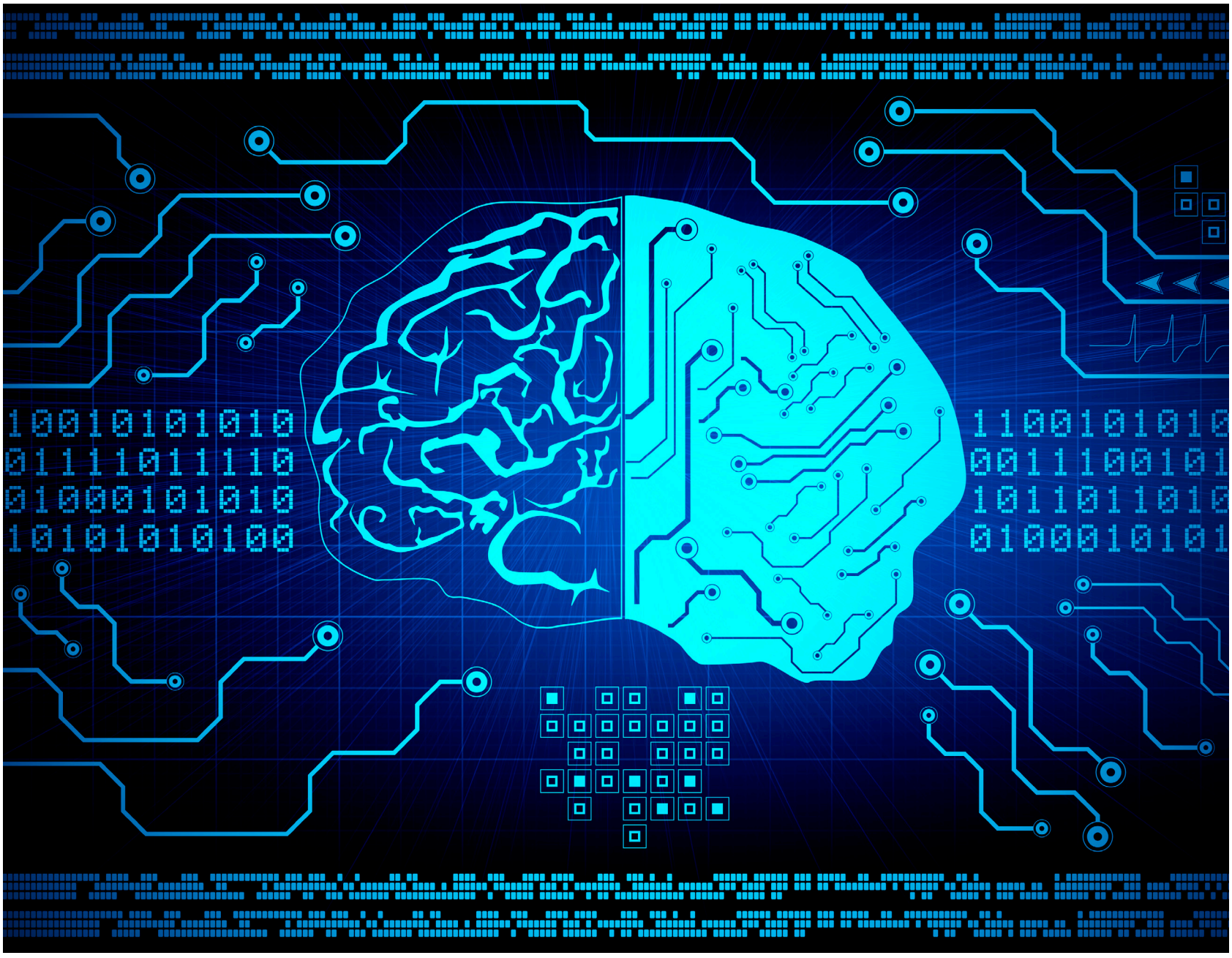
Resistive Computing: Based on the Human Brain

