The foundation of efficient robot learning

Citation: Kaelbling, Leslie Pack et al. "The foundation of efficient robot learning." Science 369, 6506 (August 2020): 915-916. © 2020 The Author

As Published: http://dx.doi.org/10.1126/science.aaz7597

Publisher: American Association for the Advancement of Science (AAAS)

Persistent URL: https://hdl.handle.net/1721.1/130244

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

ARTIFICIAL INTELLIGENCE

The foundation of efficient robot learning

Innate structure reduces data requirements and improves robustness

By **Leslie Pack Kaelbling**

The past 10 years have seen enormous breakthroughs in machine learning, resulting in game-changing applications in computer vision and language processing. The field of intelligent robotics, which aspires to construct robots that can perform a broad range of tasks in a variety of environments with general human-level intelligence, has not yet been revolutionized by these breakthroughs. A critical difficulty is that the data needed for learning comes from acting in the world, making it costly to acquire, especially because there is enormous variability in the situations a general-purpose robot must cope with. It will take a combination of new algorithmic techniques, inspiration from natural systems, and multiple levels of machine learning to revolutionize robotics with general-purpose intelligence.

Most of the successes in deep-learning applications have been in supervised machine learning, a setting in which the learning algorithm is given paired examples of an input and a desired output and it learns to associate them. For robots that execute sequences of actions in the world, a more appropriate framing of the learning problem is reinforcement learning (RL) (*1*), in which an "agent" learns to select actions to take in its environment in response to a "reward" signal that tells it when it is behaving well or poorly. One essential difference between supervised learning and RL is that the agent's actions have substantial influence over the data it acquires; the agent's ability to control its own exploration is critical to its overall success.

The original inspirations for RL were models of animal behavior learning through reward and punishment. To apply to interesting real-world problems, RL has to be extended to handle very large spaces of inputs and actions and to work when the rewards may arrive long after the critical action was chosen. New "deep" RL (DRL) methods, which employ complex neural networks with many layers, have met these challenges and resulted in stunning performance, including solving the games of Chess and Go (*2*) and physically solving Rubik's cube with a robot hand (*3*). They have also seen useful applications, including improving energy efficiency in computer installations. Based on these successes, it is tempting to imagine that RL might completely replace traditional methods of engineering for robots and other systems with complex behavior in the physical world.

Computer Science and Artificial Intelligence Laboratory and the Center for Brains, Minds, and Machines, Massachusetts Institute of Technology, 32 Vassar St., Cambridge, MA, USA. Email: lpk@csail.mit.edu

There are technical reasons to resist this temptation. Consider a robot that is designed to help in an older person's household. The robot would have to be shipped with a considerable amount of prior knowledge and ability, but it would also need to be able to learn on the job. This learning would have to be: sample-efficient (require relatively few training examples), generalizable (apply to many more situations than the one(s) it learned), compositional (be represented in a form that allows it to be combined with previous knowledge), and incremental (be capable of adding new knowledge and abilities over time). Most current DRL approaches do not have these properties: they can learn surprising new abilities, but generally require a lot of experience, do not generalize well, and are monolithic during training (non-incremental) and execution (non-compositional).

How can sample efficiency, generalizability, compositionality, and incrementality be enabled in an intelligent system? Modern neural networks have been shown to be effective at interpolating: Given a large number of parameters, they are able to remember the training data and make reliable predictions on similar examples (*4*). To obtain generalization, it is necessary to provide "inductive bias," in the form of built-in knowledge or structure, to the learning algorithm. As a simple example, an autonomous car with an inductive bias that its braking strategy need only depend on cars within a bounded distance of it could learn from relatively few examples because of the limited set of possible strategies that would fit well with the data it has observed. Inductive bias, in general, increases sample efficiency and generalizability. Compositionality and incrementality can be obtained by building in particular types of structured inductive bias, in which the "knowledge" acquired through learning is decomposed into factors with independent semantics that can be combined to address exponentially more new problems (*5*).

The idea of building in prior knowledge or structure is somewhat fraught. Richard Sutton, a pioneer of RL, asserts (*6*) that humans should not try to build any prior knowledge into a learning system, because, historically, whenever we try to build something in, it has been wrong. His essay incited strong reactions (*7*), but it gets at the critical question in the design of a system that learns: what kinds of inductive bias can be built into a learning system that will give it the leverage it needs to learn generalizable knowledge from a reasonable amount of data while not incapacitating it through inaccuracy or over-constraint?

There are two intellectually coherent strategies for finding an appropriate bias, which themselves have different time-scales and trade-offs, that can be used together to discover powerful and flexible prior structures for learning agents. One strategy is to use the techniques of machine learning at the "meta" level. That is, use machine learning offline at system design time (in the robot "factory") to discover the structures, algorithms, and prior knowledge that will enable it to learn efficiently online when it is deployed (in the "wild").

The basic idea of meta-learning has been present in machine-learning and statistics since at least the 1980s (*8*). The fundamental idea is that, in the factory, the meta-learning process has access to many samples of possible tasks or environments that the system might be confronted with in the wild. Rather than trying to learn strategies that are good for an individual environment, or even a single strategy that works well in all the environments, a meta-learner tries to learn a learning algorithm that, when faced with a new task or environment in the wild, will learn as efficiently and effectively as possible. It can do this by inducing the commonalities among the training tasks and using them to form a strong prior or inductive bias that allows the agent in the wild to learn only the aspects that differentiate the new task from the training tasks.

Meta-learning can be very beautifully and generally formalized as a type of hierarchical Bayesian (probabilistic) inference (*9*), in which the training tasks can be seen as providing evidence about what the task in the wild will be like, and using that evidence to leverage data obtained in the wild. The

Bayesian view can be computationally difficult to realize, however, because it requires reasoning over the large ensemble of tasks experienced in the factory that might potentially be the actual task in the wild.

