Science | DOI:10.1145/3495562 Chris Edwards

Shrinking Artificial Intelligence

Energy concerns push AI optimizations to the edge.

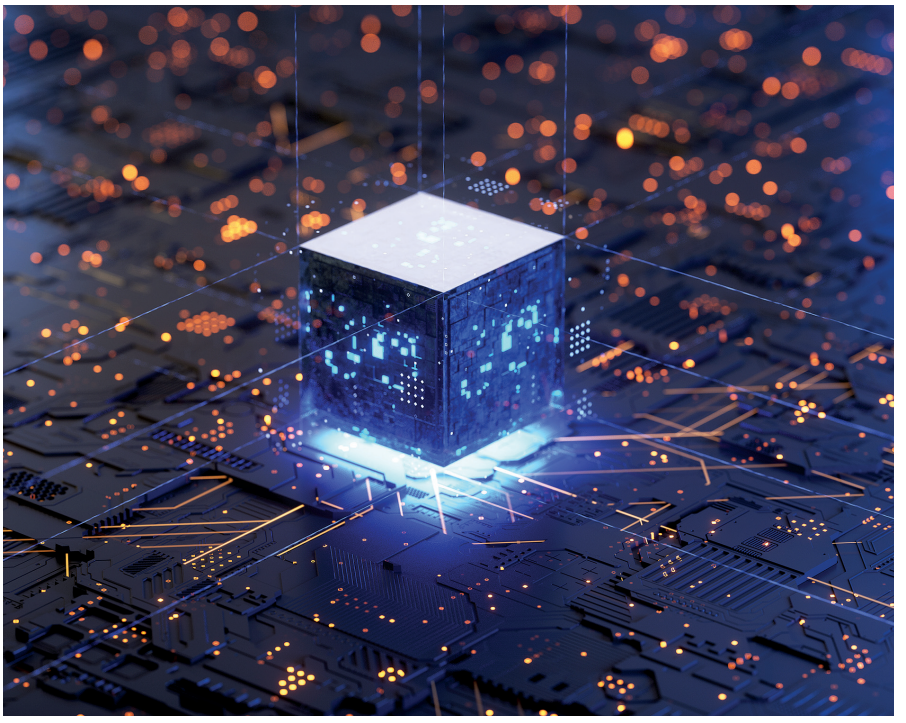


IMAGE BY BLUE ANDY

THE COMPUTATIONAL DEMAND made by artificial intelligence (AI) has soared since the introduction of deep learning more than 15 years ago. Successive experiments have demonstrated the larger the deep neural network (DNN), the more it can do. In turn, developers have seized on the availability of multiprocessor hardware to build models now incorporating billions of trainable parameters.

The growth in DNN capacity now outpaces Moore's Law, at a time when relying on silicon scaling for cost reductions is less assured than it used to be. According to data from chipmaker AMD, cost per wafer for successive nodes has increased at a faster pace in recent generations, offsetting the savings made from being able to pack transistors together more densely. "We are not getting a free lunch from Moore's Law anymore," says Yakun Sophia Shao, assistant professor in the Electrical Engineering and Computer Sciences department of the University of California, Berkeley.

Though cloud servers can support huge DNN models, the rapid growth in size causes a problem for edge computers and embedded devices. Smart speakers and similar products have demonstrated inferencing can be offloaded to cloud servers and still seem responsive, but consumers have become increasingly concerned over having the contents of their conversations transferred across the Internet to operators' databases. For self-driving vehicles and other robots, the round-trip delay incurred by moving raw data makes real-time control practically impossible.

Specialized accelerators can improve the ability of low-power processors to support complex models, making it possible to run image-recognition models in smartphones. Yet a major focus of R&D is to try to find ways to make the core models far smaller and more energy efficient than their server-based counterparts. The

work began with the development of DNN architectures such as ResNet and Mobilenet. The designers of Mobilenet recognized the filters used in the convolutional layers common to many image-recognition DNNs require many redundant applications of the multiply-add operations that form the backbone of these algorithms. The Mobilenet creators showed that by splitting these filters into smaller two-dimensional convolutions, they could cut the number of calculations required by more than 80%.

A further optimization is layer-fusing, in which successive operations funnel data through the weight calculations and activation operations of more than one layer. Though this does not reduce the number of calculations, it helps avoid repeatedly loading values from main memory; instead, they can sit temporarily in local registers or caches, which can provide a big boost to energy efficiency.

