REVISTA MEXICANA DE ANÁLISIS DE LA CONDUCTA 26, 141-158 NÚMERO 2 (SEPTIEMBRE)
MEXICAN JOURNAL OF BEHAVIOR ANALYSIS NUMBER 2 (SEPTEMBER)

BEHAVIOR ANALYSIS AND ARTIFICIAL NEURAL NETWORKS

ANÁLISIS CONDUCTUAL Y REDES NEURALES ARTIFICIALES

BILL POTTER AND MICHAEL WILSON[1]
CALIFORNIA STATE UNIVERSITY, STANISLAUS

ABSTRACT

This article provides an overview of Artificial Neural Networks (ANNs), including architecture and several learning algorithms. Reviewed algorithms include Hebbian, Rescorla-Wagner, Sutton-Barto (1981), and Hutchison's. In addition, the contributions Behavior Analysts have made to ANN development, as well as the effect ANN's have made on behavior analysis, is discussed. Finally, several applications of ANNs by Behavioral Analysts are reviewed.

Key words: behavior analysis, artificial neural networks

RESUMEN

Este artículo provee un vistazo a las redes neurales artificiales (ANNs), incluyendo su arquitectura y varios algoritmos de aprendizaje. Los algoritmos revisados incluyen el Hebbiano, el de Rescorla-Wagner, el de Sutton-Barto (1981) y el de Hutchison. Además, se discuten las contribuciones que los analistas conductuales han hecho al desarrollo de las ANNs, así como el efecto que éstas han tenido sobre los analistas conductuales. Finalmente, se revisan varias aplicaciones de las ANNs por parte de los analistas conductuales.

Palabras clave: análisis conductual, redes neurales artificiales

[1] The authors gratefully acknowledge the input of Jack Michael, Bill Hutchison and José E. Burgos in shaping and refining this article. Bill Potter may be contacted via email at wpotter@toto.csustan.edu, and Mike Wilson at carbon@prodigy.net. Either author can be contacted by regular mail at: Department of Psychology, California State University Stanislaus, 801 W. Monte Vista, Turlock, CA 95382, USA.

Artificial Neural Networks (ANNs), in general, are computer programs that learn. An ANN program consists of a network of interconnected units or nodes, each node a simple processing element. These simple elements are the artificial neurons of the ANN and loosely follow the principles that neuroscientists have learned about the nervous system.

ANNs resemble the brain in that the network learns, and that interneuronal connection strengths (numerical weights in the ANN version) is the basis of that learning (Haykin, 1994). Some Behavior Analysts have proposed that, in addition to modeling the neuronal structure of the brain, ANNs should adhere to molar characteristics as well. For example, Donahoe, Burgos, and Palmer (1993) suggest that ANNs should incorporate collections of nodes that project to each other, as the brain has projections from the ventral tegmental area to the motor association areas, to name one of many such projections (see also Donahoe & Palmer, 1994). Probably a more fitting name for such arrangements is "adaptive system" in that it is composed of more than just a neural net. This paper will only address simpler models, ones that do not incorporate such collections of nodes with projections and accordingly we will refer to these systems as ANNs although "adaptive system" is just as apt a name.

ANNs can be distinguished from early investigations into so-called "Artificial Intelligence". In general, this approach advocated the encoding of rules which the computer would follow given different inputs, presumably simulating the skills of an expert. These systems were usually designed to aid users in making decisions and generally followed a pattern of building up a knowledge base and coupling this with "if-then" rules (Hutchison, 1984; Stephens & Hutchison, 1993; Winograd, 1990). These systems were heavily influenced by cognitive science's penchant for claiming that "production rules" and "schemas" account for much of human behavior (Anderson, 1980). Although some successful applications resulted from this approach, a number of researchers have pointed out that this methodology is confined and is unlikely to ever completely model human behavior as we know it (Bechtel & Abrahamsen, 1991; Schlinger, 1992; Stephens & Hutchison, 1993).

ANNs have been used for a variety of purposes and applications. They are being successfully applied across such diverse areas as finance, medicine, engineering, geology and physics (Statsoft, 1999). ANNs are used in investment analysis, signature analysis, process control, engine monitoring (aircraft and train), and in marketing (Smith, 1998). ANNs are particulary useful in situations in which a high degree of flexibility is needed to deal with significant variations in the environment (Zomaya, 1994). That is, the specific operating parameters of the situation are unknown in advance (as well as unstable) and must be learned (and updated) from the environment, a task traditional computer programs cannot handle. ANNs, however, are capable of such tasks (behaviors), similar to an organism in its environment. Obviously, Behavior Analysis has much to contribute to how such systems interact with their environment. It is this, along with the fact that ANNs can function as a model of behavioral principles that they are of importance and utility to Behavior Analysts.

As mathematical models of real neural systems, ANNs can be employed as an interpretive tool (Donahoe & Palmer, 1989). This allows for theoretical work to be carried out on behavioral principles by means of computer simulation (Donahoe et al., 1993). Additionally, Behavior Analysis could benefit from the recognition of being instrumental in the ongoing development of ANNs and in contributing to the general understanding of how simple processes come to produce complex behaviors. In this paper we will provide an overview of ANNs, and discuss some of the reasons why Behavior Analysts have become interested in them.

