Published in final edited form as:

Neural Netw. 2008 December ; 21(10): 1493–1499. doi:10.1016/j.neunet.2008.09.004.

The Emergence of Saliency and Novelty Responses from Reinforcement Learning Principles

Patryk A. Laurent

University of Pittsburgh

Abstract

Recent attempts to map reward-based learning models, like Reinforcement Learning [17], to the brain are based on the observation that phasic increases and decreases in the spiking of dopamine-releasing neurons signal differences between predicted and received reward [16,5]. However, this *reward-prediction error* is only one of several signals communicated by that phasic activity; another involves an increase in dopaminergic spiking, reflecting the appearance of salient but unpredicted non-reward stimuli [4,6,13], especially when an organism subsequently orients towards the stimulus [16]. To explain these findings, Kakade and Dayan [7] and others have posited that novel, unexpected stimuli are intrinsically rewarding. The simulation reported in this article demonstrates that this assumption is not necessary because the effect it is intended to capture emerges from the reward-prediction learning mechanisms of Reinforcement Learning. Thus, Reinforcement Learning principles can be used to understand not just reward-related activity of the dopaminergic neurons of the basal ganglia, but also some of their apparently non-reward-related activity.

Reinforcement Learning (RL) is becoming increasingly important in the development of computational models of reward-based learning in the brain. RL is a class of computational algorithms that specifies how an artificial “agent” (e.g., a real or simulated robot) can learn to select actions in order to maximize total expected reward [17]. In these algorithms, an agent bases its actions on *values* that it learns to associate with various *states* (e.g., the perceptual cues associated with a stimulus). These values can be gradually learned through *temporal-difference learning*, which adjusts state values based on the difference between the agent’s existing reward prediction for the state and the actual reward that is subsequently obtained from the environment. This computed difference, termed *reward-prediction error*, has been shown to correlate very well with the phasic activity of dopamine-releasing neurons projecting from the substantia nigra in non-human primates [16]. Furthermore, in humans, the striatum, which is an important target of dopamine, exhibits an fMRI BOLD signal that appears to reflect reward-prediction error during reward-learning tasks [10,12,18]. This fMRI finding complements the physiology data because striatal BOLD is assumed to reflect, at least in part, afferent synaptic activity [9] and the dopamine neurons project heavily to the striatum.

Although the aforementioned physiological responses appear to be related to the reward-prediction computations of RL, there is also an increase in dopaminergic phasic activity in

Address all correspondence to: Patryk Laurent, University of Pittsburgh, 623 LRDC, 3939 O’Hara St., Pittsburgh, PA 15260 USA, E-mail: patryk@cnbc.cmu.edu, Office: (412) 624-3191, Fax: (412) 624-9149. Correspondence regarding this article should be addressed to Patryk Laurent, 623 LRDC, 3939 O’Hara St., Pittsburgh, PA, 15260; or via e-mail to: patryk@cnbc.cmu.edu.

Publisher’s Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

response to arousing and/or novel stimuli that is seemingly unrelated to reward [4,6,14,3]. A similar phenomenon has been recently observed in humans using fMRI [2]. There are several reasons why this “novelty” or “saliency” response is said to be unrelated to reward-prediction error: (1) it appears very early, before the identity of the stimulus has been assessed, so that an accurate reward prediction cannot be generated; (2) it corresponds to an *increase* in neural activity (i.e., it is positive) for both aversive and appetitive stimuli; and (3) it habituates [13]. Indeed, these saliency/novelty responses of the dopamine-releasing neurons are most reliable when the stimuli are unpredicted and result in orienting and/or approach behavior [16] regardless of the eventual outcome, highlighting the fact that they are qualitatively different from learned reward prediction. The challenge, therefore, has been to explain this apparent paradox (i.e., how novelty affects the reward-prediction error) within the theoretical framework of RL.