More than a decade ago, research presented at the 2010 International Symposium on Computer Architecture by a team from Stanford University showed the logic circuits that perform computations use far less energy compared to what is needed for transfers in and out of main memory. With its reliance on large numbers of parameters and data samples, deep learning has made the effect of memory far more apparent than with many earlier algorithms.

Accesses to caches and local scratchpads are less costly in terms of energy and latency than those made to main memory, but making best use of these local memories is difficult. Gemmini, a benchmarking system developed by Shao and colleagues, shows even the decision to split execution across parallel cores affects hardware design choices. On one test of ResNet-50, Shao notes convolutional layers "benefit massively from a larger scratchpad," but in situations where eight or more cores are working in parallel on the same layer, simulations showed larger level-two cache as more effective.

Reducing the precision of the calculations that determine each neuron's contribution to the output both cuts the required memory bandwidth and energy for computation. Most edge-AI processors now use many 8-bit integer units in parallel, rather than focusing on accelerating the 32-bit floating-point operations used during training. More than 10 8-bit multipliers can fit into the space taken up by a single 32-bit floating-point unit.

With its reliance on large numbers of parameters and data samples, deep learning has made the effect of memory far more apparent than with earlier algorithms.

To try to reduce memory bandwidth even further, core developers such as Cadence Design Systems have put compression engines into their products. "We focus a lot on weight compression, but there is also a lot of data coming in, so we compress the tensor and send that to the execution unit," says Pulin Desai, group director of business development at Cadence. The data is decompressed on the fly before being moved into the execution pipeline.

Compression and precision reduction techniques try to maintain the structure of each layer. More aggressive techniques try to exploit the redundancy found in many large models. Often, the influence of individual neurons on the output of a layer is close to zero; other neurons are far more important to the final result. Many edge-AI processors take advantage of this to cull operations that would involve a zero weight well before they reach the arithmetic units. Some pruning techniques force weights with little influence on the output of a neuron to zero, to provide even more scope for savings.

Unstructured pruning makes it hard to feed the single-instruction

ACM Member News

DESIGNING FUTURISTIC BENCHMARKS

Lizy Kurian John holds the Cullen Trust for Higher Education Endowed Professorship in Electrical Engineering in the Department of Electrical & Computer Engineering at the University of Texas at Austin.

John earned her undergraduate degree in Electronics and Communication Engineering from the University of Kerala in Thiruvananthapuram, India. She went on to obtain her master's degree from the University of Texas at El Paso, and her Ph.D. from Pennsylvania State University at State College, PA, both in Computer Engineering.

After receiving her Ph.D., in 1993 John went to work for the University of South Florida as an assistant professor. She remained there until 1996 when she joined the University of Texas at Austin faculty, where she has remained ever since.

John's research centers on the design, modeling, and benchmarking of computer architectures, focusing on multicore processors, memory systems, performance evaluation and benchmarking, workload characterization, and reconfigurable computing.

She explains, "In order to design efficient computers, you need to evaluate options and design trade-offs."

John sees applications evolving and hardware chasing after applications, and by the time the hardware catches up, the applications have advanced again. She says performance evaluation and benchmarking allows designers to capture applications' essential features that affect performance, power consumption, and other factors influencing application and hardware design.

"Designing futuristic benchmarks helps to design machines for the future," John says, adding that after more than 25 years of doing this research, she thinks the future looks brighter than ever.

—John Delaney

multiple-data (SIMD) pipelines of many processors efficiently. Even if the zero weight calculations are culled to save energy, they lead to execution units being left idle. A combination of software compilation and the use of hardware address generators at runtime reorders the operations to pack as many useful computations as possible into each SIMD group. Any operations that involve a zero operand are left untouched, but the pipeline is kept full by the hardware.

Further savings can come from not running the DNN in every case. Bert Moons, senior engineer at Qualcomm's research group in Amsterdam, says one approach is look at the differences between successive frames of a video: the DNN only operates on parts that have changed. Another uses preprocessing layers to determine which parts of an image are important before engaging the main neural network. For example, the preprocessing layers may filter out areas such as the sky, where there is little visual information that is important to an object-detection task.

