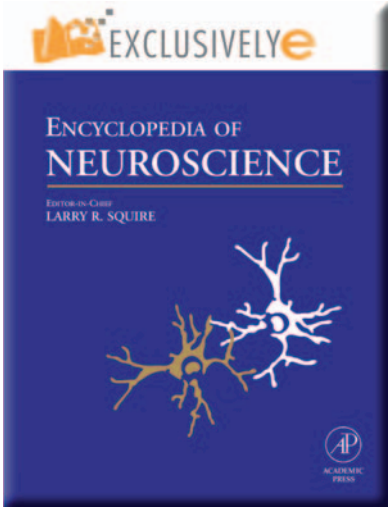

Kawato M (2009) Reinforcement Models. In: Squire LR (ed.) Encyclopedia of Neuroscience, volume 8, pp. 89-97. Oxford: Academic Press.

Reinforcement Models

M Kawato, ATR Computational Neuroscience Laboratories, Kyoto, Japan

The reinforcement learning algorithms proposed and developed by Andy Barto and Rich Sutton in the early 1980s have provided some of the most influential computational theories in neuroscience. The original theories were developed for explaining adaptive animal behaviors, but during the following decade they had a larger impact in the machine learning, computer science, and robotics communities than within the biological disciplines. In the mid-1990s, Wolfram Schultz produced epoch-making neurophysiological data on dopamine neuron activities, which led several theorists to propose computational learning models of basal ganglia and dopamine neurons based on reinforcement learning algorithms. This interaction between experiments and theories generated a steady and influential research trend, which has encompassed many previous and ongoing studies. Here, the temporal difference error plays the most important role in both experimental and theoretical studies of reinforcement learning. In brief, it is computed as the difference among the following three terms: the reward obtained at the current timing, the discounted expectation of the cumulative reward at the neighboring future timing, and the subtracted expectation of the cumulative reward at the current timing. In a class of reinforcement learning algorithms collectively called 'temporal difference learning,' the temporal difference (TD) error plays a central role in guiding acquisition of the predicted reward and behaviors. In the context of classical conditioning, the TD error is predicted to be positive at the time of the primary reward in early learning trials, but as learning progresses it is predicted to move at the time of a cue that indicates a future reward. If the expected reward from the cue is omitted, the TD error becomes negative at the time of the expected but nondelivered reward. Schultz and others confirmed all these predictions in studies of dopamine neuron activities. Because dopamine neurons innervate not only the basal ganglia but also a wide area of the cerebral cortex, learning in both regions might be able to be understood by reinforcement learning algorithms. Many experimental results supporting this hypothesis have been obtained from different levels (cell, circuit, brain, and behavior) utilizing different techniques (slice experiment, extracellular unit recording in behaving animals, lesion, and human imaging). It seems quite safe to state that at least some aspects of learning that are dependent on reward and penalty can be understood within the framework of reinforcement learning theory.

However, at least three difficult issues remain unresolved and should be studied from both theoretical and experimental standpoints. First, the plain reinforcement learning algorithm is too slow for practical problems to be considered as a realistic model of brain learning, which is not extremely slow. Second, although it is highly probable that dopamine neurons encode the TD error, the neural circuits and neural mechanisms needed to compute the TD error are still unknown. Third, although some functional aspects of the neural circuits consisting of dopamine neurons and the striatum of the basal ganglia might be understood by plain reinforcement learning algorithms, behavioral learning, which is dependent on global brain networks involving the cerebral cortex, cerebellum, basal ganglia, and midbrain, seems to necessitate much more complicated algorithms. This article discusses theoretical explorations for more efficient reinforcement learning algorithms and related experimental studies to resolve these difficulties.

Slow Learning of Plain Reinforcement Learning

The plain and basic reinforcement learning algorithm is attractive because it can learn new behaviors from only reward and penalty information without using explicit and quantitative models of controlled objects or environments. The trade-off is that the learning is extremely slow for any practical problem. Plain reinforcement learning algorithms can work reasonably well in a toy problem such as a discrete maze or in a game such as backgammon; however, even in such problems, tens of thousands of learning trials are often required, and millions of learning trials are not rare. Moreover, these algorithms fail badly for real-world problems such as those encountered in robotics, with continuous time and continuous state space. Performing even hundreds of learning trials is prohibitive for animal or robot learning. The slowness of the plain and basic reinforcement learning algorithm can be explained by either a spatial reason or a temporal reason. First, up-to-date humanoid robots often possess 50 degrees of freedom. Even if each degree of freedom is partitioned very roughly into five intervals, the total number of discrete regions necessary to cover the entire state space is $5^{(50\times 2)} = 5^{100} \approx 10^{70}$, which is astronomical since each degree of freedom necessitates position and velocity representations. Optimal (or suboptimal) solutions to the reinforcement learning problem are paths connecting some of these discrete regions;

these should be found by trial and error. Even if each region was examined with sub-millisecond duration, the solution would not be found within the life of the universe because of the enormous number of discrete regions. Human bodies possess at least 1000 degrees of freedom because the number of small muscles around the spine, important for posture control, is enormous. In the brain, neurons with population coding and overlapping tuning curves should replace the previously mentioned discrete partition of the state space, and 1000 degrees of freedom needs to be handled. Thus, a simple reinforcement learning algorithm cannot be expected to perform any faster than an artificial humanoid robot. Second, in most real-life examples of reinforcement learning, a reward is given to an animal only at the end of a long behavioral sequence. For example, meats can be eaten only at the end of a hunting tour lasting a few days, rice can be harvested only several months after seeds are sown, and entrance to university can be attained only after several years of hard study. The reward obtained at the terminal state of a sequence of behavioral decisions should be backtracked and temporal credit assignment to each decision should be computed. Some slow synaptic plasticity processes should propagate the representation of reward prediction at one time step to its preceding representations. Fully propagating the reward prediction from the terminal state to the initial state usually requires a very long time.

