Distribution Agreement

Signature:

______________________________ _________
Nicholas T. Calvin Date

Accordance of the Unified Theory of Reinforcement's Model of Behavior with the Modern Quantitative Law of Effect

By

Nicholas T. Calvin

B.S., Pennsylvania State University, 2008

Advisor: Jack J McDowell, Ph.D.

An abstract of
A thesis submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Master of Arts
in Clinical Psychology
2012

Abstract

Accordance of the Unified Theory of Reinforcement's Model of Behavior with the Modern Quantitative Law of Effect

By Nicholas T. Calvin

Virtual organisms animated by the Unified Theory of Reinforcement's neural network model of behavior responded on random interval schedules in a method that was largely consistent with the quantitative law of effect. The virtual organisms were evolved using an evolutionary algorithm to determine an optimum set of parameters that maximized the number of collected reinforcers while simultaneously minimizing the number of extraneous responses that were emitted. The behavior of the evolved virtual organisms was compared to the quantitative law of effect (Herrnstein, 1961), a modified version of the quantitative law of effect informed by the modern matching law (Soto et al., 2005), and to four comparison functions. The modern quantitative law of effect best described the data with 99.7% of the variance accounted for, but showed non-random standardized residuals. The median exponent was 0.74 for the best fits to the modern quantitative law of effect. The observed *k* was greater than the possible number of responses that the virtual organisms could emit in a time period, which supports an interpretation of *k* as simply a parameter rather than as the constant rate of responding (Dallery et al., 2000; McDowell, 2005). Although the virtual organisms exhibited very slight discrepancies from the modern quantitative law of effect, these results expand the number of phenomenon that can be demonstrated by the neural network models to include the quantitative law of effect.

Key Words: selection by consequences, behavior dynamics, quantitative law of effect, computational modeling, random interval schedules, neural networks, evolutionary algorithm

Accordance of the Unified Theory of Reinforcement's Model of Behavior with the Modern Quantitative Law of Effect

By

Nicholas T. Calvin

B.S., Pennsylvania State University, 2008

Advisor: Jack J McDowell, Ph.D.

A thesis submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Master of Arts
in Clinical Psychology
2012

Table of Contents

Accordance of the Unified Theory of Reinforcement's Model of Behavior with the Modern Quantitative Law of Effect

The Donahoe Unified Theory of Reinforcement asserts that operant and respondent behaviors are generated by the same underlying biological reinforcement model and only appear to differ due to contingencies of reinforcement (Donahoe et al., 1993). This theory is grounded in biological findings and emphasizes the ventral tegmental area (VTA) and corpus ammon area 1 (CA1) neurons of the hippocampus as detecting changes in neural activation. These changes in activation in turn alter the strength of neural connections based on the presence or absence of reinforcement provided by the environment. Over time these changes generate the behavior that is observed in experimental settings. To test the assertions of this theory, neural network models were designed with constraints based on the biological components of the theory.

It is important to understand the distinction between theories and models in this paper. As used in this paper, a theory is an explanation of how living organisms function, whereas a model is an implementation of the dynamics of a theory which can be examined in laboratory settings. Fundamentally, a model cannot prove that a theory is the true explanation of how living organisms function; only the behavior of living organisms can demonstrate that. However, it can support the theory by demonstrating that it is capable of generating behavior similar to that of live organisms.

The neural networks based on this theory have successfully mimicked a number of phenomena, but have not been applied to operant behavior. In the first paper on the Donahoe neural networks, the networks were shown to be capable of operant and respondent acquisition, extinction, and reacquisition, as well as blocking (Donahoe et al., 1993). Over the last 19 years, the Donahoe neural networks have successfully modeled revaluation, latent inhibition, and autoshaping (Donahoe & Burgos, 2000; Burgos, 2003; Burgos, 2007). Although the theory asserts that it applies to operant behavior, it has not been extensively tested using operant contingencies.

These successes have led Donahoe to assert that when neural networks are combined with genetic/neurodevelopmental algorithms (G/ND), they provide a method of examining the cumulative properties of selection by consequence (Donahoe, 2002). This method could serve as the foundation for an important behavioral perspective by demonstrating that the fundamental mechanism of selection by consequence could underlie all behavior. By showing the dynamic mechanisms of behavior, it allows for understanding why the environment and behavior correspond in very particular ways (McDowell, 2004). This method could serve as the basis for an understanding of behavior that unifies our understanding of operant and respondent behavior through the underlying method of selection by consequence. However there have been other approaches of modeling the process of selection by consequences (Catania, 2005; McDowell, 2004, e.g.).

The McDowell Theory of Selection by Consequences has been very successful at reproducing behavior in accordance with live organisms on single and concurrent schedules (McDowell, 2004; McDowell et al., 2008). The McDowell and Donahoe theories have theoretical similarities although the model implementations are very different. Both theories rely upon the concept of selection by consequences (McDowell, 2004; Donahoe et al., 1993; Skinner, 1981). They both emphasize that the dynamic processes of behavior provide important and unique information. The theories differ in their neurobiological interpretation, with the Donahoe theory emphasizing functional neuroanatomy (Donahoe, 1993) and the McDowell theory only recently drawing parallels with neurobiology through the Edelman theory of Neural Darwinism (Edelman, 1987; McDowell, 2010). When implemented as models, the Donahoe neural networks differ from the McDowell evolutionary algorithms, with the McDowell implementations being more abstract, not directly attempting to show biological processes, and currently being restricted to operant behavior.

The first phenomenon that the McDowell Theory of Selection by Consequences successfully modeled was the quantitative law of effect (McDowell, 2004). The quantitative law of effect is a robust phenomenon that has been shown to follow the hyperbolic form of,

$$R = \frac{kr}{r+r_e} \qquad (1)$$

where R represents response rate, r represents the rate of reinforcement, and k and r_e are parameters of the equation (Herrnstein, 1970). Although this form is fairly accurate, it is based on the classical matching law, which does not include the exponent and bias parameters that are found in the modern matching law (Herrnstein, 1961; Baum, 1974). The modern matching law has been found to more accurately represent data on concurrent schedules than the classic matching law (McDowell, 2005). It is possible to modify the modern matching law to create an exponentiated-hyperbola version of the quantitative law of effect in the same way that the quantitative law of effect is generated from the classical matching theory (Soto et al., 2005). The resulting exponentiated-hyperbola is

$$R = \frac{kr^a}{r^a + \frac{{r_e}^a}{b}} \qquad (2)$$

where *a* allows for deviations of responding termed undermatching and overmatching to reinforcement, *b* is the bias parameter, and all other parameters have been defined previously.

The purpose of the present article was to determine what form of behavior Donahoe neural networks generate on single component random-interval (RI) schedules. Ideally the neural networks should generate behavior demonstrating a hyperbolic or hyperbolic-exponentiated relationship between response and reinforcement. The current experiment utilizes similar methodology to that advocated by Donahoe (2002) and utilized in experiments by Burgos (1996, 1997). Experiment 1 assessed the faithfulness of the current implementation of Donahoe neural networks and experiment 2 examined what form of behavior the networks generate on single component schedules.

Virtual organisms, which are models built to interact with virtual environments, were implemented following the most current version of the neural network algorithm described in Sanchez, Galeazi, & Burgos (2010). Although the algorithm has been fairly consistent there have been some variations in the parameters and function forms to allow it to generate certain behaviors (Burgos, 2003; Sanchez et al, 2010). The algorithm can be broken down into two levels: structural and functional. The structural level of the algorithm concerns the structure of the network and how the components of the network are interfaced. The functional level delineates how the components of the network function and impact one another.

The structural level is held constant for any given experiment and is most bound by theory. One of the strengths of the model’s implementation of the Unified Theory of Reinforcement compared to other neural networks is that it constrains the structural level based on known neurobiological processes (Donahoe, 1993). Neural networks are set up with four distinct layers as can be seen in Figure 1. From left to right these are the input/stimulus, hippocampus interneuron, dopaminergic interneuron, and output/response layers, and their interconnections are unidirectional from one layer to the next. Each layer has a number of neural processing units (NPUs) that interact with the layers around it. Every NPU in this network has an activation level that roughly corresponds with the probability that the organism will respond to a stimulus. At a given time point the activation level of a NPU is a function of the previous interactions with the environment and the current activation levels of the NPUs in the previous layer. The activation level value is always within the range of 0 and 1.

There are two pathways in the structural layer that impact the behavior of the neural network. The stimulus-response (S-R) pathway determines what behavior is expressed by the virtual organism based on what stimuli are presented. The reinforcement/extinction (R/E) pathway alters the functioning of the S-R pathway based on the presence or absence of an US. Over time the R/E pathway adjusts the S-R pathway so that the virtual organism expresses a response that has resulted in receiving reinforcement after the presentation of certain stimuli. The delineation of these two pathways in the architecture of the model can be seen in Figure 1.