BY DAVID CARDINAL



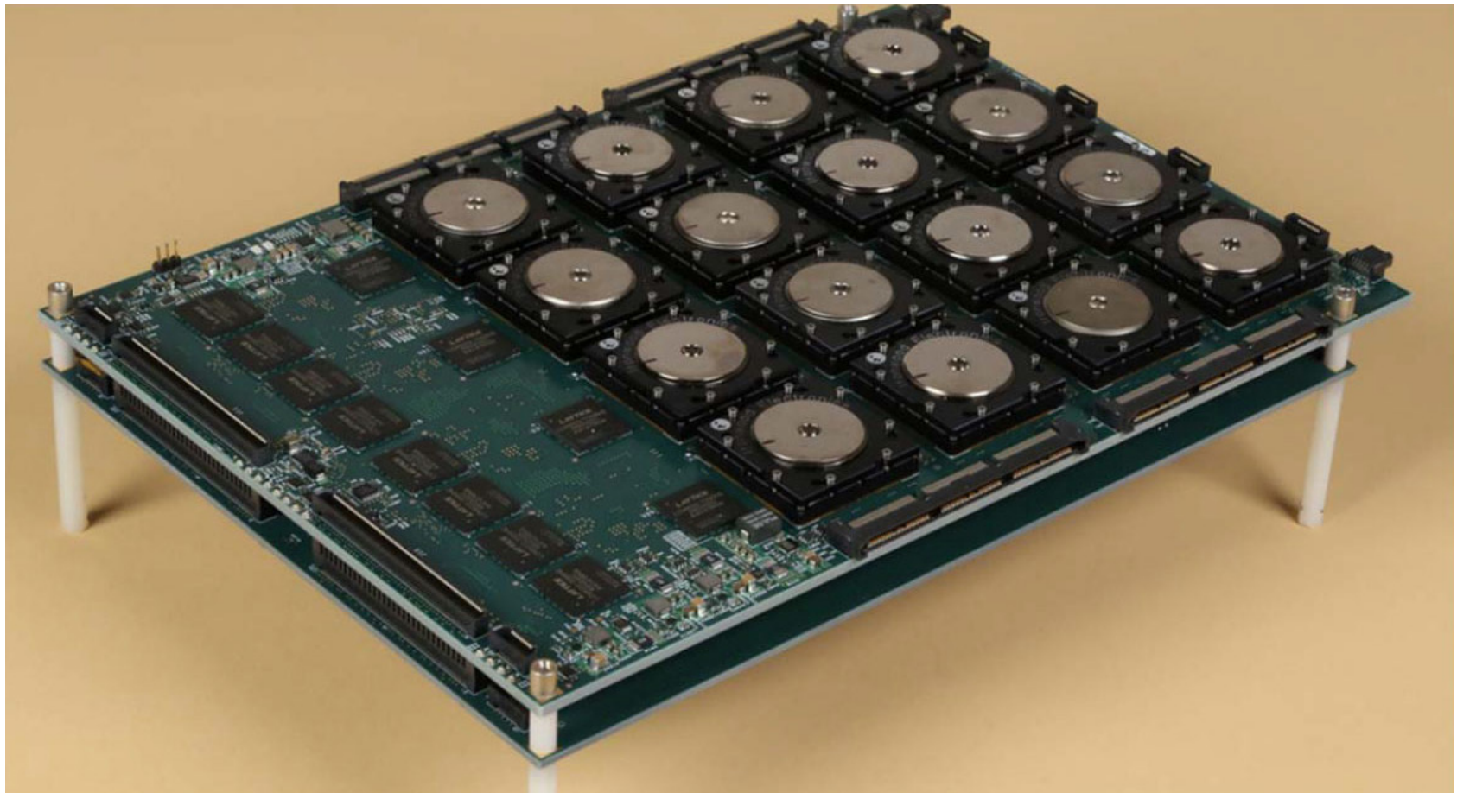
With the recent rapid advances in machine learning has come a renaissance for neural networks, computer software that solves problems similar to the way the human brain does: by employing a complex process of pattern matching distributed across many virtual nodes, or "neurons." Modern compute power has enabled neural networks to recognize images, speech, and faces, as well as to pilot self-driving cars and win at Go and *Jeopardy!*. Most computer scientists think that is only the beginning of what will ultimately be possible. Unfortunately, the hardware we use to train and run neural networks looks almost nothing like their architecture. That means it can take days or even weeks to train a neural network to solve a

problem—even on a compute cluster—and then require a large amount of power to solve the problem once they're trained.

