Beyond deep learning
Blending two approaches to artificial intelligence promises to let machines think a little more like humans

M. Mitchell Waldrop, Science Writer

Several times over the past year, Matthew Botvinick has carried out an informal test that illuminates perhaps the biggest challenge facing today’s artificial intelligence (AI). He chooses a program engineered to respond to natural human language, and asks a seemingly simple question: “What am I wearing?”

Now, these are state-of-the-art programs independently developed by his company and others, explains Botvinick, director of neuroscience research at the London, UK-based AI firm DeepMind. The programs have been trained on billions if not trillions of words of English text—the same kind of deep learning responsible for AI’s recent triumphs in game-playing computers, digital assistants, car autopilots, and image synthesis, not to mention the hype about sentient chatbots. So Botvinick has always gotten an answer—usually something like “You’re wearing khakis.”

Yet none of these machines has a camera or any other way to know how he’s actually dressed. Nor have their trillion-word feats of rote learning given them anything resembling a human-level understanding of what they’ve read, much less an ability to reason about what they know or don’t know.* “They just capture the statistics of human language,” says Botvinick.

This reliance on blind mimicry is why we still can’t trust deep learning in high-stakes applications such as driverless cars, says Gary Marcus, a psychologist, author, and frequent critic of deep learning. However beautifully the AI performs in routine traffic, he says, it could still fail when faced with something unusual.

*See “What are the limits of deep learning?” https://www.pnas.org/doi/10.1073/pnas.1821594116.
“People are going to get killed because the system still can't understand what it means for a person to carry a stop sign at a construction site.”

Nor is this a problem that can be fixed just by continuing to make the systems larger and larger. A truly flexible, robust machine intelligence will need something more, says Dan Gutfreund, who works on computer vision at the MIT-IBM Watson AI Lab in Cambridge, MA:

Indeed, alternative AI approaches have proliferated in recent years, often involving efforts to mimic human abilities. Examples include our sleep-wake cycle, when our brains consolidate memories and advance learning; and the primal rules collectively known as intuitive physics, which help curious, growing infants understand how objects interact. Such approaches may offer a way to outsmart state of the art deep learning strategies.

A Meeting of Minds

Most new approaches are tackling one fundamental challenge: finding ways to integrate two modes of thought, fast and slow.

In humans, these two modes are often called system 1 and system 2. System-1 thinking is fast, effortless, unconscious, and automatic. It encompasses hardwired responses like perception and recognition, as well as thoroughly practiced skills like driving down a familiar road, reading a word in your native language, or a chess master's know-at-a-glance intuition about the most effective way to counter an opponent's move.

Deep learning is good at emulating this kind of automatic response, which is why its most successful applications tend to include tasks in which perception and recognition skills are paramount, such as photo editing and voice recognition. The basic idea, which dates back to the first brain-inspired neural networks in the 1940s, is to process incoming data via a web of neuron-like nodes linked by synapse-like connections. This network is trained by asking the system to process a given library of inputs while adjusting the data flow through each connection until the network produces satisfactory outputs.

Early neural networks were small and limited in what they could do. But their capabilities and speed have improved dramatically since the deep-learning era began around 2010, thanks to hardware and software advances that have allowed the networks to reach gargantuan scales. DeepMind's Gopher natural language system, for example, was trained on some 2 trillion English words while tweaking 280 billion parameters in its connections.

Yet deep learning's success has highlighted its greatest weakness: the ease with which a network can be confined by input not covered in its training.

Humans can be confounded by strange situations, too. But that's when we start looking for solutions with system-2 thinking: a far slower, more difficult mode of cognition that dominates our consciousness and is anything but automatic. "Maybe you're driving in a new city where the traffic laws are different, like you have to drive on the left and not on the right," says Yoshua Bengio, an AI researcher at the University of Montreal, Canada. You can't just go by habit, he says. You have to pay attention, remember the new traffic laws, and decide where to steer next, all at once.

