficult, owing to the fact that most computational systems are designed for laboratory-oriented studies that are typically much less complex than the DSO task. The DSO task utilizes a wide variety of skills; good computational models exist for each subskill, but no single model incorporates all the skills needed for this task. Therefore, finding a single computational system to model the DSO task was quite difficult.

The two prevalent cognitive architectures in use today, neural networks and production systems, were considered during the planning stages of the computational model. Both neural networks and production systems exhibit serious shortcomings in modeling a complex task of this type. General-purpose neural networks (e.g., McClelland, Rumelhart, & the PDP Research Group, 1986; Rumelhart, McClelland, & the PDP Research Group, 1986) are excellent at very quick pattern recognition and at differentiating between signals. However, a crucial deficiency in using a neural network when modeling this type of complex task is the inability to perform both complex decision-making and problem-solving tasks. For example, a neural network would be excellent at determining which threat corresponded to a particular signal, but it would be quite poor at deciding what defense to per-

I am grateful to Douglas Eddy and Jerry Chubb for discussions on the DSO task. Paula Raymond, Rebecca Cardenas, and Douglas Eddy provided helpful comments on an earlier draft of this article. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Air Force. Correspondence and requests for reprints should be sent to J. G. Trafton, Naval Research Laboratory, Code 5513, 4555 Overlook Avenue S.W., Washington, DC 20375-5337 (e-mail: trafton@itd.nrl.navy.mil).

form on a given threat. In addition, one of the goals of this research is to improve DSO performance. It is quite difficult to interpret how a neural network performs a task and the representation that it has constructed (Churchland, 1990). Because of these two limitations, to improve a neural network's performance and then analogize to improve a DSO's performance would be virtually impossible. Because the DSO task requires a great deal of decision making and problem solving, and because the final model must be easily "inspectable," a pure neural network model was judged to be inappropriate for the DSO task.

Production systems (see, e.g., Anderson, 1993b; Newell, 1990), on the other hand, are very good at decision making and at solving problems and are easily inspectable. However, consideration of the use of a production system to model the DSO task revealed several deficiencies. First, pattern matching is, computationally, the most expensive part of executing a production (Anderson, 1993a). Pattern matching refers to the process of determining whether a production's conditions (i.e., the left-hand side of a production) match the contents of working memory. Given the fact that this task requires a great deal of pattern matching, and since this model should be fast as well as realistic, this was considered a shortcoming.

Second, a very large number of rules would be needed by a production system to achieve reasonable performance for deciding whether a signal corresponded to a particular threat. For example, different productions would be necessary for signals that had very similar characteristics. The large number of rules required for even simple systems is a very real problem for rule-based systems (see, e.g., Guha & Lenat, 1990; Lenat, Prakash, & Sheperd, 1986). Finally, a good model of how DSOs take the computer system's information into account does not currently exist. It is clear that DSOs do not ignore the ALQ-161, but it is not entirely clear how they integrate this possibly contradictory information with the signal information. Without our knowing how DSOs use the ALQ-161 information, a production system model of the DSO would be seriously deficient.

It should be noted that the main shortcomings mentioned above are primarily technical, not cognitive. For example, if a "pure" production system were built to model the DSO task, it would take an enormous amount of time to input all the rules necessary to achieve adequate performance, though it is possible (and perhaps even cognitively realistic). Therefore, because of the technical difficulties of each type of system, a hybrid system was constructed that utilized the most powerful features of each model. As mentioned before, this task requires complex pattern recognition, at which a neural network is very good, and complex decision making and problem solving, at which a production system is very good. Therefore, the hybrid system used a neural network to determine the identity of the signal (pattern recognition) and a production system to make decisions based on the

identity of the signal (problem solving and decision making).

The building of a hybrid system allows the deficiencies of each system to be resolved by the other, at least for this task. For example, the neural network does not have to perform complex problem solving and decision making, because the production system performs that aspect of the task. In addition, while a neural network is not very "inspectable," a production system is: It may be possible to examine the productions used by DSOs and construct training materials that would facilitate future DSO performance. On the other hand, the neural network, by identifying the signal, greatly reduces the number of productions and the amount of pattern matching for the production system. Finally, the neural network incorporates the information from the ALQ-161 into the model. The neural network uses the ALQ-161 information statistically to learn that the ALQ-161 is not always correct. Therefore, the network weighs the ALQ-161 information accordingly. It should be noted, however, that this "statistical" method is not necessarily cognitively realistic.

METHOD

Apparatus

Both the neural network and the production system were written in Common LISP (Steele, 1990). The hybrid system will run on any system that executes Common LISP.