The input layer of the S-R pathway has a number of NPUs that are activated if the environment expresses certain stimuli. These NPUs detect the presence of stimuli in the environment and have their activation levels set by the salience of the stimuli (usually to the maximum value of 1). The input layer interacts with the hippocampal interneuron layer by connections (shown as thin black lines in Figure 1). These interconnections between NPUs modulate the effectiveness of the transmission of the activation values from one NPU to another. Over time these connections fluctuate in their effectiveness of transmission (weight) based on activation of NPUs in the R/E pathway. The hippocampal interneuron layer determines the activation level of each of its NPUs based on the activation of the NPUs in the input layer and the strength of the connections coming from the stimulus layer. The hippocampal interneuron layer acts as a sensory-association area which processes the combinations of sensory neurons. The NPUs of the dopaminergic interneuron and response layers are updated by the activations of the NPUs and the efficiency of their projecting connections in the hippocampal interneuron and dopaminergic interneuron layers, respectively. The dopaminergic interneuron layer behaves like the motor-association area which determines which responses the neural network will emit. If the output layer NPUs have an activation level greater than 0 the virtual organism emits that response to the environment.

The R/E pathway is the part of the model that is most constrained by the Unified Theory of Reinforcement. In the input layer the R/E pathway has NPUs which detect the presence of unconditioned stimuli. When an unconditioned stimulus (US) is present in the environment the activation level of the corresponding US NPU is set based on the magnitude of the reinforcement. This usually set to 1, but it has been shown that lower values delay the acquisition of responding to the stimulus (Donahoe, 1993). Each US NPU is associated with an unconditioned response (UR) NPU in the response layer. If the US NPU’s activation level is greater than 0 the UR NPU’s activation level is set to that value. Otherwise, the UR NPU’s activation level is determined by the dopaminergic interneuron layer. This connection gives the UR NPU the dual purpose of also acting as a conditioned response (CR) NPU so that it can be activated even if the US is not present in the environment.

In the R/E pathway there are two neurons which detect changes in activation of the S-R pathway. The primary activation change detector is part of the dopaminergic interneuron layer. According to the theory, this change detecting NPU corresponds with the ventral tegmental area (VTA). The VTA detects the presence of a US in the environment and releases dopamine when it is present strengthening the interconnections of neurons that are coactivated at that time. The VTA NPU mimics this behavior by having its activation level set to that of the US NPU if the US NPU's activation is greater than 0. If the US NPU is not activated than the activation level of the VTA NPU is determined by the activations of the dopaminergic interneuron layer. The second change detector is in the hippocampal interneuron layer and its activation level is determined by the NPUs in the S/R pathway of that layer. This change detecting NPU is supposed to serve the same function as the cornu ammonis area I (CA1) in the hippocampus in that it detects changes in the environmental stimuli through the sensory-association area of the brain. Based on the amount of change in the level of activation from one time step to another, the two activation change detecting NPUs alter the effectiveness of transmission of the interconnections by modifying their weight values. The weight value of a connection is an indication of its efficiency at transmitting the activation level from one NPU to another. The CA1 NPUs affect the connection weights between the input and hippocampal interneuron layers and those between the hippocampal interneuron layer and themselves. VTA NPUs alter the connection weights between the hippocampal and dopaminergic interneuron layers, the dopaminergic interneuron and output layers, and the dopaminergic interneurons and VTA NPUs. These connection changing pathways are shown in Figure 1 as the grey areas. If the difference in activation levels is greater than a certain threshold the connections are strengthened; otherwise the connections become weaker. For the CA1 NPUs the strength of the signal to the connections that determines whether or not they become more efficient is modulated by the signal coming from the VTA NPUs.

The functional level of the model consists of the processes that implement the dynamics of the algorithm. This is largely atheoretic, with the exact forms of the equations being irrelevant so long as the correct behavior is generated. This level is additionally devoted to dealing with how to abstract the processes of the brain that are part of the Unified Theory of Reinforcement. There are three components of the functional level that influence the overall behavior of the model: number of NPUs, NPU algorithms, and connection algorithms.

The number of NPUs in the interneuron layers has been repeatedly shown to impact the model's behavior. Recently, it was shown that the numbers of NPUs in the interneuron layers modify the rate of extinction of the network with more NPUs increasing the amount of time before extinction occurs after unconditioned reinforcement is halted (Sanchez et al., 2010). This flexibility in NPUs is not limited to the number of NPUs within layers; and it also extends to the number of interneuron layers in evolutionary G/ND algorithms (Burgos, 1996, 1997). When using the standard structure with no plurality in the number of hippocampal or dopaminergic interneuron layers, the following notation can be used to indicate the number of neurons in each layer: 1_1-3_1-3_1-1. The large numbers indicate the number of NPUs for each layer of the S-R pathway and the subscripts indicate the number of NPUs for each layer of the R/E pathway going from the input to the output layers.

As a necessary level of abstraction, the neural network functions in discrete time steps. For each time step the virtual organism obtains the stimuli from the environment, randomly determines the order to process the NPUs, updates the weights of the inter-NPU connections in a random order, and then emits the response to the environment. It is very important to note that the network does not function by processing the activations of the NPUs sequentially from input to output, but by selecting NPUs in a random order. Asynchonous activation is required for this neural network model as it will not function properly with synchronous activation. Asynchronous activation has the effect of blending the time steps together which may help compensate for the abstraction of time steps.

All NPUs and their interconnections function identically. Except when assigned a specific activation level, each NPU's activation level is a function of the activation of each of the NPUs in the previous layers multiplied by the strength of the connection between the two. Based on the activations of the change detecting NPUs the connection weights are altered. When the weights of the connections are altered they compete with one another with the maximum total value of the weights projecting to a specific NPU being 1. The exact equations used to manipulate these processes are written out in the Appendix but are not necessary to understand how this model operates.

Experiment I: Confirming the current implementation

Method

Subjects

The virtual organisms in this experiment used the same parameter values as were used in the Sanchez et al. paper (2010). The parameters for the NPUs were: θ_t (μ= 0.20, σ= 0.15), τ_t = 0.1, κ_t = 0.1, Logistic Function (δ = 0.5,γ = 0.1). The parameters used for the connections were: d_θ = 0.001, α = 0.5, β = 0.1. The starting weights of the connections were 0.15 for the input layer NPUs to hippocampal interneuron NPUs and the hippocampal interneuron NPUs to hippocampal NPU. The starting weights of the other connections were 0.01. This is the same procedure that has been used in more recent testing of the neural network model (e.g., Burgos, 2003; Sanchez et al., 2010). The network structures varied for each phase of the experiment. Please see the Appendix for a detailed description of the parameters.

Apparatus and Materials

The software was written and experiments were conducted on a computer using the Windows 7 operating system. The computer had a dual core 1.6 Ghz processor with 6 GB of RAM. The neural network and G/ND algorithms were written in VB.Net 2010. The environments that the virtual organisms behaved in were also implemented in VB.Net 2010. The number of responses emitted and the number of reinforcers received by the virtual organisms were recorded and stored in standard databases and analyzed using standard software. As needed, the activation levels of the US/CR and response NPUs were recorded using standard software.

Procedure

Three tests were run to confirm that the current implementation was capable of respondent and operant acquisition, extinction, and reacquisition. These were the first three phenomena that were demonstrated by the neural network model (Donahoe, 1993). Acquisition is the acquiring of behavior to reinforcement, extinction is the ceasing of behavior after the removal of reinforcement for behavior, and reacquisition is the faster acquiring of behavior after previous acquisition. Operant and respondent refer to the conditions whereby reinforcement was delivered to the organism. In an operant contingency the virtual organism had to emit a behavior to receive reinforcement and in a respondent contingency reinforcement occurred regardless of the virtual organisms behave at a certain time. Tests were run to confirm that it was also capable of the blocking effect, which is the prevention of learning when presented compound stimuli after one of those stimuli was previously conditioned. The purpose of these tests was to ensure that the model qualitatively produced results similar to the original implementation. The networks in this experiment were not directly comparable to the original paper, because they include the hippocampal layer, which was not implemented in the original presentation of these behaviors.

Respondent acquisition, extinction, and reacquisition consisted of 600 trials evenly divided between the three phases of acquisition, extinction, and reacquisition. For each trial the environment emitted the CS for 7 time steps. On the 7th time step (ts) of the acquisition and reacquisition trials the virtual organism received an US with a magnitude of 1 (maximum) giving an interreinforcement interval of 6 ts. On extinction trials no US was presented. At the 6th ts of each trial the activation level of the CR/UR NPU was measured. After the 7th ts the activation levels of the NPUs were reset to 0 to simulate the effect of an intertrial interval. The neural network architecture used for this test was 1_1-3_1-3_1-1.

