# New technique enables on-device training using less than a quarter of a megabyte of memory

October 4 2022, by Adam Zewe


Researchers tested their framework by training a computer vision model to detect people in images. After only 10 minutes of training, it learned to complete the task successfully. Credit: Ji Lin et al

Microcontrollers, miniature computers that can run simple commands, are the basis for billions of connected devices, from internet-of-things (IoT) devices to sensors in automobiles. But cheap, low-power microcontrollers have extremely limited memory and no operating system, making it challenging to train artificial intelligence models on "edge devices" that work independently from central computing resources.

Training a machine-learning model on an intelligent edge device allows it to adapt to new data and make better predictions. For instance, training a model on a smart keyboard could enable the keyboard to continually learn from the user's writing. However, the training process requires so much memory that it is typically done using powerful computers at a data center, before the model is deployed on a device. This is more costly and raises privacy issues since user data must be sent to a central server.

To address this problem, researchers at MIT and the MIT-IBM Watson AI Lab have developed a new technique that enables on-device training using less than a quarter of a megabyte of memory. Other training solutions designed for connected devices can use more than 500 megabytes of memory, greatly exceeding the 256-kilobyte capacity of most microcontrollers (there are 1,024 kilobytes in one megabyte).

The intelligent algorithms and framework the researchers developed reduce the amount of computation required to train a model, which makes the process faster and more memory-efficient. Their technique can be used to train a machine-learning model on a microcontroller in a matter of minutes.

This technique also preserves privacy by keeping data on the device, which could be especially beneficial when data are sensitive, such as in medical applications. It also could enable customization of a model

based on the needs of users. Moreover, the framework preserves or improves the accuracy of the model when compared to other training approaches.

"Our study enables IoT devices to not only perform inference but also continuously update the AI models to newly collected data, paving the way for lifelong on-device learning. The low resource utilization makes deep learning more accessible and can have a broader reach, especially for low-power edge devices," says Song Han, an associate professor in the Department of Electrical Engineering and Computer Science (EECS), a member of the MIT-IBM Watson AI Lab, and senior author of the paper describing this innovation.

Joining Han on the paper are co-lead authors and EECS Ph.D. students Ji Lin and Ligeng Zhu, as well as MIT postdocs Wei-Ming Chen and Wei-Chen Wang, and Chuang Gan, a principal research staff member at the MIT-IBM Watson AI Lab. The research will be presented at the Conference on Neural Information Processing Systems.

Han and his team had previously addressed the memory and computational bottlenecks that exist when trying to run machine-learning models on tiny edge devices, as part of their TinyML initiative.

## Lightweight training

A common type of machine-learning model is known as a neural network. Loosely based on the human brain, these models contain layers of interconnected nodes, or neurons, that process data to complete a task, such as recognizing people in photos. The model must be trained first, which involves showing it millions of examples so it can learn the task. As it learns, the model increases or decreases the strength of the connections between neurons, which are known as weights.

The model may undergo hundreds of updates as it learns, and the intermediate activations must be stored during each round. In a neural network, activation is the middle layer's intermediate results. Because there may be millions of weights and activations, training a model requires much more memory than running a pre-trained model, Han explains.

Han and his collaborators employed two algorithmic solutions to make the training process more efficient and less memory-intensive. The first, known as sparse update, uses an algorithm that identifies the most important weights to update at each round of training. The algorithm starts freezing the weights one at a time until it sees the accuracy dip to a set threshold, then it stops. The remaining weights are updated, while the activations corresponding to the frozen weights don't need to be stored in memory.

"Updating the whole model is very expensive because there are a lot of activations, so people tend to update only the last layer, but as you can imagine, this hurts the accuracy. For our method, we selectively update those important weights and make sure the accuracy is fully preserved," Han says.

Their second solution involves quantized training and simplifying the weights, which are typically 32 bits. An algorithm rounds the weights so they are only eight bits, through a process known as quantization, which cuts the amount of memory for both training and inference. Inference is the process of applying a model to a dataset and generating a prediction. Then the algorithm applies a technique called quantization-aware scaling (QAS), which acts like a multiplier to adjust the ratio between weight and gradient, to avoid any drop in accuracy that may come from quantized training.

The researchers developed a system, called a tiny training engine, that

can run these algorithmic innovations on a simple microcontroller that lacks an operating system. This system changes the order of steps in the training process so more work is completed in the compilation stage, before the model is deployed on the edge device.

"We push a lot of the computation, such as auto-differentiation and graph optimization, to compile time. We also aggressively prune the redundant operators to support sparse updates. Once at runtime, we have much less workload to do on the device," Han explains.

## A successful speedup

Their optimization only required 157 kilobytes of memory to train a machine-learning model on a microcontroller, whereas other techniques designed for lightweight training would still need between 300 and 600 megabytes.

They tested their framework by training a computer vision model to detect people in images. After only 10 minutes of training, it learned to complete the task successfully. Their method was able to train a model more than 20 times faster than other approaches.

Now that they have demonstrated the success of these techniques for computer vision models, the researchers want to apply them to language models and different types of data, such as time-series data. At the same time, they want to use what they've learned to shrink the size of larger models without sacrificing accuracy, which could help reduce the carbon footprint of training large-scale machine-learning models.

**More information:** Ji Lin et al, On-Device Training Under 256KB Memory, (2022).

*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