

[ABOUT](#) [CONTACT](#) [SPONSORS](#) [PRIVACY POLICY](#)[ANALYSIS](#) / [TECHNOLOGY](#) / [GLOBAL](#)

Artificial Intelligence's Next Big Step: Reinforcement Learning

25 Jan 2017 8:00am, by [Mary Branscombe](#)

1

Almost every [machine learning](#) breakthrough you hear about (and most of what's currently called "[artificial intelligence](#)") is supervised learning; where you start with a curated and labeled data set. But another technique, reinforcement learning, is just starting to make its way out of the research lab.

Reinforcement learning is where an agent learns by interacting with its environment. It isn't told by a trainer what to do and it learns what actions to take to get the highest reward in the situation by trial and error, even when the reward isn't obvious and immediate. It learns how to solve problems rather than being taught what solutions look like.

Reinforcement learning is how [DeepMind](#) created the [AlphaGo](#) system that beat a high-ranking Go player (and has recently been winning online Go matches anonymously). It's how [University of California Berkeley's BREETT robot](#) learns how to move its hands and arms to perform physical tasks like stacking blocks or screwing the lid onto a bottle, in just three hours (or ten minutes if it's told where the objects are that it's going to work with, and where they need to end up). Developers at a hackathon built a smart trash can be called [AutoTrash](#) that used reinforcement learning to sort compostable and recyclable rubbish into the right compartments.

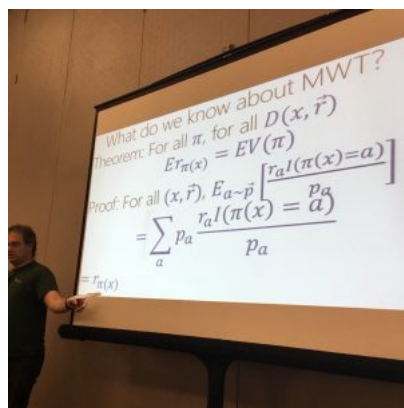
Reinforcement learning is the reason [Microsoft just bought Maluuba](#), which Microsoft plans to use it to aid in understanding natural language for search and chatbots, as a stepping stone to general intelligence.

Commercial deployments are far rarer, though. In 2016, Google started using

DeepMind's reinforcement learning to save power in some of its data centers by learning how to optimize around 120 different settings like how the fans and cooling systems run, adding up to a 15 percent improvement in power usage efficiency.

And without anyone really noticing, back in January 2016 Microsoft started using a very specific subset of reinforcement learning called contextual bandits to pick the personalized headlines for MSN.com; something multiple machine learning systems had failed to improve.

The contextual bandit system increased clickthrough by 25 percent — and a few months later, Microsoft turned it into an open source [Multiworld Testing Decision Service](#) built on the [Vowpal Wabbit machine learning](#) system, that you can run on [Azure](#).



Microsoft's John Langford discusses multiworld testing at QCon NYC last June.

"We have a deployable system, which I think is the first anywhere in the world," claims [John Langford](#), the Microsoft researcher who started work on Vowpal Wabbit when he was at Yahoo.

Multiworld testing runs multiple context-sensitive experiments at the same time and it lets you answer much more detailed questions than standard A/B testing. Contextual bandits are a mathematical representation of a

slot machine with multiple arms, and before choosing which arm to pull each time the agent sees a feature vector showing the current context (from the multiworld testing), as well as the rewards for arms it's played in the past.

Contextual bandits are one of two 'islands of tractability' in the reinforcement learning space "which is clearly still more of a research endeavor than supervised learning," warned Langford. "There are a lot of problems that are readily posed as reinforcement learning problems for which we have no effective solution."

"They work in situations where the reward is immediately clear and you get feedback on your actions," he explained; "Where you have contextual control over small numbers of actions and where you get feedback about that action. We want to try to tame these techniques, to normalize them and make them easy to use; that's what we're trying to do with the decision service."

Sometimes the feedback isn't immediate, so you might need to use reward shaping, which "lets you take a long-term goal and decompose the reward for that into a bunch of short-term rewards — the sum of which, if you get it right, gives you the long-term goal. This is a key problem you need to resolve when trying to solve reinforcement learning problems."

Sticking to those situations is how the team was able to create a reinforcement learning solution that works in the real world rather than just a research problem. "It's very much a limitation of scope that makes it tractable," Langford points out. "There's a particular subset, contextual bandits, where we know that things are tractable and the decision service

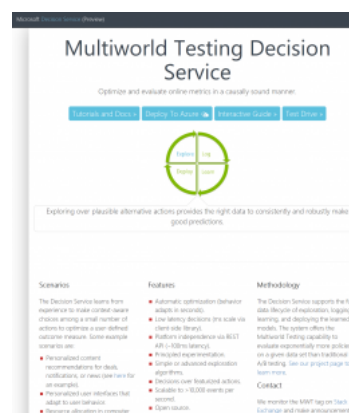
tackles that subset.”