The Nature of Artificial Neural Networks

First, let us look at ANNs from a behavioral/organismic perspective. First, we would be interested in the nature of the organism we are dealing with – does it have good hearing (thus we might use auditiory stimuli effectively) and color perception? Does the organism have a keen sense of smell? Researchers typically examine these issues prior to working with an organism to determine the sensory capacities of the organism. When using a computer, we are talking about inputs, such as keyboard input or input from a mouse (the computer is designed to detect mouse clicks and key presses). Small video cameras are becoming more prevalent, which allow visual signals to be input into the computer, as well as sounds via the use of a microphone. Therefore, computers are able to sense the world to the extent that we design instruments that detect environmental changes and create the appropriate interface for the computer. Here is a short list, other than those already mentioned, of types of inputs that can be arranged for a computer:

- Temperature and changes in temperature
- Electromagnetic fluctuations
- Touch, including hardness, texture, etc.
- Position (e.g. vestibular)
- Kinesthetic (position of limbs, etc.)
- Acceleration/deceleration
- Movement sensors
- Edge detectors
- Distance meters

Although this is not a comprehensive list, one can see that a computer can be fitted with a wide array of input/sensory devices.

In examining an organism, we are also interested in what type of responses such an organism can emit. It is clearly understood that pigeons will more readily learn a keypeck, versus a treadle press. Humans will never learn to swing from trees by their tails, for obvious reasons. Typical computer outputs include images on a screen, sounds from speakers, characters on a printer and packages of data sent over the internet. However, while these are standard outputs (which we can think of as responses), there are innumerable types of outputs a computer can have, limited only by the technology available. Here is a list of potential outputs:

- Motors that turn in horizontal and vertical directions, for specified amounts of time. These can do such things as lift, open and close a mechanical hand or leg; turn, lower and raise an input camera; provide for various forms of locomotion for the computer.
- Open and close any other type of circuit, thus controlling many types of electrical/mechanical instruments.
- Output to a speaker, thus allowing the computer to produce speech-like sounds, as well as other audio outputs.

Although researchers seldom examine the neurophysiological system of an organism they would like to work with, this is the essence of ANNs and will be discussed here in some detail. As noted earlier, this system is loosely modeled after what we know about brain neurophysiology. Namely, artificial neurons function in a manner similar to biological neurons – they receive inputs and if activated, emit an output. When implemented in a computer program, these neuronal-like objects are called nodes, and a single ANN may consist of only a few nodes or a million or more. ANNs are often represented by diagrams such as Figure 1. This diagram illustrates the relationships between inputs (represented by the "I" nodes, and outputs (represented by the "O" nodes, with nodes sometimes being placed between the two, which are analogous to interneurons in an organism, and constitute what are often called hidden layers by ANN researchers (Wasserman, 1989). The basic system is an attempt to roughly model the nervous system of an organism. That is, the inputs are roughly analogous to the sensory inputs of an organism, which on a computer may be visual input (a video camera), sound input (audio files or live audio via a microphone), while the outputs may be a mechanical arm movement, or a sound from a speaker as discussed earlier. For our purposes, the inputs and outputs used in these networks are arbitrarily chosen (i.e. a "1" or an "A"), as is often the case in the experimental analysis of behavior (see Skinner, 1969 for a complete analysis).

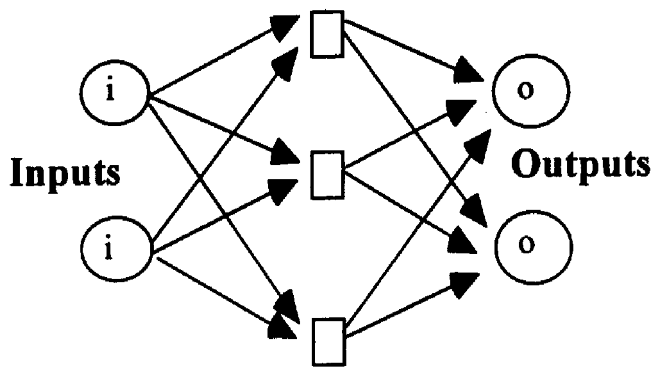

Figure 1. Simple illustration of an ANN. The "i" refers to inputs, the "o" to outputs. All layers actually receive inputs and generate outputs, so these labels are for convenience only and to indicate the direction of processing. See text for details.

Consulting Figure 1, it can be seen that often all nodes in a ANN are connected to all other nodes, in a "feed-forward" manner, that is, they do not have lateral connections, or reverse or self-stimulating connections, although some researchers do incorporate these designs, but they will not be discussed here. One can think of the connections as presynaptic (feeding into a node) and postsynaptic connections (an output from a node and functioning as an input – presynaptic – for a trailing node). The inputs and outputs, keeping with math as the language of computers, is often just a number. For example, if an "A" is presented to an input node, and that node is activated by such a visual stimulus, then the node may send a "1" along all of its connections. If that node is activated by only auditory stimuli, then it may send a "0" along its connections, or simply not be active. The connections between nodes (e.g. in Figure 1 between the inputs and the internodes) are often weighted. In fact, it is changes in these weights what generally defines learning in ANNs. The values of these weights are often set between 0 and 1, and influence how much of an impact an input has. For example, say that we present a bright visual stimulus to one of the input nodes in Figure 1. The node interprets (according to the visual sensor we have supplied the computer with) the input as a value of 2, that is, a relatively bright light. This value will be sent along all of the connections that node has, and each of those is weighted. If the first has a weight of 1 then the entire 2 value will be sent to the internode along that connection ($2*1=2$). If the second has a weight of 0.5, then the node at the end of that connection will only receive a value of 1 ($2*0.5=1$) (Caudill & Butler, 1993; Wasserman, 1989). A threshold value is often incorporated within a node, which is simply a criterion level for input values to meet before the node emits a response. Thus if a node has a threshold value of 3 and receives a total input value from all the nodes feeding into it of only 2.95, then that node will not "fire", that is send any values along to its connections. The network models discussed in this paper utilize an "all-or-none" output pattern, thus a node generates an output or it does not (thus 1 or 0). Some researchers have created networks that utilize continuous output, although they will not be discussed here (Wasserman, 1989). It is also possible to have negative (or inhibitory) connection weights and input values, but these will not be discussed in this introductory paper.

