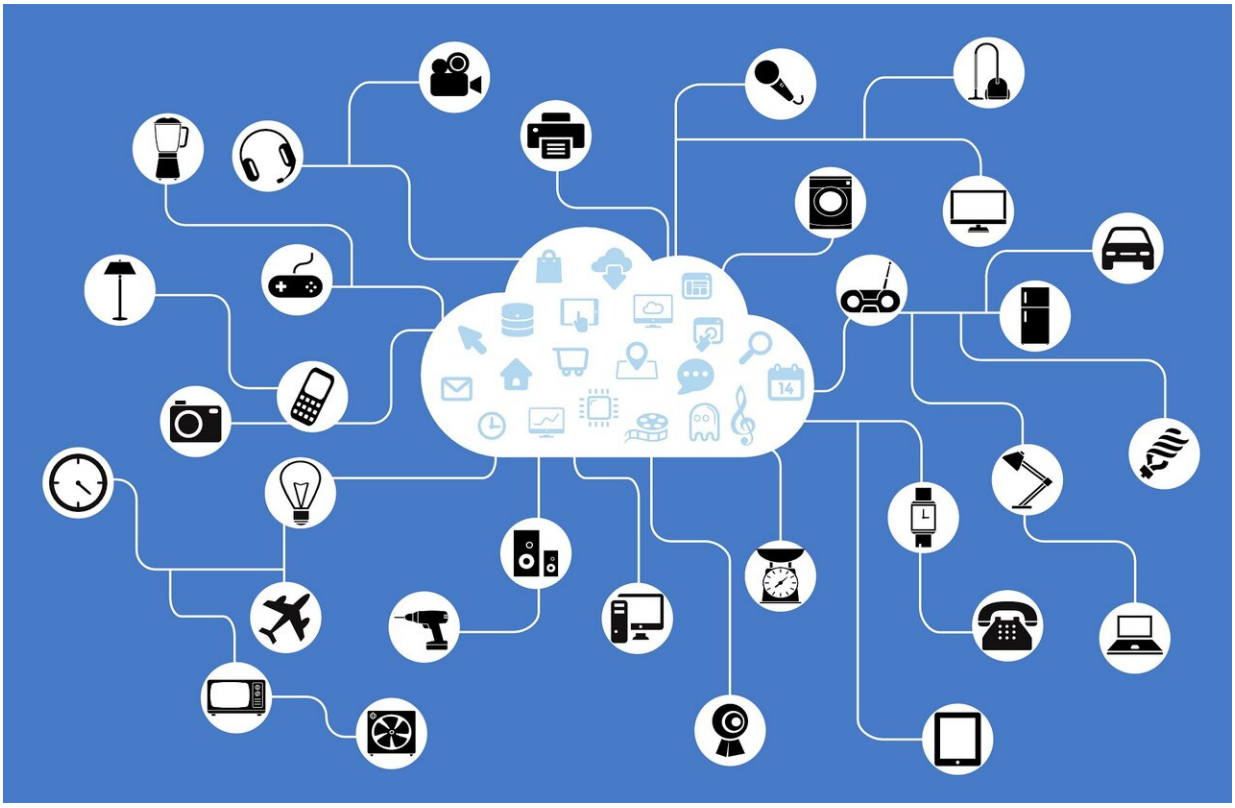
System brings deep learning to Internet of Things devices

November 13 2020, by Daniel Ackerman



Credit: Pixabay/CC0 Public Domain

Deep learning is everywhere. This branch of artificial intelligence curates your social media and serves your Google search results. Soon, deep learning could also check your vitals or set your thermostat. MIT researchers have developed a system that could bring deep learning

neural networks to new—and much smaller—places, like the tiny computer chips in wearable medical devices, household appliances, and the 250 billion other objects that constitute the "internet of things" (IoT).

The system, called MCUNet, designs compact neural networks that deliver unprecedented speed and accuracy for deep learning on IoT devices, despite limited memory and processing power. The technology could facilitate the expansion of the IoT universe while saving energy and improving data security.

The Internet of Things

The IoT was born in the early 1980s. Grad students at Carnegie Mellon University, including Mike Kazar '78, connected a Cola-Cola machine to the internet. The group's motivation was simple: laziness. They wanted to use their computers to confirm the machine was stocked before trekking from their office to make a purchase. It was the world's first internet-connected appliance. "This was pretty much treated as the punchline of a joke," says Kazar, now a Microsoft engineer. "No one expected billions of devices on the internet."

Since that Coke machine, everyday objects have become increasingly networked into the growing IoT. That includes everything from wearable heart monitors to smart fridges that tell you when you're low on milk. IoT devices often run on microcontrollers—simple computer chips with no operating system, minimal processing power, and less than one thousandth of the memory of a typical smartphone. So pattern-recognition tasks like deep learning are difficult to run locally on IoT devices. For complex analysis, IoT-collected data is often sent to the cloud, making it vulnerable to hacking.