“The important thing to note is there is no long-term decision-making aspect,” further explained [Alek Aggarwal](#) of Microsoft Research, who also worked on the decision service. “It’s a sequential process where you observe some context and take an action and immediately get a reward; there’s no long-term reward to worry about. There’s still a limitation; if you take an action you only see the reward for that action, not what would have happened if you took another action. You’re trying to optimize your choice of actions given the contextual information you have.” That’s another way reinforcement learning differs from supervised learning, he notes.

The problem isn’t just long-term rewards, but “the credit assignment across multiple actions,” added Langford. “If you take 30 actions and any of those can affect the rewards you observe, that’s a different situation. It’s easy to come up with problems where all the existing algorithms will fail badly.”

The Right Problems

There are problems where reinforcement learning is better than the better known supervised learning, though. “Right now there are a lot of problems people are trying to solve that don’t fit the supervised learning paradigm but they’re used to supervised learning so they try to use it,” Langford explained. “The canonical one is trying to predict clicks and optimize which ad you place.” Contextual bandits are ideal for that, and for making other personalized content recommendations.



The Multiworld Decision Service is live and costs about 20 cents an hour to run on Azure (click to enlarge).

Another applicable area is personalized user interfaces that adapt to your behavior. Imagine that you wanted to use an EEG to control a computer. “Every time you put the EEG on your head, it’s in a slightly different position and the system needs to learn to very quickly adjust to how where it’s been placed on your head,” suggested Langford. “The person using the interface is going to notice when things go wrong and they can issue a correction; that was right, that was wrong. There would be a natural co-learning process.”

Personalized healthcare is a trickier problem, especially given privacy issues and fragmented medical records, but at least theoretically, contextual bandits might help. “Not all healthcare is the sort where a single action leads to a clear outcome, but a portion is,” he noted. “Instead of the drug trials we have today, imagine trials that are ten or twenty times larger that learn a policy for deploying treatments personalized to individual subjects rather than the one-size fits all policy we do right now.”

Resource allocation — for humans or computers — is applicable well beyond Google’s data center management trials and it’s also a good fit for contextual bandits, says Agarwal. “When you send a request to a website, which server should handle it? Operating systems solve many resource allocation problems. In many of these cases you have to do some work to define the right functions, but some of them end up being reasonable fits for bandits.”

Getting the rewards right is key; “it tends to be where the art is in trying to solve reinforcement learning problems,” says Langford. “Sometimes it’s dead obvious, like clicks. Sometimes it’s more subtle. But figuring out how to frame the problem is the silver bullet for solving the problem.”

Show Me

If that’s just too difficult, researchers turn to the second type of reinforcement learning that we can currently do well: imitation learning, by demonstrating a technique. “It may be easier for a human to supply the information about the right thing to do than to come up with a good reward function that matches the problem.”

“You see this a lot in robotics where you demonstrate to a robot what you want it to do and it can learn to do something like the same thing, sometimes even better than a human can do,” he noted. You need to make a long sequence of decisions to succeed and it’s hard to break down the value of incremental decisions. Robots work from sensory feedback; they have cameras and sensors for where the actuators are and they translate this feedback into short-term rewards. The beauty of this is that the demos keep you out of local minima and maxima.”

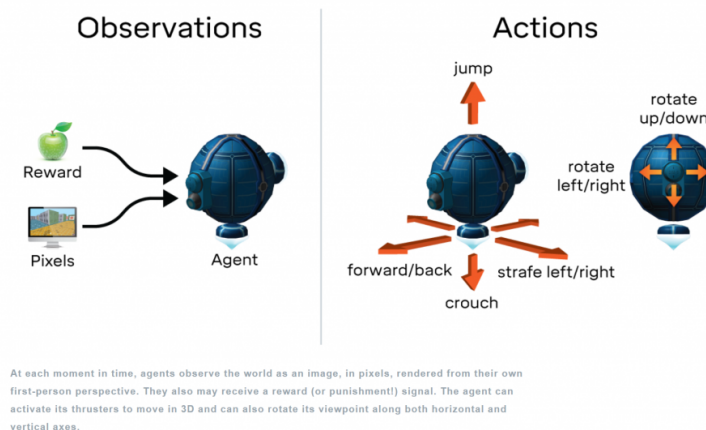
Self-driving car systems work in the same way, noted Agarwal, and he points out that to be effective, imitation learning needs high-quality demos. “If you’re getting very expert demonstrations with optimal sequences of actions most of the time, you can learn to imitate them well and generalize them to unseen situations and not get stuck.”