Operant acquisition, extinction, and reacquisition followed nearly the same environmental framework as the respondent acquisition, extinction, and reacquisition. Unlike the respondent test, the emission of an US from the environment was conditional upon the virtual organism's behavior at the 6th ts. At the 6th ts, if the activation of the response NPU was greater than 0 the virtual organism emitted the required response for reinforcement. If the virtual organism emitted a response at the 6th ts it received reinforcement from the environment by the emission of an US on the 7th ts. The neural network architecture used for this test was 1_1-3_1-3_1-2.

The neural network architecture used to test for blocking was 2_1-3_1-3_1-1. For this test 600 trials were broken across three phases. Each trial consisted of 7 time steps of a stimulus presentation with an US presentation on the 7th ts. For the first phase, only stimulus 1 was emitted for each trial. During the second phase, stimulus 1 and 2 were emitted simultaneously. For the final phase only stimulus 2 was emitted. On the 6th ts of each trial the activation level of the CR/UR NPU was measured and recorded.

Results

Respondent conditioning was exhibited by the neural network, consistent with Donahoe (1993). As can be seen in Figure 2, the virtual organism successfully acquired conditioned responding after 180 trials (first panel), extinguished 150 trials after CS-US pairings ceased (second panel), and reacquired in approximately 30 trials after CS-US pairings were reintroduced (third panel),. As in the original experiment, the virtual organism more quickly acquired responding during the reaquisition phase than it did during the acquisition phase. Unlike the original paper, this implementation of the neural network took much longer to acquire CR to the stimulus. This difference is primarily due to the longer time it takes to build up the weights in the hippocampal interneuron layer which was not present in the original neural network. According to Dr. Burgos, the neural network model is "extremely sensitive to parameter and architecture changes" (personal communication, September 19, 2011). This precluded a stricter quantitative analysis of the timing of acquisition, extinction, and reacquisition.

This implementation of the neural network model also exhibited operant conditioning. The behavior of the neural network can be seen in Figure 3 with the activation levels of the UR/CR NPU shown as the filled points and the operant response NPU as the unfilled points. As was observed in the respondent conditioning test, the virtual organism successfully acquired the behavior, extinguished, and reacquired operant responding. As was shown previously by Donahoe (1993) the virtual organism began showing increased activation on the UR/CR NPU sooner than on the response NPU. This was the major feature of the original implementation when the virtual organisms were placed in this environment. It is noteworthy that activation of the operant response NPUs continued to increase even after the extinction procedure was implemented (second panel of Figure 3). Continued increases in the weights between NPUs can still increase even after CS-US pairings cease, because the VTA NPUs can still be activated by the NPUs in the dopaminergic interneuron layer.

The virtual organism also demonstrated the phenomenon of blocking. As can be seen in Figure 4, the virtual organism effectively acquired conditioning to stimulus 1 after 180 trials. After 200 paired associations with stimulus 1 and the US, stimulus 2 was unable to acquire conditioned responding to the US within 200 trials despite repeated pairings. This result was consistent with Donahoe (1993) which showed no responding on probe trials when stimulus 2 was presented.

Overall, this implementation of the Donahoe neural network has the same qualitative characteristics as Donahoe's (1993) implementation. The virtual organism qualitatively demonstrated acquisition, extinction, and reacquisition for both operant and respondent procedures, and these phenomena had the same features as Donahoe (1993) reported for his implementation. The virtual organism also successfully demonstrated blocking, with no responding occurring in 200 trials where stimulus 2 was presented alone (third panel of Figure 4). Although this implementation is not quantitatively identical to the original implementation, it is a faithful reimplementation of the Donohoe group's current model.

Experiment II: Behavior on single RI schedules

Method

Subjects

The subjects for this experiment consisted of 1_1-3_1-3_1-2 architecture neural networks. These neural networks were identical to those used in Experiment 1 except for the addition of one parameter. Previous research has used the assumption that an activation level greater than 0 was sufficient for the organism to emit a response. In preliminary testing, this resulted in the number of measureable responses on single random interval (RI) schedules always being equal to the number of opportunities to respond, because the activation levels of the output layer NPUs were always above 0.

To counteract this, a threshold parameter that determined what constituted a response was added to the organism. When determining whether the virtual organism emitted a response, the activation levels of the CR/UR and response NPUs were compared to the value of this parameter and if they exceeded, it the virtual organism emitted that response to the environment. In preliminary testing a very small value of 0.0005 was sufficient to prevent constant responding. Additionally, a response activation threshold (r_t) parameter value this low does not prevent spontaneous activation from occurring. The addition of this parameter did not affect the ability of the model to demonstrate the findings in Experiment 1.

Procedure

For this experiment, 11 single component random interval (RI) schedules were presented to virtual organisms in a random order within a virtual operant chamber. The scheduled RI rates were 2, 3, 5, 8, 12, 17, 25, 45, 85, 145, and 225. An RI rate of 1 was not included, because in preliminary testing, an RI 1 schedule resulted in slightly fewer emitted behaviors than were observed on poorer schedules of reinforcement due to the difficulty of differentiating between reinforcement for responding and non-responding.

Previous experiments that have tested Donahoe neural networks have made the simplifying assumption that every stimulus-response trial is distinct. This assumption does not hold for single component RI schedules where the timing of reinforcer delivery depends on when a reinforcer is available, and the organism's behavior at that time. In an operant chamber with a single component RI schedule, the stimulus is presented constantly and there are no breaks in stimulus presentation between reinforcements. The absence of distinct trials does not allow the activation levels of the NPUs to be set to 0.

Thus it was necessary to modify the environment-organism interaction so that a stimulus could be presented continuously. Each schedule presented a stimulus to the virtual organism for either 10,500 or 20,500 ts. During this time if the virtual organism emitted a response, the environment checked to see if a reinforcer was available. If a reinforcer was available the virtual organism was presented with an US that maximally activated the US NPU. For analyses, the first 500 ts of a schedule were discarded and only the remaining ts were analyzed.

Prior to testing on single RI schedules the virtual organisms were conditioned to acquire responding to the stimulus. The virtual organism was presented with 500 trials of stimulus-US pairings. For each trial the environment emitted the stimulus for 5 time steps. On the 4th time step if the response NPU's activation of the virtual organism exceeded the response threshold it was presented with an US that maximally activated the US NPU. If the activation level of the response NPU was never greater than 0.8 at the 4th ts during the 500 trials it was deemed to have failed at acquiring responding to the CS and the virtual organism was not run on the RI schedules.

This experiment can be broken down into two phases. During the first phase, virtual organisms were evolved using an evolutionary algorithm to determine an optimal set of parameters. During the second step of the experiment, this set of optimal parameter values was used in 10 virtual organisms to examine what function form best described the responding of a virtual organism. Each RI schedule was presented for 10,500 ts during the evolutionary algorithm fitness testing and 20,500 ts for the testing of the evolved parameters.

Evolutionary Algorithm

An evolutionary algorithm was used to evolve a population of neural networks. Evolutionary algorithms are an engineering method that recursively determine a point of optimality. This technique is especially useful if it is not well understood how the parameters affect the results of an algorithm. In this case, the parameter space of the neural networks that the Donahoe model uses has not been thoroughly examined. The goal of this evolutionary algorithm was to determine the set of parameters which when implemented in the neural network maximized the number of collected rewards while simultaneously minimizing the number of extraneous responses.

The evolutionary algorithm implemented virtual organisms in test environments and observed their behavior. With every generation, virtual organisms were selected to be parents based on how fit they were, children were created from the selected organisms, and the children were then subjected to mutation. Every virtual organism was exposed to the virtual environment of 11 single RI schedules to determine their fitness. A population of 100 organisms (neural networks) was evolved over 100 generations generating a total of 10,000 neural networks.

The initial generation of the evolutionary algorithm was produced using a prototype organism and heavily mutating it. The prototype organism was based on some preliminary sampling, because it generated a curvilinear pattern of behavior that showed increasing responding with increasing reinforcement (See figure 5). This pattern of behavior was not ideal, because it showed a slight decrease in responding on very rich schedules. The parameters for the NPUs were: θ_t (μ= 0.20, σ= 0.15), $\tau_t = 0.1$, $\kappa_t = 0.1$, Logistic Function ($\delta = 0.5$, $\gamma = 0.1$), & $r_t = 0.0005$. The parameters used for the connections were: $d_\theta = 0.005$, $\alpha = 0.7$, $\beta = 0.15$. This prototype organism was copied 100 times and these copies were mutated (see below) with a mutation rate of 5% to generate the initial population.

Fitness

The fitness of every organism was a function of the costs and benefits of responding when placed in the aforementioned set of environments. For each reinforcer presented by the environment due to the virtual organism's behavior, the virtual organism's fitness value was increased by 20 points. To simulate the opportunity cost of behavior, each time the organism emitted a behavior its fitness value was reduced by one point. The observed fitness can thus be expressed as,

$$Fitness_{obs} = (r * 20) - (R * 1)$$

where *R* represents the number of responses and *r* represents the number of reinforcements. This was how the fitness of every virtual organism was determined, except in cases where the virtual organism was unable to acquire operant responding. If the virtual organism was unable to acquire responding within 500 trials of operant CS-US reinforcement pairings it was assigned the minimum possible fitness value.

