immense but the true impact of drivers depends on their prevalence. Currently, it is unknown how many drivers are present. Answering this question is challenging because drive phenotypes can only be observed in heterozygotes, not all drivers are linked to an observable phenotype (e.g. sex), and drive may require a particular genetic background (e.g. the absence of drive suppressors). Even amongst the known meiotic drive systems, not much molecular information is known about many of them beyond their presence on a chromosome. Understanding the prevalence of drivers and the molecular mechanisms they use will lead to a greater understanding of how these parasites shape the evolution of eukaryotic biology, particularly gametogenesis.

Where can I find out more?

Akera, T., Trimm, E., and Lampson, M.A. (2019). Molecular strategies of meiotic cheating by selfish centromeres. Cell *178*, 1132–1144.e10.

Burt, A., and Crisanti, A. (2018) Gene drive: evolved and synthetic. ACS Chem. Biol. *13*, 343–346.

Cazemajor, M., Joly, D., and Montchamp-Moreau, C. (2000). Sex-ratio meiotic drive in *Drosophila simulans* is related to equational nondisjunction of the Y chromosome. Genetics *154*, 229–236.

Crow, J.F. (1991). Why is mendelian segregation so exact? *BioEssays 13*, 305–312.

Dawe, R.K., Lowry, E.G., Gent, J.I., Stitzer, M.C., Swentowsky, K.W., Higgins, D.M., Ross-Ibarra, J., Wallace, J.G., Kanizay, L.B., Alabady, M., *et al*. (2018). A kinesin-14 motor activates neocentromeres to promote meiotic drive in maize. Cell *173*, 839–850.

Dyer, K.A., Charlesworth, B., and Jaenike, J. (2007). Chromosome-wide linkage disequilibrium as a consequence of meiotic drive. Proc. Natl. Acad. Sci. USA *104*, 1587–1592.

Herrmann, B.G., Koschorz, B., Wertz, K., McLaughlin, K.J., and Kispert, A. (1999). A protein kinase encoded by the *t complex responder* gene causes non-mendelian inheritance. Nature *402*, 141–146.

Larracuente, A.M., and Presgraves, D.C. (2012). The selfish Segregation Distorter gene complex of Drosophila melanogaster. Genetics *192*, 33–53.

Lindholm, A.K., Dyer, K.A., Firman, R.C., Fishman, L., Forstmeier, W., Holman, L., Johannesson, H., Knief, U., Kokko, H., Larracuente, A.M. *et al*. (2016). The ecology and evolutionary dynamics of meiotic drive. Trends Ecol. Evol. *31*, 315–326.

Sandler, L., and Novitski, E. (1957). Meiotic drive as an evolutionary force. Am. Nat. *91*, 105–110.

Zanders, S.E., and Unckless, R.L. (2019). Fertility costs of meiotic drivers. Curr. Biol. *29*, R512–R520.

[1]Stowers Institute for Medical Research, Kansas City, MO 64110, USA. [2]Department of Molecular and Integrative Physiology, University of Kansas Medical Center, Kansas City, KS 66160, USA.
*E-mail: sez@stowers.edu

Primer

The neurobiology of deep reinforcement learning

Samuel J. Gershman[1]
and Bence P. Ölveczky[2,*]

To generate adaptive behaviors, animals must learn from their interactions with the environment. Describing the algorithms that govern this learning process and how they are implemented in the brain is a major goal of neuroscience. Careful and controlled observations of animal learning by Thorndike, Pavlov and others, now more than a century ago, identified intuitive rules by which animals (including humans) can learn from their experiences by associating sensory stimuli and motor actions with rewards. But going from explaining learning in simple paradigms to deciphering how complex problems are solved in rich and dynamic environments has proven difficult (Figure 1). Recently, this effort has received help from computer scientists and engineers hoping to emulate intelligent adaptive behaviors in machines. Inspired by the animal behavior literature, pioneers in artificial intelligence developed a rigorous and mathematically principled framework within which reward-based learning can be formalized and studied. Not only has the field of reinforcement learning become a boon to machine learning and artificial intelligence, it has also provided a theoretical foundation for biologists interested in deciphering how the brain implements reinforcement learning algorithms.