Unlike contextual bandits, there isn’t just one technique for imitation learning. There isn’t a standardized platform like the Multiworld Decision Service that you can use on your own problems. But we’re starting to get platforms to help researchers experiment.

Play a Game?

Games are a common way to train reinforcement learning systems because they have built-in rewards. Atari 2600 games have been a popular choice, but often they’re fairly simplistic environments. At the end of 2016, both Google and [Open.AI](#) announced that they were opening up their reinforcement learning training systems to researchers, giving far more people access to complex simulated environments for training AI agents previously reserved for companies with the budget to build them.

Google’s [DeepMind Lab](#) — known internally as Labyrinth — looks like a simple 3-D game, with a floating orb representing the AI agent. The world includes 14 levels and four kinds of learning tasks like navigating a maze (static or generated on the fly), playing laser tag and collecting fruit, but researchers can get the code for this virtual environment from [GitHub](#), create their own levels (using a game editor or programmatically in C and Python) and experiment with different reward schemes and gameplay logic.



The AI agent in DeepMind Lab is an orb that can view and navigate the 3D world (Photo: Google).

[OpenAI's Universe](#) is also an experimentation platform for working on AI agents that try to learn to use computers the way humans do; by looking at the pixels on screen and operating virtual controls. As with Lab, the aim of Universe is to develop an AI agent that can not only learn to deal with one situation but use the learning techniques it's developed to tackle unfamiliar environments as a stepping stone to creating AI that goes beyond a single, narrow domain — and OpenAI's approach is to give researchers access to a lot of environments that were created for humans, not specially crafted for AI agents to learn in. Not only does that turn games and apps we already have into a training ground; it also means AI agents can watch people using the software to kick-start their learning — and we can compare the AI to human performance rather than just each other.

Universe lets you use any program with [OpenAI's Gym toolkit](#) for building reinforcement learning agents in frameworks like TensorFlow and [Theano](#). Gym already included simulated robots, Go and a range of classic Atari games and Universe extends that to over a thousand environments, including Flash games, 80 common browser tasks like typing in a password or a booking a flight, and games like Grand Theft Auto V.

Universe packages them up as Docker images, launches them on a [VNC remote desktop](#) and controls them through Python scripts — although not all of them support reinforcement learning yet. OCR runs in the [Python](#) program that controls the Docker container to scrape the game scores to use as rewards; of the 1,000 Flash games, 100 have reward functions and OpenAI has plans to use human players to demonstrate more of the games to AI agents, to make it easier to analyze what the rewards should be. In the future, Universe AI agents will get access to games like Portal and Wing Commander III, as well as [Wolfram Mathematica](#), and maybe Android and Unity games as well.

They're also going to be able to run inside [Project Malmö](#), Microsoft's reinforcement learning experimentation platform which runs on top of [Minecraft](#) (which it started work on in 2014 and open sourced in mid-2016).

“Some AI techniques that were purely on the research side for decades are starting to move closer and closer to real world applications,” says [Katja Hofmann](#), from Microsoft's research lab in Cambridge. “That's very exciting. To really push those techniques forward, we need to flexibly, rapidly be able to experiment with techniques. That need for pushing forward experimentation was the motivation for Project Malmö. Now there are more

and more of these platforms, which is exciting — and important for both pushing research forward and opening that research up to the broader community of developers and enthusiasts who can join in and start productizing.”

Currently, Universe and Project Malmö use slightly different APIs to integrate bots and agents in games and to run experiments. The first step will be making it easier to train an agent on one platform and then test it on the other. “There’s a lot to be gained by standardizing some of those APIs to make it as easy as possible for the community to switch back and forth.”

In the long run, that will let researchers create portable agent architectures. “We’re working with variants of deep reinforcement learning agents that can not only learn 2-D Atari games but also plug into agents that navigate the 3-D Minecraft world where they can look around and see where to go. Having the same kind of architecture for both will translate to effective learning in both those scenarios, so we can rapidly make progress, though experimentation, on aspects that focus on interactive learning.”

The two platforms have different research agendas. Project Malmö is about what Hofmann calls flexible learning.