Parental Selection

Parental selection occurred after all of the organisms were assigned fitness values based on their behavior. Tournament selection was used with 5 competitors per tournament. In tournament selection a number of competitors are selected from the population of virtual organisms. Among these competitors the one with the highest fitness was selected to be one of parents of the next generation. In the event of a tie among the competitors, no parent was selected and another set of competitors was drawn. This process was repeated until 100 parents were selected. It is important to note that theoretically this process could select the same organism to be each of the 100 parents. To ensure that this did not occur, if there was only one unique parent, then the entire set of 100 parents was discarded and the process was repeated until a set of 100 parents was selected that had at least two unique parents.

Reproduction

Two parents were randomly chosen with replacement from the selected parents to generate a child. A method of generating children is to translate the parameter values of both parents into bit strings and then use crossover recombination. To translate the parents into bit strings each of the parent's 10 parameter values was translated into a 10-bit string creating a total parent bit string consisting of 100 bits. The parent's parameter values, which ranged between 0 and 1, were multiplied by 1023 (max binary value of 10-bits) and then translated to binary. This method gave the parameter values a fixed precision of about 0.001 . The exceptions to this were the response and reinforcement threshold parameter values, which were divided by 10 because in the model their values were required to be much smaller than the other parameters. These two parameters were precise to 0.0001 and ranged in value from 0 to 0.1.

After the two parents were translated into bit strings, single point crossover recombination occurred. In single point crossover recombination a single point along one of the parent's bit string is selected. To the left and including this point the bits from one of the parents and to the right of it the bits from the other parent are used. These are then combined to create a new bit string that is a child of both parents. To ensure that no clones of the parents were created the crossover points that could be selected were restricted to the range from the second to the next to last bit. This process of reproduction was repeated 100 times to create the next generation of virtual organisms.

Mutation

After the new generation of organisms was created it underwent random bitwise mutation. For each of the children, every bit of its bit string representation was subjected to a percentage chance of its bit value flipping from 0 to 1 or 1 to 0. The percentage chance of each bit mutating was set at 1%. At one percent mutation, the probability of an organism having at least one mutation in its bit string representation is 64% ($1 - 0.99^{100}$). Although this may seem high, the probability of any given parameter value mutating was about 10% ($1 - 0.99^{10}$); with 10 parameters this gives an expected value of 1 parameter changing per organism mutation. After each of the organisms in the new generation was mutated the bit string representation was translated back into parameter values. Once the new generation was returned to parameter values, the entire evolutionary process was repeated.

Testing the Evolved Parameters

Once all 100 generations of virtual organisms had been tested, the ideal parameters that the evolution had converged upon were used to animate 10 virtual organisms. These virtual organisms were tested in the same virtual environments that were used during the evolutionary algorithm run. Each schedule was run for 20500 time steps and the number of reinforcements and operant responses were counted. The resulting patterns of behavior were then examined to see what function form best described their distribution of responses and reinforcements.

Results

Evolutionary Algorithm

The evolutionary algorithm converged on a set of parameters within the first 20 generations. The parameter values found by the evolutionary algorithm for the NPUs were: θ_t (μ= 0.20, σ= 0.15), τ_t = 0.10, κ_t = 0.10, Log. Function (δ = 0.50, γ = 0.10), & r_t = 0.0005. The evolved parameter values for the connections were: d_θ = 0.0054, α = 0.83, β = 0.15. Each of the evolved parameter values was rounded to the nearest hundredth except for the delta threshold and response threshold that were rounded to the nearest ten thousandth before being used in phase 2. The effect of the evolution was localized to the connection parameters. Compared to the prototype virtual organism used to seed the evolutionary algorithm, the connection parameters of the delta threshold, acquisition, and extinction parameters were higher. This set of parameters still showed the same qualitative features in operant and respondent acquisition, extinction, and reaquisition and blocking as was shown in Donahoe (1993).

Evolved Parameter Values

The response and reinforcement frequencies of each scheduled RI for each of the 10 virtual organisms was averaged over 40 500-ts blocks after discarding the first 500-ts block. This gave a total of 440 blocks of data for each of the virtual organisms. Averages of fewer blocks of data result in more variability in the observed responses, which is more typical of live organisms. Equations 1 and 2 were fitted to data from each of the 10 virtual organisms by the method of least squares. The proportion of variance accounted for and the parameter values estimated from the fits are listed in Table 1. The parameters for Equation 1 (hyperbola) are listed in columns 2 and 3 and the parameters for Equation 2 (exponentiated-hyperbola) are listed in columns 5, 6, and 7. The mean of the proportion of variance accounted for by both equations was nearly 1.00. Based on a Wilcoxon matched-pairs signed-rank test, the exponentiated-hyperbola accounted for a higher proportion of variance than the hyperbola ($T_{obs} = 0$, $p < 0.01$).

The observed exponents showed systematic undermatching with values that ranged from 0.82 0.64 with a median of 0.73 across the 10 virtual organisms. The value of the exponentiated-hyperbola's asymptote for every virtual organism exceeded the possible number of responses per 500-ts block with a median of 572. The value of the asymptote of the hyperbolic fit for each virtual organism also exceeded the possible number of responses per 500-ts block with a median of 516.

Four additional function forms were fitted to the data of the ten virtual organisms to examine the uniqueness of the exponentiated-hyperbola fit. The four functions were an asymptotic exponential,

$$R = a(1 - e^{-br}),$$

an asymptotic power,

$$R = a(1 - r^{-b}),$$

a logarithmic,

$$R = log_b(r),$$

and a ramp,

$$R = \begin{cases} br, & 0 \leq r \leq \dfrac{a}{b} \\ a, & 0 > \dfrac{a}{b} \end{cases}.$$

For all of these equations, R and r represent the rates of response and reinforcement, respectively. a represents the asymptote or upper limit of each function and b represents the rate of change. The asymptotic exponential, asymptotic power, and logarithmic have differential properties that are similar to those of a hyperbola. The piecewise continuous ramp function provides a base line for comparison to continuous, monotoically increasing functions. It is the simplest description of data that increase rapidly and then reach an asymptote (Beardsley & McDowell, 1992). The ramp function consists of a line with slope b that connects to a horizontal line with a constant value of a once the reinforcement rate is greater than a / b.

The proportions of variance accounted for by least squares fit of these four equations are shown in Table 1. The median proportion of variance accounted for by the asymptotic exponential, asymptotic power, logarithmic, and ramp functions were 0.989, 0.989, 0.985, and 0.971, respectively. Wilcoxon matched-pairs signed-ranks tests confirmed that both the hyperbola and hyperbolic-exponentiated accounted for a greater proportion of the variance than the four comparison functions ($T_{obs} = 0$, $p = 0$ for all eight comparisons).

The randomness of the residuals for all 6 function forms were tested by fitting cubic polynomials to the pooled standardized residuals. Both the hyperbolic-exponentiated and the hyperbola showed significant cubic polynomial trends in the standardized residuals ($R^2 = 0.72$ and 0.78, respectively). The standardized residuals for the exponentiated-hyperbola and the hyperbola are shown in Figures 6. Significant polynomial trends were also found in the standardized residuals of the asymptotic exponential, asymptotic power, logarithmic, and ramp fits.

The evolutionary algorithm successfully evolved a set of parameter values that when implemented in Donahoe neural networks generated behavior similar to that of biological organisms. The exponentiated-hyperbola best accounted for the observed data with only 0.3% of the variance unaccounted for. However, the remaining 0.3% was non-random with significant polynomial trends in the residuals. Despite this weakness this function form is the best description of the data when compared to the other function forms tested.

Discussion

There are two parts to the following discussion which separately discuss the current implementation of the model and this methodological approach. The first part focuses on the success of the current implementation and the likely ways to modify it to generate behavior that is more in accordance with live organisms. The second part of the discussion deals with some potential problems with the methodology and proposes an alternative method of building the model.

Accordance with the Modern Quantitative Law of Effect

This implementation of Donahoe neural networks generated behavior that was consistent with previous research. A G/ND algorithm was used to discover parameters that when implemented in a 1_1-3_1-3_1-2 neural network architecture were capable of generating behavior that is well described by the exponentiated-hyperbola version of the quantitative law of effect. This was accomplished by balancing the benefits of attained reinforcements while simultaneously minimizing the opportunity costs of responding. The estimated exponents of the exponentiated-hyperbola fits (median = 0.73) were within the range of exponents typically observed in live organisms. In a recent meta-analysis of the concatenated matching law the average exponent of the matching to reinforcement rate component was found to be 0.74 (Cording et al., 2011).

