Temporal Difference Based Actor Critic Learning - Convergence and Neural Implementation

Authored by:

Ron Meir
Dmitry Volkinshtein
Dotan D. Castro

Abstract

Actor-critic algorithms for reinforcement learning are achieving renewed popularity due to their good convergence properties in situations where other approaches often fail (e.g., when function approximation is involved). Interestingly, there is growing evidence that actor-critic approaches based on phasic dopamine signals play a key role in biological learning through the cortical and basal ganglia. We derive a temporal difference based actor critic learning algorithm, for which convergence can be proved without assuming separate time scales for the actor and the critic. The approach is demonstrated by applying it to networks of spiking neurons. The established relation between phasic dopamine and the temporal difference signal lends support to the biological relevance of such algorithms.

1 Paper Body

Actor-critic (AC) algorithms [22] were probably among the first algorithmic approaches to reinforcement learning (RL). In recent years much work focused on state, or state-action, value functions as a basis for learning. These methods, while possessing desirable convergence attributes in the context of table lookup representation, led to convergence problems when function approximation was involved. A more recent line of research is based on directly (and usually parametrically) representing the policy, and performing stochastic gradient ascent on the expected reward, estimated through trying out various actions and sampling trajectories [3, 15, 23]. However, such direct policy methods often lead to very slow convergence due to large estimation variance. One approach suggested in recent years to remedy this problem is the utilization of AC approaches, where the value function is estimated by a critic, and passed to an actor which selects an appropriate action, based on the approximated value function. The first convergence result for a policy gradient AC algorithm based on function

approximation was established in [13], and extended recently in [5, 6]. At this stage it seems that AC based algorithms provide a solid foundation for provably effective approaches to RL based on function approximation. Whether these methods will yield useful solutions to practical problems remains to be seen. RL has also been playing an increasingly important role in neuroscience, and experimentalists have directly recorded the activities of neurons while animals perform learning tasks [20], and used imaging techniques to characterize human brain activities [17, 24] during learning. It was suggested long ago that the basal ganglia, a set of ancient sub-cortical brain nuclei, are implicated in RL. Moreover, these nuclei are naturally divided into two components, based on the separation of the striatum (the main input channel to the basal ganglia) into the ventral and dorsal components. Several imaging studies [17, 24] have suggested that the ventral stream is associated with value estimation by a so called critic, while the dorsal stream has been implicated in motor output, action selection, and learning by a so called actor. Two further experimental findings support the view taken in this work.

First, it has been observed [20] that the short latency phasic response of the dopamine neurons in the midbrain strongly resembles the temporal difference (TD) signal introduced in theory of TDlearning [22], which can be used by AC algorithms for both the actor and the critic. Since mid-brain dopaminergic neurons project diffusively to both the ventral and dorsal components of the striatum, these results are consistent with a TD-based AC learning interpretation of the basal ganglia. Second, recent results suggest that synaptic plasticity occurring at the cortico-striatal synapses is strongly modulated by dopamine [18]. Based on these observations it has been suggested that the basal ganglia take part in TD based RL, with the (global) phasic dopamine signal serving as the TD signal [16] modulating synaptic plasticity. Some recent work has been devoted to implementing RL in networks of spiking neurons (e.g., [1, 9, 12]). Such an approach may lead to specific and experimentally verifiable hypotheses regarding the interaction of known synaptic plasticity rules and RL. In fact, one tantalizing possibility is to test these derived rules in the context of ex-vivo cultured neural networks (e.g., [19]), which are connected to the environment through input (sensory) and output (motor) channels. We then envision dopamine serving as a biological substrate for implementing the TD signal in such a system. The work cited above is mostly based on direct policy gradient algorithms, (e.g., [3]), leading to nonAC approaches. Moreover, these algorithms were based directly on the reward, rather than on the biologically better motivated TD signal, which provides more information than the reward itself, and is expected to lead to improved convergence.

