UC SANTA BARBARA

THE Current

January 8, 2025
James Badham

Innovative hardware for rapidly solving optimization problems

The rise of AI, graphic processing, combinatorial optimization and other data-intensive applications has resulted in data-processing bottlenecks, as ever greater amounts of data must be shuttled back and forth between the memory and compute elements in a computer. The physical distance is small, but the process can occur billions of times per second. Inevitably, the energy and time required to move so much data adds up. In response, computer engineers are designing specialized hardware accelerators with innovative architectures to improve the performance of such applications.

Prior efforts to develop hardware for optimization problems have involved Ising machines, a category of hardware solvers that incorporate the Ising model to find the absolute or approximate “ground state,” as in, the energy minimum. Until now, hardware architectures for Ising machines could efficiently solve problems with quadratic polynomial objective functions but were not scalable to increasingly relevant higher-order problems, such as protein folding, electronic-structure prediction, AI-model verification, circuit routing, fault diagnosis and scheduling.

Conducting research in this area is Tinish Bhattacharya, a doctoral student in the UC Santa Barbara lab of electrical and computer engineering professor Dmitri Strukov. He and several industry collaborators, along with academic colleagues in Europe and industrial collaborator Hewlett Packard Labs, have developed specialized function gradient computing hardware to accelerate the rate at which complex high-order

optimization problems can be solved. A paper describing their work, "Computing High-degree Polynomial Gradients in Memory," appeared in the journal Nature Communications. "The objective function of any optimization problem, such as an AI workload, represents an N-dimensional 'energy landscape,' where each combination of variable values represents a unique point in that landscape," Bhattacharya said, noting, "The goal is to find the set of variable assignments that corresponds to the lowest — or more generally, as close as possible to the lowest — point in that landscape."

By way of a parallel, he suggests an actual landscape. "Imagine yourself high in the Sierra Nevada mountains, and your objective is to find the lowest point in a given area, as quickly as possible and with the least possible effort. To achieve that, obviously, you will follow the steepest downward slope. The information about the steepness and the direction in which the steepest slope lies with respect to where you are standing is given by the function's gradient at that point. You proceed by taking incremental steps and recalculating the gradient after each one to confirm that you're still on the steepest slope." This example posits a three-dimensional landscape that could be represented by x, y and z axes, and the gradient calculation is relatively simple. Practical optimization problems, however, may have hundreds of thousands of variables.

"The gradient calculation operation is performed iteratively, over and over, and we need to be able to do it fast and efficiently," he added.

According to Battacharya, much of the currently proposed, state-of-the-art hardware for solving these kinds of issues are limited to second-order problems. The main benefit of their hardware, he noted, is that it can solve problems like Boolean Satisfiability in their native high-order space without having to do any pre-processing, potentially providing exponential speedup over current hardware architectures that are limited to second-order objective functions.

How they do it

A key element of the new hardware is its ability to perform in-memory computing, within the memory array itself, mitigating the bottleneck that results from moving vast amounts of data back and forth between memory and processor in a classic computer. The researchers accelerate operations by performing matrix vector multiplication, the mathematical operation behind the gradient-computation step, by using crossbar arrays of specialized memristor devices.

The great advantage of in-memory computing is that it can be done in a time independent of the size of the matrix. It always requires only one step, with no shuttling of data back and forth, dramatically reducing the time to solve.

The hardware consists of crossbar memories — actual raised surfaces lithographed onto the chip — where several word lines (wires) run horizontally and several bit lines run vertically. Placing a memristor at every location where a word line and a bit line intersect, with one terminal of the device connected to the word line and the other to the bit line, forms a memristor crossbar array. The matrix encoding the problem is stored in the states of these memristors. The vector is applied as proportional read pulses on the word lines. The resulting currents, which flow in the bit lines, then depict the result of the vector-matrix multiplication.

The core innovation that enables gradient computation of high-order polynomials in the native (high-order) space is using two such crossbar arrays back to back. Both crossbars store the matrix depicting the high-order polynomial. The first crossbar computes the high-order monomials of the polynomial. The second crossbar uses this result as its input to compute the high-order gradient for all the variables in each of its bit lines.

This “massively parallel” element of the group’s approach is key to their success. “By that, we mean that our hardware can compute the gradients for each of those variables at the same time, rather than sequentially, as a lot of current hardware does,” Bhattacharya said. “That's the optimization, in one respect, the fact that we have retained that massively parallel property even when going to that high-order space.”

From an algorithmic point of view, the ability to optimize a native high-order function, as opposed to the reduced second-order version, can result in a speed advantage of nearly two orders of magnitude for problems having only 150 variables. That is still an order of magnitude smaller than most practically relevant problems encountered in real-world scenarios, and the speed advantage is expected

to increase exponentially with the addition of more variables.

Tags
Innovation & Entrepreneurship

Media Contact
Shelly Leachman
Editorial Director
(805) 893-2191
sleachman@ucsb.edu

About UC Santa Barbara

The University of California, Santa Barbara is a leading research institution that also provides a comprehensive liberal arts learning experience. Our academic community of faculty, students, and staff is characterized by a culture of interdisciplinary collaboration that is responsive to the needs of our multicultural and global society. All of this takes place within a living and learning environment like no other, as we draw inspiration from the beauty and resources of our extraordinary location at the edge of the Pacific Ocean.