Procedure

Neural network. The neural network code was adapted from Fahlman's quickprop code (Fahlman, 1988). The neural network contained three layers: an input layer, a hidden layer, and an output layer. The input layer contained three units, one for the signal of the threat, one containing some spatial location information, and one with the ALQ-161 "best guess" information. The hidden layer contained five units, all interconnected to both the input layer and the output layer. Finally, the output layer contained an output unit for each of the 12 possible threats.² The upper portion of Figure 1 shows a schematic of the simple neural network.

Initially, 12 "idealized signals" were constructed by taking the nominal signal for each of the 12 threats, the appropriate spatial location information, and the correct system "guess."³ Noise was then added to these 12 basic signals to construct a total of 400 total input-output vectors. Specifically, noise was added to the signal to simulate possible ranges of the signal. In addition, the system "guess" was incorrect approximately 25% of the time. No changes were made to the spatial location information since the spatial location information is always correct and always available to the DSO. Half of these signals were then used to train the network; the other half were used to test the network after successful training.

Production system. ACT-R (Anderson, 1993b) was used as the production system. ACT-R is a model of cognitive skill that makes a fundamental distinction between declarative knowledge (knowledge of facts) and procedural knowledge (knowledge of how to do something). Declarative knowledge is represented as an associative memory network (i.e., Anderson & Bower, 1973), while procedural knowledge is represented as production rules.

Hybrid system. The neural network takes an input and then outputs an activation level for each possible threat. In order to take the outputs from the neural network and apply them to ACT-R, a single simple function was created. This function was the only code

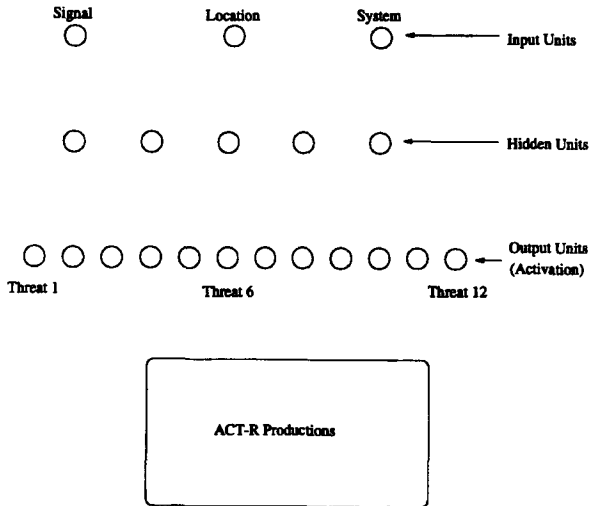


Figure 1. Schematic model of the hybrid system. The top half of the figure shows the neural network (each circle is a unit), and the bottom box represents the production system. The input units correspond to what the DSO sees, and the output units represent levels of activation for each threat. The production system decides what action to take, depending on the activation of the output units.

in the simulation that was not directly part of either the neural network or ACT-R. The unit that has the highest level of activation corresponds to the threat that the inputs most closely match. This level of activation then becomes part of the associative memory network of ACT-R. Which productions fire depend on which threat has the highest level of activation. If two or more threats have the same activation level, a random decision is made. Figure 1 shows a graphical description of the hybrid system.

For example, assume that a particular series of inputs is fed into the neural network. The neural network outputs a signal between 0.0 and 1.0 for each of the 12 threats. This output level becomes the activation level for each threat. A series of productions then fire, depending on which threat has the highest level of activation. For example, if the threat that has the highest level of activation is a true danger, the production system may decide that the plane should change course and avoid the threat.

RESULTS AND DISCUSSION

The neural network successfully learned the 200 input-output training vectors with error $\leq .10$, suggesting that the neural network was able to create a consistent representation for the training set.

Overall, the hybrid system was quite successful at recognizing the inputs and choosing an appropriate defense against the threat. When tested against the original idealized signals, the model chose a successful defense 100% of the time. When tested against the half of the data that the neural network had not been trained on, it was successful 98% of the time. The success of the network on data that it had never seen before suggests that the network correctly abstracted the necessary features from the training data.

In addition, pilot data suggest that subjects are confused by some of the same signals that the system gets wrong. This result is consistent with past research if the

neural network is considered a simple associative memory network; similar signals are likely to be confused because of their similarity. Because the neural network becomes part of the declarative memory structure of ACT-R, the finding that human subjects and the hybrid system make similar errors is not entirely surprising, given the vast amount of research on associative memory (Anderson, 1983; Anderson & Bower, 1973).