2

A Temporal Difference Based Actor-Critic Algorithm

The TD-based AC algorithm developed in this section is related to the one presented in [5, 6]. While the derivation of the present algorithm differs from the latter work (which also stressed the issue of the natural gradient) , the essential novel theoretical feature here is the establishment of convergence1 without the restriction to two time scales which was used in [5, 6, 13]. This result is also

important in a biological context, where, as far as we are aware, there is no evidence for such a time scale separation. 2.1

Problem Formulation

We consider a finite Markov Decision Process (MDP) in discrete time with a finite state set X of size —X — and a finite action set U. The MDP models the environment in which the agent acts. Each selected action u ? U determines a stochastic matrix P (u) = [P (y—x, u)]x,y?X where P (y—x, u) is the transition probability from a state x ? X to a state y ? X given the control u. A parameterized policy is described by a conditional probability function, denoted by ?(u—x, ?), which maps observation x ? X into a control u ? U given a parameter ? ? RK . For each state x ? X the agent receives a corresponding reward r(x). The agent?s goal is to adjust the parameter ? in order to attain maximum average reward over time. For each ? ? RK , we have a Markov Chain (MC) induced by P (y—x, u) and ?(u—x, ?). The state transitions of the MC are obtained by first generating an action u according to ?(u—x, ?), and then generating the next state according to P (y—x, u)]x,y?X . Thus, the MC has a transition matrix P (?) = R [P (y—x, ?)]x,y?X which is given by P (y—x, ?) = U P (y—x, u)d?(u—x, ?). We denote the set of these ? We denote by P (x, u, y) transition probabilities by P = {P (?)—? ? RK }, and its closure by P. the stationary probability to be in state x, choose action u and go to state y. Several technical assumptions are required in the proofs below. ? is aperiodic, recurrent, and contains a single Assumption 2.1. (i) Each MC P (?), P (?) ? P, equivalence class. (ii) The function ?(u—x, ?) is twice differentiable. Moreover, there exist positive constants Br and B? , such that for all x ? X , u ? U, ? ? RK and 1 ? k1 , k2 ? K, we have —r(x)— ? Br , —??(u—x, ?)/??k — ? B? , —? 2 ?(u—x, ?)/??k1 ?k2 — ? B? . As a result of assumption 2.1(i), we have the following lemma regarding the stationary distribution (Theorem 3.1 in [8]). 1 Throughout this paper convergence refers to convergence to a small ball around a stationary point; see Theorem 2.6 for a precise definition.

? has a unique stationary distribution, Lemma 2.1. Under Assumption 2.1(i), each MC, P (?) ? P, denoted by ?(?), satisfying ?(?)0 P (?) = ?(?)0 , where x0 is the transpose of vector x. Next, we define a measure for performance of an agent in an environment. The average reward per stage of a MC starting from an initial state x0 ? X is defined by # " T ? 1 X ? r(xn)?x0 = x , J(x—?) , lim E? T ?? T n=0 where E? [?] denotes the expectation under the probability measure P (?), and xn is the state at time n. The agent?s goal is to find ? ? RK which maximizes J(x—?). The following lemma shows that under Assumption 2.1, the average reward per stage does not depend on the initial states (see Theorem 4.7 in [10]). Lemma 2.2. Under Assumption 2.1 and Lemma 2.1, the average reward per stage, J(x—?), is independent of the starting state, is denoted by ?(?), and satisfies ?(?) = ?(?)0 r. Based on Lemma 2.2, the agent?s goal is to find a parameter vector ?, which maximizes the average reward per stage ?(?). Performing the maximization directly on ?(?) is hard. In the sequel we show how this maximization can be performed by optimizing ?(?), using ??(?). A consequence of Assumption 2.1 and the definition of ?(?) is the following lemma (see Lemma 1 in [15]). Lemma 2.3. For each x, y ? X and

for each ? ? RK , the functions P (y—x, ?), ?(x—?), and ?(?), are bounded, twice differentiable, and have bounded first and second derivatives. Next, we define the differential value function of state x ? X which represents the average reward the agent receives upon starting from a state x0 and reaching a recurrent state x? for the first time. Mathematically, " T # ? X ? h(x—?) , E? (r(xn) ? ?(?))?x0 = x , (1) n=0 ?