Compared with system-1 thinking, system-2 constitutes a tiny fraction of what's going on in our brains. But because it's the only kind of cognition we're aware of, most of AI's founders in the 1940s and 1950s assumed that it was all machine intelligence needed. This inspired the symbol-processing paradigm that dominated AI for its first half-century until the deep-learning revolution of the past decade. The idea was to give the computer some kind of model of the problem at hand, in the form of a data structure analogous to the mental models of the world that we build in our conscious minds, then look for a solution to the problem by manipulating the model with algorithms that mimic our system-2 reasoning.

The symbolic approach to AI has had many successes, such as expert systems for applications like medical diagnosis. But it can also bog down pretty quickly, notes Swarat Chaudhuri, a computer scientist who works on AI-powered program synthesis at the University of Texas in Austin. These algorithms tend to lead to a rapidly branching tree of possibilities, and searching for the best solution can quickly become hopeless, even for the most powerful computers imaginable.

A much better strategy, says Chaudhuri, is to invoke some system-1 intuition about useful strategies to try—the kind of heuristic knowledge that is second nature to human experts. In expert systems and other forms of symbolic AI, this intuition was often encoded in a series of hand-crafted rules of thumb: "If such-and-such is the case, THEN try that."

But these days it's common to train these rules into a neural network, which will typically operate much faster. Over the past half-decade, this method has become a favorite way for researchers to join neural and symbolic AI. Instead of asking their neural networks to turn input into output in one leap, they use the networks as guides for symbolic reasoning. Or as Bengio puts it, "system 1 is your imagination machine."

One example of this strategy was AlphaGo, an AI system that DeepMind developed in 2016 to play the notoriously complex board game Go (1). AlphaGo explored potential moves with a symbolic look-ahead algorithm but avoided the explosion of possibilities by guiding the search for solutions with two neural networks: one trained to evaluate board positions, and another to recognize the most promising moves. For both networks, the training consisted of watching amateur games, followed by practice sessions in which copies of AlphaGo were pitted against one another to refine their play and learn from their mistakes. The pay-off came in March 2016, when AlphaGo became the first AI program to defeat a human world champion, South Korea's Lee Sedol.
Since then, DeepMind has continued to refine and generalize this approach (2, 3). The company’s latest iteration is MuZero, which has taught itself a better-than-human-level play on Go, chess, and dozens of 1980s-era Atari video games—with no previous knowledge of the rules in any of them (4). The program accomplishes this with neural networks that are trained on prerecorded games, where the networks learn to ignore nonessential details and instead keep a tight focus on the moves: Will a given sequence help or hurt its chances of winning?

**Real-World Messiness**

Although MuZero’s generality and power represent an undeniable advance for AI, even ultra-challenging games like chess and Go still exist in a finite world—the board. And they follow just a handful of precisely defined rules. The same could be said about Atari games. None of them has anything like the complexity of the real world—a challenge that AI researchers have been wrestling with for decades.

A prime example is the task of scene perception: looking at a real-world image and trying to recognize what’s in it. “Perception is exactly the inverse of computer graphics,” says Gutfreund. Instead of specifying some three-dimensional (3D) arrangement of objects and then rendering how they’d look from a certain perspective, as in an effects-heavy Hollywood blockbuster, you’re trying to start with the rendering—the image you see—and work backward to recreate the original objects.

What makes this hard for a computer is that the reconstruction is mathematically impossible. In general, there are infinitely many solutions that could fit the data, especially in natural scenes where objects block one another and boundaries get confused by light and shadow. What makes perception seem easy to us is that our system-1 processes kick in and resolve the ambiguities with rules of thumb like “objects have to be supported or they will fall” or “two objects can’t occupy the same space at the same time.”

These rules, known collectively as intuitive physics, start to appear in infancy and develop rapidly throughout our first years of life. By adulthood, they are so ingrained that we hardly know they exist. But AI researchers have to think about them carefully: What heuristic knowledge does your scene-understanding system need, and how do you include it in the algorithm?