Notice that "knowledge" or responses are not hardwired or stored in a particular place in an adaptive system, but are distributed as a pattern of weights across nodes (Caudill, 1987). Feedback introduced back into the network allows the network to "learn" (for connection weights to change) from its responses. A number of techniques have been used which change the connection weight between inputs and nodes; several are discussed below.

Learning Algorithms

The Hebbian Model

Many current adaptive systems incorporate all or part of Hebb's (cited in Wasserman, 1989) general rule that if the source and destination neuron (or node in the ANN case) are both activated on any given trial, then the pathways in that connection are strengthened, or in the case of ANNs, the connection weights are increased (Wasserman, 1989). An equation for the Hebbian model follows:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha \cdot [o_i(t) \cdot o_j(t)]$$

Where $w_{ij}(t+1)$ refers to the weight of the connection between any two nodes "i" and "j" at time-step $t+1$ (that is, the new weight at the moment immediately after learning occurs), and $w_{ij}(t)$ refers to the weight of the connection between i and j at t (the current trial). The 'α' is a learn-rate coefficient (a rate at which modification of the weights occurs, usually set between 0 and 1), while $o_i(t)$ and $o_j(t)$ refer to the outputs of i and j, respectively, which are generally valued at 1 (an output occurred) or 0 (no output occurred). Given the above equation, the weight of a particular connection will change (i.e., be increased by the amount of the learning rate, e.g., 0.1) if an output is generated by both i and j. If an output is generated by i but not by j, then $o_i(t) \cdot o_j(t)$ will be zero and no weight change will occur (Caudill & Butler, 1993; Wasserman 1989).

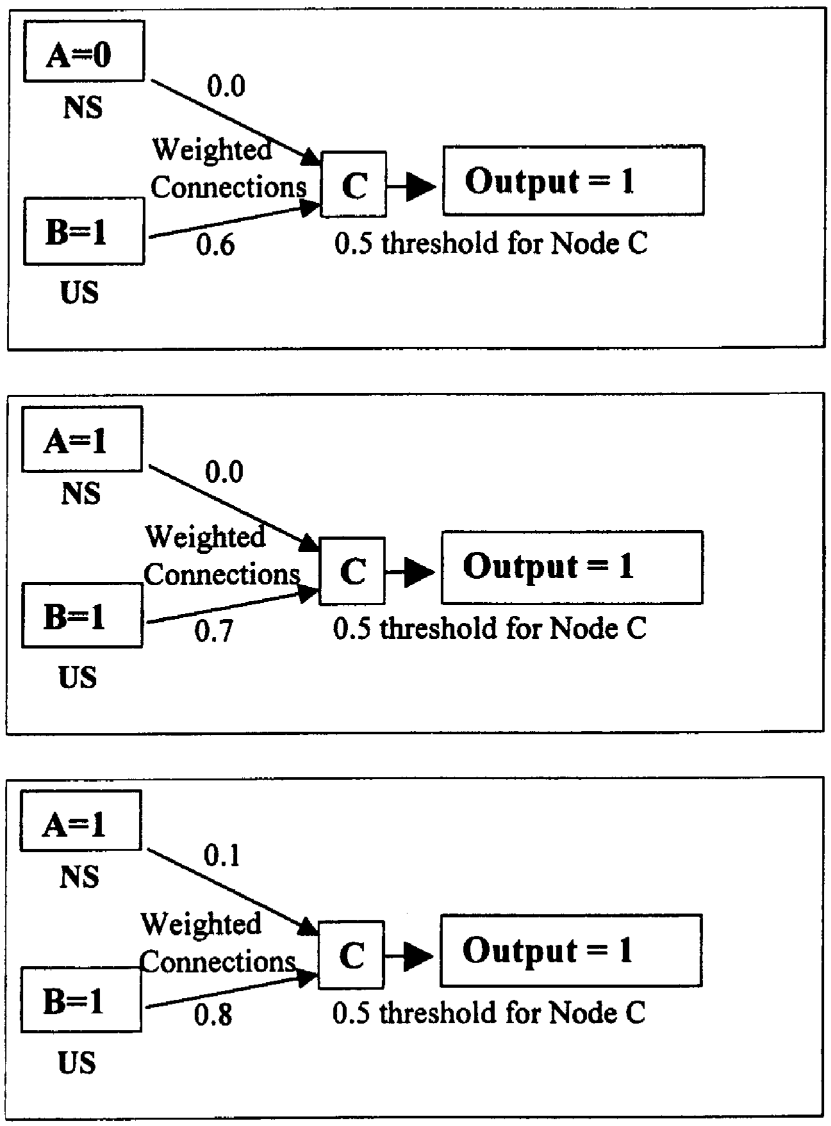

Figure 2. An illustration of the Hebbian learning algorithm. Each panel represents a trial. In the upper panel (first trial), Node A is not active, meaning that the neutral stimulus (NS) is not present. The equation for each trial is: $w_{ij}(t+1) = w_{ij}(t) + \alpha \cdot [o_i(t) \cdot o_j(t)]$, where $\alpha = 0.1$ in this case. Eventually, A's weight will reach 0.5, surpassing the threshold value for C and triggering the response. At this point we could label the NS a conditioned stimulus (CS). See text for details.

