Technique enables AI on edge devices to keep learning over time

November 16 2023, by Adam Zewe

A machine-learning technique developed by researchers from MIT and elsewhere enables deep learning models, like those that underlie AI chatbots or smart keyboards, to efficiently and continuously learn from new user data directly on an edge device like a smartphone. Credit: MIT News

Personalized deep-learning models can enable artificial intelligence

chatbots that adapt to understand a user's accent or smart keyboards that continuously update to better predict the next word based on someone's typing history. This customization requires constant fine-tuning of a machine-learning model with new data.

Because smartphones and other edge devices lack the memory and computational power necessary for this fine-tuning process, user data are typically uploaded to cloud servers where the model is updated. But data transmission uses a great deal of energy, and sending sensitive user data to a cloud server poses a security risk.

Researchers from MIT, the MIT-IBM Watson AI Lab, and elsewhere developed a technique that enables deep-learning models to efficiently adapt to new sensor data directly on an edge device.

Their on-device training method, called PockEngine, determines which parts of a huge machine-learning model need to be updated to improve accuracy, and only stores and computes with those specific pieces. It performs the bulk of these computations while the model is being prepared, before runtime, which minimizes computational overhead and boosts the speed of the fine-tuning process.

When compared to other methods, PockEngine significantly sped up on-device training, performing up to 15 times faster on some hardware platforms. Moreover, PockEngine didn't cause models to have any dip in accuracy. The researchers also found that their fine-tuning method enabled a popular AI chatbot to answer complex questions more accurately.

"On-device fine-tuning can enable better privacy, lower costs, customization ability, and also lifelong learning, but it is not easy. Everything has to happen with a limited number of resources. We want to be able to run not only inference but also training on an edge device.

With PockEngine, now we can," 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, a distinguished scientist at NVIDIA, and senior author of an open-access paper describing PockEngine posted to the *arXiv* preprint server.

Han is joined on the paper by lead author Ligeng Zhu, an EECS graduate student, as well as others at MIT, the MIT-IBM Watson AI Lab, and the University of California San Diego. The paper was recently presented at the IEEE/ACM International Symposium on Microarchitecture.

Layer by layer

Deep-learning models are based on neural networks, which comprise many interconnected layers of nodes, or "neurons," that process data to make a prediction. When the model is run, a process called inference, a data input (such as an image) is passed from layer to layer until the prediction (perhaps the image label) is output at the end. During inference, each layer no longer needs to be stored after it processes the input.

But during training and fine-tuning, the model undergoes a process known as backpropagation. In backpropagation, the output is compared to the correct answer, and then the model is run in reverse. Each layer is updated as the model's output gets closer to the correct answer.

Because each layer may need to be updated, the entire model and intermediate results must be stored, making fine-tuning more memory demanding than inference

However, not all layers in the neural network are important for improving accuracy. And even for layers that are important, the entire layer may not need to be updated. Those layers, and pieces of layers,

don't need to be stored. Furthermore, one may not need to go all the way back to the first layer to improve accuracy—the process could be stopped somewhere in the middle.

PockEngine takes advantage of these factors to speed up the fine-tuning process and cut down on the amount of computation and memory required.

The system first fine-tunes each layer, one at a time, on a certain task and measures the accuracy improvement after each individual layer. In this way, PockEngine identifies the contribution of each layer, as well as trade-offs between accuracy and fine-tuning cost, and automatically determines the percentage of each layer that needs to be fine-tuned.

"This method matches the accuracy very well compared to full back propagation on different tasks and different neural networks," Han adds.

A pared-down model

Conventionally, the backpropagation graph is generated during runtime, which involves a great deal of computation. Instead, PockEngine does this during compile time, while the model is being prepared for deployment.

PockEngine deletes bits of code to remove unnecessary layers or pieces of layers, creating a pared-down graph of the model to be used during runtime. It then performs other optimizations on this graph to further improve efficiency.

Since all this only needs to be done once, it saves on computational overhead for runtime.

"It is like before setting out on a hiking trip. At home, you would do

careful planning—which trails are you going to go on, which trails are you going to ignore. So then at execution time, when you are actually hiking, you already have a very careful plan to follow," Han explains.

When they applied PockEngine to deep-learning models on different edge devices, including Apple M1 Chips and the digital signal processors common in many smartphones and Raspberry Pi computers, it performed on-device training up to 15 times faster, without any drop in accuracy. PockEngine also significantly slashed the amount of memory required for fine-tuning.

The team also applied the technique to the large language model Llama-V2. With large language models, the fine-tuning process involves providing many examples, and it's crucial for the model to learn how to interact with users, Han says. The process is also important for models tasked with solving complex problems or reasoning about solutions.

For instance, Llama-V2 models that were fine-tuned using PockEngine answered the question "What was Michael Jackson's last album?" correctly, while models that weren't fine-tuned failed. PockEngine cut the time it took for each iteration of the fine-tuning process from about seven seconds to less than one second on a NVIDIA Jetson Orin, an edge GPU platform.

In the future, the researchers want to use PockEngine to fine-tune even larger models designed to process text and images together.

"This work addresses growing efficiency challenges posed by the adoption of large AI models such as LLMs across diverse applications in many different industries. It not only holds promise for edge applications that incorporate larger models, but also for lowering the cost of maintaining and updating large AI models in the cloud," says Ehry MacRostie, a senior manager in Amazon's Artificial General Intelligence

division who was not involved in this study but works with MIT on related AI research through the MIT-Amazon Science Hub.

More information: Ligeng Zhu et al, PockEngine: Sparse and Efficient Fine-tuning in a Pocket, *arXiv* (2023). DOI: 10.48550/arxiv.2310.17752

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