The asymptotes of the exponentiated-hyperbola and hyperbola fits exceeded the number of responses that the virtual organisms could possibly express. The parameter that governs the asymptote, k, was originally conceptualized as the sum of the number instrumental responses (R) plus all extraneous responses (R_e) (Herrnstein, 1970), and was theoretically required to remain constant. However, more recent experiments have shown that k is not constant and thus cannot be equal to the sum of R and R_e (Dallery et al., 2000; McDowell, 2005). To the extent that this model reflects the behavior of live organisms, the finding that k is greater than the possible number of responses supports the notion that *k* is simply a parameter that can be used to describe the form of behavior that organisms express in RI schedules.

The non-randomness of the residuals is the most problematic feature of the exponentiated-hyperbola and hyperbola fits. There are no existing data from live organisms that suggest there is a significant cubic polynomial trend in the residuals for the modern quantitative law of effect (McDowell, 2005), and competing theories do not show this trend (McDowell, 2004). Although unlikely, previous examinations of the residuals may have failed to detect these very small trends in the residuals due to the larger amount of error in measurement that is not averaged out in smaller samples of behavior.

The combination of non-random residuals and the very high *k* are not ideal despite the high percentage of variance accounted for. Future research should focus on eliminating the pattern in the residuals and possibly generating a set of parameters that would reduce the value of *k*. The pattern of results obtained in this study may have been affected by features of the experimental design, such as the choices of the reinforcement schedules and the cost-benefit ratio that was used to calculate fitness. Leaner schedules of reinforcement and a higher cost-benefit ratio might reduce the observed *k*. It also might be possible to remove the non-random residuals by using different network architectures or by modifying the equations that are used by the algorithm. Another approach would be to add the inhibitory pathways that have been discussed conceptually, but have never been implemented in the model. The current model does not produce behavior that is fully in accordance with what has been observed in live organisms, but it is very close, and it is possible that small changes would render the model fully consistent with the behavior of live organisms.

If these details can be corrected, then the modern Matching Law (Baum, 1974) is the next step in expanding the number of phenomena that the neural network models of the Unified Theory of Reinforcement can reproduce. Examining the behavior of these neural networks on concurrent schedules would expand the flexibility of the model by allowing them to dynamically handle two or more mutually exclusive responses. Generating behavior that followed the modern Matching Law while simultaneously being capable of respondent phenomena would allow for unique, testable hypotheses. Additionally, by being capable of generating behavior that is in accordance with the modern matching law it would strengthen the argument that operant and respondent behavioral phenomenon can be explained by the Unified Theory of Reinforcement. However this is only the case if there is an architecture and set of parameters that can be implemented in the model that is simultaneously capable of generating both respondent and operant behavior, which has not been demonstrated yet.

Methodological Weaknesses

Although the combination of G/ND algorithms and neural networks as a method of inquiry is capable of generating behavior that describes important phenomenon, there are some weaknesses to this approach. This experiment limited the number of parameters that could be varied and the structure of the network, unlike previous efforts which have permitted much more parameter variance across individual NPU layers, number of interneuron layers, and number of NPUs in those layers (Burgos, 1996; Donahoe and

Burgos, 1999; Donahoe, 2002). The methodological approach that Donahoe advocates allows for infinite degrees of freedom, although it is somewhat restricted by the format of the neural network structure. Furthermore, the exact forms of the equations that are used to animate the neural networks are also not part of the theory and are thus also subject to change if the model is not capable of generating appropriate behavior. For this reason the current approach to testing the model is nearly unfalsifiable and may provide misleading evidence for the Unified Theory of Reinforcement.

The main problem with this approach is that, at best, it can only demonstrate that something is possible, but not necessarily that it is probable. As has been pointed out previously there are a large number of parameters in this model that can be manipulated to generate behavior, which suggests that it might be possible to obtain any desired outcome (Marr, 1997). With an infinite number of possible parameters and possible structures, there are also an infinite number of possible combinations that could generate similar results. As a clear example, if the fitness of a G/ND algorithm were set to maximize the length of time a virtual organism could maintain flight in a virtual environment with perfect physics implemented, all that one could say about the resulting virtual organism would be that it can fly. Without further testing it would be impossible to know whether it flew like a plane, helicopter, glider, balloon, bird, or some other mechanism. All of these options could generate behavior that met the criterion, but if our goal is to mimic the dynamic behavior of living organisms we would still be no closer to that goal. The purpose of using this model is to provide evidence that the Unified Theory of Reinforcement is possibly correct, but by allowing an infinite number of possibilities it in effect provides no evidence that it is probably correct.

In previous research Donahoe neural networks have been allowed to vary in structure across experiments. The usual justification of this approach is that the important components of the neural networks can be separated from a theoretically complete network that would exhibit all behaviors. However, this assumption is inconsistent with the actual implementation of the model, and it limits its future utility. The competitiveness of the connections between NPUs violates the assumption of the independence of components of a complete neural network because the strength of connections that are not being activated, but that nevertheless connect to the same NPUs, can interfere with the process of learning. This is further exacerbated by the effect of the number of NPUs on the rate of extinction (Sanchez et al., 2010). Thus the assumption that parts of a theoretical complete network can be separated from one another is false. This does not invalidate previous research, but it does suggest that future research should focus on a specific network structure that can be added to and developed, so long as it maintains the ability to generate the behaviors it has previously expressed. Eventually this could lead to a neural network model that would simultaneously generate a wide range of both operant and respondent behaviors, which is more in accordance with the Unified Theory of Selection by Reinforcement than the current mix and match approach.

References

Baum, W.M. (1974). On two types of deviation from the matching law: Bias and undermatching. *Journal of the Experimental Analysis of Behavior*, *22*, 231-242.

Beardsley, S.D., & McDowell, J.J (1992). Application of Herrnstein’s hyperbola to time allocation of naturalistic human behavior maintained by naturalistic social reinforcement. *Journal of the Experimental Analysis of Behavior*, *57*, 177-185.

Burgos, J.E. (1996). Computational explorations of the evolution of artificial neural networks in Pavlovian environments. Unpublished doctoral dissertation, University of Massachusetts, MA.

Burgos, J.E. (1997). Evolving artificial neural networks in Pavlovian environments. In: Donahoe, J.W., Dorsel, V.P. (Eds.), Neural-Network Approaches to Cognition: Biobehavioral Foundations. Amsterdam, Elsevier, pp. 58-79.

Burgos, J.E. (2003). Theoretical note: simulating latent inhibition with selection neural networks. *Behavioral Processes*, *62*, 183-192.

Burgos, J.E. (2007). Autoshaping and automaintenance: a neural-network approach. *Journal of the Experimental Analysis of Behavior*, *88*, 115-130.

Catania, A.C. (2005). The operant reserve: a computer simulation in (accelerated) real time. *Behavioral Processes*, *69*, 257-278.

Cording, J.R., McLean, A.P., & Grace, R.C. (2011). Testing the linearity and independence assumptions of the generalized matching law for reinforcer magnitude: a residual meta-analysis. *Behavioral Processes, 87,* 64-70.

Dallery, J., McDowell, J.J, & Lancaster, J.S. (2000). Falsification of matching theory's account of single-alternative responding: Herrnstein's k varies with sucrose concentration. *Journal of the Experimental Analysis of Behavior, 73*, 23-43.

Donahoe, J.W. (2002) Behavior analysis and neuroscience. *Behavioral Processes*, *57,* 241-259.

Donahoe, J.W., & Burgos, J.E.(1999). Timing without a timer. *Journal of the Experimental Analysis of Behavior*, *71*, 257-301.

Donahoe, J.W. & Burgos, J.E. (2000). Behavior analysis and revaluation. *Journal of the Experimental Analysis of Behavior, 74,* 332-346

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

Edelman, G. (1987). *Neural Darwinism*. New York: Basic Books.

Herrnstein, R.J. (1961). Relative and absolute strength of response as a function of frequency of reinforcement. *Journal of the Experimental Analysis of Behavior*, *4*, 267-272.

Herrnstein, R.J. (1970). On the law of effect. *Journal of the Experimental Analysis of Behavior*, *13*, 243-266.

Marr, J. (1997). The eternal antithesis: a commentary on Donahoe, Palmer, and Burgos. *Journal of the Experimental Analysis of Behavior*, *67*, 232-235.

McDowell, J.J (2004). A computational model of selection by consequences. *Journal of the Experimental Analysis of Behavior, 81*, 297-317.

McDowell, J.J (2005). On the classic and modern theories of matching. *Journal of the Experimental Analysis of Behavior, 84*, 111-127.

McDowell, J.J (2010). Behavioral and neural Darwinism: selectionist function and mechanism in adaptive behavior dynamics. *Behavioral Processes, 84*, 358-365.

McDowell, J.J, Caron, M.L., Kulubekova, S., & Berg, J.P. (2008). A computational theory of selection by consequences applied to concurrent schdules. *Journal of the Experimental Analysis of Behavior*, *90*, 387-403.

Sanchez, J.M., Galeazzi, J.M., & Burgos, J.E. (2010). Some structural determinants of Pavlovian conditioning in artificial neural networks. *Behavioral Processes*, *84* 526-535.