The three difficulties associated with plain reinforcement learning – slowness, computation mechanism for TD error, and incompatibility with brain structure – all necessitate specific structures, frameworks, and neural mechanisms in the brain for efficient learning algorithms, if the brain still adopts the essence of plain reinforcement learning algorithms. In computational and engineering studies aimed at finding efficient reinforcement learning algorithms, the following factors, at least, have been proposed and examined as accelerating mechanisms: hierarchy, modularity, blending with supervised learning, internal models, imitation learning, and preconditioning of search space. The first three factors are reviewed with several new models to resolve the difficulties.

Hierarchy and Modularity

If the number of regions to be explored for finding optimal solutions is astronomical, then solutions cannot be found within a reasonable amount of time, as stated previously. One possible solution to resolve this difficulty is to introduce a hierarchy in reinforcement learning algorithms. Consider a simple two-level hierarchy. In the upper hierarchy, the state space for the problem is very coarsely represented by a reasonable number of discrete regions, cells, or neurons. Because the number is not astronomically large, an optimal solution can be found in a reasonable amount of time at the upper hierarchy, but the representation of the solution is very approximate due to the coarse representation. In the lower hierarchy, the state space is finely represented by an enormous number of discrete regions, cells, or neurons. The approximate solution obtained at the upper level can constrain the possible range of the space in which the optimal solution at the fine level is explored. Thus, the reinforcement learning problem could be solved within a reasonable amount of time and with the required accuracy. The large-scale versus small-scale contrast between the higher and lower hierarchies applies not only to spatial representation but also to temporal representation. Multiple levels of hierarchy can also be introduced. However, the following theoretical issues arise from the introduction of a hierarchy. First, how to introduce coarse representations in the upper hierarchies is not at all trivial, especially if this process should be executed automatically. Second, some consistency should be maintained between different levels of the hierarchy regarding the state space representation, the trajectories and optimal solutions, the reward prediction, the reward prediction error, and the reward. These theoretical issues were studied in several proposals of hierarchical reinforcement learning algorithms (Composite Q, Feudal Q, HQ-learning, HAM, and Option). In most of these proposals, the researchers needed to specify coarse representations of the state space in the upper hierarchy. In some of them, reinforcement learning does not take place in either the upper or the lower hierarchy; that is, behaviors (policy) and reward expectation (value function) need to be prespecified by researchers. In many of the proposals, even local optimality of the obtained solutions is not theoretically guaranteed due to the lack of consistency between higher and lower hierarchies. Because of these limitations, most of the examples examined in previous proposals were discrete time and discrete space problems, such as navigation in a two-dimensional maze. The shortcomings of previous hierarchical reinforcement learning algorithms were quite serious obstacles to viewing the algorithms as possible computational theories of brain learning. In particular, the fact that researchers need to specify coarse representations in the upper hierarchy, or need to predetermine motor primitives or subgoals in the lower hierarchy, is nearly equivalent to assuming the existence of a homunculus or that those factors are all predetermined genetically, both of which are unacceptable from the viewpoint of neuroscience. One of the very rare exceptions among the continuous time and continuous state space examples was a hierarchical

reinforcement learning algorithm for standing up by a real multijoint robot.

One of the strategies common to the military, politics, and engineering is to deal with complicated and large-scale problems in a 'divide and conquer' manner. That is, to solve a difficult large-scale problem, it is first partitioned into several smaller and more tractable subproblems. Then, a responsible expert deals with each subproblem to obtain a partial solution. Finally, the obtained partial solutions are combined and a global solution to the original problem is obtained. For example, if a given problem is to move from a southern area of Kyoto prefecture in Japan to an Italian village in Sicily, taking an airport limousine to Kansai International airport is a subproblem and should be much more easily solved than the original problem. This modular approach is another promising strategy that has been explored in the context of reinforcement learning. Actually, most hierarchical reinforcement learning methods either explicitly or implicitly adopt the 'modular' approach. However, not all modular approaches are hierarchical. The most critical and difficult issue in modular reinforcement learning is how to divide or modularize the original problem into subproblems. Again, it is unsatisfactory from the viewpoint of neuroscience if researchers deal with this manually. Another difficult and important theoretical issue is to guarantee the global optimality of the entire problem since each expert deals mainly with a single subproblem and is essentially narrow-scoped. Several theoretical studies have addressed this global optimality issue.