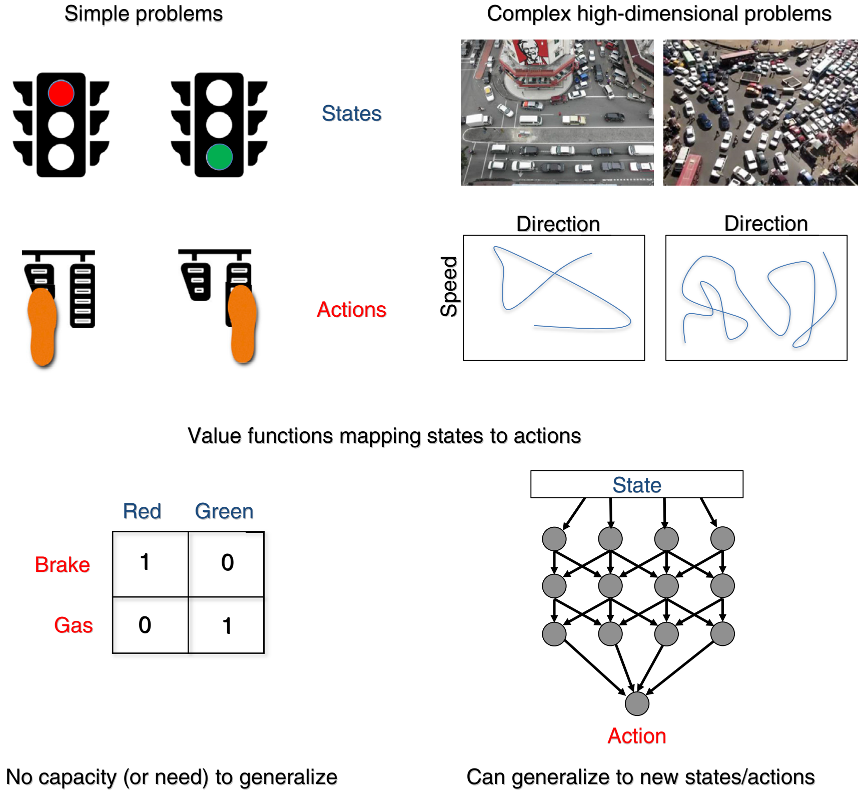
The ability of reinforcement learning agents to solve complex, high-dimensional learning problems has been dramatically enhanced by using deep neural networks (deep reinforcement learning, Figure 1). Indeed, aided by ever-increasing computational resources, deep reinforcement learning algorithms can now outperform human experts on a host of well-defined complex tasks, although significant gaps remain. The aim of this primer is not to review progress in this fast-moving field or compare various algorithmic implementations. Rather, we believe familiarity with the algorithms developed for machine learning can help neuroscientists better understand, in computationally precise ways, how humans and animals learn from interactions with their environments. Importantly, developments in deep reinforcement learning can help inspire new ideas about how the brain implements neural circuit-level solutions to these challenges.

In this primer, we will briefly review basic concepts of reinforcement learning, and discuss some of the shortcomings of traditional approaches and ways in which they can be overcome by using deep reinforcement learning. We then consider how the brain might implement some of the ideas from deep reinforcement learning, specifically: relative value coding; policy regularization; and efficient exploration of large solution spaces.

Solving reinforcement learning problems with deep neural networks

In reinforcement learning, the problem of learning from experience can be reduced to an agent (which can be a machine or an animal) interacting with its environment. At each step, t, of this interaction, the agent observes the state of the world, s(t), and enacts a policy which determines its action, a(t). Each action, in turn, results in a reward, r(t), and changes the state to s(t+1). The goal is to learn a policy that maps states into actions in a way that maximizes cumulative reward (value) over some time horizon. One way to learn the optimal policy is to explore different actions in different states and use reward feedback to update the estimated value of taking a given action in a given state. Deviation from the value estimate is called the reward prediction error, which is used to update the estimates, and through it, the policy.

In the real world, both states and actions are continuous and high-dimensional, which means an agent cannot construct a look-up table for the values of each state–action pairing (the ‘value function’, Figure 1). This is known as the *curse of dimensionality*. One way to overcome this problem is to find low-dimensional representations


Simple problems
Complex high-dimensional problems
States
Actions
Direction
Direction
Speed
Value functions mapping states to actions
Red
Green
Brake
Gas
1
0
0
1
State
Action
No capacity (or need) to generalize
Can generalize to new states/actions

	Red	Green
Brake	1	0
Gas	0	1

Current Biology

Figure 1. Deep reinforcement learning can find general solutions to complex real-world problems.
Left: simpler problems, such as selecting the right action in response to traffic lights, can be represented in a tabular form. Right: for more complicated problems, such as how to negotiate traffic in a crowded city, both the states (road condition, traffic, weather, and so on) and the possible actions (speed, direction, and so on) are continuous and high-dimensional. Deep neural networks are effective tools for approximating the complex high-dimensional mappings between state–action pairs and their values. If properly trained and regularized, these networks can generalize to new states and actions, such as driving in a new city. Traffic image used with permission from Jason Thien (CC BY 2.0).

of high-dimensional state and action spaces, such that good policies can be learned in more tractable domains. For example, if rewards vary smoothly across space (a plausible assumption for a foraging animal), then representing states by their coordinates in Euclidean space will allow an agent to generalize effectively: discovering reward in one region of space increases the value of neighboring regions. However, a useful representation cannot always be determined *a priori*. If the spatial distribution of rewards depends on other factors (for example, terrain, season, patch depletion and renewal rates), then the value function will not be well approximated by a simple function of spatial location.