Skinner, B.F. (1981). Selection by consequences. *Science*, *213*, 501-504.

Soto, P.L., McDowell, J.J, & Dallery, J. (2005). Effects of adding a second reinforcement alternative: implications for Herrnstein's interpretation of r_e. *Journal of the Experimental Analysis of Behavior*, *84*, 185-225.

Figures

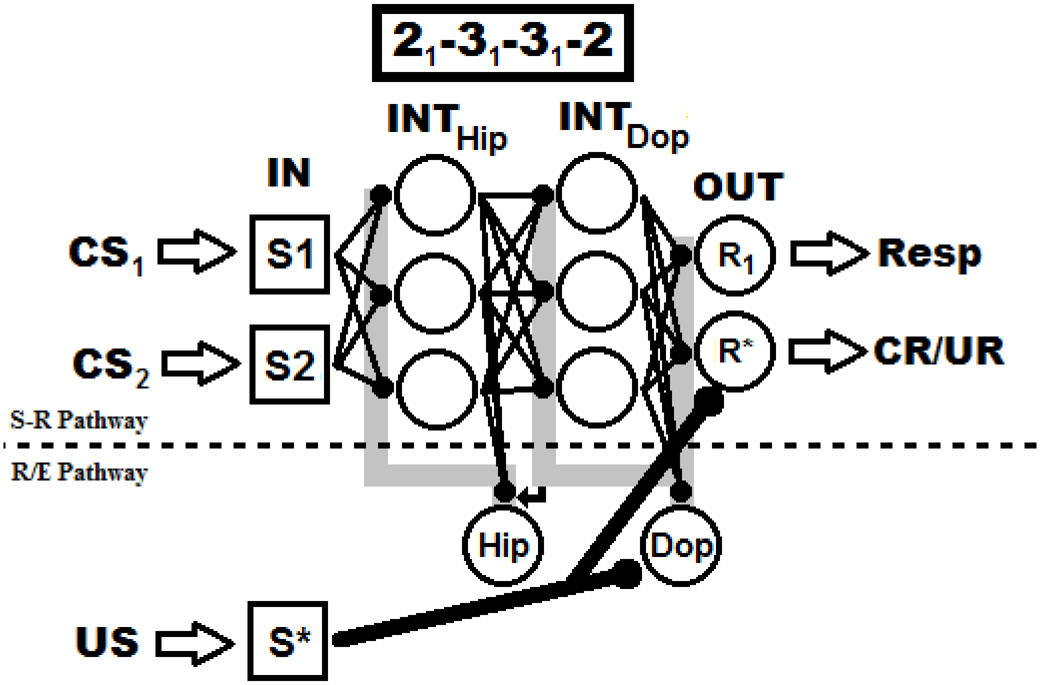


Figure 1. Example of a standard neural network architecture. The structure of the network is denoted by 2_1-3_1-3_1-2. The large numbers indicate the number of NPUs in the S-R pathway and the subscripts indicate the number of NPUs in the R/E pathway. The order of the numbers corresponds to the individual layers (input, hippocampal interneuron, dopaminergic interneuron, and output). The division between the stimulus-response (S-R) and extinction/reinforcement (E/R) pathways is shown by the dashed line. NPUs are symbolized by squares (sensory units) and circles (processing units). S = sensory input NPU, S* = US input NPU, Hip = hippocampal like NPU, Dop = dopaminergic like NPU, INT = interneuron, R = response NPU, R* = CR/UR NPU. Shaded areas indicate the connection weight changing pathways and the small arrow connecting them indicates the modulatory effect of the dopaminergic discrepancy signal on the hippocampal weight changing

pathway. The thick black lines indicate fixed, maximally strong connections and the thinner lines indicate flexible connections.

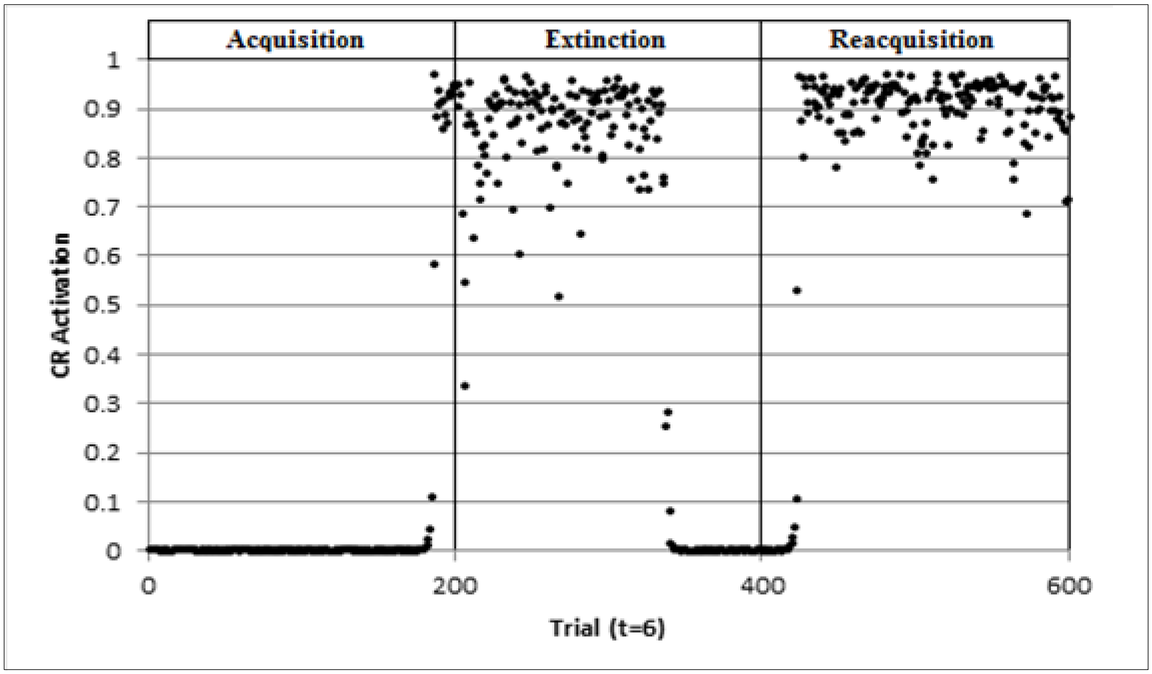


Figure 2. CR/UR NPU activation level at time step 6 during respondent based acquisition, extinction, and reacquisition training using a 1_1-3_1-3_1-1 architecture with standard parameters.

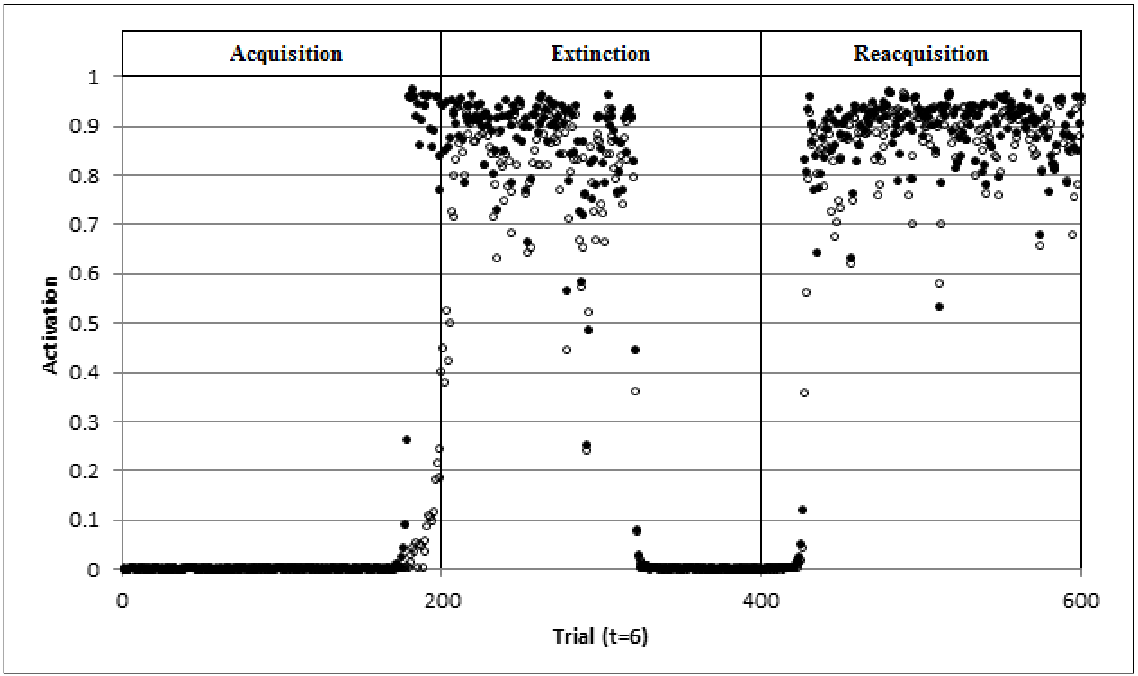


Figure 3. Activation levels of CR/UR and response NPUs at time step 6 during operant based acquisition, extinction, and reacquisition training using a 1_1-3_1-3_1-2 architecture with standard parameters. The filled and unfilled points are the activation level for each trial of the CR/UR and response NPUs, respectively.