Dividing the original problem based on space or time seems to be the most natural and efficient method for a modular approach. In a discrete state and discrete time example, such as a maze, this approach is frequently used because the division point is often very obvious, such as doors segregating different rooms and/or corridors, and the time division could follow the space division. However, note that the division of even discrete examples is still executed by researchers. Continuous state and continuous time examples, with which brains must deal, are much more complicated to modularize. In the continuous case, there is no apparent division point in either space or time. Furthermore, for neurons in the brain, which are involved in reinforcement learning of some problem, the coordinate system appropriate for the problem and its subdivisions is not explicitly provided. Each neuron receives sensory, motor, and reward information related to the problem, but there is no global picture of the whole problem and the coordinate system available to each neuron. Even in this difficult situation, modularity should spontaneously emerge via self-organization within neural networks.

One promising approach for self-organization of modularity is a modular selection and identification control (MOSAIC) model. MOSAIC was originally proposed for supervised learning in sensorimotor integration. Doya and colleagues extended MOSAIC to reinforcement learning, and further developments have been made. In the most advanced form of reinforcement MOSAIC, three different sets of expert networks self-organize through reinforcement learning (**Figure 1**). One set of experts includes predictors of state transitions. These can formally be called internal forward models. An internal forward model receives the current state and the current motor command and predicts the next time step state. In interpretations of neuroscience, forward models receive sensory feedback and an efference copy of motor commands and then predict the sensory feedback from the next time step. In MOSAIC, many forward models compete and cooperate to predict the state change based on the goodness of prediction of each forward model. That is, many forward models are different from each other; thus, they make different predictions despite receiving identical sensory feedback and motor command. Their predictions are compared with the actual sensory feedback at the next time point, and a responsibility signal for each forward model is computed. The responsibility signal for a forward model is larger when its prediction is better. The entire prediction by MOSAIC is the linear weighted sum of predictions made by its forward models, which are weighted by their responsibility signals. Two other sets of experts are for approximators of actual rewards and reinforcement learning controllers. The responsibility signals and linear weighted summation are computed similarly for these two sets of experts; however, the prediction error for the approximators of rewards involves errors in reward approximation, whereas that for the controllers involves reward prediction errors such as the TD error. The responsibility signal also gates the learning of forward models, reward approximators, and controllers by regulating learning rate by its amplitude. That is, if the responsibility signal of one expert is large, then the expert learns more. On the other hand, if its responsibility signal is small and close to zero, the expert learns little. Within this framework, three sets of experts self-organize while guided by three different kinds of error signal: prediction error in dynamics, approximation error in rewards, and reward prediction error. Consequently, modularity self-emerges without ad hoc manual tuning by researchers.

If the original reinforcement learning problem is rather simple, with a simple linear dynamics and a simple reward function shape, then MOSAIC will utilize only one forward model and only one reward