Classical reinforcement learning algorithms typically use fixed and predetermined representations of state and action spaces (such as the Euclidean coordinate system in our example above). As we've already seen, the choice of representation can limit the capability of reinforcement learning agents. Modern deep reinforcement learning algorithms overcome this problem by *learning* the state and action representations alongside the values of state–action pairs and the policy, a process known as 'end-to-end' learning. This is accomplished by using deep neural networks, which consist of neuron-like non-linear processing units that, when connected in a network, can mimic aspects of the computations our brains perform (Figure 1). For reinforcement learning, a commonly used architecture is a feedforward network, in which units are arranged into layers, with dense connections between units in adjacent layers and no connections between units within a layer.

Such deep neural networks learn efficient representations by mapping states and actions to values and adjusting the parameters of the network to maximize future rewards. Signals are propagated in one direction through a hierarchy of layers, starting with 'raw' sensory inputs (state information) and culminating with a scalar value estimate for each action. At each layer of the network, some details of the input are lost, creating an 'information bottleneck' that forces the network (during learning) to find compact (low-dimensional) representations of the state information useful for reinforcement learning.

Importantly, deep neural networks can learn to represent any continuous function, making them well-suited to approximate complex high-dimensional mappings between state–action pairs and their values. This property comes from large neural networks having many free parameters — the weights of connections between neurons — that can be tuned to approximate nearly any input–output function (the universal function approximation property). The drawback is that deep networks can 'overfit' sparse data and hence may fail to properly generalize to new situations. This tension between the ability of a large network to represent any complex function (its *expressivity*) and the risk of overfitting with small data sets is a major issue in deep reinforcement learning, and one we discuss below.

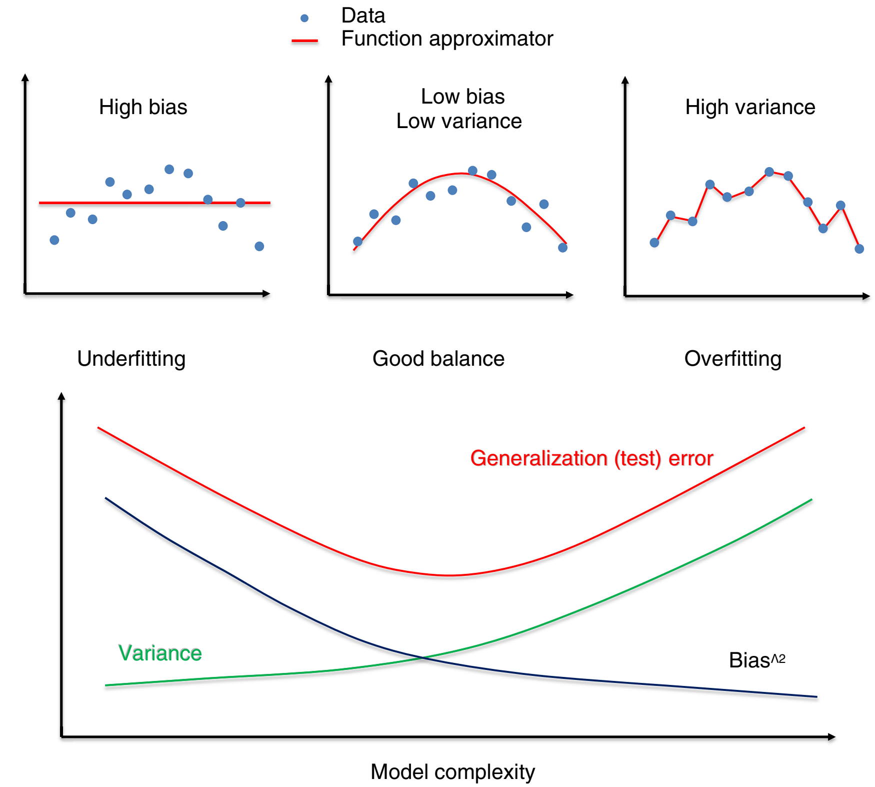
In the next sections, we highlight three algorithmic ideas that have shown promise for deep reinforcement learning and discuss the insights they provide for neuroscience. These examples are meant to be illustrative rather than exhaustive. We will return to some general conclusions at the end of the primer.