NEUROMORPHIC COMPUTING MAY BE KEY TO ADVANCING AI

Researchers at IBM aim to change all that by perfecting another technology that, like neural networks, first appeared decades ago. Loosely called resistive computing, the concept is to have compute units that are analog in nature, small in substance, and can retain their history so they can learn during the training process. Accelerating neural networks with hardware isn't new to IBM. It recently announced the sale of some of its TrueNorth chips to Lawrence National Labs for AI research. TrueNorth's design is neuromorphic, meaning that the chips roughly approximate the brain's architecture of neurons and synapses. Despite its slow clock rate of 1KHz, TrueNorth can run neural networks very efficiently because of its million tiny processing units that each emulate a neuron.

FINDING TRUENORTH
This 16-chip DARPA SyNAPSE board uses IBM's neuromorphic TrueNorth chip.



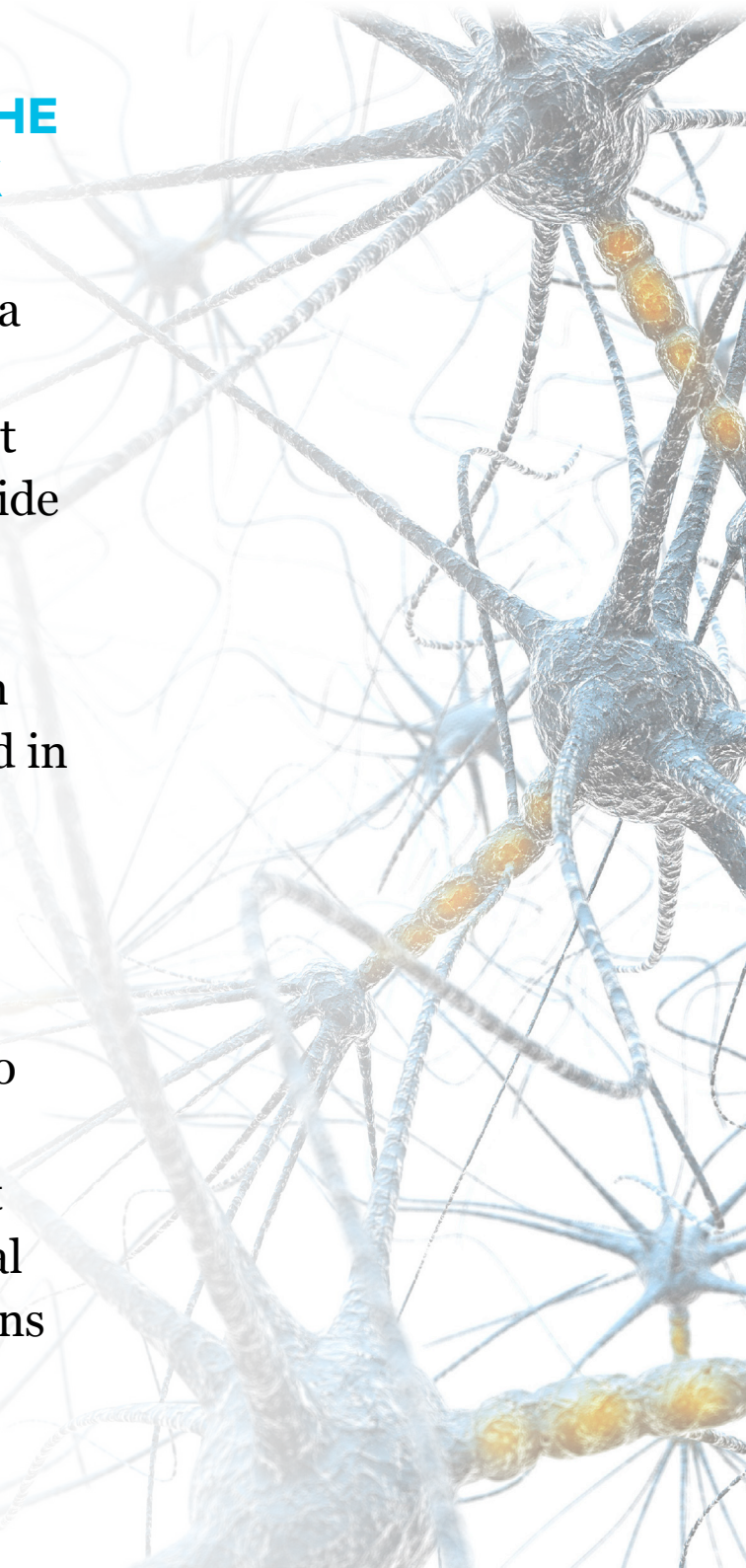
Until now, though, neural network accelerators like TrueNorth have been limited to the problem-solving portion of deploying a neural network. Training—the painstaking process of letting the system grade itself on a test data set, and then tweaking parameters (called weights) until it achieves success—still needs to be done on traditional computers. Moving from CPUs to GPUs and custom silicon has increased performance and reduced the power consumption required, but the process is still expensive and time-consuming. That is where new work by IBM researchers Tayfun Gokmen and Yuri Vlasov comes in. They propose a new chip architecture, using resistive computing to create tiles of millions of Resistive Processing Units (RPUs), which can be used for both training and running neural networks.

