Simple Artificial Neural Networks That Match Probability and Exploit and Explore When Confronting a Multiarmed Bandit

Michael R. W. Dawson, Brian Dupuis, Marcia L. Spetch, and Debbie M. Kelly

***Abstract*—The matching law (Herrnstein 1961) states that response rates become proportional to reinforcement rates; this is related to the empirical phenomenon called probability matching (Vulkan 2000). Here, we show that a simple artificial neural network generates responses consistent with probability matching. This behavior was then used to create an operant procedure for network learning. We use the multiarmed bandit (Gittins 1989), a classic problem of choice behavior, to illustrate that operant training balances exploiting the bandit arm expected to pay off most frequently with exploring other arms. Perceptrons provide a medium for relating results from neural networks, genetic algorithms, animal learning, contingency theory, reinforcement learning, and theories of choice.**

***Index Terms*—Instrumental learning, multiarmed bandit, operant conditioning, perceptron, probability matching.**

I. Introduction

The matching law states that the rate of a response reflects the rate of its obtained reinforcement: if response A is reinforced twice as frequently as response B, then A will appear twice as frequently as B [1]. While modern variations exist [4], the matching law is usually expressed as $R = kr/(r + re)$, where R is response rate, k and re are parameters, and r is reinforcement rate. Intended, to explain response frequency, the matching law also predicts how response strength varies with reinforcement frequency [5]. The matching law is a foundational regularity, applying to many tasks in psychology and economics [6]–[8]. An empirical phenomenon that is formally related [9] to the matching law is probability matching, in which the probability that an agent makes a choice among alternatives mirrors the probability associated with the outcome or reward of that choice [2]. This brief investigates whether a simple artificial neural network can vary response strengths in accordance with such probability matching.

A perceptron [10], [11] is a simple artificial neural network whose input units send signals about detected stimuli through weighted connections to an output unit, which converts them into a response ranging from 0 to 1 using a nonlinear activation function. Modern perceptrons typically use the logistic equation $aj = 1/(1 + e - netj)$, where aj is the activity of output unit j, and $netj$ is the incoming signal. Perceptrons can be trained to produce desired responses to stimuli with a gradient-descent learning rule [12] that modifies network weights using response error scaled by the derivative of the activation function. Perceptrons can simulate a large number of classic results in the learning literature [13]. Given the ubiquity and importance of the probability matching, is it possible that perceptrons can exhibit such matching as well? Our first simulation attempted to answer this question.

Manuscript received January 05, 2009; accepted June 09, 2009. First published July 10, 2009; current version published August 05, 2009. The work of work M. R. W. Dawson, M. L. Spetch, and D. M. Kelly was supported by NSERC Discovery Grants.

M. R. W. Dawson, B. Dupuis, and M. L. Spetch are with the Department of Psychology, University of Alberta, Edmonton, AB T6G 2P9 Canada (e-mail: mdawson@ualberta.ca; bdupuis@ualberta.ca; mspetch@ualberta.ca).

D. M. Kelly is with the Department of Psychology, University of Saskatchewan, Saskatoon, SK S7N 5A5 Canada (e-mail: debbie.kelly@usask.ca).

Digital Object Identifier 10.1109/TNN.2009.2025588

II. Simulation 1: Matching Differential Reinforcement Probabilities

A. Method

1) Network Architecture and Training Set: Perceptrons, with a single output unit and four input units, were trained. The input units were turned on or off to represent the presence or absence of four different discriminative stimuli (DSs). For example, the input pattern $[1\ 0\ 0\ 0]$ indicated that DS1 was present and that all other DSs were absent.

Each DS was reinforced at different frequencies. For the first 300 epochs of training, DS1 was reinforced on 20% of its presentations while DS2, DS3, and DS4 received 40%, 60%, and 80% reinforcement, respectively. For the second 300 epochs of training, reinforcement frequencies were reversed, so that DS1 was reinforced 80% of its presentations while DS2, DS3, and DS4 received 60%, 40%, and 20% reinforcement, respectively.