where T , min{k ¿ 0—xk = x }. We define h(?) , (h(x1 —?), . . . , h(x—X — —?)) ? R—X — . For each ? ? RK and x ? X , h(x—?), r(x), and ?(?) satisfy Poisson?s equation (see Theorem 7.4.1 in [4]), X h(x—?) = r(x) ? ?(?) + P (y—x, ?)h(y—?). (2) y?X

Based on the differential value definition we define the temporal difference (TD) between the states x ? X and y ? X . Formally, d(x, y) , r(x) ? ?(?) + h(y—?) ? h(x—?).

(3)

The TD measures the difference between the differential value estimate following the receipt of reward r(x) and a move to a new state y, and the estimate of the current differential state value at state x. 2.2

Algorithmic details and single time scale convergence

We start with a definition of the likelihood ratio derivative, ?(x, u—?) , ??(u—x, ?)/?(u—x, ?), which we assume to be bounded. Assumption 2.2. For all x ? X , u ? U, and ? ? RK , there exists a positive constant, B? , such that —?(x, u—?)— ? B? ¡ ?. In order to improve the agent?s performance, we need to follow the gradient direction. The following theorem shows how the gradient of the average reward per stage can be calculated by the TD signal. Similar variants of the theorem were proved using the Q-value [23] or state value [15] instead of the TD-signal. Theorem 2.4. The gradient of the average reward per stage for ? ? RK can be expressed by X ??(?) = P (x, u, y)?(x, u—?) (d(x, y) + f (x)) (f (x) arbitrary). (4) x,y?X ,u?U

The theorem was proved using an advantage function argument in [6]. We provide a direct proof in section A of the supplementary material. The flexibility resulting from the function f (x) allows us to encode the TD signal using biologically realistic positive values only, without influencing the convergence proof. In this paper, for simplicity, we use f (x) = 0. Based on Theorem 2.4, we suggest an TD-based AC algorithm. This algorithm is motivated by [15] where an actor only algorithm was proposed. In [15] the differential value function was re-estimated afresh for each regenerative cycle leading to a large estimation variance. Using the continuity of the actor?s policy function in ?, the difference between the estimates between regenerative cycles is small. Thus, the critic has a good initial estimate at the beginning of each cycle, which is used here in order to reduce the variance. A related AC algorithm was proposed in [5, 6], where two time scales were assumed in order to use Borkar?s two time scales convergence theorem [7]. In our proposed algorithm, and associated convergence theorem, we do not assume different time scales for the actor and for the critic. We present batch mode update equations2 in Algorithm 1 for the actor and the critic. The algorithm is based on some recurrent state x? ; the visit times to this state are denoted by t0 , t1 , ... Updated occur only at these times (batch

mode). We define a cycle of the algorithm by the time indices which ? h(x), ? satisfy tm ? n ¡ tm+1 . The variables d, and ?? are the critic?s estimates for d, h(x—?), and ?(?) respectively. Algorithm 1 Temporal Difference Based Actor Critic Algorithm 1: Given ? An MDP with finite set X of states and a recurrent state x? , satisfying 2.1(i). ? Hitting times t0 ¡ t1 ¡ t2 ¡ ?P ? ? for the state x? . P ? ? 2 ¡ ?. ? Step coefficients ?m such that m=1 ?m = ? and m=1 ?m K ? A parameterized policy ?(u—x, ?), ? ? R , which satisfies Assumption 2.1(ii). ? A set H, constants Bh? and B? , and an operator ?H according to Assumption B.1. ? Step parameters ?? and ?h satisfying Theorem 2.6. 2: Initiate the critic?s variables: ? ??0 = 0 (the estimate of the average reward per stage) ? 0 (x) = 0, ?x ? X (the estimate of the differential value function) ? h 3: Initiate the actor: ?0 = 0 and choose f (x) (see (4)) 4: for each state xtm+1 visited do 5: Critic: For all x ? X , Nm (x) , min{tm ¡ k ¡ tm+1 —xk = x}, (min(?) = ?) ? n , xn+1) = r(xn) ? ??m + h ? m (xn+1) ? h ? m (xn), d(x ? ? tm+1 ?1 X ? m+1 (x) = h ? m (x) + ?m ?h ? ? n , xn+1)? , h d(x