Variance reduction using relative value coding

One way of formulating the tension between generalizability and overfitting is to say that deep neural networks have low bias (they can closely approximate the correct function given enough data) and high variance (approximation errors will tend to be large for small data sets). Low bias is good, because it means your predictions won't be systematically wrong (Figure 2). High variance is a problem because it leads to unreliable generalization accuracy, and results from deep networks being very sensitive to randomness in small data sets: each time you train them with different samples, they'll give you a different approximation. To ensure that deep reinforcement learning acquires good policies from relatively few samples, it is important to manage this bias–variance tradeoff by reducing the variance without introducing too much bias (Figure 2).

One source of variance comes from action-independent fluctuations in reward. As an illustration, imagine optimizing your choice of food (for example, chicken or steak) at restaurants in your city. Because the restaurants (the 'states') will vary in their 'goodness', the reward of choosing a given type of food (the 'action') can be largely independent of the action itself. This effectively adds noise to your policy, in the sense that it will take longer to learn the value of each state–action pair from experience. This is particularly problematic for deep reinforcement learning, because the deep function approximator can more easily fit noise compared to 'shallow' function approximators. An effective way to reduce this variance is to learn the values of actions (chicken or steak) relative to the average action value in a particular state (the 'goodness' of the restaurant). These relative values are known as 'advantages', and learning them helps deep networks generalize from sparse observations without adding excessive bias.

There is increasing evidence that the brain learns and codes the value of actions and stimuli in terms of 'advantages'. For example, neurons in orbitofrontal cortex — an area of the brain involved in decision making and reward processing — represent outcome values relative to other options available in a given state. Much like sensory neurons, the firing rates of value coding neurons are further normalized to fit the range of values likely to be encountered. Deep reinforcement learning gives a computational rationale for why the brain, a network tasked with generalizing from sparse observations, should use relative value codes.



Current Biology

Figure 2. The bias–variance tradeoff.
Top: illustrative one-dimensional function learning example. Each dot corresponds to an input–output pair generated from a noisy function (in reinforcement learning problems, the relevant function would map state–action pairs to values). The lines in each plot show functions from different families (polynomials of different orders) fit to the data. Simple functions, such as linear or lower-order polynomials, have high bias but low variance: they underfit both signal and noise. Complex functions, such as higher-order polynomials, have low bias but high variance: they overfit both signal and noise. In this example, a second-order (quadratic) polynomial function appears to adequately balance bias and variance. Deep neural networks are typically thought of as low bias and high variance estimators, necessitating regularization to prevent overfitting. Bottom: generalization error can be decomposed into the sum of bias and variance, implying that minimum error is achieved when the two are balanced.

Policy regularization in neural networks

Another way to manage the bias–variance trade-off is to make sure policies don't overfit sparse data by 'regularizing' them — penalizing them based on their complexity. Take, for example, superstitious rituals like always eating orange food out of a cardboard box while wearing a pointy hat before your team plays a match. Such a policy might appear to be weakly correlated with your team winning, but that's most likely due to chance. Regularization techniques ensure that such policies are disfavored unless there is strong evidence that they work.

Clues as to whether to implement policy regularization can come from considering reward prediction errors. When predictions of a deep network are consistently poor, it is typically due to overfitting — a situation in which low variance solutions (stronger policy regularization) would be preferred (Figure 2). On the other hand, if predictions are good, increased variance could help nudge the system towards more complex (and possibly more accurate) policies. Applying this idea to neurobiology could help explain why dopamine depletion, as for example in Parkinson's disease, causes increased motor variability. In a normal brain, reduction in phasic dopamine means predictions are becoming better (errors are small) — a regime in which higher variability (weaker regularization) could be beneficial.

The link between prediction errors and regularization could also explain why variation in genes that regulate the effects of striatal dopamine are linked to choice variability in reinforcement learning tasks. However, deep networks with high bias would also produce consistently poor predictions, which could be remedied by the agent exploring new parts of policy space, meaning higher — not lower — policy variance. Some of this ambiguity in interpreting the prediction error signal can, in theory at least, be resolved by considering its statistics. Whether the brain does this is not known.