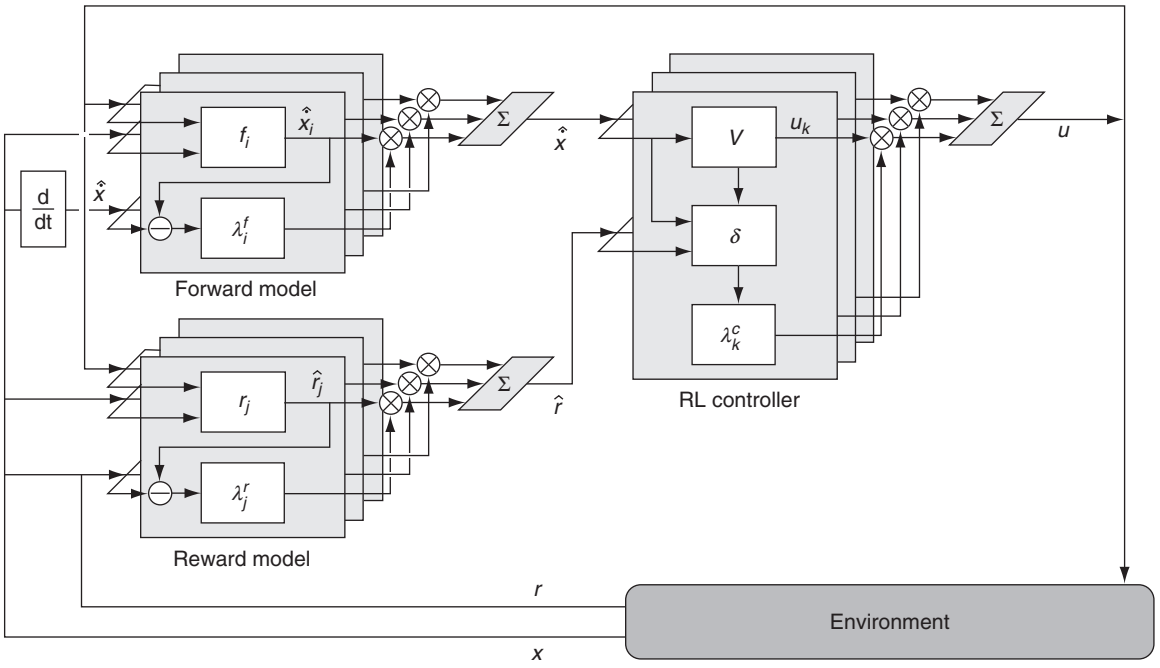

Figure 1 Overall organization of the combinatorial module-based reinforcement learning (CMRL) architecture. The architecture consists of forward modules (top left), reward modules (bottom left), and reinforcement learning (RL) controllers (right). The forward modules receive the state and action, and they segregate the environmental dynamics into several subdynamics based on the prediction error of the dynamics. Each reward module represents the local reward function, conditioned on the state variable (position and velocity) and the error of approximated reward. Then, an RL controller approximates the local value function for each configuration of a forward module and a reward module and outputs the local action. Finally, the local actions are weighted based on the squared TD errors of each RL controller. In real-world control problems, reinforcement learning has to handle nonstationary environments, in which both the dynamics of control and how the consequence of the control is rewarded change over time. One of promising strategies to deal with nonstationary environments is to decompose a complex task into several subtasks and adapt multiple RL modules for each of these subtasks. A basic assumption of this modular architecture is that the environment can be approximated as stationary within a single subtask. Previous studies of modular architecture for control attempted to separate a complex task based on the prediction error of the environmental dynamics and reported fairly good results for nonstationary dynamics. However, none of these studies addressed the dual nonstationary problem consisting of the dynamics and reward. To make RL applicable to such situations, the CMRL decomposes a complex task based on both the prediction error of the environmental dynamics and the approximation error of the reward function. First, the forward module (f_i) predicts the environmental dynamics ($\dot{x}$) from the current state and action (x, u). Second, the reward module (r_j) represents the local reward function conditioned on the state. The RL module, receiving outputs from forward and reward modules, approximates local value functions (V_k) and outputs the local greedy action (u_k). Finally, the CMRL selects actions by the sum of these local actions weighted by the responsibility signal of each RL module (λ^c_k), a posterior probability obtained from the squared TD errors (δ^2_k) of each RL module (likelihood) and the usage history of component forward and reward modules (priors). These priors can be recomputed as the responsibility signals of forward and reward modules (λ^f_i and λ^r_j) by applying Bayes rules to the prediction error of the environmental dynamics and reward approximation error, respectively. The CMRL can successfully cope with nonstationary problems such as the swinging of a pendulum, in multiple dynamics and reward conditions, and the bipedal locomotion of robots.

approximator. Accordingly, only one controller suffices to compute the necessary control commands. By contrast, if the dynamics of the original problem are highly nonlinear and consist of several qualitatively different dynamics, a set of forward models is recruited to approximate different segments of the original dynamics. If the dynamics are different, the necessary controllers are also different, even for the same reward function. Thus, a set of reinforcement learning controllers is recruited with corresponding forward models. In this nonlinear case, modular architecture dealing with nonlinear dynamics self-organizes. If the objective of the entire task changes from time to time, such as earning money and then spending it on a fan, then different reward approximators are recruited for different times, and different controllers are also needed. In this nonstationary case, modular architecture dealing with different task goals self-organizes. Thus, if the original problem contains nonlinearity and/or nonstationarity, MOSAIC automatically recruits its modular architecture to cope with the situation and self-organizes a set of controllers. A reinforcement MOSAIC model has also been extended to a hierarchical model.

Hierarchy and modularity are generally conceived as a common design principle for many different areas of the brain, based on an enormous number of previous anatomical and neurophysiological studies,

but direct support of their significance in the context of reinforcement learning mainly comes from neuroimaging studies on humans. For example, Haruno and colleagues found that the activities of many brain areas, such as the prefrontal cortex, orbitofrontal cortex, premotor cortex, supplementary motor area, cerebellum, and basal ganglia, are correlated with important variables in reinforcement learning, such as accumulated reward, learned behavior, short-term reward, and behavioral learning, even in a very simple Markov decision process with monetary reward. Furthermore, whereas the putamen is correlated with the reward prediction by action representation, which is consistent with a monkey study by Samejima and colleagues, the caudate is more correlated with the reward prediction error. Tanaka and colleagues found topographic representations of reward prediction in the medial prefrontal cortex and the insular cortex, and they found topographic representations of reward prediction error in the basal ganglia, with different values of discount factor, which is a very important parameter balancing future and immediate rewards in reinforcement learning algorithms.

How Can TD Error be Computed?

TD error plays the most critical role in solving the temporal credit assignment problem in reinforcement learning theory. Thus, Houk et al. proposed an explicit neural circuit model for computing the TD error, as shown in Figure 2(b), when they first proposed the reinforcement learning model of the basal ganglia and dopamine neurons shown in Figure 2(a). Doya then proposed the modified model shown in Figure 2(c). In both models, the primary reward information, $r(t)$, is assumed to be carried to the substantia nigra compacta (SNc) dopamine neurons by direct excitatory inputs. By contrast, the reward expectation of the next time step, $V(t+1)$, minus the reward expectation at the current time step, $-V(t)$, is assumed to be computed within the basal ganglia network utilizing double-inhibitory and inhibitory connections, respectively, from the common source of the value function $V(t)$ representation in the striatum. These models are attractive from a theoretical standpoint because the same value function $V(t)$ is the common source for positive and negative terms in the TD error, thus satisfying the most basic assumption of reinforcement learning algorithms: consistency of Bellman equation is the source of the error signal that drives learning. Moreover, these models utilize the known neural network within the basal ganglia. However, no direct physiological evidence has been obtained to support these models. Furthermore, double inhibition may not be able to explain the burst firing of dopamine neurons in response to a cue signal predicting a future reward, as in Schultz's experiment, due to the membrane properties of dopamine neurons.