Reinforcement probabilities were manipulated by repeating the pattern that coded the presence of one of the DSs ten times, building a total training set of 40 input patterns. Each pattern was reinforced (or not) by being paired with a desired network output value of either 1 or 0. Differential probabilities of reinforcement were produced by varying the number of positive reinforcements applied to a DS's set of ten input patterns. For example, setting the desired response to DS1 to 1 for two of its input patterns, and to 0 for its remaining eight patterns, produced a 20% reinforcement probability.

2) Network Training: Ten different perceptrons were trained using the gradient-descent rule with a learning rate of 0.1, and with connection weights randomly set in the range from -0.1 to 0.1 prior to training. Training was accomplished with the Rosenblatt program that is available as freeware [14]. During an epoch of training, a network was presented each of the 40 patterns; connection weights were modified after each presentation. The order of DS presentations was randomized in each epoch. Network responses to each DS were recorded every 20 epochs of training. After 300 epochs of training, the reinforcement contingencies associated with the DSs were inverted without reinitializing connection weights. Training continued for an additional 300 epochs.

B. Results

The results, presented in Fig. 1(a), indicated that perceptrons matched response strength to reinforcement probabilities, and quickly adjusted their behavior when reinforcement probabilities were altered. After 60 epochs, the perceptrons generated output activity that equalled probability of reinforcement for each DS (e.g., generating activity of 0.20 to DS1). When reinforcement probabilities were changed, the perceptrons adjusted and again matched their responses to the new reinforcement contingencies within 60 epochs.

A number of experiments have studied probability matching under conditions in which reinforcement probabilities are changed or reversed at the midpoint of the study; subjects in these experiments have included insects [15]–[18], fish [19], turtles [20], pigeons [21], and humans [22]. The simulation results reported in Fig. 1 are very similar to the results obtained in these experiments. For example, in their classic study of probability matching in the fish [19], Behrend and Bitterman found that their subjects quickly matched their choice preference of two alternatives to the probability of reinforcement of the two. When reinforcement probabilities were altered, the animals quickly altered their choice of behavior to reflect the new contingencies. Behrend and Bitterman's graph of this choice of behavior over time [19, Fig. 2] is strikingly similar in shape to the curves illustrated in Fig. 1(a).

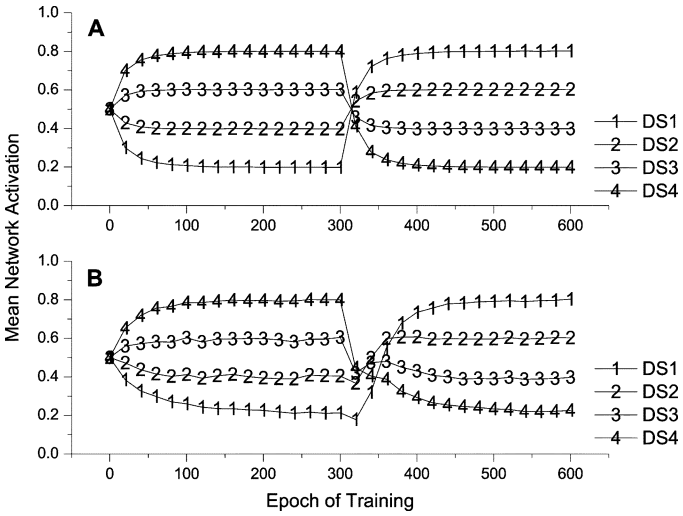

Fig. 1. Average responses of ten different perceptrons to each of the four stimuli as a function of training epoch. (a) Responses for networks from the first simulation which used standard training procedures. (b) Responses for networks from the second simulation which used an operant training procedure.