?x ? X ,

n=Nm (x) tm+1 ?1

??m+1 = ??m + ?m ??

X

(r(xn) ? ??m).

n=tm

Ptm+1 ?1 ? n , xn+1) + f (xn)) Actor: ?m+1 = ?m + ?m n=t ?(xn , un —?m)(d(x m ? m+1 and ?m+1 onto H (see Assumption B.1.). 7: Project each component of h 8: end for

6:

In order to prove the convergence of Algorithm 1, we establish two basic results. The first shows that the algorithm converges to the set of ordinary differential equations (5), and the second establishes conditions under which the differential equations converge locally. 2 In order to prove convergence certain boundedness conditions need to be imposed, which appear as step 7 in the algorithm. For lack of space, the precise definition of the set H is given in Assumption B.1 of the supplementary material.

Theorem 2.5. Under Assumptions 2.1 and B.1, Algorithm 1 converges to the following set of ODE?s ? ? ? X (x) ? ? ? ? = T (?)??(?) + C(?) (?(?) ? ? ?) + D (?) h(x—?) ? h(x) , ? ? ? ? x?X ? ? (5) ?? ? h(x) = ?h h(x—?) ? h(x) + ?h T (?) (?(?) ? ??) , x ? X ? ? ? ? ? ??? = ?? T (?) (?(?) ? ??) , with probability 1, where ?

?

T = min{k ¿ 0—x0 = x , xk = x }, D(x) (?) = E?

"T ?1 X

T (?) = E? [T],

C(?) = E?

"T ?1 X

? ? ? ?(xn , un —?)?x0 = x ,

n=0

1 {xn+1

"T ?1 # ? ? X ? ? ? ? = x} ?(xn , un —?)?x0 = x + E? (1 {xn = x} ?(xn , un —?)?x0 = x ,

n=0

n=0

and where T (?), C(?), and D(x) (?) are continuous with respect to ?. Theorem 2.5 is proved in section B of the supplementary material, based on the theory of stochastic approximation, and more specifically, on Theorem 5.2.1 in [14]. An advantage of the proof technique is that it does not need to assume two time scales. The second theorem, proved in section C of the supplementary material, states the conditions for which ?(?t) converges to a ball around the local optimum. Theorem 2.6. If we choose ?? ? B?2? /?? and ?h ? Bh2? /?h , for some positive constants ?h and ?? , then lim supt?? k??(?(t))k ? ?, where ? , BC ?? + —X —BD ?h . The constants B?? and Bh? are defined in Section C of the supplementary material.

3

A Neural Algorithm for the Actor Using McCulloch-Pitts Neurons

In this section we apply the previously developed algorithm to the case of neural networks. We start with the classic binary valued McCulloch-Pitts neuron, and then consider a more realistic spiking neuron model. While the algorithm presented in Section 2 was derived and proved to converge in batch mode, we apply it here in an online fashion. The derivation of an online learning algorithm from the batch version is immediate (e.g., [15]), and a proof of convergence in this setting is currently underway. A McCulloch-Pitts actor network The dynamics of the binary valued neurons, given at time n by {ui (n)}N i=1 , ui (n) ? {0, 1}, is assumed to be based on stochastic discrete time parallel updates, given by Pr(ui (n) = 1) = ?(vi (n))

where vi (n) =

N X

wij uj (n ? 1)