Previous anatomical studies have shown that major excitatory synaptic inputs to SNc originate from the pedunculopontine tegmental nucleus (PPN), as shown in Figure 3(a). Neurophysiological studies by Kobayashi and colleagues further suggest that excitatory inputs from the PPN to the SNc might play the most central role in computing TD error. In their experiments, monkeys performed saccade tasks with variable reward amounts, which were indicated by the shape of a fixation spot. Two populations of neurons whose firing rates varied with reward amount were observed. One population of neurons seemed to encode the primary or current reward $r(t)$, whereas the other population of neurons started to fire upon the presentation of the fixation spot and maintained their firing until the end of saccade or the delivery of the primary reward, even after the fixation spot was extinguished. In addition, the firing rates predicted the amount of the future reward, thus seeming to encode the predicted reward or the value function $V(t)$. These remarkable findings directly suggest two possible neural mechanisms for computing the TD error, as shown in Figures 3(b) and 3(c). In Figure 3(b), some intranuclear circuits within the PPN or SNc, or some membrane properties of dopamine neurons in SNc, execute either temporal difference or temporal differentiation (in the case of Doya's continuous time reinforcement learning) as shown by the box in Figure 3(b). In this case, the PPN provides all of the components necessary for computing the TD error. By contrast, the model shown in Figure 3(c) predicts that the primary reward information $r(t)$ and the expected reward at the next time step $V(t+1)$ are carried by excitatory inputs from the PPN to the SNc, whereas the inhibitory input from the striatum conveys the subtracted and predicted reward information at the current time-$V(t)$. The model shown in Figure 3(c) seems more plausible because (1) no cellular or circuit mechanisms for temporal difference or differentiation are known that involve the PPN or SNc and (2) the inhibitory inputs from the striatum to the SNc are well established and it would be quite odd if they did not contribute to computing the TD error.

Heterarchical Reinforcement Learning Model

The fact that $V(t)$ and $V(t+1)$ arise from different sources (Str and PPN) in the model of Figure 3(c) casts doubt on its fidelity to reinforcement learning theory: consistency of the Bellman equation. By contrast, in the models shown in Figures 2(b), 2(c), and 3(b),