These results are comparable to other algorithms in the machine learning literature that do not involve artificial neural networks. For example, at trial t, the pursuit algorithm [23] updates the expected probability of source i delivering reinforcement using the equation $Pt + 1(i) = Pt(i) + \alpha \cdot (\lambda) - Pt(i))$, where λ is equal to 1 if reinforcement is received and 0 otherwise, and α is a constant. When α equals 0.005 and the pursuit algorithm is given the identical training set used with the perceptrons, the results were indistinguishable from Fig. 1(a); the total sum of squared differences between the 120 points plotted in Fig. 1(a) and the same data from the pursuit algorithm was 0.005. In short, the matching behavior of the perceptrons was identical to that obtained using a standard machine learning algorithm.

III. Simulation 2: Operant Learning of the Multiarmed Bandit

The training set used above is similar to the classic multiarmed bandit problem [3]. In this problem, an agent is in a room with k different "one-armed bandit" gambling machines. When a machine's arm is pulled, the machine pays either 1 or 0 units; each machine has a different (and usually fixed) probability of paying off, which is not known by the agent. At each time t, the agent pulls the arm of one machine. The goal is to choose machines to maximize the total payoff over the game's duration. To do this, the agent must explore the different machines to determine payoff probabilities. The agent must also exploit the results of this exploration in order to maximize reward. As a result, there is a tradeoff between exploration and exploitation that must be balanced [24]. A "greedy strategy" only pulls the arm of the machine with the highest expected payoff probability. However, as the duration of the game increases, an alternative strategy would be to explore other machines as well, in case early probability estimates were inaccurate.

Simulation 1 was not identical to the multiarmed bandit problem because there was no balance between exploitation and exploration: every time a "machine" was presented to a perceptron, it "pulled the machine's arm" and learned new information. This can be changed using the knowledge gleaned from Simulation 1 that network responses estimate reward likelihood.

Rather than modifying weights on every DS presentation, one can implement operant network learning as follows. On every trial, compute the network's response to a presented stimulus. The magnitude of this response is the network's current estimate of reward likelihood. This response is used as the probability of updating weights (i.e., of learning on the trial). That is, connections weights are not always modified, as was the case in Simulation 1. Sometimes they will be changed, but other times they will remain the same. As training proceeds, connection weights will be updated more frequently for those DSs associated with a higher frequency of reinforcement than for those receiving less reinforcement. Therefore, a perceptron trained (operantly) on this problem would be functionally equivalent to an agent playing a multiarmed bandit. Operant learning also imposes a simple balance between exploitation and exploration, because the perceptron will occasionally modify connection weights using a DS associated with a low (but nonzero) estimated reinforcement contingency.

A. Method

The method for Simulation 2 was identical to that used in Simulation 1, except that output unit activity was used as a probability to determine whether connection weights were modified. This was accomplished as follows. After the network's response to a pattern was calculated, a random number between 0 and 1 was generated. If this number was less than or equal to the network's response, then the learning rule was used to update all of the connection weights. Otherwise, the connection weights were not changed, and the next pattern was presented to the network.

It might be argued that this method is a major departure from Simulation 1, in the sense that an external controller is generating the random number that is used, in conjunction with output unit activity, to determine whether a particular trial will involve learning. From this perspective, the perceptron itself is incapable of operant learning, because it requires this external control. However, while it is possible to elaborate artificial neural network architectures to build learning rules directly into them, it is almost always the case that these learning rules exist as controllers that are external to the network [25]. Thus, in our view, Simulation 2 uses a slightly elaborated learning rule that is no more external to the perceptron than was the learning rule employed in Simulation 1, or than any learning rule that is typically used to train artificial neural networks.