Kakade and Dayan [7] attempted to do exactly this; in their article, they postulate two ways in which novelty responses could be incorporated into RL models of dopaminergic function—both involved the inclusion of new theoretical assumptions. The first assumption, referred to as *novelty bonuses*, involves introducing an additional reward when novel stimuli are present, above and beyond the usual reward received by the agent. This additional reward enters into the computation so that learning is based on the difference between the agent’s existing reward prediction and the sum of both the usual reward from the environment and the novelty bonus. Thus, the novelty becomes part of the reward that the agent is attempting to maximize. The second assumption, termed *shaping bonuses*, can be implemented by artificially increasing the values of states associated with novel stimuli. Because the temporal-difference learning rule used in RL is based on the difference in reward-prediction between successive states, the addition of a constant shaping bonus to states concerned with the novel stimuli has no effect on the final behavior of the agent. However, a novelty response still emerges when the agent enters the part of the state space that has been “shaped” (i.e., that is associated with novelty).

Although the addition of each of these assumptions is sufficient to explain many observed effects of novelty, the assumptions also interfere with the progression of learning. As Kakade and Dayan [7] point out, novelty bonuses can distort the value function (i.e., the values associated with each state by the agent) and affect what is ultimately learned because they are implemented as an additional reward that is intrinsically associated with novel states. The problem is that the agent learns to predict both the primary and novelty components of the reward. Although Kakade and Dayan point out that shaping bonuses do not cause this type of problem because they become incorporated into the reward predictions from preceding states, their addition is still problematic because shaping bonuses introduce biases into the way an agent will explore its state space. Thus, although these additional assumptions may explain how novelty affects the reward-prediction error in RL, they are problematic. Further, the explanations come at the cost of reducing the parsimony of modeling work that attempts to use RL to understand the behavior of real biological organisms.

The simulation reported below was carried out in order to test the hypothesis that a simple RL agent, without any additional assumptions, would develop a reward-prediction error response that is similar to the non-reward-related dopamine responses that are observed in biological organisms. An RL agent was given the task of interacting with two types of object—one positive and the other negative—that appeared at random locations in its environment. In order to maximize its reward, the agent had to learn to approach and “consume” the positive object, and to avoid (i.e., not “consume”) the negative object. There were three main predictions for the simulation.

The first prediction was simply that, in order to maximize its reward, the agent would in fact learn to approach and “consume” the positive, rewarding objects while simultaneously learning

to avoid the negative, punishing objects. The second prediction was slightly less obvious: that the agent would exhibit an orienting response (i.e., learn to shift its orientation) towards both negative and positive objects. This prediction was made because although the agent could “sense” the appearance of an object and its location, the positive or negative identity of the object (i.e., the cue that the agent would eventually learn to associate with the reward value of the object) could not be determined by the agent until *after* the agent had actually oriented towards the object. Finally, the third (and most important) prediction was related to the simulated dopaminergic phasic response in the model; this prediction was that, when the object appeared, the agent would exhibit a reward-prediction error that was computationally analogous to the phasic dopamine response observed in biological organisms, being positive for both positive *and* negative objects. This response was also predicted to vary as a function of the distance between the agent and the stimulus, which in the context of the simulation was a proxy measure for stimulus “intensity” or salience. As will be demonstrated below, these predictions were confirmed by the simulation results, demonstrating that the apparently non-reward-related dopamine responses can in principle emerge from the basic principles of RL. The theoretical implications of these results for using RL to understand non-reward-related activity in biological organisms will be discussed in the final section of this article.

Method

As already mentioned, RL algorithms specify how an agent can use moment-to-moment numerical rewards to learn which actions it should take in order to maximize the total amount of reward that it receives. In most formulations, this learning is achieved by using reward-prediction errors (i.e., the difference between an agent’s current reward prediction and the actual reward that is obtained) to update the agent’s reward predictions. As the reward predictions are learned, the predictions can also be used by an agent to select its next action. The usual *policy* (defined in Equation 2) is for the agent to select the action that is predicted to result in the largest reward. The actual reward that is provided to the agent at any given time is the sum of the immediate reward plus some portion of the value of the state that the agent enters when the action is completed. Thus, if the agent eventually experiences positive rewards after having been in a particular state, the agent will select actions in the future that are likely to result in those rewarded states; conversely, if the agent experiences negative rewards (i.e., punishment) it will avoid actions in the future that lead to those “punished” states.