The Hebbian model has been espoused as a very simple model of respondent learning (Caudill & Butler, 1993; Sutton & Barto, 1981). Take the example of two nodes A and B (input nodes) that are connected to a third node C. The connection weight between B and C is of sufficient magnitude to cause C to produce an output when B is activated. However, the connection weight between A and C is too small for A alone to produce an output in C. Stated in behavioral terms, analogously this reads: stimulus B elicits a response in C, but A does not. We can refer to B as an unconditioned stimulus (US) and A as a neutral stimulus (NS). Given the learning algorithm stated above, if A and B are both activated, the fact that B will generate an output in C will cause the connection weight from A to C to increase (by the amount of the a). With repeated pairings, A, when activated alone, will come to generate an output in C (see Figure 2 for an illustration of this process). If A's activation was paired with another node's activation that did not activate C, that new node would do so after repeated pairings (i.e., which would simulate a form of second-order conditioning. (Sutton & Barto, 1981). As noted above, the Hebbian model is very simple. While demonstrating some of the basic elements of respondent conditioning, it fails to demonstrate others such as blocking and overshadowing, or even the necessary temporal relationship of stimuli (Caudill & Butler, 1993; Sutton & Barto, 1981). Researchers have attempted to rectify these shortcomings as noted below.

Sutton and Barto's Model

Sutton and Barto (1981) were familiar with research in nonhuman learning and applied this to the design of their adaptive systems. The authors incorporated both the Hebbian model and the Rescorla-Wagner (1972) model of respondent learning in their learning algorithm. In general, the Rescorla-Wagner model states that learning is a function of the difference in the amount that a conditioned stimulus (CS) elicits a conditioned response, and the amount that an US elicits an unconditioned response (UR) (Mazur, 1986). "Amount" refers to the magnitude of the response under consideration and may include a variety of dimensions. These researchers assume that a US elicits a UR of a specific and maximum or asymptotic magnitude, labeled "Aj". A general equation for their proposition is:

$$\Delta V_i = a_i \cdot (\lambda - V)$$

ΔV_i refers the amount of change in the power of a particular CS i to elicit the CR (this is simply a connection weight), a_i is a coefficient indicating the salience of the CS (normally set between 0 and 1), λ refers to the maximum (or asymptotic) magnitude of response elicitation by the US, and V is the sum of response strengths generated by all CS (or potential CSs) present on any given trial (cited in Mazur, 1986; Sutton & Barto, 1981). Three general rules can be generated from this model:

1. If λ - $V > 0$, then the CS will be strengthened;
2. If λ - $V < 0$, then the CS will be weakened;
3. If λ - $V = 0$ (i.e., the sum of all CSs elicit the same response magnitude as the US), then the CS will be not be changed. That is, no learning will take place (Mazur, 1986).

This model has been shown effective in predicting blocking, overshadowing and conditioned inhibition (Mazur, 1986), although it does have weaknesses, namely in predicting conditioning changes based on the temporal relationships of the CS and the US (Mazur, 1986; Sutton & Barto, 1981). These shortcomings are unimportant for our current analysis.

Sutton and Barto (1981) incorporated the Rescorla-Wagner model in their adaptive system learning algorithms. The addition of this model allows for increments and decrements in the weights assigned to a particular network connection, unlike the basic Hebbian model, which only allows for increasing connection weights. A simple version of Sutton and Barto's learning algorithm as compared to the Rescorla-Wagner equation appears below:

Rescorla & Wagner: $\Delta V_i = \alpha_i \cdot (\lambda - V)$

Sutton & Barto: $\Delta W_i(t) = c \cdot [Y(t) - \Sigma W_j X_j(t\text{-}1)] \cdot x_i(t\text{-}1)$

Where $\Delta W_i(t)$ refers to the change in the connection weight of i at t. The c is a learning-rate coefficient, similar to the α used in the Hebbian model and the Rescorla-Wagner model. The $Y(t)$ is the maximum value of the US output at t. The $\Sigma W_j X_j(t\text{-}1)$ refers to the sum of all the weighted connections for all inputs X_j and weights W_j at t-1. The last part, $x_i(t\text{-}1)$, is similar to the Hebbian model. This indicates whether or not x_i was activated (i.e., generated an output) at t-1. If an activation occurred, then $x_i(t\text{-}1) = 1$, allowing for the change in connection weight to be modified by the difference in the value of the output by the US and the sum of the outputs by all CSs present on any given trial. If an activation occurred, then $x_i(t\text{-}1) = 0$ and the total change in $W_i(t)$ will also be 0. That is, $[Y(t) - \Sigma W_j X_j(t\text{-}1)]$ will evaluate to 0.

Sutton and Barto created an ANN using this algorithm and found that this model performed similarly to the Rescorla-Wagner model, demonstrating blocking, overshadowing and conditioned inhibition (Sutton & Barto, 1981). These researchers went on to implement modifications to their model which overcome some of the temporal issues discussed above, although they will not be discussed here.