B. Results

Fig. 1(b) illustrates the results of using this operant procedure to train perceptrons on exactly the same task used in Simulation 1. The results were qualitatively very similar to the results in Fig. 1(a): perceptrons quickly adjusted responses to match reinforcement contingency, and then quickly readapted when reinforcement contingencies were inverted. One difference in results was that the operant networks were slightly slower at achieving matching behavior. Also, when reinforcement contingencies changed, operant networks adapted to DS_4 earlier than the other DSs, because, at epoch 300, operant training was changing weights on the basis of DS_4 information more frequently than on the basis of the other DSs. Quantitatively, the total sum of squared differences between the 120 data points used to create Fig. 1(a) and the corresponding data points in Fig. 1(b) was 0.557.

IV. General Discussion

In summary, we have shown that perceptrons generate responses that accord with probability matching, and that this can be used to create an operant training paradigm for these networks. There are three main implications of these results.

First, formal accounts of matching are typically stated in terms of observables (rates of reinforcement and responses) and not by appealing to underlying mechanisms [6], [8]. Because the matching law can emerge from modifying a perceptron's connection weights, it

might be explained by appealing to general mechanisms of associative learning.

Other mechanistic accounts of matching have been proposed. McDowell *et al.* have developed a genetic algorithm that evolves a population of behaviors over time, skewing the distribution of possible behaviors in the direction of those that have been reinforced [26]–[29]. The matching law is an emergent property of this selectionist account of adaptive behavior. Such selectionist accounts are usually taken as radical alternatives to instructionist theories such as artificial neural networks [30]. However, it is possible to create neural networks that are consistent with selectionist theory [31], [32]. Clearly, one area deserving future research is an examination of the computational and algorithmic similarities and differences between selectionist and instructionist mechanisms that are capable of producing matching behavior.

A second implication of our results is a response to Herrnstein's position [8, p. 68] that the matching law can be distinguished from other theories of learning, such as the Rescorla–Wagner model [33]. That perceptrons produce probability matching indicates that this distinctness needs to be reevaluated. Formal equivalences between perceptron learning and the Rescorla–Wagner model [34] and contingency theory [35] have already been established. Genetic algorithms that produce matching behavior reach the same equilibria as the Rescorla–Wagner model [29], [36]. Matching behavior would be expected from these approaches to learning, as well as from networks adapted via reinforcement learning [37] Perceptrons might mediate a formal account of the relationships between neural networks, these important theories of learning, and the matching law and probability matching.

A third implication of our results concerns further explorations of matching behavior in perceptrons, particularly with the goal of using perceptrons to emulate animal behaviors that have been used to study the matching law. The perceptrons reported here represent idealized systems that demonstrated probability matching. They were trained in a situation in which they distinguished between four discriminative stimuli, and in which they were essentially trained using a random-ratio reinforcement schedule. Animal subjects produce behavior in accord with Herrnstein's matching law when trained under different reinforcement schedules (in particular, concurrent interval schedules) [6]. Furthermore, under a variety of conditions, animals produce systematic deviations from the strict matching law [6], [38] including undermatching, where they respond less frequently to a DS than the matching law would predict, and bias, where they have a stronger preference for a DS than the matching law would predict. Importantly, the goal of this brief was not to emulate extant data in the animal literature, but instead to determine whether perceptrons were capable of probability matching. Given that our results indicate that perceptrons can match probabilities, this suggests a future line of research in which constraints on network architecture, learning procedures, and learning rules can be explored in an explicit attempt to emulate the complexity of the data that is to be found in the experimental literature on matching.

A fourth implication of our results is that the ability of perceptrons to simulate multiarmed bandit problems might serve to link statistical theories of choice [3], and models of reinforcement learning [24], [37], to other theories of learning, including artificial neural networks and standard associative models. Theories of multiarmed bandits usually view each machine as a unique whole. However, machines could be viewed as feature collections, with the machine's reinforcement estimate based upon the sum of the estimates associated with each feature, and not upon the (whole) machine itself. The reinforcement estimate associated with a feature can depend on the payoff of several machines, because a feature may be shared by more than one machine.