(i = 1, 2, . . . , N).

j=1

Here ?(v) = 1/(1 + exp(?v)), and the parameters ? in Algorithm 1 are given by {wij }, where wij (n) is the j 7? i synaptic weight at time n. Each neuron?s stochastic output ui is viewed as an action. Applying the actor update from Algorithm 1 we obtain the following online learning rule wij (n + 1) = wij (n) + ?d(x(n), x(n + 1)) (ui (n) ? ?(vi (n))) uj (n ? 1).

(6)

where d(x(n), x(n + 1)) is the TD signal. The update (6) can be interpreted as an error-driven Hebbian-like learning rule modulated by the TD signal. It resembles the direct policy update rule presented in [2], except that in this rule the reward signal is replaced by the TD signal (computed by the critic). Moreover, the eligibility trace formalism in [2] differs from our formulation.

We describe a simulation experiment conducted using a one layered feed-forward artificial neural network which functions as an actor, combined with a non biologically motivated critic. The purpose of the experiment is to examine

a simple neuronal model, using different actor and critic architectures. The actor network consists of a single layered feed-forward network of McCullochPitts neurons, and TD modulated synapses as described above, where the TD signal is calculated by a critic. The environment is a maze with barriers consisting of 36 states, see Figure 1(b), where a reward of value 1 is provided at the top right corner, and is zero elsewhere. Every time the agent receives a reward, it is transferred randomly to a different location in the maze. At each time step, the agent is given an input vector which represents the state. The output layer consists of 4 output neurons where each neuron represents an action from the action set U = {up, down, left, right}. We used two different input representations for the actor, consisting either of 12 or 36 neurons (note that the minimum number of input neurons to represent 36 states is 6, and the maximum number is 36). The architecture with 36 input neurons represents each maze state with one exclusive neuron, thus, there is no overlap between input vectors. The architecture with 12 input neurons uses a representation where each state is represented by two neurons, leading to overlaps between the input vectors. We tested two types of critic: a table based critic which performs iterates according to Algorithm 1, and an exact TD which provides the TD of the optimal policy. The results are shown in Figure 1(c), averaged over 25 runs, and demonstrate the importance of good input representations and precise value estimates. 0.12 Average Reward per Stage

0.1 0.08 0.06 0.04 0.02 0

(a)

(b)

0

5 10 Number of Steps

15 5

x 10

(c)

Figure 1: (a) A illustration of the McCulloch-Pitts network. (b) A diagram of the maze where the agent needs to reach the reward at the upper right corner. (c) The average reward per stage in four different cases: an actor consisting of 12 input neurons and a table based critic (blue crosses), an actor consisting of 36 input neurons and a table based critic (green stars), an actor consisting of 12 input neurons and exact critic (red circles), and an actor consisting of 36 input neurons and an exact TD (black crosses). The optimal average reward per stage is denoted by the dotted line, while a random agent achieves a reward of 0.005.

A spiking neuron actor Actual neurons function in continuous time producing action potentials. In extension of [1, 9], we developed an update rule which is based on the Spike Response Model (SRM) [11]. For each neuron we define a state variable vi (t) which represents the membrane potential. The dynamics of vi (t) is given by N X X ?ij (t ? t?i , t ? tfj), (7) vi (t) = ?i (t ? t?i) + wij (t) j=1

tfj

where wij (t) is the synaptic efficacy, t?i is the last spike time of neuron i prior o t, ?i (t) is the refractory response, tfj are the times of the presynaptic

spikes emitted prior to time t, and ?ij (t ? t?i , t ? tfj) is the response induced by neuron j at neuron i. The second summation in (7) is over all spike times of neuron j emitted prior to time t. The neuron model is assumed to have a noisy threshold, which we model by an escape noise model [11]. According to this model, the neuron fires in the time interval [t, t + ?t) with probability ui (t)?t = ?i (vi (t) ? vth)?t, where vth is the firing threshold and ?i (?) is a monotonically increasing function. When the neuron reaches the threshold it is assumed to fire and the membrane potential is reset to vr .