Hutchison & Stephens' General Model

Hutchison (1984, 1998a) and Stephens and Hutchison (1993) have described an adaptive system which simulates operant conditioning. For our purposes, a series of arbitrary stimuli can be first defined for use by the system, for example, "S1, S2, S3, S4". Then a series of responses is also defined, for example "R1, R2, R3, R4" (they can also be acquired responses, but for this example we will assume an existing repertoire). Each stimulus is connected to each response. These connections are weighted, that is they are assigned a value. However, instead of using the weight to calculate whether or not a threshold value is reached, the weight determines the strength of each response available to the system. When a stimulus is presented, the computer determines the value of the weights between stimuli and responses, then emits the response with the highest value. A clarifying analogy is the probability that a pigeon in an operant chamber will peck the lighted right key, versus emitting some other response such as pecking the left key or turning in a circle. In a single trial situation, the user, environment, or the computer if it is set up to do so, reinforces, punishes or provides no consequence for the response. This is accomplished by associating a response with a value, for example +1 as a mild reinforcer, -1 as a mild punisher, and 0 for no consequence. The absolute value of the positive or negative number determines the magnitude of the reinforcer or the punisher. A slight negative cost (e.g. -.2) is implemented upon each response emitted to account for response cost and the prolonged effect of no consequence (extinction). These consequence values are used to modify the weights between appropriate stimuli and responses. As in the Rescorla-Wagner model, an asymptotic value is used. This asymptotic value is the value of the consequence delivered on any given trial, or as shown later, the consequential value of response-produced stimuli. A simple equation for this type of algorithm is:

$$W_{ij}(t+1) = W_{ij}(t) + \alpha[Cv - W_{ij}(t)]$$

Where $W_{ij}(t+1)$ refers to the connection weight between an appropriate stimulus i and a response j at $t+1$. $W_{ij}(t)$ is the current weight before the consequence is delivered. As before, α refers to a learning-rate coefficient. Cv refers to the consequence value (e.g., +2). Consider this example: the computer is arranged in such a manner to recognize the stimulus S1 and can respond with responses R1 and R2. Initial response values (or connection weights) in the presence of S1 is 0.3 for R1 and 0.1 for R2. When S1 is presented, R1 will be emitted by the system, because it has the highest value (0.3) of all responses available to the system. The value of the consequence delivered to the system is +1. Given the equation above, the new value of R1 in the presence of S1 will be the result of the asymptotic value of the response (the value of the consequence, in this instance +1) minus the prior value of the response (0.3), multiplied by the learning rate (0.2 for this example). This will all be added to the previous value of the response (0.3). For clarity's sake we are ignoring the response cost described above. The calculations follow:

$W_{S1,R1}(t+1) = 0.3 + 0.2(1 - 0.3)$
$W_{S1,R1}(t+1) = 0.3 + 0.14$
$W_{S1,R1}(t+1) = 0.44.$

An interesting difference between this model and the Rescorla-Wagner model is that the asymptote value is not fixed from trial to trial, it is the value of the consequence, which may or may not remain constant, depending on the contingencies in effect. Also note that this equation does not specify which connections are strengthened, but as noted before, it is the connection between the stimuli presented and response emitted (or more behaviorally, the strength of R1 in the presence of S1). In Hutchison's model often there are many stimuli present on a trial, each contributing to the value of each response. It should also be noted that a negative or a zero value for a consequence will likely reduce the S1→ R1 connection, which will increase the likelihood that R2 will be emitted.

Hutchison (1984, 1998a) has combined elements of respondent and operant conditioning into his model. An example will illustrate this. Using $W_{S1,R1}(t+1) = W_i(t) + a[Cv - W_{S1,R1}(t)]$ from above, with $a = .2$, the following scenario depicts a two-link chain of responses shaped by backward chaining. For our purposes it is not necessary to discuss the shaping process. The process is picked up after responding is nearly stable, but not at asymptotic values.

An ANN is arranged to recognize S1 and to emit R1 and R2, and the network has been shaped to emit R2 after R1 in the presence of S1, at which point a consequence of +2 will be delivered. The values of R1 and R2 in the presence of S1 are 0.3 and 0.1, respectively. Since the value of R1 is largest, this response will be emitted. As it is emitted, the system is arranged in such a way as to also treat the occurrence of that response as a response-produced stimulus, which for convenience Hutchison calls "Did R1". In the presence of "Did R1" the value of R1 is 0.1 and the value of R2 is 0.6. Since the value of R2 is a higher value than R1, it will be emitted. Before it is emitted, however, we can calculate the new value of the relationship between S1 and R1. This is calculated by using the value of R2, which is 0.6. This value can be thought of as the conditioned reinforcing value of R2 (or the response-produced stimulation of R2) on R1. The calculation for this is:

$W_{S1,R1}(t+1) = 0.3 + 0.2(0.6-0.3)$
$W_{S1,R1}(t+1) = 0.3 + .06$

$W_{S1,R1}(t+1) = .36$

Notice that the asymptotic value is the value of R2 (0.6), which Hutchison refers to as the value of the new situation presented. Then, when R2 occurs a consequence value of +2 is delivered and the relationship between stimulus "Did R1" (R1's response-produced stimuli) and R2 is strengthened as follows (where S2 = Did R1):

$W_{S2,R2}(t+1) = 0.6 + 0.2(2\text{-}0.6)$

$W_{S2,R2}(t+1) = 0.6 + 0 .28$

$W_{S2,R2}(t+1) = 0.88$

If a new trial is started in which S1 is again presented, the process will occur again with these new stimulus-response values being used. It can be seen that the response-produced stimulus "Did R1" is functioning similarly to a conditioned reinforcer, and with repeated trials will ultimately arrive at a value equal to the value of the reinforcer, if the reinforcer remains constant. This is similar to the Rescorla-Wagner model in that the CS eventually comes to elicit a response of the same magnitude as does the US.