"How do we deploy neural nets directly on these tiny devices? It's a new research area that's getting very hot," says Han. "Companies like Google

and ARM are all working in this direction." Han is too.

With MCUNet, Han's group codesigned two components needed for "tiny deep learning"—the operation of neural networks on microcontrollers. One component is TinyEngine, an inference engine that directs resource management, akin to an operating system. TinyEngine is optimized to run a particular neural network structure, which is selected by MCUNet's other component: TinyNAS, a neural architecture search algorithm.

System-algorithm co-design

Designing a deep network for microcontrollers isn't easy. Existing neural architecture search techniques start with a big pool of possible network structures based on a predefined template, then they gradually find the one with high accuracy and low cost. While the method works, it's not the most efficient. "It can work pretty well for GPUs or smartphones," says Lin. "But it's been difficult to directly apply these techniques to tiny microcontrollers, because they are too small."

So Lin developed TinyNAS, a neural architecture search method that creates custom-sized networks. "We have a lot of microcontrollers that come with different power capacities and different memory sizes," says Lin. "So we developed the algorithm [TinyNAS] to optimize the search space for different microcontrollers." The customized nature of TinyNAS means it can generate compact neural networks with the best possible performance for a given microcontroller—with no unnecessary parameters. "Then we deliver the final, efficient model to the microcontroller," say Lin.

To run that tiny neural network, a microcontroller also needs a lean inference engine. A typical inference engine carries some dead weight—instructions for tasks it may rarely run. The extra code poses no

problem for a laptop or smartphone, but it could easily overwhelm a microcontroller. "It doesn't have off-chip memory, and it doesn't have a disk," says Han. "Everything put together is just one megabyte of flash, so we have to really carefully manage such a small resource." Cue TinyEngine.

The researchers developed their inference engine in conjunction with TinyNAS. TinyEngine generates the essential code necessary to run TinyNAS' customized neural network. Any deadweight code is discarded, which cuts down on compile-time. "We keep only what we need," says Han. "And since we designed the neural network, we know exactly what we need. That's the advantage of system-algorithm codesign." In the group's tests of TinyEngine, the size of the compiled binary code was between 1.9 and five times smaller than comparable microcontroller inference engines from Google and ARM. TinyEngine also contains innovations that reduce runtime, including in-place depth-wise convolution, which cuts peak memory usage nearly in half. After codesigning TinyNAS and TinyEngine, Han's team put MCUNet to the test.

MCUNet's first challenge was image classification. The researchers used the ImageNet database to train the system with labeled images, then to test its ability to classify novel ones. On a commercial microcontroller they tested, MCUNet successfully classified 70.7 percent of the novel images—the previous state-of-the-art neural network and inference engine combo was just 54 percent accurate. "Even a 1 percent improvement is considered significant," says Lin. "So this is a giant leap for microcontroller settings."

The team found similar results in ImageNet tests of three other microcontrollers. And on both speed and accuracy, MCUNet beat the competition for audio and visual "wake-word" tasks, where a user initiates an interaction with a computer using vocal cues (think: "Hey,

Siri") or simply by entering a room. The experiments highlight MCUNet's adaptability to numerous applications.

"Huge potential"

The promising test results give Han hope that it will become the new industry standard for microcontrollers. "It has huge potential," he says.

The advance "extends the frontier of deep neural network design even farther into the computational domain of small energy-efficient microcontrollers," says Kurt Keutzer, a computer scientist at the University of California at Berkeley, who was not involved in the work. He adds that MCUNet could "bring intelligent computer-vision capabilities to even the simplest kitchen appliances, or enable more intelligent motion sensors."

MCUNet could also make IoT devices more secure. "A key advantage is preserving privacy," says Han. "You don't need to transmit the data to the cloud."

Analyzing data locally reduces the risk of personal information being stolen—including personal health data. Han envisions smart watches with MCUNet that don't just sense users' heartbeat, blood pressure, and oxygen levels, but also analyze and help them understand that information. MCUNet could also bring deep learning to IoT devices in vehicles and rural areas with limited internet access.

Plus, MCUNet's slim computing footprint translates into a slim carbon footprint. "Our big dream is for green AI," says Han, adding that training a large neural network can burn carbon equivalent to the lifetime emissions of five cars. MCUNet on a microcontroller would require a small fraction of that energy. "Our end goal is to enable efficient, tiny AI with less computational resources, less human resources, and less data,"

says Han.

More information: MCUNet: Tiny Deep Learning on IoT Devices. arXiv:2007.10319 [cs.CV] arxiv.org/abs/2007.10319

This story is republished courtesy of MIT News (web.mit.edu/newsoffice/), a popular site that covers news about MIT research, innovation and teaching.

Provided by Massachusetts Institute of Technology