One of the most interesting questions that this research addresses is "How can a neural network and a production system be combined?" This is an important question, both from a pragmatic and cognitive viewpoint. In the present research, this problem has a very simple pragmatic solution: both programs were written in LISP. Therefore, it was a very simple matter for the neural network to feed information into the production system.

The cognitive viewpoint is more difficult and controversial; a great deal of discussion has gone into determining whether rules are a basic part of human cognition (Churchland, 1990; Harnad, 1990; Rumelhart & McClelland, 1986) or whether productions are instantiated through neural-network-like structures (Lébière & Anderson, 1993). For this hybrid system, it is assumed that declarative memory is implemented in terms of a neural-network-like structure and that productions are instantiated via rule-like structures.⁴ This assumption, however, makes it more difficult for the neural network to communicate to the production system, because they differ in terms of their accessible representations.

Some researchers (e.g., Bremner & Gotts, 1995; Gotts & Bremner, 1995) use a "black box" approach for which they transform the output of a neural network into a symbolic representation that an expert system can use. The problem with this approach is that because the transformation program is a black box, it is not cognitively realistic and the black box is not inspectable. In the present research, it is very clear what the relationship between the neural network and the production system is: the neural network is simply a more elaborate form of declarative memory than is typically used in ACT-R.

REFERENCES

- ANDERSON, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.
- ANDERSON, J. R. (1993a). Performance. In J. R. Anderson (Ed.), *Rules of the mind* (pp. 45-67). Hillsdale, NJ: Erlbaum.
- ANDERSON, J. R. (1993b). *Rules of the mind*. Hillsdale, NJ: Erlbaum.
- ANDERSON, J. R., & BOWER, G. H. (1973). *Human associative memory*. Hillsdale, NJ: Erlbaum.
- BREMNER, F. J., & GOTTS, S. J. (1995). Shades of Braitenberg's Vehicle 5. *Behavior Research Methods, Instruments, & Computers*, 27, 169-172.
- CHURCHLAND, P. M. (1990). Cognitive activity in artificial neural networks. In D. N. Osherson & E. E. Smith (Eds.), *An invitation to cognitive science: Vol. 3. Thinking* (pp. 199-227). Cambridge, MA: MIT Press.
- FAHLMAN, S. E. (1988). Faster-learning variations on back-propagation. In D. S. Touretzky, G. E. Hinton, & T. J. Sejnowski (Eds.), *Proceedings of the 1988 Connectionist Models Summer School* (pp. 38-51). Los Altos, CA: Morgan-Kaufmann.
- GOTTS, S. J., & BREMNER, F. J. (1995). Neural network-expert system models of relative-depth perception. *Behavior Research Methods, Instruments, & Computers*, 27, 173-177.

- GUHA, R. V., & LENAT, D. B. (1990). *Building large knowledge-based systems: Representation and inference in the Cyc project*. Reading, MA: Addison-Wesley.
- HARNAD, S. (1990). Symbols and nets: Cooperation vs. competition. *Connection Science*, 2, 257-260.
- LEBIÈRE, C., & ANDERSON, J. R. (1993). A connectionist implementation of the ACT-R production system. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society* (pp. 635-640). Hillsdale, NJ: Erlbaum.
- LENAT, D., PRAKASH, M., & SHEPERD, M. (1986). CYC: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *Artificial Intelligence Magazine*, 6(4), 65-85.
- MCCLELLAND, J. L., RUMELHART, D. E., & THE PDP RESEARCH GROUP (1986). *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 2). Cambridge, MA: MIT Press.
- NEWELL, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.
- RUMELHART, D. E., & MCCLELLAND, J. L. (1986). On learning the past tenses of English verbs. In J. L. McClelland, D. E. Rumelhart, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 2, pp. 216-271). Cambridge, MA: MIT Press.
- RUMELHART, D. E., MCCLELLAND, J. L., & THE PDP RESEARCH GROUP (1986). *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 1). Cambridge, MA: MIT Press.
- STEELE, JR., G. L. (1990). *Common LISP: The language* (2nd ed.). Bedford, MA: Digital Press.

NOTES

1. A DSO's job is much more complicated than described above, but this initial model was built to model the task described above.
2. Please note that there are many more than 12 threats in the actual DSO task. However, for simplicity's sake the number of threats has been reduced to 12.
3. The actual signals seen by the DSO are confidential. In order to avoid revealing confidential data, simple analogues of the actual data were created as training data. These analogues were created by transforming the numeric information and converting the spatial information into numbers.
4. Of course, because productions are implemented in the human brain, at some level they also must be implemented in neural-like structures. See Anderson (1993b) for a fuller discussion of this issue.

(Manuscript received November 18, 1994;
revision accepted for publication January 31, 1995.)