The specific algorithm that determines the reward predictions that are learned for the various states (i.e., the *value function* V) is called *Value Iteration* [Footnote 1] and can be formally described as:

¹Another Reinforcement Learning algorithm, called *Trajectory Sampling* [17], is frequently used instead of Value Iteration when the state space becomes so large that it cannot be exhaustively iterated or easily stored in a computer’s memory. Rather than iterating over every state in the state space and applying the value function update equation based on the actions that appear to lead to the most reward, Trajectory Sampling works by following paths through the state space. Similarly to Value Iteration, the actions leading to the most reward are usually selected from each state, but occasionally a random exploratory action is chosen with some small probability. Thus the algorithm is: *From some starting state s , select an action leading to the most reward [e.g., $\text{reward} + \gamma V(s')$] with probability ϵ , or select a random exploratory action with probability $1 - \epsilon$. Apply $V(s) \rightarrow V(s) + \alpha[\text{reward} + \gamma V(s') - V(s)]$ during non-exploratory actions from state s .*

Besides overcoming the technical limitations of computational time and memory, Trajectory Sampling may be appealing because it may better reflect the manner in which real biological organisms learn: by exploring paths in a state space. On the task described in this paper, Trajectory Sampling yields results that are qualitatively identical to those obtained with Value Iteration. However, for conciseness those results are not reported here in detail. Value Iteration was selected for the simulation in this paper for two main reasons. First, because Trajectory Sampling involves stochasticity in the selection of trajectories, the large amount of branching that is due to the many possible sequences of actions in this task may result in agents that lack experience with some states unless the exploration-exploitation parameter (i.e., ϵ -greediness [17]) is carefully selected. This lack of experience with particular states can be disruptive of an agent’s performance when a lookup table memory structure is used because of the lack of generalization of value to similar (but possibly unvisited) states. Thus, it was preferred to take advantage of the exhaustive exploration of state space that is guaranteed with Value Iteration. Second, the use of Value Iteration obviated the need to specify that additional exploration-exploitation parameter, thereby simplifying the simulation. Note that Trajectory Sampling can ultimately approximate Value Iteration as the number of trajectories approaches infinity [17].

For all possible states s ,

$$V(s) \leftarrow V(s) + \alpha [\max_{\text{action} \in M} \{\text{reward} + \gamma V(s')\} - V(s)] \quad (\text{Equation 1})$$

where s corresponds to the current state, $V(s)$ is the current reward prediction for state s that has been learned by the agent, $\max_{\text{action} \in M} \{ \}$ is an operator for the maximum value of the bracketed quantity over the set of all actions M available to the agent, $V(s')$ is the agent's current reward prediction for the next state s' , α is some learning rate (between 0 and 1), and γ is a discount factor reflecting how future rewards are to be weighted relative to immediate rewards. The initial value function was set so that $V(s)$ was 0 for all states s .

The value function $V(s)$ was implemented as a lookup table, which is formally equivalent to the assumption of perfect memory. Although function approximators such as neural networks have been used with some success to represent value functions [1], a lookup table was used to ensure that the results were not dependent on the types of generalization mechanism that are provided by various function approximators. The agent was trained for 1,500 learning iterations over its state space. Because of the unpredictability of the identity of the objects, a value function update parameter of less than one ($\alpha = 0.01$) was used during the learning to allow for averaging of different outcomes. Finally, the discount factor was set to $\gamma = 0.99$ to encourage the agent to seek reward sooner rather than delay its approach behavior until the end of the trial (although changing it from a default value of 1 had no effect on the results reported here.) In order to independently determine whether 1,500 learning iterations were sufficient for learning to complete, the average amount of change in the learned was monitored and was found to have converged before this number of iterations.

After training, the specific algorithm that governs the agent's behavior (i.e., the *policy* of actions that it takes from each given state) is:

$$\pi(s) = \arg\max_{\text{action} \in M} [\text{reinforcement} + \gamma V(s')] \quad (\text{Equation 2})$$

where $\pi(s)$ is the action the agent will select from state s , and the right side of the equation returns the action (e.g., change of orientation, movement, or no action) which maximizes the sum of the reward and the discounted value of the resulting state s' .

In the simulation that are reported below, all of the states that were visited by the agent were encoded as 7-dimensional vectors that represented information about both the external "physical" state of the agent and its internal "knowledge" state. The physical information included both the agent's current position in space and its orientation. The knowledge information included the position of the object (if one was present) and the identity of that object (if it had been determined by the agent). The specific types of information that were represented by the agent are shown in Table 1.