Deep reinforcement learning may also lend some intuition into how the brain implements policy regularization. One way to regularize deep networks is by penalizing strong connections, which can sparsify network connectivity (many connection strengths are pushed towards zero). Beyond policy regularization, such sparsification can make network implementations faster and more reliable. Similarly, synaptic pruning in neural circuits is associated with many forms of learning and is a hallmark of critical period plasticity. Analogies to deep reinforcement learning suggest that it may be a mechanism for the brain to regularize learned policies, making their implementations more efficient and robust.

Balancing exploration and exploitation in neural networks

Linked to the issue of how variability is regulated is the exploration–exploitation dilemma in reinforcement learning. At each time step, the agent must decide whether to go with the action predicting the highest reward or to explore actions previously not taken. Exploration can lead to the identification of options that are better in the long run. A classic approach for balancing exploration and exploitation is to guide exploration towards states and actions that haven't been visited very often (ones for which the uncertainty about the value is high). This is sometimes referred to as 'count-based exploration' and has strong theoretical guarantees. However, this cannot be directly applied to high-dimensional continuous state-action spaces where the same state–action pair is rarely (if ever) revisited.

One way to implement efficient exploration in vast state/action spaces is to randomize the value function. Because policies in deep reinforcement learning are usually parametrized by a neural network with trained weights, randomizing the value function has the effect of adding exploratory noise to the policy (a strategy known as *random exploration*). It turns out that efficient exploration can be accomplished by adding noise to network weights at the time of action selection.

In line with reinforcement learning theory, behavioral studies in both humans and animal models have suggested that the brain regulates exploratory variability to promote learning. As in deep reinforcement learning, adding noise to the weights in the underlying neural network is a plausible mechanism. For short timescales, this noise could be driven by the probabilistic nature of synaptic transmission, whereas longer timescale exploration could be caused by slower structural changes in dendritic spine volume and morphology, which are known to fluctuate continuously and stochastically over time. We conjecture that these stochastic fluctuations drive noise in valuation circuits, thus randomizing the value function. The contributions of different timescales have not yet been systematically addressed in the deep reinforcement learning literature; doing so may lead to new and more effective multi-scale exploration algorithms.

Conclusions

Given that deep networks were inspired by neuroscience and are effective in part because they have properties also found in real neural circuits, a seductive hypothesis is that the brain implements deep reinforcement learning. As we've alluded to here, this analogy can shed light on several aspects of brain function. Although the computational issues we've addressed — bias–variance and exploration–exploitation dilemmas — apply to any reinforcement learning algorithm, they are especially salient for deep neural networks (and presumably also biological networks), because of their high degree of expressivity.

At the same time, there are ways in which the analogy breaks down: deep networks require much more training data than biological brains (particularly those of humans), their learning algorithms (in particular backpropagation) make biologically implausible assumptions, and they lack the cognitive flexibility of real brains. All of these challenges are currently the focus of intense research efforts. Thus, we are hopeful that the analogy, once suitably nuanced, will provide a useful framework for thinking about reinforcement learning in the brain.

FURTHER READING

Cheng, R., Verma, A., Orosz, G., Chaudhuri, S., Yue, Y., and Burdick, J. (2019). Control Regularization for Reduced Variance Reinforcement Learning. In International Conference on Machine Learning. pp. 1141–1150.

François-Lavet, V., Henderson, P., Islam, R., Bellemare, M.G., and Pineau, J. (2018). An introduction to deep reinforcement learning. Found. Trends Mach. Learn. *11*, 219–354.

Llera-Montero, M., Sacramento, J., and Costa, R.P. (2019). Computational roles of plastic probabilistic synapses. Curr. Opin. Neurobiol. *54*, 90–97.

Louie, K., Glimcher, P.W., and Webb, R. (2015). Adaptive neural coding: from biological to behavioral decision-making. Curr. Opin. Behav. Sci. *5*, 91–99.

Niv, Y. (2009). Reinforcement learning in the brain. J. Math. Psychol. *53*, 139–154.

Plappert, M., Houthooft, R., Dhariwal, P., Sidor, S., Chen, R.Y., Chen, X., Asfour, T., Abbeel, P. and Andrychowicz, M. (2018). Parameter Space Noise for Exploration. International Conference on Learning Representations (ICLR).

Schulman, J., Moritz, P., Levine, S., Jordan, M.I., and Abbeel., P. (2016). High-dimensional continuous control using generalized advantage estimation. International Conference on Learning Representations (ICLR).

[1]Department of Psychology and the Center for Brain Science, Harvard University, Cambridge, MA 02138, USA. [2]Department of Organismic and Evolutionary Biology and the Center for Brain Science, Harvard University, Cambridge, MA 02138, USA.
*E-mail: olveczky@fas.harvard.edu