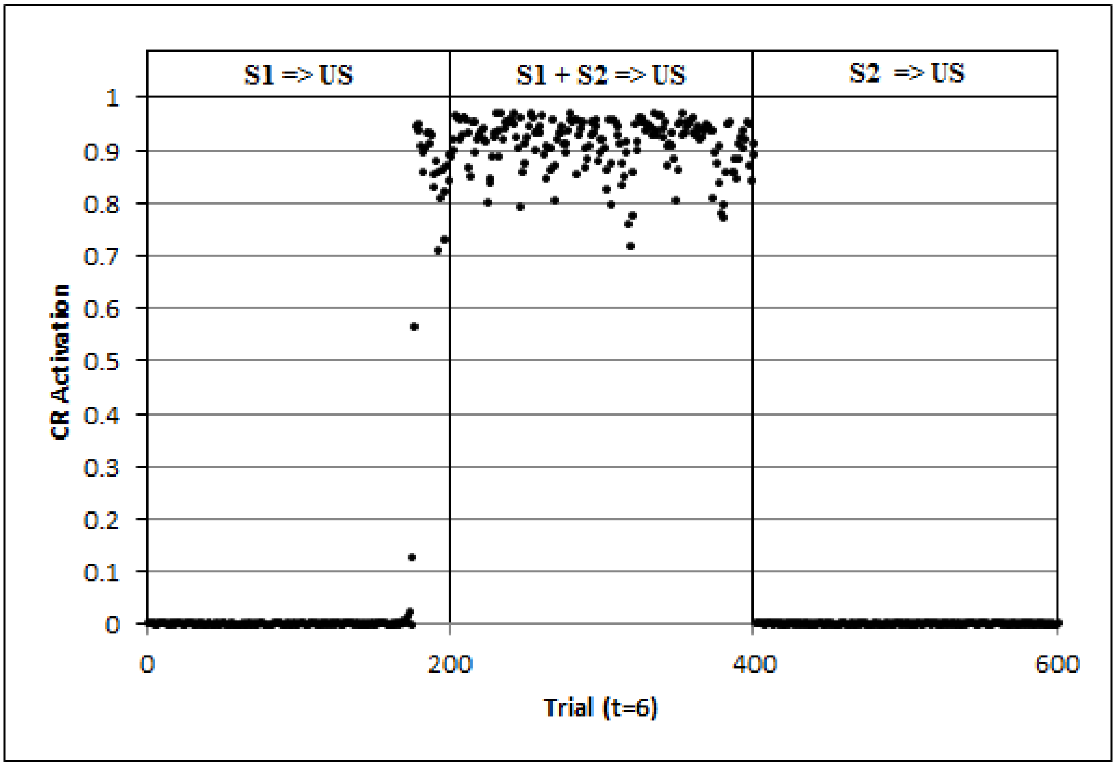


Figure 4. Activation levels of the CR/UR NPU at time step 6 during a blocking demonstration using 2_1-3_1-3_1-1 architecture with standard parameters. In the first panel (S1 => US) the environment emits the first conditioned stimulus and pairs that with an US. In the second panel (S1 + S2 => US), the environment emits a compound stimuli and pairs that with an US. During the final panel just the second stimulus was paired with the US.

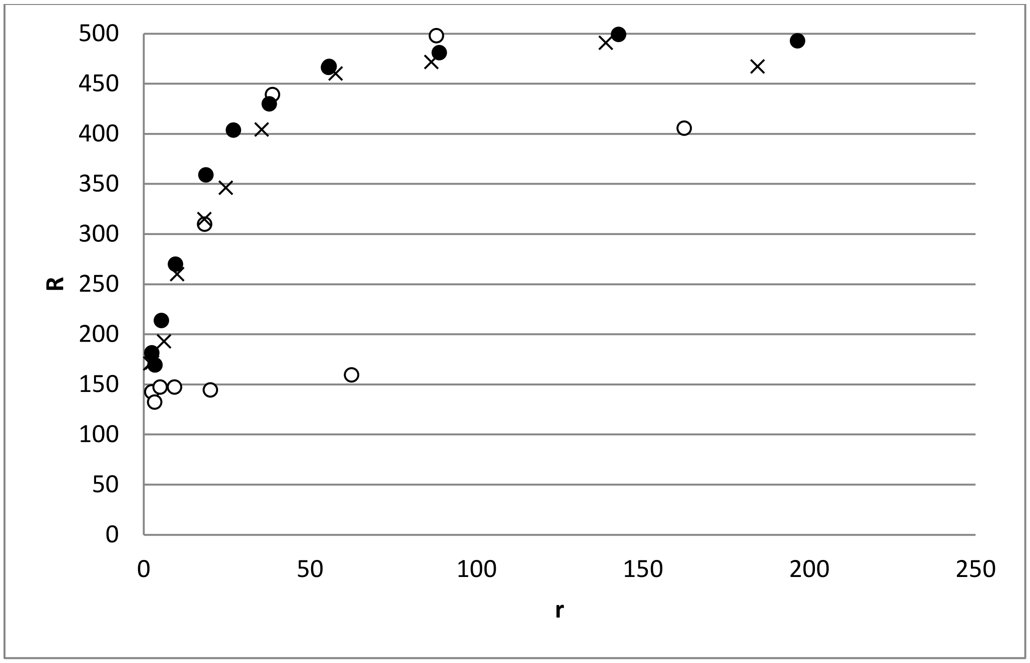
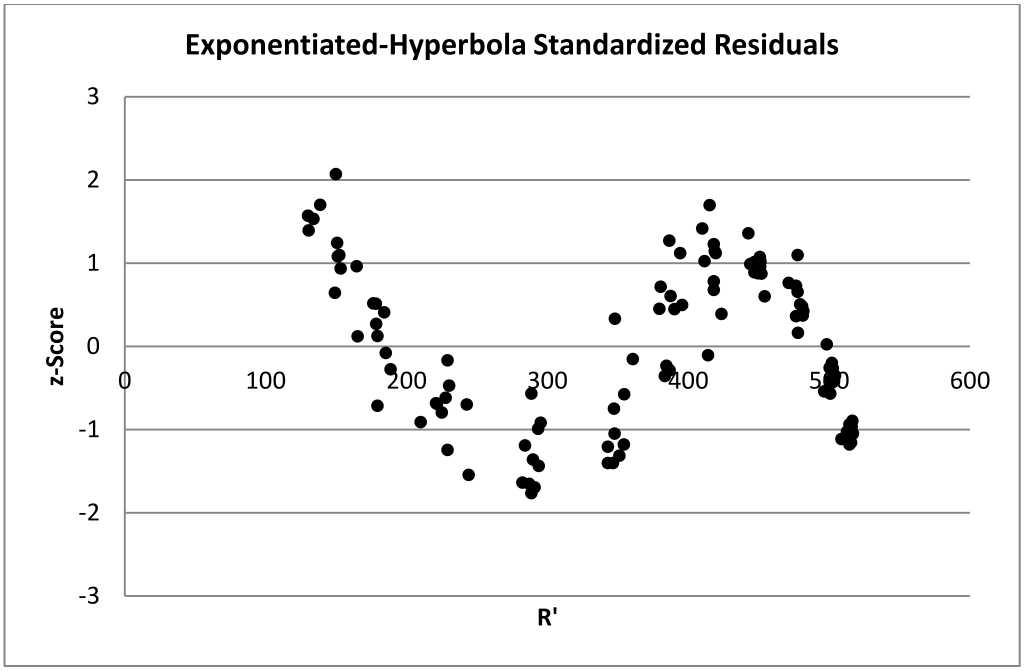


Figure 5. Plot of the response rate (R) versus reinforcement rate (r) on RI schedules of three virtual oganisms. The unfilled points represent the behavior of a virtual organism using standard parameters. The x's represent the behavior or the prototype virtual organism used to seed the evolutionary algorithm. The filled points represent the behavior of the evolved virtual organism.