Although this is a simplified version of the Hutchison and Stephens' algorithm, it functions to show how simple operant and respondent conditioning can be implemented in an adaptive system. Hutchison (1984,1998a) notes that the above described adaptive system demonstrates many established behavioral phenomena such as stimulus control, chaining, generalization and discrimination. The current system Hutchison has developed incorporates real-time audio and visual inputs, and has response outputs which are grouped into five movement areas: lips, glottals, tongue, velum, and aspiration. These correspond to the major components of the human articulatory system. One response (or position) can occur from each category on each time step, and each of the five is arranged to occur in parallel. While this is much more complex system than described in this paper, the basic learning algorithm is similar.

Behavioral Analysis Contributions to Adaptive Systems

Behavior Analysis has as one of its focuses the problem of accounting for complex behavior by the learning of simpler behavior (Donahoe, 1991; Donahoe and Palmer, 1989, 1994; Hutchison, 1984). Such techniques as shaping, chaining, and fading are noted to be training techniques which may lead from simpler to more complex behaviors. Donahoe and Palmer (1989) report that little or no work has been conducted with networks utilizing shaping or training. They did report one experiment in which fading was used to train an ANN, resulting in a 75% reduction in trials needed as compared to normal training procedures. To date few researchers are using ANNs in behavioral research, although the number has increased (e.g., this issue; and Donahoe, 1998; Donahoe & Palmer, 1994; Donahoe, Palmer, & Burgos, 1997; Hutchison, 1998b).

Schlinger (1992) notes that the field of "Artificial Intelligence" has not been successful in modeling "intelligence" because the researchers have adopted an essentialist model of natural intelligence. He suggests that these failures might be overturned when a selectionist model is adopted. Palmer and Donahoe (1992) argue that the features of ANNs classify them as operating within the selectionist framework, although some connectionists deny any affiliation with radical behaviorism (Rummelhart & McClelland, 1986).

In a similar vein, Hutchison (1985) and Hutchison and Stephens (1987a) note that at least one researcher has divided "artificial intelligence" into two broad camps, structuralists and functionalists. Hutchison and Stephens believe this division roughly corresponds to the division between the behavioral and cognitive communities and conclude that "the case is compelling that behavior analysis provides a more natural model for adaptive systems than does cognitive science" (p 14). Kehoe (1989) in illustrating connectionist adaptive system architecture, draws the conclusion that the algorithms show that behavioral principles have been incorporated into their formulation. Sutton and Barto (1981) concur with this thinking stating "animal learning theory constitutes a large body of carefully explored and tested theory about fundamental processes of learning" (p 135). While the authors are not behaviorists, their work incorporates many basic behavioral principles. Finally, Bechtel and Abrahamsen (1991), both connectionists, have noted that a return to parsimony is being advocated within their field, and note that parsimony is a characteristic of behaviorism.

Contribution of ANNs to Behavior Analysis

The use of models in psychology is a long accepted practice. Models aid science in organizing and aiding in the interpretation of data, in addition to pointing to new research directions (Bush & Mosteller, 1955). The introduction of computer simulations into behavioral psychology prompted one behaviorist to call the computer application that embodies behavioral principles as "a theory of behavior" (Cook, 1994). Cook adds that computer simulations of behavior are useful in that they allow for multiple derivations to be carried out rapidly and at low cost and effort, and for access to covert behaviors. Cook warns, however, that unchecked "theorizing" may lead to disenchantment from the original purposes of the research. Donahoe (1991) also notes that networks allow for the extension of behavioral principles observed overtly to covert events, which he argues demonstrate that such principles are sufficient to explain certain behaviors.

Hutchison (1984) notes that ANNs can be an invaluable tool in ascertaining the training necessary for developing complex behaviors. On a more personal note, he claims that developing his ANN model forced him to consider theoretical and practical issues not normally raised in the course of discussing behavioral issues, in essence shaping his behavior toward a more molecular analysis of behavior (personal communication, July 31, 1994). He notes that this may have value by developing a more exact science of behavior.

Donahoe and Palmer (1989; 1994) argue that ANNs are a powerful tool for formal interpretation in the behavioral sciences. The authors define formal interpretation as a method of using principles derived from experimental analyses, versus mere speculation that may not be constrained by those principles. They note that new principles are not uncovered in formal interpretations, but that new aspects of existing principles may be revealed. Donahoe, Burgos and Palmer (1993) in concurrence with Hutchison, argue that complex behavior, built up from a prolonged history of more simple reinforcement processes, often precludes the use of experimental regimens. Given the assumption that complex behavior is built from simpler processes, the authors argue that an ANN is an efficient method of conducting such formal interpretations.

Shimp (1989) adds to these arguments, stating "A theory that behaves, that produces a stream of behavior, would seem in an intriguing way, to fit better with Skinner's criterion for a good theory than do many other sorts of behavioral theory" (p. 169). Shimp is referring to computer simulations of behavior and with his reference to "other sorts of behavioral theory" to mathematical models.

Some Behavior Analysts have argued against the use of computers in this manner. Vargas (1991) argues that the computer's responses are mechanical, that its responses are set prior to an event that might trigger that response. He notes that the computer's responses would have to come in contact with and be shaped by the environment to be effective in demonstrating complex human behaviors. It is interesting to note that his criticisms are directed (although not explicitly) against the use of rule-based computer systems designed to demonstrate human behavior, and his suggestions for improvement point in the direction of currently available ANNs. Epstein's (1986) criticisms are nearly identical to Vargas, noting that a premise behind most computer applications designed to model human behavior is that humans behave the way they do because they are information processors, using rules that result in complex behaviors. It is interesting to note that Epstein (1993) is now using computer simulations to aid his research in generativity theory.