There were a total of 21,120 states in the simulation [Footnote 2]. However, the states in which there was an unidentified positive and unidentified negative object are, from the perspective of the agent, identical, so there are therefore only 16,280 distinct states. Thus, during each iteration of learning, it was necessary to visit some of those "identical" states twice to allow for the fact that half of the time they might be followed with the discovery of a positive object, and half of the time they might be followed with the discovery of a negative object [Footnote 3].

²The number of 21,120 states can be calculated as follows: 11 possible agent locations $\times$ 4 possible agent orientations $\times$ (10 time-steps before an object might appear + 10 time-steps where no object appeared + 10 time-steps where the agent had been positively reinforced + 10 time-steps where the object had been negatively reinforced + 11 possible object locations $\times$ (10 time-steps with a positive identified object + 10 time-steps with a negative identified object + 10 time-steps with an unidentified positive object + 10 time-steps with an unidentified negative object))).

At the beginning of each simulated testing trial, the agent was placed in the center of a simulated linear 11×1 unit track with five spaces to the “east” (i.e., to the right) of the agent and five spaces to the “west” (i.e., to the left) of the agent. As Table 1 shows, the agent’s state-vector included an element indicating its current location on the track (i.e., an integer from 0 to 10), as well as an element (i.e., a character “n”, “s”, “e”, or “w”) representing its current orientation (i.e., north, south, east, or west, respectively). The agent’s initial orientation was always set to be “north,” and no other object was present in the environment (i.e., the value of “OBJECT” in the agent’s state-vector was set to equal to “0”).

During each time-step of the simulation, the agent could perform one of the following actions: (1) do nothing, and remain in the current location and orientation; (2) orient to the north, south, east or west; or (3) move one space in the environment (east or west). The result of each action took place on the subsequent simulated time-step. All changes in the location and/or orientation of the agent in space occurred through the selection of actions by the agent. However, during every time-step of the simulation, even when a “do nothing” action was selected, time was incremented by 1 until the end of the trial (i.e., time-step 20).

The agent’s environment was set up so that half of the time, an object appeared at a random location (but not in the same location as the agent) after ten time steps; 50% of the objects were positive (represented by a “+”; see Table 1) and 50% of the objects were negative (represented by a “-”). The delay before the object appeared was introduced to allow the observation of any behavior the agent may have exhibited before the appearance of the object. If the agent was not oriented towards the object when it appeared, then the element representing the “OBJECT” identity in the agent’s state vector was changed from “0” to “?” to reflect the fact that the identity of the object that was now present was currently unknown. However, if the agent was oriented towards the object, then on the subsequent time-step the “OBJECT” element was set to equal to the identity of the object, so that “0” became either “+” or “-” for positive and negative objects, respectively.

If the agent moved to an object’s location, then during the next time step the object vanished. If the object had been positive, then the agent’s “CONSUMED” flag was set equal to true and the agent was rewarded (reward = +10); however, if the object had been negative, then the “SHOCKED” flag was set to true and the agent was punished (reward = -10). (Note that the flags were set in this way regardless of whether the agent had or had not identified the object; e.g., the agent could consume an object without ever orienting towards it.) On the subsequent time-step, the “SHOCKED” or “CONSUMED” flag was cleared. The agent was also given a small penalty (reinforcement = -1) for each movement or orienting action, and received no reward or punishment (reinforcement = 0) if it performed no action.

Both the overt behaviors (i.e., orienting and movement) and a measure of reward-prediction error were quantified for the agent. The overt behavior (i.e., the list of actions selected by the agent) was used as an indication of whether the task had been learned. The measure of reward-prediction error was used to test the hypothesis about the emergence of the non-reward dopaminergic phasic signal. The reward-prediction error, δ , was measured at the time t of the appearance of an object by subtracting the reward prediction at the previous time-step, i.e., $V(s)$ at time step $t-1$, from the reward prediction when the object appeared, i.e., $V(s)$ at time t , yielding the quantity $\delta = V(s_t) - V(s_{t-1})$.

³The existence of these “hidden” states must be considered during training because Value Iteration only looks “one step ahead” from each state in the state space. The fact that states with negative and positive unidentified objects are effectively identical would prevent learning about and averaging the values in the two different subsequent states in which either the positive or negative object becomes identified. A Trajectory Sampling approach on the other hand maintains the hidden state information (i.e., the identity of the unidentified stimulus) throughout the trial and so with that variant of RL the hidden states are not a concern.