“The idea is to develop AI that that doesn’t just address a single task but that flexibly learn and build up common sense knowledge and use that to tackle more and more complex problems. We think Minecraft is fantastic for this because it creates a sandbox. You have an entire world that’s infinite and that’s procedurally generated. You put the agent in the environment and its experiences are from a first-person perspective. Today, the agent can learn basic survival — how to navigate and avoid lava. As the technology matures, it can build up more complex skills like construction, learning how to build up complex items. Agents will be able to reuse their knowledge and basic understanding of the Minecraft world when they learn new tasks. That’s similar to how people learn about the world in one particular way and adopt that to the task at hand.”

Ultimately, she hopes that work will lead to collaborative AI, where agents learn to collaborate with human users. “If you want to achieve a goal, we’ll need an agent that can understand the goal and reason about the steps it needs to take in order to help you achieve that goal. That’s one of our key motivations.”

The OpenAI project has a rather different goal; they’re hoping to create a single AI agent with a generic problem-solving strategy, a first step towards general AI.

Experimentation Is the Way Forward

Like so much of AI, reinforcement learning isn’t new; the first textbook covering it dates to 1998 (and the second edition will finally come out this year). What’s different now is partly that we have experience with some problems that are well understood, particularly in the two areas of contextual bandits and imitation learning. But we also need these new experimentation platforms like Universe and Project Malmö and DeepMind Lab to give more researchers access, and to compare solutions in the same environment to benchmark progress.

Agarwal compares the availability of experimentation platforms for

reinforcement learning to the impact large labeled data sets like [ImageNet](#) had on supervised learning. “The way we make a lot of progress in supervised learning was that we started accumulating large data sets and repositories and once we had those, we could try algorithms out on them reliably and iterate those algorithms.” A static data set isn’t useful for evaluating more general reinforcement learning; “two different agents will take two different trajectories through an environment.”

Instead, researchers need a large, diverse set of environments that’s also standardized so everyone in the field works against them. “Flexible, diverse platforms can serve the same function as a repository for reinforcement learning tasks where we can evaluate and iterate on ideas coming out of research much faster than was possible in the past, when we had to restrict the algorithms to simple evaluation problems because more complex ones weren’t available. Now we can take ideas to the platforms and see whether or not they do a good job,” Agarwal said.

Reinforcement learning will often be only one of the machine learning strategies in a solution. Even AlphaGo was initially trained to mimic human play using deep learning and a database of around 30 million moves from 160,000 games played by human Go masters. It was only once it reached a certain level of skill that it began playing against other instances of AlphaGo and using reinforcement learning to improve.

“It’s important in the long run to understand how an AI agent could be able to learn about those goals and about the peculiarities and abilities of the person it’s working with” — Katja Hofmann

That pattern might actually be key to making reinforcement learning ready for wider use, Hofmann suggested — either by using deep learning to prepare the actions and rewards for a reinforcement learning system, or by using reinforcement learning to reduce the work it takes to apply supervised learning to a domain.

“At the moment, if you put a reinforcement learning agent in a 3-D environment they would reason at the granularity of individual abstraction, taking one step forward, not on a higher level using the concept of walking to the next door. There are questions about abstraction that still need to be addressed before we can put the next pieces of reinforcement learning into applications, like understanding how to set goals in situations where a clean scoring function might not be available,” Hofmann explained.

The experimentation platforms help with that research, but so do the recent advances in deep learning. “For decades, the reinforcement learning problem of learning an appropriate presentation of a domain couldn’t be tackled in a systematic way because for every new app you could envision, you would need domain experts to find a representation [for that domain]. That was very resource intensive and didn’t scale to the breadth of applications we would like to unlock,” Hofmann explained, adding that “We’ve had major advances in learning representations and automatically extracting the features that

There is still plenty of research to be done, like how to get agents to explore

their environment systematically, giving them the equivalent of curiosity or the intrinsic motivation to explore. And there's also the problem that few developers are familiar with reinforcement learning at this point. It's likely that systems like the Multiworld Decision Service will help bring reinforcement learning to a broader developer audience, Agarwal suggested.