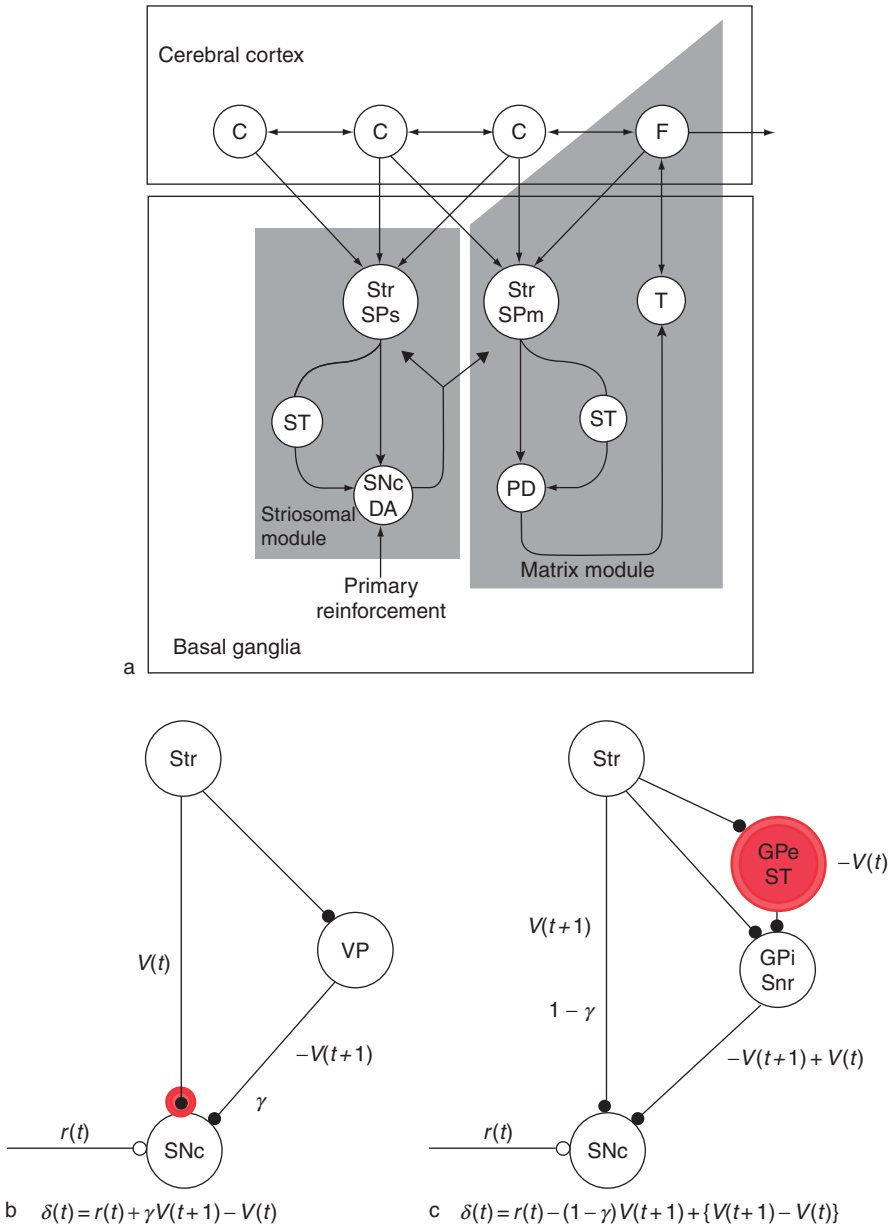

Figure 2 (a) Reinforcement learning model of the basal ganglia The modular organization of the basal ganglia, including both striosomal and matrix modules, is shown. C, cortical column composed of corticocoritical, corticothalamic, and other projection neurons; F, cortical column in the frontal cortex; Str, striatum; SPm, spiny neurons in the matrix compartment of the striatum; SPs, spiny neurons in the striosomal compartment of the striatum; SNc, substantia nigra pars compacta; DA, dopamine neurons; ST, subthalamic nucleus neuron; T, thalamocortical neuron; PD, pallidal neurons. (b, c) Houk versus Doya models of TD error computations. Possible mechanisms for TD computation of the value function represented in the basal ganglia are shown. Open circles ○ represent excitatory connections, whereas black dots ● represent inhibitory connections. The large red circle represents the source of possible time delay. SNr, substantia nigra pars reticulata; GPi and GPe, internal and external segments of the globus pallidus, respectively; VP, ventral pallidum. (a) From Houk JC, Adams JL, and Barto AG (1995) A model of how the basal ganglia generates and uses neural signals that predict reinforcement. In: Houk JC, Davis JL, and Beiser DG (eds.) *Models of Information Processing in the Basal Ganglia*, pp. 249–270. Cambridge, MA: MIT Press.

the TD error is computed from the same source of the value function; thus, they conform to the most basic principle of temporal difference learning. However, deviation from this principle may lend efficiency and power to the model in Figure 3(c). If subtraction can be computed in the SNc between PPN and striatal inputs, we can introduce a supervised learning aspect into reinforcement learning. That is, the learner $V(t)$ tries to approximate the teaching signal $r(t) + V(t+1)$ and the error signal is conveyed by dopamine neurons. Indeed, many robotics applications of reinforcement learning algorithms have been forced to incorporate