Results

Simulated Behavior

The overt behavior of the agents was first quantified. The results of this analysis showed that, after training, the agent approached and obtained positive reinforcement from all of the positive objects and never approached any of the negative objects. Together, these results provide behavioral confirmation that the agents learned to perform the task correctly. This conclusion is bolstered by the additional observation that, during the trials when no object appeared, the agent remained motionless. As predicted, the agent oriented to both positive and negative objects.

Simulated Reward-Prediction Error

The central hypothesis of this paper is that the appearance of an unpredictable stimulus will consistently generate a positive reward-prediction error, even if that object happens to be a “negative” object that is always punishing. In support of this hypothesis, the agent exhibited a positive reward-prediction error whenever an (unidentified) object appeared, but not when nothing appeared. Also consistent with the central hypothesis is the fact that the magnitude of the agent’s phasic response (δ , measured as described in the Method section) was sensitive to the simulated “intensity” of the stimulus, defined using the distance between the agent and the object (see Figure 1). A regression analysis indicated that the magnitude of δ was inversely related to the distance from the object, so that closer objects caused a stronger response ($r = -0.999$, $p < 0.001$; $\beta = 0.82$). This negative correlation was caused by the small penalty (reinforcement = -1) that was imposed for each movement that the agent was required to make in order to move to the positive object, consume it, and thereby obtain reward.

Given that positive and negative objects appeared in this simulation with equal probability ($p = .25$), the question arises: Why was the agent’s reward-prediction error signal positive at the time of the object’s appearance? Reasoning along the lines of Kakade and Dayan [7], one might predict that the signal should reflect the average of all of the learned rewards from such situations, and therefore be equal to zero. The key to understanding this result is to note that not only does RL make an agent less likely to choose actions that result in negative reinforcement, it also makes an agent less likely to enter states which eventually lead to negative reinforcement. This results in a kind of “higher-order” form of learning that is depicted in Figure 2 and described next.

At the beginning of learning (see Figure 2A), the agent orients to both “+” and “−” objects, approaches them, and is both rewarded and punished by consuming the each type of object. If the agent’s learned state values were unable to influence the agent’s actions (see Figure 2B), then the agent would continue to approach and consume the objects. The appearance of the cue would then predict an average reward of 0 and there would be a sudden increase in reward-prediction error. However, the agent in this simulation does use learned state values to influence its actions (see Figure 2C), and although the agent still has to orient to the unknown object to determine its identity, it will no longer consume a negative object if it approached it (as it might if trained with a random exploration algorithm like *trajectory sampling* [Footnote 1]). Furthermore, because temporal-difference learning allows the negative reward prediction to “propagate” back to preceding states, and because there is a small cost for moving in space, the agent learns to avoid approaching the negative object entirely. Thus, after this information has been learned, the value of the state when the object first appears (indicated as “V” in the first circle in each sequence) is not based on the average of the positive and negative outcome state values, but is instead based on the average of positive and the “neutral” outcome that is attained once the agent learns to avoid the negative objects. This is why the average of all rewards actually obtained by the trained agent was greater than zero, and explains why the

agent's reward prediction (and therefore reward-prediction error when the object suddenly appears) was a net positive. This is illustrated in Figure 3. In fact, as long as the agent can learn to change its behavior and avoid the negative object, the value of the negative object is ultimately irrelevant to the final behavior of the agent and the magnitude of the novelty/saliency response.

The simulation results are critically dependant on three assumptions. First, the stimuli had to be "salient" in that the magnitude of the reinforcement predicted by the initial cue was sufficiently large (e.g., +10) relative to the costs of orienting and approaching (e.g., -1). If the magnitude had been relatively small, the agent would not have learned to orient, nor would it have generated the positive reward-prediction error response. Second, a delay prior to recognizing the stimuli was also necessary. (Delay is a proxy for "novelty" under the reasoning that a familiar stimulus would be quickly recognized.) Without a delay, the agent would have simply generated the appropriate positive or negative reward prediction error appropriate for the actual perceived object. Finally, the agent's behavior had to be determined by the values that it had learned. If the agent could not control its own behavior (i.e., whether to approach the stimuli), then its reward prediction when an object appeared would have equaled 0, the average of the equiprobable positive and negative outcomes.