Staddon (1988) criticized the use of a computer simulation to model choice behavior. Although the specific simulation he criticized did not utilized an ANN, his arguments hold for ANNs. Staddon notes that the computer simulation forces the programmer to make specific assumptions, for example about the length of "short-term memory" or temporal distributions of responding. This is true of any system that attempts to model phenomenon that are not completely describable. Hutchison (1984), as noted above, lists this as an advantage of using computer modeling as it forces him to address those assumptions. Additionally, Cook (1994) notes that the computer offers the flexibility to modify or adjust those assumptions quickly and observe the outcomes. Staddon (1988) notes however, that the success or failure of such models may depend on some unaddresssed assumption made on the programmer's part. Staddon (1993; Staddon & Zhang,1991) has also recently become interested in the use of computer simulations of adaptive behavior.

Some Applications of ANNs

Behavior Analysts have begun to apply ANN technology in both experimental (Donahoe, et al., 1993; Kehoe, 1989) and applied settings (Hutchison, 1998a; Hutchison & Stephens, 1987b; Stephens & Hutchison, 1992, 1993). Donahoe et al. (1993) have used an ANN to examine the utility of the unified reinforcement principle. Kehoe (1989) has examined the use of connectionist models of conditioning for investigating operant and respondent conditioning. Other researchers have used ANNs to also investigate operant and respondent behavior, although these researchers do not affiliate themselves with the behavioral sciences. Maki and Abunawass (1991) investigated the use of an ANN to simulate matching to sample (MTS), although his interpretations are from a connectionist framework. Rummelhart and McClelland (1986) successfully trained an ANN to learn the past tense of English verbs. Sutton and Barto (1981) examined the use of a network to model various aspects of respondent conditioning, while Barto, Sutton, and Anderson (1983) successfully trained an ANN to increase its ability to balance a pole. Although these examples are not exhaustive, they illustrate some successful implementations of ANNs applicable to the experimental analysis of behavior.

Other Behavior Analysts have successfully implemented ANNs based on behavioral, economic and evolutionary sciences (Hutchison & Stephens, 1987b; Stephens & Hutchison, 1993). Used in the airline industry, Stephens and Hutchison's system predicted optimal seat allocations on commercial airlines. The same authors are in the process of developing a program using an ANN to teach language-impaired children to learn the elementary verbal operants. The ANN is designed to recognize vocal verbal behavior and respond appropriately.

The logical extension of tnis, as described by Stephens and Hutchison (1992) in a related paper, is to demonstrate proficiency in responding to natural language through "emulating the functional properties of verbal behavior as described by Skinner (1957)" (p 152).

It is hoped that this paper has demonstrated the general utility and nature of ANNs. It seems to us that Behavior Analysts are in a prime position to contribute to the ongoing development of ANNs. The data obtained from the experimental analysis of behavior has already made great contributions to ANN development, and is likely to contribute more in the future. Considering the proliferation of computers and computer technology, this is also likely to increase the exposure and general acceptance of behavior analysis.

REFERENCES

Anderson, J. R. (1980). *Cognitive psychology and its implications*. San Francisco: W. H. Freeman and Co.

Barto, A. G., Sutton, R. S., & Anderson C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics, Vol. 13*, 835-846.

Bechtel, W., & Abrahamsen, A. (1991). *Connectionism and the mind*. Oxford, England: Blackwell, Ltd.

Bush, R. R., & Mosteller, F. (1955). A mathematical model for simple learning. In R. D. Luce, R. R. Bush, and E. Galanter, *Handbook of mathematical psychology, Vol. 1*, (pp 278- 305). New York: Wiley.

Caudill, M. (1987). Neural networks primer. *AI Expert, Dec.*, 46-52.

Caudill, M., & Butler, C. (1993). *Understanding neural networks*. Vol. 1. Cambridge, MA: MIT Press.

Cook, D. A. (1994, May). *The behavior of organisms studying the behavior of organisms*. Paper presented at the annual meeting of the Association for Behavior Analysis, Atlanta, GA.

Donahoe, J. W. (1991). Selectionist approach to verbal behavior: potential contributions of neuropsychology and connectionism. In L. J. Hayes, & P. N. Chase (Eds.), *Dialogues on Verbal Behavior* (pp. 119-145). Reno, NV: Context Press.

Donahoe, J. (1998). Interpreting verbal behavior. *Analysis-of-Verbal-Behavior, 15*, 107-112.

Donahoe, J. W., Burgos, J. E., & Palmer, D. C. (1993). A selectionist approach to reinforcement. *Journal of the Experimental Analysis of Behavior, 60*, 17-40.

Donahoe, J. W., & Palmer, D. C. (1989). The interpretation of complex human behavior: Some reactions to 'Parrallel Distributed Processing', edited by J. L. McClelland, D. E. Rummelhart, and the PDP Research Group. *Journal of the Experimental Analysis of Behavior, 51*, 399-416.

Donahoe, J. W., & Palmer, D. C. (1994). *Learning and Complex Behavior*. Allyn and Bacon: Nedham Heights: MA.

Donahoe, J. W., & Palmer, D. C. & Burgos, J. E. (1997). The S-R issue: its status in behavior analysis and in Donahoe's and Palmer's Learning and Complex Behavior. *Journal of the Experimental Analysis of Behavior, 67*, 193-211.