Knowledge distillation provides a different avenue for finding savings in large models. In this case, the original model acts as the teacher for a more compact student model that is trained using not just the original input data, but also information from inside the teacher's neural network. For image-recognition tasks, a common technique is to have the teacher give the student the probabilities it generated of a training image being in various classes, instead of the final prediction it would provide to a user. With this extra information, student networks usually perform better than they would compared to those trained purely on the teacher's classifications.

The amount of compression that knowledge distillation provides varies widely. In some cases the savings are minimal, but in work on language, models students have made are 18 times smaller than the teacher, and still only suffer minor losses in accuracy.

In many of the experiments performed so far using distillation, size and structure have a major influence on the ability of smaller student networks to approach the accuracy of the teacher. This observation is helping to drive work on automated neural architecture search, in which machine learning and evolutionary algorithms are used to try to find the best combination of model parameters for a given level of accuracy and computational overhead.

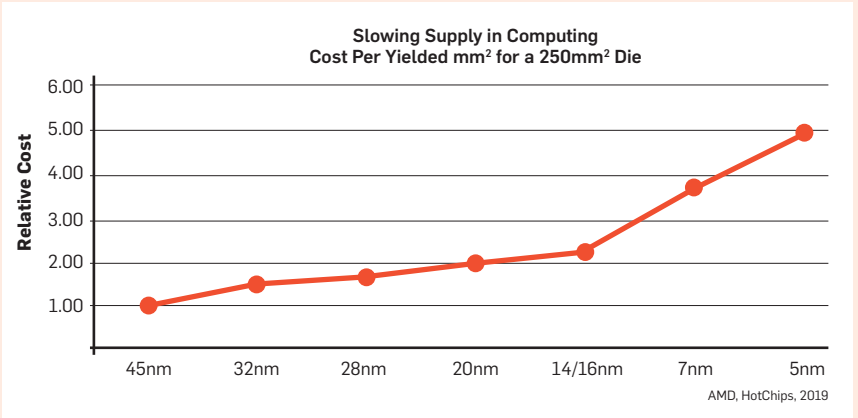
In their work, a group from DarkMatter AI Research and Monash University decided to focus distillation on blocks inside a trained DNN, rather on the network overall, to try to find the best combination of structures for a given size-and-accuracy trade-off. The choice of teacher may not make as much of a difference using current approaches to distillation. The DarkMatter AI group found they could switch to a teacher that was 10 times smaller than the original and build a student with the same level of accuracy as the one derived from the much larger model.

Research by a team from Andrew Gordon Wilson's group at New York University and Google Research published in the summer of 2021 questioned whether the teacher passes on the knowledge that people expect. Student networks often can perform well based on the training they receive, but in tests, provide different answers to the teacher, even when the teacher and student networks have the same structure and capacity.

"It means we can't generally expect the student to behave like the teacher. So, if we are confident that the teacher is a good model, it's hard to have the same confidence about the student," says Wilson. "I find it interesting that it is so hard to achieve a good distillation loss. It turns out to be a much harder optimization problem than we usually encounter in classification. It indicates it may be much harder to train modern deep networks when we move outside standard loss functions."

As with much of the world of DNNs, more work will be needed to understand how these systems generalize and how that can translate into improved compression. However, research has already found numerous ways to avoid joining the parameter arms race that is happening in DNNs that are being trained to run on cloud servers.

Figure 1. Cost per square millimeter for recent semiconductor nodes.



Further Reading

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H.
Mobilenets: Efficient Convolutional Neural networks for Mobile Vision Applications
***arXiv preprint arXiv:1704.04861*, 2017**

Genc, H. et al.
Gemmini: Enabling Systematic Deep Learning Architecture Evaluation via Full-Stack Integration
***Proceedings of the 58th Annual Design Automation Conference (DAC)*, December 2021**

Li, C., Peng, J., Yuan, L., Wang, G., Liang, X., Lin, L., and Chang, X.
Block-wisely Supervised Neural Architecture Search with Knowledge Distillation
***Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020**

Stanton S., Izmailov, P., Kirichenko, P., Alemi, A.A. and Wilson, A.G.
Does Knowledge Distillation Really Work?
***arXiv preprint arXiv:2106.05945*, 2021**

Chris Edwards is a Surrey, U.K.-based writer who reports on electronics, IT, and synthetic biology.