General Discussion

The simulation reported in this article demonstrated that a positive reward-prediction error occurs when an unpredictable stimulus, either rewarding or punishing, appears but cannot be immediately identified. Furthermore, the simulation indicated that the size of the reward-prediction error increases with proximity of the stimulus to the agent, which in the context of the simulation is a proxy measure for stimulus intensity and is thus related to salience. In the theoretical framework of RL, reward predictions are normally understood to reflect the learned value of recognized stimuli, or of the physical and/or cognitive states of an agent [15]. However, the reward-prediction error reported here has a qualitatively different interpretation because it is generated before the agent has recognized the object. Together, these results support the hypothesis that RL principles are sufficient to produce a response that is seemingly unrelated to reward, but instead related to the properties of novelty and saliency. This conclusion has several important ramifications for our general understanding of RL and for our interpretation of RL as an account of reward learning in real biological organisms.

First, the reward prediction that is generated by an RL agent when an unidentified stimulus appears is not necessarily a strict average of the obtainable rewards as suggested by Kakade and Dayan [7], but can in fact be greater in magnitude than that particular average. Kakade and Dayan would predict that the average reward prediction should be equal to zero because, because the trials were rewarded and punished equally often. This surprising result emerged because the agent learned in an "on-policy" manner; that is, the agent learned not only about negative outcomes, but also about its ability to avoid those outcomes. This ability of the reward system to cause an agent to avoid negative outcomes should be carefully considered in translating our understanding of RL to real organisms. This fact is potentially even more important given the apparent asymmetry in the capacity of the dopaminergic phasic response to represent positive reward prediction error better than negative reward prediction error [11]. It may be sufficient to indicate that a particular sequence of events leads to a negative outcome, but that for the purposes of action selection, the magnitude of that outcome is unimportant.

A second ramification of the current simulation is that the novelty response may emerge from an interaction between perceptual processing systems and reward-prediction systems. Specifically, the novelty response may be due to a form of similarity between novel objects and objects that have not yet undergone complete perceptual processing [Footnote 4]. In this

simulation, novelty was implemented by introducing a delay before the object's identity (and consequently, its rewarding or punishing nature) became apparent to the agent. This was done under the assumption that novel objects take longer to identify, but this assumption also resulted in the positive and negative objects being perceived similarly when they first appeared (i.e., they were both encoded as "?"). In contrast, Kakade and Dayan [7] suggest that novelty responses and "generalization" responses are essentially different despite being manifested similarly in the neurophysiology data.

A third ramification of the current simulation results is that they show that the additional assumptions of novelty and shaping bonuses that were proposed by Kakade and Dayan [7] are not necessary. Instead, novelty-like responses can emerge from realistic perceptual processing limitations and the knowledge of being able to avoid negative outcomes. This is fortunate because, as pointed out by Kakade and Dayan, novelty bonuses distort the value function that is learned by an agent, and shaping bonuses affect the way in which agents explore their state spaces. The inclusion of either of these assumptions thus reduces the parsimony of models based on RL theory. Interestingly, the results presented here also help explain why the biological novelty response might not be disruptive to reward-based learning in real organisms: the novelty response is in fact already predicted by RL. That is, the novelty response reflects behaviors and reward predictions that are inherent in an agent that has already learned something about its environment.

An alternative (but not mutually exclusive) interpretation of the present simulation results is that there is indeed an abstract (perhaps cognitive) reward that agents obtain by orienting towards and identifying objects. In studies of dopaminergic activity, positive phasic responses can occur to unanticipated cues that are known to predict a reward. This simulation, however, demonstrates how these kinds of responses can also occur in response to a cue that could ultimately predict either reward or punishment. The only consistent benefit that is predicted by the cue is the gain in information obtained when the agent determines the identity of the object. Thus, if there is a valid, learned "reward prediction" when the unidentified object appears, it is one that is satisfied after the agent obtains the knowledge about whether to approach or avoid the stimulus. The value of this information is based not on the average of the obtainable outcomes, but is instead based on the knowledge of the effective outcomes—that the agent can either consume the positive reward or avoid the negative reward (see Figure 2).