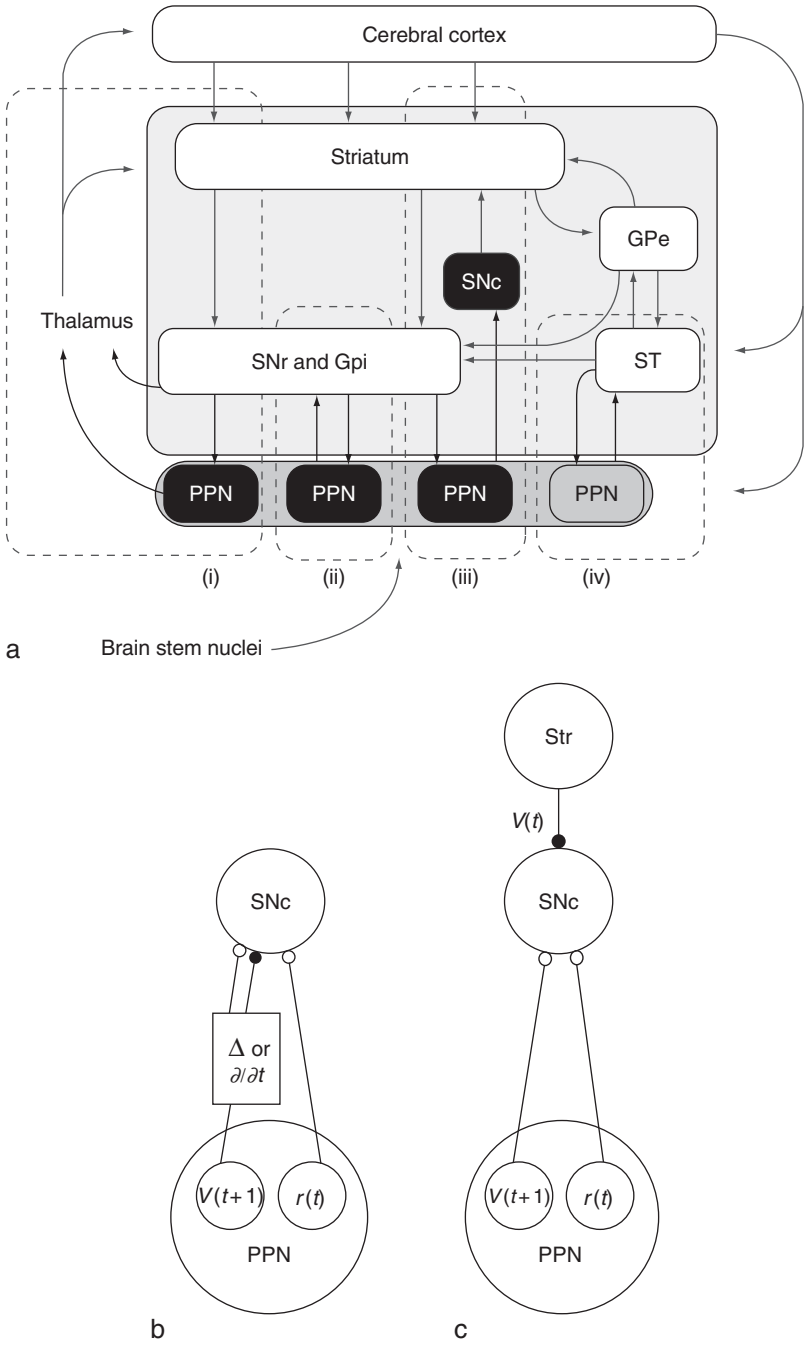

Figure 3 (a) Connections of the pedunculopontine tegmental nucleus (PPN) with the basal ganglia and cortex. (i) The PPN sends projections to the thalamic cells that project to the striatum; (ii) there are reciprocal connections between the PPN and the basal ganglia output nuclei (substantia nigra pars reticulata (SNr) and internal segment of the globus pallidus (GPi)), with the latter nuclei providing a dense inhibitory projection to the PPN; (iii) a substantial projection from the PPN innervates dopaminergic neurons of the substantia nigra pars compacta (SNc), which in turn modulate striatonigral and striatopallidal pathways; and (iv) a reciprocal, putative excitatory loop is formed between the PPN and the subthalamic nucleus neuron (ST). These complex interconnections imply that the PPN has a profound and widespread influence on basal ganglia activity and, similarly, that the basal ganglia are in a position to modulate PPN activity at many different levels. (b, c) Two possible neural circuit models for computing the TD error with an emphasis on excitatory inputs from the PPN to the SNc. Str, striatum. (a) From Mena-Segovia J, Bolam JP, and Magill PJ (2004) Pedunculopontine nucleus and basal ganglia: Distant relatives or part of the same family? *Trends in Neuroscience* 27: 585–588.

aspects of supervised learning to attain allowable learning time. The model shown in Figure 3(c) might represent a bridge between neuroscience and these efforts in robotics; furthermore, this model led to the proposal of the 'heterarchical' reinforcement learning model shown in Figure 4. In this figure, there exist two independent closed loops: the prefrontal cortex–caudate–ventral tegmental area–PPN cognitive loop

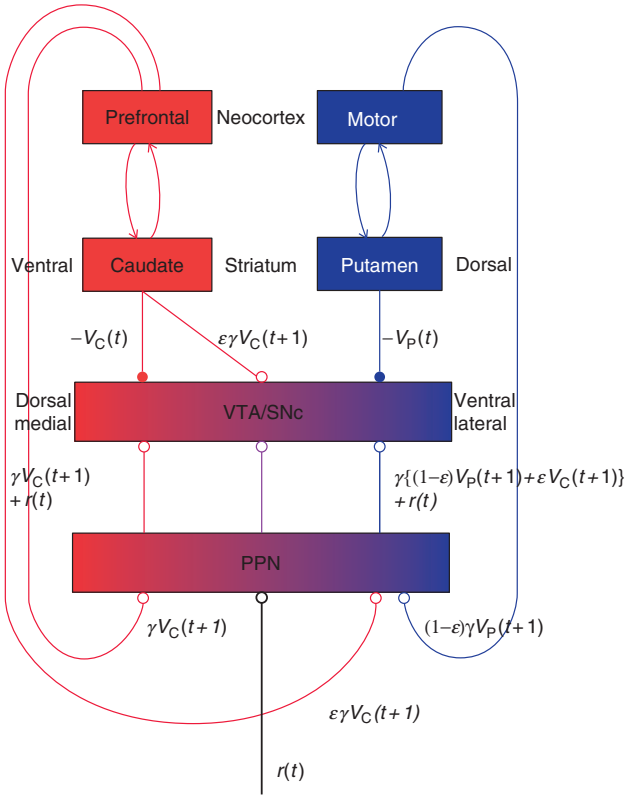

Figure 4 A basic heterarchical reinforcement learning model consisting of caudate–prefrontal and motor–putamen loops, the ventral tegmental area (VTA), the substantia nigra pars compacta (SNc), and the pedunculopontine tegmental nucleus (PPN). Open and solid circles show excitatory (disinhibitory) and inhibitory projections, respectively. V_C, a coarse value function in the caudate nucleus; V_P, a fine value function in the putamen; r, a reward; γ, a discount factor for future rewards; ε, a time-varying weighting factor between V_C and V_P.