When cast in this way, the multiarmed bandit can be related to other learning problems, such as the reorientation task used to study spatial representations [39]. In the reorientation task, an agent explores different locations, each describable as a feature set, with some features present at multiple locations. Not all locations are rewarded, and the agent must use feature sets to learn where rewards might be placed. Perceptrons thus provide an opportunity to explore a featural elaboration of the multiarmed bandit, relating it to a broader set of learning paradigms than has been previously considered.

This featural elaboration of choice tasks such as the multiarmed bandit is also crucial to comparing perceptrons to other models, such as genetic algorithms [26]. In the perceptron, there is a very limited behavioral repertoire (i.e., choose or not choose), but behavior can be in principle selected by a potentially huge variety of stimuli. In contrast, the genetic algorithm has an enormous variety of behaviors to select from, but these are selected randomly without considering stimulus properties. The fact that these two different approaches produce matching is very interesting, and raises the possibility that they represent complementary mechanisms for generating such behavior as probability matching.

REFERENCES

[1] R. J. Herrnstein, "Relative and absolute strength of response as a function of frequency of reinforcement," *J. Exp. Anal. Behav.*, vol. 4, pp. 267–272, 1961.

[2] N. Vulkan, "An economist's perspective on probability matching," *J. Econom. Surv.*, vol. 14, pp. 101–118, Feb. 2000.

[3] J. C. Gittins, *Multi-Armed Bandit Allocation Indices*. Chichester, U.K.: Wiley, 1989.

[4] J. J. McDowell, "On the classic and modern theories of matching," *J. Exp. Anal. Behav.*, vol. 84, pp. 111–127, Jul. 2005.

[5] P. de Villiers and R. J. Herrnstein, "Toward a law of response strength," *Psychol. Bull.*, vol. 83, pp. 1131–1153, 1976.

[6] M. Davison and D. McCarthy, *The Matching Law : A Research Review*. Hillsdale, N.J.: L. Erlbaum, 1988.

[7] P. de Villiers, "Choice in concurrent schedules and a quantitative formulation of the law of effect," in *Handbook of Operant Behavior*, W. K. Honig and J. E. R. Staddon, Eds. Englewood Cliffs, NJ: Prentice-Hall, 1977, pp. 233–287.

[8] R. J. Herrnstein, *The Matching Law : Papers in Psychology and Economics*. New York: Harvard Univ. Press, 1997.

[9] R. J. Herrnstein and D. H. Loveland, "Maximizing and matching on concurrent ratio schedules," *J. Exp. Anal. Behav.*, vol. 24, pp. 107–116, 1975.

[10] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, pp. 386–408, 1958.

[11] F. Rosenblatt, *Principles of Neurodynamics*. Washington, DC: Spartan Books, 1962.

[12] M. R. W. Dawson, *Minds and Machines : Connectionism and Psychological Modeling*. Malden, MA: Blackwell, 2004.

[13] M. R. W. Dawson, "Connectionism and classical conditioning," *Comparat. Cogn. Behav. Rev.*, vol. 3, Monograph, pp. 1–115, 2008.

[14] M. R. W. Dawson, *Connectionism : A Hands-on Approach*, 1st ed. Malden, MA: Blackwell, 2005.

[15] M. E. Fischer, P. A. Couvillon, and M. E. Bitterman, "Choice in honeybees as a function of the probability of reward," *Animal Learn. Behav.*, vol. 21, pp. 187–195, Aug. 1993.

[16] T. Keasar, E. Rashkovich, D. Cohen, and A. Shmida, "Bees in two-armed bandit situations: Foraging choices and possible decision mechanisms," *Behav. Ecol.*, vol. 13, pp. 757–765, Nov.-Dec. 2002.

[17] N. Longo, "Probability-learning and habit-reversal in the cockroach," *Amer. J. Psychol.*, vol. 77, pp. 29–41, 1964.

[18] Y. Niv, D. Joel, I. Meilijson, and E. Ruppin, "Evolution of reinforcement learning in uncertain environments: A simple explanation for complex foraging behaviors," *Adapt. Behav.*, vol. 10, pp. 5–24, 2002.