One common answer is to train a large neural network with a lot of data, says Gutfreund. But neural networks are only as good as the training data they get. And when it comes to 3D scenes, the data is often incomplete or ambiguous. How can AI systems deal with these ambiguities?

**Dreaming in Code**

One example of how this might work is DreamCoder, another neuro-symbolic-Bayesian system from 2021 (6). DreamCoder’s basic task is a classic in AI research: Given a handful of input–output examples like [1 2 3] → [2 4 6], figure out the general rule—in this case, “double every number.” Then generate the simplest computer code that will apply the rule to any input.

But DreamCoder is much more of a generalist than its predecessors. It can handle examples from eight different domains, including text editing, drawing simple graphics, planning a robot’s movements, and other scene complications. This ability could have big payoffs for autonomous vehicles, robotics, assistants for visually impaired individuals, and other such applications.

Perhaps most importantly, 3D3P’s novel combination of existing ideas—symbols, networks, and Bayesian inference—offers much more scope for incorporating capabilities that go beyond scene recognition.
rearranging a stack of blocks to form a new stack, and turning physics data into equations. And it is almost unique in the way it learns from experience, with each success or failure helping the program refine its skills and identify new subroutines to improve its performance next time. “That’s really the big claim to novelty,” says Armando Solar-Lezama, a computer scientist who studies AI-assisted program synthesis at MIT in Cambridge, MA, and who is a co-author of the DreamCoder article. Existing code-generation techniques “just assume a given vocabulary and a given set of symbols,” he says. DreamCoder can invent its own—in effect, discovering new languages to improve its thinking.

The program accomplishes all this in a wake–sleep cycle roughly analogous to the one we experience. During its waking phase, DreamCoder uses a 3DP3-like combination of neural networks and Bayesian inference to search for programs that solve the tasks it has been given. Then in a first sleeping phase, “abstraction,” the program goes through these programs in search of useful subroutines: frequently appearing sequences of operations that can be packaged together to make programs shorter, easier to discover, and easier to understand. This is similar to the way human sleep is thought to help us consolidate memories and skills. Then in a second sleep phase, “dreaming,” DreamCoder solves replays of its previous tasks, as well as multiple random tasks—and in the process, trains its code-search neural network to use these new subroutines.

In tests of all eight domains, the DreamCoder team found that the program typically requires fewer than five wake–sleep cycles to go from a rank beginner that takes hours to solve only a few percent of its tasks to being an expert that can quickly solve 60% or more.

Together, these neuro-symbolic programs point in promising directions, combining different ways of thinking to form more flexible AI. But there’s a long way to go before anyone assembles anything like a fluid, human-like intelligence, with the abilities to reason and make plans across multiple domains; to understand and learn from natural language; to integrate perception, action, and movement in a physical body; and to recognize other people as agents acting according to their own beliefs and desires.

Such an intelligence should also incorporate some sense of ethics—the machine equivalent of values such as equity, respect for life, and subservience to (lawful) human control. “I think this is something that the entire AI field is really pivoting to,” says Botvinick. As the interactions between humans and machines are getting richer and richer, he says, “the question is not only how we make our systems more intelligent but how we grow their intelligence in a way that’s going to benefit us.”

1. D. Silver et al., Mastering the game of Go with deep neural networks and tree search. Nature 529, 484–489 (2016).
2. D. Silver et al., Mastering the game of Go without human knowledge. Nature 550, 354–359 (2017).
3. D. Silver et al., A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science 362, 1140–1144 (2018).
4. J. Schrittwieser et al., Mastering Atari, Go, chess and shogi by planning with a learned model. Nature 588, 604–609 (2020).
5. N. Gothoskar et al., “3DP3: 3D scene perception via probabilistic programming” in Advances in Neural Information Processing Systems, M. Schrittwieser et al., Eds. (Curran Associates, Inc., 2021), pp. 9600–9612.
6. K. Ellis et al., DreamCoder: Growing generalizable, interpretable knowledge with wake-sleep Bayesian program learning. arXiv:2006.08381 (15 June 2020).