We consider a network of continuous time neurons and synapses. Based on Algorithm 1, using a small time step ?t, we find wij (t + ?t) = wij (t) + ?d(t)?ij (t).

(8)

We define the output of the neuron (interpreted as an action) at time t by ui (t). We note that the neuron?s output is discrete and that at each time t, a neuron can fire, ui (t) = 1, or be quiescent, ui (t) = 0. Using the definition of ? from Section 2.2, yields (similar to [9]) ? 0 P f ? ?i (t) ? if ui (t) = 1 Htj ?ij (t ? ti , t ? tj), ?i (t) ?ij (t) = 0 P ?t? (t) f i ? ? ? if ui (t) = 0 Ht ?ij (t ? ti , t ? tj), 1??t?i (t) j

Taking the limit ?t ? 0, yields the following continuous time update rule Fpost ({tfi })

}—

z ?

{

!

X dwij (t) = ?d(t) (1/?i (t)) ?(t ? tfi) ? 1 ?0i (t) dt Hi

z X

Fpre ({tfj })

}— { f ? ?ij (t ? ti , t ? tj) .

(9)

Htj

Similarly to [1, 9] we interpret the update rule (9) as a TD modulated spike time dependent plasticity rule. A detailed discussion and interpretation of this update in a more biological context will be left to the full paper. We applied the update rule (9) to an actor network consisting of spiking neurons based on (7). The network?s goal was to reach a circle at the center of a 2D plain =, where the agent can move, using Newtonian dynamics, in the four principle directions. The actor is composed of an input layer and a single layer of modifiable weights. The input layer consists of ?sensory? neurons which fire according to the agent?s location in the environment. The synaptic dynamics of the actor is determined by (9). The critic receives the same inputs as the actor, but uses a linear function approximation architecture rather than the table lookup used in Algorithm 1. A standard parameter update rule appropriate for this architecture (e.g., ch. 8 in [22]) was used to update the critic?s parameters3 . The output layer of the actor consists of four neuronal groups, representing the directions in which the agent can move, coded based on a firing rate model using Gaussian tuning curves. The TD signal is calculated according to (3). Whenever it reaches the centered

circle, it receives a reward, and is transferred randomly to a new position in the environment. Results of such a simulation are presented in Figure 3. Figure 3-a displays the agent?s typical random walk like behavior prior to learning, . Figure 3-b depicts four typical trajectories representing the agent?s actions after a learning phase. Finally, Figure 3-c demonstrates the increase of the average reward per stage, ?, vs. time. 20

0.02

15

15

0.015

10

10

5

?

20

0.01 0.005

5

0 0

0

10

(a)

20

0

0

10

(b)

20

0

200 400 time[sec]

600

(c)

Figure 2: (a) Typical agent tracks prior to learning. (b) Agent trajectories following learning. (c) Average reward per stage plotted against time.

4

Discussion

We have presented a temporal difference based actor critic learning algorithm for reinforcement learning. The algorithm was derived from first principles based on following a noisy gradient of the 3 Algorithm 1 relies on a table lookup critic, while in this example we used a function approximation based critic, due to the large (continuous) state space.

average reward, and a convergence proof was presented without relying on the widely used two time scale separation for the actor and the critic. The derived algorithm was applied to neural networks, demonstrating their effective operation in maze problems. The motivation for the proposed algorithm was biological, providing a coherent computational explanation for several recently observed phenomena: actor critic architectures in the basal ganglia, the relation

of phasic dopaminergic neuromodulators to the TD signal, and the modulation of the spike time dependent plasticity rules by dopamine. While a great deal of further work needs to be done on both the theoretical and biological components of the framework, we hope that these results provide a tentative step in the (noisy!) direction of explaining biological RL.