Finally, it is important to note that the opportunities to take particular actions (e.g., to orient) may themselves take on rewarding properties through some generalization or learning mechanism not included in this simulation. For example, the very act of orienting and determining "what's out there" could become rewarding to an organism based on the association between that action and the above-demonstrated emergent, always-positive reward-prediction error when new stimuli appear. A similar idea has been recently advanced by Redgrave and Gurney [13] who hypothesize that an important purpose of the phasic dopamine response is to reinforce actions that occur before unpredicted salient events. The results here are not incompatible with that hypothesis, however it should be noted that Redgrave and Gurney's hypothesis is not directly tested in this simulation because no actions (i.e., exploration) were required of the agent in order for the salient event (the appearance of the

⁴One potential objection to the present work is that the orienting response appears to be hard-wired in the mammalian brain, for example, in projections from the superior colliculus [3,14]. In the present simulation, the agents were not hard-wired to orient to objects but instead learned an orienting behavior that permitted the eventual selection of an action (e.g., either approach or avoidance) that maximized reward. Similarly to hard-wired responses, these orienting behaviors occurred very rapidly, before the objects were identified, and were directed towards all objects. The goal of this work was not to make the claim that all such responses are learned, but rather that they can co-exist within the RL framework. Nevertheless, it would be interesting to investigate whether reward-related mechanisms might be involved in setting up connectivity in brainstem areas in order to generate this phasic dopamine response.

object) to occur. However, the simulated phasic signal coincided with the time of the orienting response suggesting that the two may be strongly related.

In closing, this article has demonstrated that RL principles can be used to explain a type of seemingly non-reward related activity of the dopaminergic neurons. This result emerged from the fact that the temporal-difference learning rule (such as that used by Kakade and Dayan [7]) was embedded in a simulation in which the agent could select actions that had an effect on the eventual outcome. In the simulation, the agent learned that the outcome of orienting to an object that suddenly appeared could always either be rewarding or neutral because the negative outcome could be avoided. Therefore when the agent had an opportunity to orient, its reward-prediction error was always positive, computationally analogous to the novelty and saliency responses observed in biological organisms.

Acknowledgements

The work described in this article was supported by NIH R01 HD053639 and by NSF Training Grant DGE-9987588. I would like to thank Erik Reichle, Tessa Warren, and an anonymous reviewer for helpful comments on an earlier version of this article.

References

1. Baird, LC. Residual Algorithms: Reinforcement Learning with Function Approximation. In: Priedetis, A.; Russell, S., editors. Machine Learning: Proceedings of the Twelfth International Conference; 9–12 July; 1995.
2. Bunzeck N, Düzel E. Absolute coding of stimulus novelty in the human substantia nigra/VTa. *Neuron* 2006;51(3):369–379. [PubMed: 16880131]
3. Dommett E, Coizet V, Blaha CD, Martindale J, Lefebvre V, Walton N, Mayhew JEW, Overton PG, Redgrave P. How visual stimuli activate dopaminergic neurons at short latency. *Science* 2005;307(5714):1476–1479. [PubMed: 15746431]
4. Doya K. Metalearning and neuromodulation. *Neural Networks* 2002 Jun–Jul;15(4–6):495–506. [PubMed: 12371507]
5. Gillies A, Arbutnott G. Computational models of the basal ganglia. *Movement Disorders* 2000;15(5):762–770. [PubMed: 11009178]
6. Horvitz JC. Mesolimbocortical and nigrostriatal dopamine responses to salient non-reward events. *Neuroscience* 2000;96(4):651–656. [PubMed: 10727783]
7. Kakade S, Dayan P. Dopamine: generalization and bonuses. *Neural Networks* 2002;15(4–6):549–559. [PubMed: 12371511]
8. Knutson B, Cooper JC. The lure of the unknown. *Neuron* 2006;51(3):280–282. [PubMed: 16880122]
9. Logothetis NK, Pauls J, Augath M, Trinath T, Oeltermann A. Neurophysiological investigation of the basis of the fMRI signal. *Nature* 2001;412(6843):150–157. [PubMed: 11449264]
10. McClure SM, Berns GS, Montague PR. Temporal prediction errors in a passive learning task activate human striatum. *Neuron* 2003;38(2):339–346. [PubMed: 12718866]
11. Niv Y, Duff MO, Dayan P. Dopamine, uncertainty and TD learning. *Behavioral and Brain Functions* 2005 May 4;1:6. [PubMed: 15953384]
12. O’Doherty JP, Dayan P, Friston K, Critchley H, Dolan RJ. Temporal difference models and reward-related learning in the human brain. *Neuron* 2003;38(2):329–337. [PubMed: 12718865]
13. Redgrave P, Gurney K. The short-latency dopamine signal: a role in discovering novel actions? *Nature Reviews Neuroscience* 2006 Dec;7(12):967–975.
14. Redgrave P, Prescott TJ, Gurney K. Is the short-latency dopamine response too short to signal reward error? *Trends in Neurosciences* 1999 Apr;22(4):146–151. [PubMed: 10203849]
15. Reichle ED, Laurent PA. Using reinforcement learning to understand the emergence of “intelligent” eye-movement behavior during reading. *Psychological Review* 2006;113(2):390–408. [PubMed: 16637766]