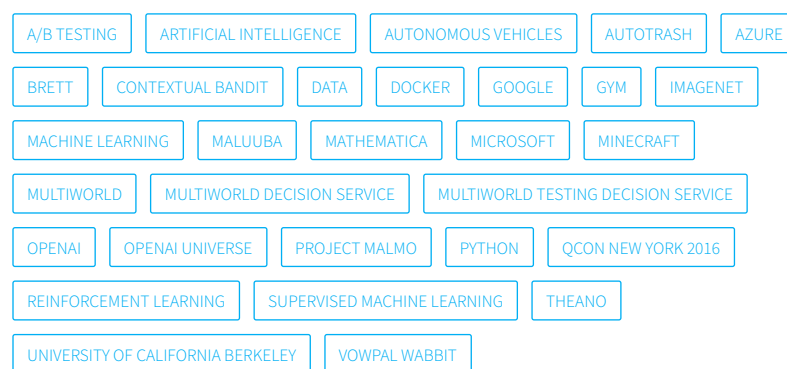
One of the differences between supervised learning and reinforcement learning is that in the supervised learning world it's usually a machine learning person providing the algorithm and the user bringing the data. "In reinforcement learning, it's more that you bring your domain to the table and the agent by acting in the world is creating its own data. That creates a lot of complexity; the app developer has to build a lot more pieces of the solution if they want an algorithm to work in the right way and there are many things that can go wrong. It's often better to design a whole system," Agarwal suggested.

Langford is also optimistic. "Over the next say five years, I expect that education will happen and that we will see many more successful applications and see these kinds of systems become much more standardized in the world."

And Hofmann has some big ambitions of her own. "You can envision an AI that's able to learn; it would have some general knowledge about the environment, about what kinds of tasks people want to achieve and it would also be able to learn on the fly so it can personalize its help and support towards the goals of the person or the player."

"In the real world, every person has different knowledge and abilities and desires," she explained. "It's important in the long run to understand how an AI agent could be able to learn about those goals and about the peculiarities and abilities of the person it's working with and be able to personalize its assistance and actions to help that particular person achieve their goals."

Feature image: Laser tag levels in DeepMind Lab test agents on fine control, strategy, planning and dealing with a complex visual environment. Image from Google.



1

THE
NEW
STACK UPDATE

A digest of the week's most important stories & analyses.

Subscribe

We don't sell or share your email. Occasionally, we send updates and useful info.

RELATED STORIES

INTERVIEWS / PODCAST / TOP STORIES /
GLOBAL

A Skeptical Look at Kubernetes

18 Aug 2017 1:00pm, by [Alex Williams](#) and [Scott M. Fulton III](#)

NEWS / TECHNOLOGY / TOP STORIES

Box Debuts an Image Recognition Service Built From Google's AI Technology












18 Aug 2017 5:00am, by [TC Currie](#)[View / Add Comments](#)

SPONSORED FEED

vSpeaking Podcast Episode 55: Automate all the things

AUGUST 18, 2017

OpenShift Commons Briefing #91: CRI-O Kubernetes Runtime Deep Dive with Mrunal Patel and Dan Walsh (Red Hat)
AUGUST 18, 2017Updates from the OpenStack Ops Midcycle
AUGUST 18, 2017Sell more properties with these real estate website design tips
AUGUST 18, 2017Stepping Up to the Plate: A Story About Being On-Call
AUGUST 18, 2017Twistlock Defender Architecture
AUGUST 17, 2017Querying and Pagination with DynamoDB
AUGUST 17, 2017CoreOS launches Tectonic 1.7.1 with support for Microsoft Azure cloud
AUGUST 17, 2017Container Forensics
AUGUST 16, 2017Liberty Mutual Creates Underwriter Portal MVP in Just 28 Days with Cloud Foundry
AUGUST 16, 2017Business Becomes Data and Data Becomes the Business
AUGUST 16, 2017

	Slalom's approach to breaking down silos between DevOps and Security Teams AUGUST 16, 2017
	Part 3 - Continuous Deployment Strategies: Implementation Techniques for Canary Releases AUGUST 14, 2017
	Hybrid IT Workload Placement Guide AUGUST 14, 2017
	Fortune: "Amazon Joins Google, Goldman Sachs, and Twitter in This Cloud Foundation" AUGUST 09, 2017
	Marley Spoon: A Look into Their Stack and Team Structure JULY 31, 2017
	Harnessing the Power of Kubernetes the Smart Way JULY 31, 2017
	Achieving Continuous Improvement Through Containers JULY 27, 2017
	Looking over the Edge JULY 25, 2017
	Why Health Care Research Needs Multitenant Orchestration to Save Lives JULY 07, 2017
	Cloud Native Apps and Security: The Case for CoreOS Rkt and Xen JUNE 21, 2017
	Deploying ASP.NET Core Applications on Apcera MAY 16, 2017

TOPIC HUBS	ECOSYSTEMS	CONTENT	THE NEW STACK
Architecture	AI	Analysis	About
Data	Cloud	Ebooks	Contact
Development	Containers	Events	Sponsors
Do-It-Yourself	Kubernetes	Podcasts	Privacy Policy
Operations	Microservices	Research	
Security	Node.js		
Tech Culture	Serverless		

© 2017 The New Stack. All rights reserved.