2 References

[1] D. Baras and R. Meir. Reinforcement learning, spike time dependent plasticity and the bcm rule. Neural Comput., 19(8):22452279, 2007 [2] J. Baxter and P.L. Bartlett. Hebbian synaptic modifications in spiking neurons that learn. (Technical rep.). Canberra: Research School of Information Sciences and Engineering, Australian National University, 1999. [3] J. Baxter and P.L. Bartlett. Infinite-Horizon Policy-Gradient Estimation. J. of Artificial Intelligence Research, 15:319?350, 2001. [4] D.P. Bertsekas. Dynamic Programming and Optimal Control, Vol I., 3rd Ed. Athena Scinetific, 2006. [5] S. Bhatnagar, R. Sutton, M. Ghavamzadeh, and M. Lee. Incremental natural actor-critic algorithms. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, Advances in Neural Information Processing Systems 20, pages 105?112. MIT Press, Cambridge, MA, 2008. [6] S. Bhatnagar, R.S. Sutton, M. Ghavamzadeh, and M. Lee. Natural actor-critic algorithms. Automatica, To appear, 2008. [7] V.S. Borkar. Stochastic approximation with two time scales. Syst. Control Lett., 29(5):291294, 1997. [8] P. Bremaud. Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues. Springer, 1999. [9] R.V. Florian. Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity. Neural Computation, 19:14681502, 2007. [10] R.G. Gallager. Discrete Stochastic Processes. Kluwer Academic Publishers, 1995. [11] W. Gerstner and W.M. Kistler. Spinking Neuron Models. Cambridge University Press, Cambridge, 2002. [12] E.M. Izhikevich. Solving the Distal Reward Problem through Linkage of STDP and Dopamine Signaling. Cerebral Cortex, 17(10):2443-52, 2007. [13] V.R. Konda and J. Tsitsiklis. On actor critic algorithms. SIAM J. Control Optim., 42(4):11431166, 2003. [14] H.J. Kushner and G.G. Yin. Stochastic Approximation Algorithms and Applications. Springer, 1997. [15] P. Marbach and J. Tsitsiklis. Simulation-Based Optimization of Markov Reward Processes. IEEE. Trans. Auto. Cont., 46:191?209, 1998. [16] P.R. Montague, P. Dayan, and T.J. Sejnowski. A framework for mesencephalic dopamine systems based on predictive hebbian learning. Journal of Neuroscience, 16:19361947, 1996. [17] J. ODoherty, P. Dayan, J. Schultz, R. Deichmann, K. Friston, and R.J. Dolan. Dissociable roles of ventral and dorsal striatum in instrumental conditioning. Science, 304:452454, 2004. [18] J.N.J. Reynolds and J.R. Wickens. Dopamine-dependent plasticity of corticostriatal synapses. Neural Networks, 15(4-6):507521, 2002. [19] S. Marom and G. Shahaf. Development, learning and memory in large random networks of cortical neurons: lessons beyond anatomy. Quarterly Reviews of Biophysics, 35:6387, 2002. [20] W. Schultz. Multiple reward signals in the brain. Nature Reviews Neuroscience, 1:199207,

Dec. 2000. [21] S. Singh and P. Dayan. Analytical mean squared error curves for temporal difference learning. Machine Learning, 32:540, 1998. [22] R. S. Sutton and A. G. Barto. Reinforcement Learning. MIT Press, 1998. [23] R. Sutton, D. McAllester, S. Singh and Y. Mansour. Policy-Gradient Methods for Reinforcement Learning with Function Approximation. Advances in Neural Information Processing Systems, 12:1057?1063, 2000. [24] E.M. Tricomi, M.R. Delgado, and J.A. Fiez. Modulation of caudate activity by action contingency. Neuron, 41(2):281292, 2004.