16. Schultz W. Predictive reward signal of dopamine neurons. *Journal of Neurophysiology* 1998;80(1): 1–27. [PubMed: 9658025]
17. Sutton, RS.; Barto, AG. *Reinforcement Learning: An Introduction*. MIT Press; Cambridge: 1998.
18. Tanaka SC, Doya K, Okada G, Ueda K, Okamoto Y, Yamawaki S. Prediction of immediate and future rewards differentially recruits cortico-basal ganglia loops. *Nature Neuroscience* 2004;7(8):887–893.

Figure 1.

This figure shows the reward-prediction error (i.e., δ) when the object appeared as a function of the location of the object relative to the location of the agent. The responses are identical for both positive and negative objects. When no object appeared, the response was 0. Note that the size of the response is inversely correlated with distance from the object when it appeared. There is no data for location 0 because the object would be immediately consumed had it appeared there.

Figure 2.

Illustration showing how an RL agent develops positive reward-prediction error when an it is trained with both rewarding and punishing stimuli in its environment and is able to choose whether to approach and consume them. (A) The situation before learning: all states begin with a value of 0, and the agent has not yet learned the rewarding and punishing values of the "+" and "-" stimuli. (B) A temporal-difference learning algorithm is used without allowing those values to affect the actions of the agent: the agent learns reward predictions based on experience but is unable to use the learned values to influence its own behavior. In this case, the reward-prediction error when the object appears will be the average of the positive and negative outcomes (i.e., 0). (C) We show what happened in the present simulation. The agent quickly learns to avoid consuming, or even approaching, the negative object. The result is that when the stimulus appears, the reward-prediction error is based on the average of the positive stimulus and a neutral outcome in which the negative stimulus is avoided and is consistently greater than 0. Note: This figure does not illustrate the fact that in the present simulation, more distant objects require more actions (and therefore, more intervening small punishments) in

order to approach them. That fact is what causes the decreasing magnitude of the novelty/saliency response for object that appear more distantly from the agent (e.g., as plotted in Figure 2.)

Figure 3.

(A) Demonstrates the changes in reward prediction that would have occurred if RL did not result in higher-order learning (i.e., if the agent could not take measures to avoid the negative outcome), so that the agent was forced to consume all the objects that appeared. When an object appears, the agent does not know yet its identity but generates a net reward prediction of zero because the reward prediction is the average of the positive and negative consequences (i.e., half the time the object has been positive, and half the time it has been negative). (B) Demonstrates what actually occurred: higher-order learning permitted the agent to avoid the negative object, so that when the stimulus appeared, the agent had a greater-than-0 reward prediction because it is the average of the positive outcome and null outcomes. The curly brace spans the difference in reward-prediction values that represents this reward-prediction error.

Table 1

The dimensions used in the RL simulations and the possible values of those dimensions.

Dimension No.	Description	Possible Values
1	Position of Agent	integer (0–10)
2	Orientation of Agent	character (“n”, “s”, “e”, or “w”)
3	Position of Object	integer (0–10)
4	Identity of Object	character (“0”, “?”, “+”, or “-”)
5	“Shocked” by Object	Boolean (true or false)
6	“Consumed” Object	Boolean (true or false)
7	Time Since Trial Onset	Integer (0–20)