Epstein, R. (1986). Simulation research in the analysis of behavior. In A. Poling & R. W. Fuqua (Eds.), *Research Methods in Applied Behavior Analysis: Issues and Advances* (pp. 135-156). New York: Plenum Press.

Epstein, R. (1993). Generativity theory and education. *Educational Technology, 23(10)*, 40-45.

Haykin, S. (1994). *Neural networks: A comprehensive foundation*. New York: Macmillan College Publishing Company.

Hutchison, W. R. (1984, December).Cognitive versus behavioral approaches to artifical intelligence. *Newsletter for the Behavioral Artificial Intelligence Network*.

Hutchison, W. R. (1985, March). Behavior analysis and other adaptive systems theories within the "functional camp" of artificial intelligence. *Newsletter for the Behavior Artificial Intelligence Network, 2*, 3-18.

Hutchison, W. R. (1998a). *Adaptive autonomous agent with verbal learning*. U.S. Patent #5,802,506.

Hutchison, W. R. (1998b). Computer simulations of verbal behavior for research and persuasion. *The Analysis of Verbal Behavior, 15*, 117-121.

Hutchison, W. R., & Stephens, K. R. (1987a). Integration of distributed and symbolic knowledge representations. *Proceedings of the First International Conference on Neural Networks, 2*, 395-398. IEEE press.

Hutchison, W. R., & Stephens, K. R. (1987b). The airline marketing tactician (AMT): A commercial application of adaptive networking. *Proceedings of the First International Conference on Neural Networks, 4*, 753-756. IEEE press.

Kehoe, J. E. (1989). Connectionist models of conditioning: A tutorial. *Journal of the Experimental Analysis of Behavior, 52*, 427-440.

Maki, W. S., & Abunawass, A. M. (1991). A connectionist approach to conditional discriminations: Learning, short-term memory, and attention. In Commons, M. L., Grossberg, S., & Staddon, J. E. R. (Eds.). *Neural network models of conditioning and action*. Hillsdale, NJ: Erlbaum.

Mazur, J. E. (1986). *Learning and Behavior*. Englewood Cliffs, NJ: Prentice Hall Inc.

Palmer, D. C., & Donahoe, J. W. (1992). Essentialism and selectionism in cognitive science and behavioral analysis. *American Psychologist, 47*, 1344-1358.

Rescorla, R. A., & Wagner, A. R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In A. H. Black and W. F. Prokasy (Eds.) *Classical Conditioning II: Current research and theory*. New York: APPLETON-Century-Crofts.

Rummelhart, D. E., McClelland, J. L., & the PDP Research Group (1986). *Parallel distributed processing: Explorations in the microstructures of cognition*. (Vol. 1), Cambridge, MA: MIT Press.

Schlinger, H. (1992). Intelligence: Real or artificial? *The Analysis of Verbal Behavior, 10*, 125-133.

Shimp, C. P. (1989). Contemporary behaviorism versus the old behavioral straw man in gardner's the mind's new science: A history of the cognitive revolution. *Journal of the Experimental Analysis of Behavior, 51*, 163-171.

Skinner, B. F. (1957). *Verbal Behavior.* New York: Appleton-Century-Crofts.

Skinner, B. F. (1969). *Contingencies of reinforcement: A theoretical analysis.* Englewood Cliffs, NJ: Prentice-Hall, Inc.

Smith, L. (1998). *An introduction to neural networks* [On-line]. Available: *http://www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html.*

Staddon, J. E. R. (1988). Quasi-dynamic choice models: Melioration and ratio invariance. *Journal of the Experimental Analysis of Behavior, 49*, 303-320.

Staddon, J. E. R., & Zhang, Y. (1991). On the assignment-of-credit problem in operant learning. In M. L. Commons, S. Grossberg, & J. E. R. Staddon (Eds.) *Neural networks of conditioning and action, the XIIth Harvard Symposium.* Hillsdale, NJ: Erlbaum Associates. Pp. 279-293.

Staddon, J. E. R. (1993). A note on rate-sensitive habituation. In J. E. Meyer, H. L. Roitblat, & S. W. Wilson (Eds), *From animals to animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Behavior* (pp. 203-209). Cambridge, MA: Bradford/MIT Press.

Statsoft (1999). Neural Networks [On-line]. Available: *http://www.statsoft.com/textbook/steunet.html.*

Stephens, K. R., & Hutchison, W. R. (1992). Behavioral personal digital assistants: The seventh generation of computing. *The Analysis of Verbal Behavior, 10*, 149-156.

Stephens, K. R., & Hutchison, W. R. (1993). Behavior analysis and intelligent machines. *Educational Technology, 23(10)*, 52-61.

Sutton, R. S., & Barto, A. G. (1981). Toward a modern theory of adaptive networks: Expectation and prediction. *Psychological Review, 88*, 135-171.

Vargas, E. (1991). Verbal Behavior and Årtificial intelligence. In L. J. Hayes, & P. N. Chase (Eds.), *Dialogues on Verbal Behavior* (pp. 287-305). Reno, NV: Context Press.

Wasserman, P. D. (1989). *Neural computing: Theory and practice.* New York: Van Nostrand Reinhold.

Winograd, T. (1990). Thinking machines: Can there be? Are we? In D. Partridge, & Y. Wilks (Eds.), *The foundations of artificial intelligence: A sourcebook.* Cambridge, MA: Cambridge University Press.

Zomaya, A. Y. (1994). Reinforcement learning for the adaptive control of nonlinear systems. *IEEE Transactions on Systems, Man, and Cybernetics, 24*, 357-363.