Another approach is to explicitly characterize meta-learning as two nested optimization problems. The inner optimization happens in the wild: The agent tries to find the hypothesis from some set of hypotheses generated in the factory that has the best “score" on the data it has in the wild. This inner optimization is characterized by the hypothesis space, the scoring metric, and the computer algorithm that will be used to search for the best hypothesis. In traditional machine learning, these ingredients are supplied by a human engineer. In meta-learning, at least some aspects are instead supplied by an outer “meta" optimization process that takes place in the factory. Meta-optimization tries to find parameters of the inner learning process itself that will enable the learning to work well in new environments that were drawn from the same distribution as the ones that were used for meta-learning.

Recently, a useful formulation of meta-learning, called "model-agnostic meta-learning" (MAML) has been reported (*10*). MAML is a nested optimization framework in which the outer optimization selects initial values of some internal neural network weights that will be further adjusted by a standard gradient-descent optimization method in the wild. The RL2 algorithm (*11*) uses DRL in the factory to learn a general small program that runs in the wild but does not necessarily have the form of a machine-learning program. Another variation (*12*), seeks to discover, in the factory, modular building blocks, such as small neural networks, that can be combined to solve problems presented in the wild.

The process of evolution in nature can be considered an extreme version of meta-learning, in which nature searches a highly unconstrained space of possible learning algorithms for an animal (of course, in nature, the physiology of the agent can change as well). The more flexibility there is in the inner optimization problem solved during a robot's lifetime, the more resources, including example environments in the factory, broken robots in the wild, and computing capacity in both phases, is needed to learn robustly. In some ways, this returns us to the initial problem. Standard RL was rejected because, although it is a general-purpose learning method, it requires an enormous amount of experience in the wild. However, meta-RL requires substantial experience in the factory which could make development infeasibly slow and costly. Thus, perhaps meta-learning is not a good solution, either.

What is left? There are a variety of good directions to turn, including teaching by humans, collaborative learning with other robots, and changing the robot hardware along with the software. In all these cases it remains important to design an effective methodology for developing robot software. Applying insights gained from computer science and engineering together with inspiration from cognitive neuroscience can help to find algorithms and structures that can be built into learning agents and provide leverage to both learning in the factory and in the wild.

A paradigmatic example of this approach has been the development of convolutional neural networks (*13*). The idea is to design a neural network for processing images in such a way that it performs "convolutions:" Local processing of patches of the image using the same computational pattern across the whole image. This design simultaneously encodes the prior knowledge that objects have basically the same appearance no matter where they are in an image (translation invariance), and that it is groups of nearby pixels that are jointly informative about the content of the image (spatial locality). Designing a neural network this way means that it requires many fewer parameters and, hence, much less training, than doing so without convolutional structure. Where did the idea of image convolution come from? Both from engineers and from nature. It was a foundational concept in early signal processing and computer vision (*14*). And it has long been understood that there are cells in the mammalian visual cortex that seem to be performing a similar kind of computation (*15*).

It is necessary to discover more ideas like convolution, fundamental structural or algorithmic constraints that provide substantial leverage for learning but will not prevent robots from reaching their potential for generally intelligent behavior. Some candidate ideas include the ability to do some form of forward search using a "mental model" of the effects of actions, similar to planning or reasoning; the ability to learn and represent knowledge that is abstracted away from individual objects, but can be applied much more generally (e.g., for all A and B, if A is on top of B and I move B then A will probably move too); and the ability to reason about 3-dimensional space, including planning and executing motions through it as well as using it as an organizing principle for memory. There are likely many other such plausible candidate principles. Many other problems will also need to be addressed, including developing infrastructure for training both in the factory and in the wild, and methodologies for helping humans specify the rewards and for maintaining safety. It will be through a combination of engineering principles, biological inspiration, learning in the factory, and ultimately learning in the wild that generally intelligent robots can finally be created.

REFERENCES AND NOTES

1. A. Barto *et al.*, *IEEE Trans. Syst. Man Cyb. SMC*-13, 834 (1983).
2. D. Silver *et al.*, *Science* **362**, 1140 (2018).
3. OpenAI, *arXiv* 1910.07113 (2019).
4. M. Belkin *et al.*, *Proc. Natl. Acad. Sci. U.S.A.* **116**, 15849 (2019).
5. P. Battaglia *et al.*, *arXiv* 1806.01261 (2018).
6. R. Sutton, www.incompleteideas.net/IncIdeas/BitterLesson.html
7. R. Brooks, https://rodneybrooks.com/a-better-lesson/
8. J. Schmidhuber, Evolutionary principles in self-referential learning, T.U.Muenchen, 1987.
9. D. Lindley *et al.*, *J. Royal Stat. Soc*. B **34**, 1 (1972).
10. C. Finn *et al.*, *Proc. Intl. Conf. Mach. Learn*, (2017).
11. Y. Duan *et al.*, arXiv 1611.02779 (2016).
12. F. Alet *et al.*, *Proc. Mach. Learn. Res* **87**, 856 (2018).
13. Y. LeCun *et al. Proc. IEEE* **86**, 2278 (1998).
14. A. Rosenfeld, *ACM Comput. Surv*. **1**, 147 (1969).
15. D. Hubel *et al.*, *J. Physiol*. **195**, 215 (1968).

ACKNOWLEDGMENTS

The author is supported by NSF, ONR, AFOSR, Honda Research, and IBM. Thanks for insightful discussions with T. Lozano-Perez and students and colleagues in the CSAIL Embodied Intelligence group.

10.1126/science. aaz7597

Caption for photo