[19] E. R. Behrend and M. E. Bitterman, "Probability-matching in the fish," *Amer. J. Psychol.*, vol. 74, pp. 542–551, 1961.

[20] K. L. Kirk and M. E. Bitterman, "Probability-learning by the turtle," *Science*, vol. 148, pp. 1484–1485, 1965.

[21] V. Graf, D. H. Bullock, and M. E. Bitterman, "Further experiments on probability-matching in the pigeon," *J. Exp. Anal. Behav.*, vol. 7, pp. 151–157, 1964.

[22] W. K. Estes and J. H. Straughan, "Analysis of a verbal conditioning situation in terms of statistical learning theory," *J. Exp. Psychol.*, vol. 47, pp. 225–234, 1954.

[23] M. A. L. Thathachar and P. S. Sastry, "A new approach to the design of reinforcement schemes for learning automata," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-15, no. 1, pp. 168–175, Feb. 1985.
[24] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, pp. 237–285, 1996.
[25] M. R. W. Dawson and D. P. Schopflocher, "Autonomous processing in PDP networks," *Philosoph. Psychol.*, vol. 5, pp. 199–219, 1992.
[26] J. J. McDowell, "Computational model of selection by consequences," *J. Exp. Anal. Behav.*, vol. 81, pp. 297–317, May 2004.
[27] J. J. McDowell and Z. Ansari, "The quantitative law of effect is a robust emergent property of an evolutionary algorithm for reinforcement learning," in *Advances in Artificial Life*. Cambridge, MA: MIT Press, 2005, vol. 3630, pp. 413–422.
[28] J. J. McDowell and M. L. Caron, "Undermatching is an emergent property of selection by consequences," *Behav. Processes*, vol. 75, pp. 97–106, Jun. 2007.
[29] J. J. McDowell, P. L. Soto, J. Dallery, and S. Kulubekova, M. Keijzer, Ed., "A computational theory of adaptive behavior based on an evolutionary reinforcement mechanism," in *Proc. Conf. Genetic Evol. Comput.*, New York, 2006, pp. 175–182.
[30] M. Piattelli-Palmarini, "Evolution, selection and cognition: From "learning" to parameter setting in biology and in the study of language," *Cognition*, vol. 31, pp. 1–44, 1989.
[31] J. W. Donahoe and D. C. Palmer, *Learning and Complex Behavior*. Boston, MA: Allyn and Bacon, 1994.
[32] R. B. T. Lowry and M. R. W. Dawson, "Connectionist selectionism: A case study of parity," *Neural Inf. Process.—Lett. Rev.*, vol. 9, pp. 59–67, 2005.
[33] R. J. Herrnstein, "Derivatives of matching," *Psychol. Rev.*, vol. 86, pp. 486–495, 1979.
[34] R. S. Sutton and A. G. Barto, "Toward a modern theory of adaptive networks: Expectation and prediction," *Psychol. Rev.*, vol. 88, pp. 135–170, 1981.
[35] D. R. Shanks, *The Psychology of Associative Learning*. Cambridge, U.K.: Cambridge Univ. Press, 1995.
[36] D. Danks, "Equilibria of the Rescorla-Wagner model," *J. Math. Psychol.*, vol. 47, pp. 109–121, Apr. 2003.
[37] R. S. Sutton and A. G. Barto, *Reinforcement Learning*. Cambridge, MA: MIT Press, 1998.
[38] W. M. Baum, "On two types of deviation from the matching law: Bias and undermatching," *J. Exp. Anal. Behav.*, vol. 22, pp. 231–242, 1974.
[39] K. Cheng and N. S. Newcombe, "Is there a geometric module for spatial orientation? Squaring theory and evidence," *Psychonom. Bull. Rev.*, vol. 12, pp. 1–23, Feb. 2005.