and the motor cortex–putamen–SNc–PPN sensorimotor loop. If these two closed loops are entirely independent, then they can provide neural mechanisms for computing the TD error in a strictly reinforcement learning manner. However, the heterarchical structure shown in Figure 4 allows the cognitive loop value function to spread to the sensorimotor loop via neural connections from the caudate to the ventrolateral part of SNc, as well as through connections from the prefrontal cortex to the part of PPN innervating the ventrolateral part of SNc. This heterarchical model captures some aspects of the hierarchical reinforcement learning models introduced previously, but it does not have a strict hierarchy; thus, the name heterarchy was adopted. Representations of the state and actions in the cognitive loop are much coarser than those in the sensorimotor loop, so reinforcement learning progresses much faster in the cognitive loop than in the sensorimotor loop, just as with the higher and lower levels of hierarchical reinforcement learning algorithms. Then, the value function $V_C(t+1)$, which was learned earlier within the cognitive loop, plays the role of a teaching signal for the sensorimotor loop. That is, the sensorimotor (putamen) loop value function $V_P(t)$ is trained to approximate $V_C(t+1)$ during the early stages of learning. However, a much more fine-grained value function will be ultimately learned using the proper reinforcement learning implemented by the sensorimotor loop. Thus, fast but accurate learning is expected in the heterarchical reinforcement learning model, and no distinct boundaries should be manually set by researchers or homunculus. For simplicity, we explained the model using only two distinct closed loops, but a full model is rather continuous and realistic. This model was motivated by the neurophysiological studies of Kobayashi and colleagues and also by the anatomical studies of Haber on the spiral connections of corticostriatal and striatonigral loops. Kobayashi and colleagues characterized PPN neural firings as encoding intrinsic motivation or internal rewards. Because the PPN also receives inputs from brain stem nuclei, some portion of the future expected reward $V(t+1)$ could be represented based on quite coarse and endogenous factors, almost excluding sensorimotor or cognitive inputs. The coarse $V(t+1)$ could still be very helpful for developing more fine-tuned value functions based on sensorimotor and cognitive inputs, in a way similar to the technique known

as value function shaping in computational studies, and it could be a clue to understand the computational advantages of intrinsic motivation.

See also: Conditioning: Theories; Conditioning: Simple Neural Circuits in the Honey Bee; Delayed Reinforcement: Neuroscience; Delayed Reinforcement: Economics; Neuropsychology of Primate Reward Processes; Prediction Errors in Neural Processing: Imaging in Humans; Reward and Learning; Reward Processing: Human Imaging; Segawa Dopa Responsive Dystonia; Visual Cortical Models of Orientation Tuning.

Further Reading

Barto AG, Sutton RS, and Anderson CW (1983) Neuron-like elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics* 13: 835–846.

Dayan P and Hinton G (1993) Feudal reinforcement learning. *Advances in Neural Information Processing Systems* 5: 271–278.

Doya K (2000) Complementary roles of basal ganglia and cerebellum in learning and motor control. *Current Opinion in Neurobiology* 10: 732–739.

Haruno M and Kawato M (2006) Heterarchical reinforcement-learning model for integration of multiple cortico-striatal loops; fMRI examination in stimulus–action–reward association learning. *Neural Networks* 19: 1242–1254.

Houk JC, Adams JL, and Barto AG (1995) A model of how the basal ganglia generates and uses neural signals that predict reinforcement. In: Houk JC, Davis JL, and Beiser DG (eds.) *Models of Information Processing in the Basal Ganglia*, pp. 249–270. Cambridge, MA: MIT Press.

Kawato M (1999) Internal models for motor control and trajectory planning. *Current Opinion in Neurobiology* 9: 718–727.

Kobayashi Y, Inoue Y, Yamamoto M, et al. (2002) Contribution of pedunculopontine tegmental nucleus neurons to performance of visually guided saccade tasks in monkeys. *Journal of Neurophysiology* 88: 715–731.

Kobayashi Y, Okada K, Inoue Y, et al. (2002) Reward predicting activity of pedunculopontine tegmental nucleus neurons during visually guided saccade tasks. In: *Abstract of the 35th Annual Meeting of the Society for Neuroscience*, No. 890.5.

Mena-Segovia J, Bolam JP, and Magill PJ (2004) Pedunculopontine nucleus and basal ganglia: Distant relatives or part of the same family? *Trends in Neuroscience* 27: 585–588.

Morimoto J and Doya K (1999) Hierarchical reinforcement learning for motion learning: Learning 'stand up' trajectories. *Advances in Robotics* 13: 267–268.

Parr R and Russell S (1997) Reinforcement learning with hierarchies of machines. *Advanced in Neural Information Processing Systems* 10: 1043–1049.

Samejima K, Veda Y, Doya K, et al. (2005) Representation of action-specific reward values in the striatum. *Science* 310: 1337–1340.

Schultz W and Dickinson A (2000) Neuronal coding of prediction errors. *Annual Review of Neuroscience* 23: 473–500.

Singh S (1992) Transfer of learning by composing solutions of elemental sequential tasks. *Machine Learning* 8: 323–339.

Sugimoto N (2006) *Hierarchical Reinforcement Learning: Computational Model of Communication*. Nara Institute of Science and Technology, PhD Dissertation, No. NAIST-IS-DD0361013.

Sutton RS (1991) Planning by incremental dynamic programming. In: Brinbaum LA, Sutton RS, and Collins GC (eds.) *Proceedings of the Eighteenth International Workshop on Machine Learning*, pp. 353–357. San Mateo, CA: Morgan Kaufmann.

Sutton RS and Barto AG (1998) *Reinforcement Learning*. Cambridge, MA: MIT Press.

Sutton R, Precup D, and Singh S (1999) Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence* 112: 181–211.

Tanaka SC, Doya K, Okada K, et al. (2004) Prediction of immediate and future rewards differentially recruits cortico-basal ganglia loops. *Nature Neuroscience* 7: 887–893.

Wiering M and Schmidhuber J (1997) HQ-learning. *Adaptive Behavior* 6: 219–246.