Virtual Org.	Hyperbola			Hyperbola-Exp				Proportion VAF			
	k	r_e	pVAF	k	r_e/b	a	pVAF	A Exp	A Pow	Log	Ramp
1	514	7.2	0.995	582	5.0	0.70	0.997	0.987	0.990	0.984	0.966
2	514	7.1	0.998	549	5.5	0.81	0.999	0.992	0.989	0.986	0.980
3	515	7.7	0.995	579	5.3	0.71	0.997	0.988	0.990	0.986	0.967
4	516	7.3	0.995	576	5.1	0.72	0.997	0.988	0.990	0.984	0.969
5	527	8.4	0.997	560	6.5	0.82	0.998	0.992	0.988	0.988	0.975
6	517	7.1	0.995	571	5.1	0.74	0.997	0.988	0.989	0.984	0.968
7	516	7.6	0.996	570	5.5	0.74	0.998	0.989	0.990	0.987	0.970
8	523	8.5	0.996	569	6.2	0.78	0.997	0.990	0.988	0.987	0.973
9	503	6.5	0.995	589	4.4	0.64	0.998	0.985	0.992	0.982	0.969
10	517	7.3	0.995	572	5.1	0.73	0.997	0.989	0.989	0.982	0.975
Median	516	7.3	0.995	572	5.2	0.73	0.997	0.989	0.989	0.985	0.971

Table 1. Parameters of the best fitting hyperbola and exponentiated-hyperbola and the proportion of variance accounted for (pVAF) by the hyperbola, exponentiated-hyperbola, asymptotic exponential (A Exp), asymptotic power (A Pow), logarithmic (Log), and ramp functions.



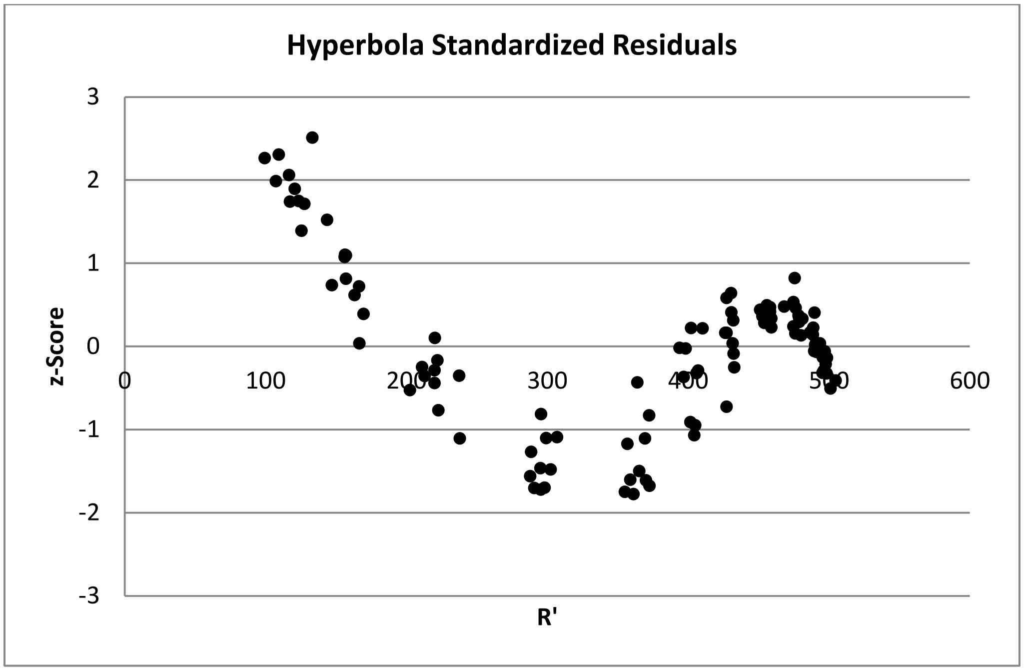


Figure 6. Standardized residuals of the exponentiated-hyperbola and hyperbola fits to the data across all 10 virtual organisms.

APPENDIX

Let *j* be a generic NPU as can be seen in Figure 1A. This NPU can be influenced by excitatory and inhibitory connections that are projected to it. The weight of each connection governs its efficiency of transmission to the NPU. Although listed for completeness inhibitory connections were not used in this experiment, because there has not been a published study that implemented them.

At a specific instance in time (*t*) the NPUs activation level is determined by the stimulation that is transmitted to it via connections and its previous activation level. The incoming stimulation is the inner dot product of the activation levels of the NPUs that project a connection to the NPU and the weights of the associated connection. This inhibitory or excitatory stimulation can be expressed as, $s = \sum_{i=1}^{n} a_{i,t} w_{i,j,t}$, where a is the activation level of a NPU that projects to *j* (*i*) at time *t*, *w* is the weight of the connection from NPUs *i* to *j* at time *t*, and n is the number of units that project to *j*. The logistic of the resulting excitatory and inhibitory stimulations is then calculated using the formula, $L(x) = 1/\left(1 + e^{-(s-\mu)/\sigma}\right)$. This logistic function ensures that the resulting inhibition and excitation values are in the range of 0 and 1. The resulting excitation and inhibition values are then used to determine whether or not the activation level of the NPU should increase (reactivation), decrease (decay), or be set to 0 (deactivation). The activation value of the NPU at *t* ($a_{j,t}$) is determined using the following equation,

$$a_{j,t} = \begin{cases} L(exc_{j,t}) + \tau L(exc_{j,t-1})\left[1 - L(exc_{j,t})\right] - L(inh_{j,t}), \\ \qquad if\ L(exc_{j,t}) > L(inh_{j,t})\ and\ L(exc_{j,t}) > \theta_t\ (reactivation); \\ a_{j,t-1} - \kappa a_{j,t-1}(1 - a_{j,t-1}), \\ \qquad if\ L(exc_{j,t}) > L(inh_{j,t})\ and\ L(exc_{j,t}) < \theta_t\ (decay); \\ 0, \quad if\ L(exc_{j,t}) < L(inh_{j,t})\ (deactivation). \end{cases} \tag{1A}$$

where τ is the temporal summation parameter, κ is the decay parameter, and θ_t is the Gaussian activation threshold. The Gaussian activation threshold, θ_t, is a randomly generated value that follows a Gaussian distribution with a mean, μ and standard deviation, σ. Equation 1A is overridden if the NPU receives activation greater than 0 from the US NPU, in which case $a_{j,t}$ is equal to that value.

Equation 2A shows the formula used to calculate the change in weights.

$$\Delta w_{i,j,t+1} = \begin{cases} Gain\colon \alpha(a_{j,t}d_t p_{i,t} r_{j,t}), if\ d_t \geq\ d_\theta; otherwise, \\ Loss\colon\ -\beta(w_{i,j,t}a_{i,t}a_{j,t}) \end{cases} \tag{2A}$$

where α is the rate of weight gain, β is the rate of weight loss, d_θ is the threshold between weight gain and loss. The other rules of the term are defined as follows:

$$d_t = \begin{cases} d_{Dop} =\ \ a_{j,t} - a_{j,t-1}, if\ the\ connection\ is\ altered\ by\ the\ VTA\ NPU \\ d_{Hip} =\ \left|a_{j,t} - a_{j,t-1}\right| + d_{Dop,t}(1 -\ \left|a_{j,t} - a_{j,t-1}\right|a_{j,t-1}, if\ by\ the\ CA1\ NPU \end{cases};$$

$p_{i,t} = a_{i,t}w_{i,j,t}\ /stim$, where $s = exc_{j,i}$ if i is excitatory, or $N = inh_{j,i}$ if i is inhibitory;

$r_{j,t} = 1 -\ \sum_{i=1}^{n} w_{i,j,t}$, where n is the total number of units connected to j and $w_{i,j,t}$ is the weight of the connection from i to j at t.

Appendix Figure

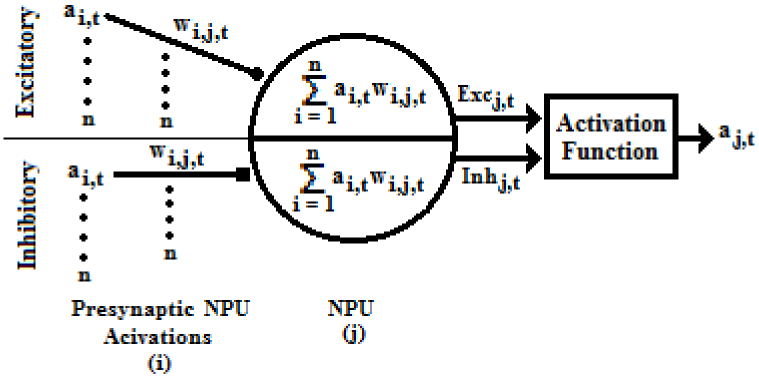


Figure 1A. A generic neural processing unit (NPU). The NPU (*j*) receives stimulation from NPUs that project to it (*i*). These projections can be either excitatory or inhibitory. The efficiency of these connections (weight) is indicated by $w_{i,j,t}$. The total inhibitory and excitatory stimulations are calculated separately by taking the inner dot product of the activations and connection weights of the NPUs that project to NPU *j*. The resulting total excitation ($Exc_{j,t}$) and inhibition ($Inh_{j,t}$) of NPU unit *j* at time *t* are then passed to the activation function (Equation 1A). Based on the values of $Exc_{j,t}$ and $Inh_{j,t}$ the NPU's activation level will either increase (reactivation), decrease (decay), or be set to 0 (deactivation).