> **In the IBM design each small RPU mimics a synapse in the brain.**

USING RESISTIVE COMPUTING TO BREAK THE NEURAL NETWORK TRAINING BOTTLENECK

Resistive Computing is a large topic, but roughly speaking, in the IBM design each small RPU mimics a synapse in the brain. It receives a variety of analog inputs—in the form of voltages—and based on its past "experience" uses a weighted function of them to decide what result to pass along to the next set of compute elements. Synapses have a bewildering, and not yet totally understood, layout in the brain, but chips with resistive elements tend to have them neatly organized in two-dimensional arrays. For example, IBM's recent work shows how it is possible to organize them in 4,096-by-4,096 arrays.

Because resistive compute units are specialized (compared with CPU or GPU cores) and don't need to either convert information from analog to digital or access memory other than their own, they can be fast and consume little power. In theory, a complex neural network—such as the kind used to recognize road signs in a self-driving car—can be directly modeled by



dedicating a resistive compute element to each of the software-described nodes. But because their analog nature and a certain amount of noise in their circuitry makes RPUs imprecise, any algorithm run on them needs to be made resistant to the imprecision inherent in resistive computing elements.

Traditional neural network algorithms, both for execution and training, have been written assuming high-precision digital processing units that could easily call on any needed memory values. Rewriting them so that each local node can execute largely on its own, and be imprecise while producing a result that is still sufficiently accurate, required a lot of software innovation.

For these new software algorithms to work at scale, advances were also needed in hardware. Existing technologies weren't adequate for creating "synapses" that could be packed together closely enough, and operate with low power in a noisy environment, to make resistive processing a practical alternative to existing approaches. Runtime execution happened first, with the logic for training a neural net on a hybrid resistive computer not developed until 2014. At the time, researchers at the University of Pittsburg and Tsinghua University claimed that such a solution could result in a three-to-four-order-of-magnitude gain in power efficiency at the cost of only about 5 percent in accuracy.

THE THINK SYSTEM

Resistive computing adopts the model of a synapse in the brain by using its previous experience to determine how to process new inputs it receives.

MOVING FROM EXECUTION TO TRAINING

This new work from IBM pushes the use of resistive computing even further, postulating a system where almost all computation is done on RPUs and traditional circuitry is only needed for support functions and input and output. This innovation relies on combining a version of a neural network training algorithm that can run on an RPU-based architecture with a hardware specification for an RPU that could run it.

As far as putting the ideas into practice, to date resistive compute has been mostly a theoretical construct. The first resistive memory (RRAM) became available for prototyping in 2012, and isn't expected to be a mainstream product for several more years. And although those chips will help scale memory systems and show the viability of using resistive technology in computing, they don't address the issue of synapse-like processing.

IF RPUS CAN BE BUILT, THE SKY IS THE LIMIT

The proposed RPU design is expected to accommodate a variety of deep neural network (DNN) architectures, including fully connected and convolutional, which makes them potentially useful across nearly the entire spectrum of neural network applications. Using existing CMOS technology, and assuming RPUs in 4,096-by-4,096-element tiles with an 80ns cycle time, one of these tiles would be able to execute about 51 gigaops per second, using a minuscule amount of power. A chip with 100 tiles and a single complementary CPU core could handle a network with up to 16 billion weights while consuming only 22 watts (only two of which are actually from the RPUs—the rest are from the CPU core needed to help get data into and out of the chip and provide overall control).

That is a staggering number compared with what is possible when chugging data through the relatively lesser number of cores in even a GPU (think about 16 million compute elements, as opposed to a few thousand). The researchers claim that, once built, a resistive-computing-based AI system using chips densely packed with these RPU tiles could achieve performance improvements of up to 30,000 times compared with current architectures, all with a power efficiency of 84,000 gigaops per second per watt. If this becomes a reality, we could be on our way to realizing Isaac Asimov's fantasy vision of the robotic positronic brain.