In-Memory Computing Challenges Come Into Focus

by Technical Paper Link 07/09/2023 18:13:59 本文共 1749 字 阅读完需 46 分钟

> In-Memory Computing Challenges Come Into Focus Researchers digging into ways around the von Neumann bottleneck.

For the last several decades, gains in computing performance have come by processing larger volumes of data more quickly and with superior precision.

Memory and storage space are measured in gigabytes and terabytes now, not kilobytes and megabytes. Processors operate on 64-bit rather than 8-bit chunks of data. And yet the semiconductor industry's ability to create and collect high quality data is growing more quickly than the ability to analyze it.

On one hand, the Internet and the Internet of Things are driving a data explosion. In a presentation at the Leti Devices Workshop (a side event to December's IEEE Electron Device Meeting), John Paul Strachan, research scientist at Hewlett Packard Labs noted that Facebook users alone collectively generate 4 petabytes (1 petabyte = 10^{15} bytes) of data per day.

The digital capture of reality — through sensors, cameras, and all the rest — generates much more. A single autonomous vehicle might collect 4 TB of data per day, and in the future a large city might have millions of them. The energy and bandwidth required to merely capture and upload all this information to central data centers is staggering.

Neural networks and the von Neumann bottleneck

Meanwhile, much of the analysis of such large data sets falls to neural networks.

Neural networks work by calculating matrix products and sums. A data matrix is loaded into an array, and each element is multiplied by a predetermined weight. In most cases, the result is passed to the next layer of the network and multiplied by a new set of weights. After several such steps, the result is a conclusion about what the data is. That could be an image of a cat, a suspicious pattern of behaviors, or a particular kind of electrical activity.

In the training stage, the network's conclusion is compared to the previously known "correct" answer. Then a process called back-propagation uses the difference between the predicted and correct values to adjust each weight in each layer of the network up or down.

Conceptually, this approach is very simple. In practice, though, the data sets are huge and the number of calculation steps is enormous. The best performers on the ImageNet image classification benchmark use an 8-layer neural network with 60 million parameters. One pass through the algorithm for one image requires 20 billion operations.

For each layer of the network, both the existing weights and the elements of each training example are loaded into the processor's registers, multiplied, and the result written back out to memory. The performance bottleneck is not the computation, but the bandwidth between the processor and the memory array. This separation between the memory and the processor is one of the defining characteristics of von Neumann architectures, and exists in almost all modern computing systems.

The combination of large data sets, bandwidth-constrained machine learning workloads, and the end of Dennard scaling is shifting industry benchmarks from raw computing performance to computational efficiency. For a given task, what is the best balance among silicon real estate, power consumption, and computational precision?

Low precision, analog memory, high accuracy
In an IEDM presentation, Jeff Welser, vice president and lab director at IBM Research Almaden, noted that high computational precision generally is not needed for neural network calculations. A 16-bit computational block uses a quarter of the circuit real estate needed by an equivalent 32-bit block and cuts the required data volume in half. Reduced precision arithmetic can significantly improve computational efficiency, even with conventional architectures.

The need to overcome the memory bottleneck also is driving more radical compute-in-memory architectures. In the simplest view of such an architecture, pre-determined weights are stored in an array of non-volatile memory elements. Input data is loaded on the memory wordlines, and the currents from individual cells are summed.

Exactly how such a scheme might be implemented in hardware is a topic of ongoing research. Both digital and analog solutions have been proposed. A digital array, for example, might be assembled from flash memory elements. Researchers at the University of Minnesota demonstrated CMOS-compatible eflash memory cells, which store charge on a floating gate between the control gate and the channel. In such an array, both the specific weight values and the rate at which they change (the learning rate) can be precisely controlled through well-established integrated circuit designs. This approach is appealing because it depends on mature, well-understood component technologies.

However, much of the data of interest in machine learning applications is inherently analog. Researcher Xin Zheng and colleagues at Stanford University and UC Berkeley observed that analog-to-digital and digital-to-analog conversions — and their associated energy consumption and silicon footprint — can be avoided by using memory elements like RRAM that inherently store analog values. The analog memory elements that are currently available introduce a host of new challenges, though.

While digital elements are either on or off, analog elements can take on a range of values. The value that is stored by a given signal depends on the properties of the device. In filamentary RRAM, the resistance drops as conductive filaments form between the terminals of the device. A series of weak programming pulses might create weak filaments, while a strong pulse will create a stronger one. Thus, the strength and number of pulses needed to store a given value depends on the kinetics of filament formation. The learning rate depends on the separation between resistance states and the number of pulses necessary to move from one state to the next.

For inference tasks, weights might be calculated using conventional CMOS logic, then stored in an RRAM array. The exact value achieved with a given number of programming pulses might vary from device to device, but simulations suggest overall accuracy is robust in the face of such variations. For learning tasks, though, individual weights need to be adjusted both upward and downward as corrections propagate backward through the network. Unfortunately, current RRAM devices often have asymmetric responses to SET and RESET pulses. Simply changing the sign of a programming pulse won't produce an equal adjustment in the opposite direction. This asymmetry is a major issue for in-memory implementation of learning tasks.

Endurance, stability, and repeatability

As discussed above, learning tasks also require enormous amounts of data and very high numbers of weight updates, somewhere on the order of 10^5 and 10^7, according to Meiran Zhao, a graduate student at Tsinghua University. Tests of RRAM arrays designed for conventional storage applications place the device lifetime in that same range. However, data storage applications require digital values — a device is either on or off — and typically use strong enough SET and RESET pulses to create or remove a strong conductive filament. If weak pulses are used instead, Zhao's group showed that analog switching did not fail after more than 10^{11} update pulses, though learning accuracy did degrade above 10^9 update pulses.

The large number of training cycles required also threatens the stability of the stored weight values. In RRAM devices, the conductivity of the filament is defined by the concentration of oxygen vacancies inside the filament volume. This concentration is in turn controlled by the applied voltage pulses. However, it is not possible to precisely control the locations of individual vacancies. As they migrate within the device, either under the influence of a voltage gradient or after thermal excitation, the exact resistance can vary.

An alternative non-volatile memory, the electrochemical RAM, seeks to address the limitations of filamentary RRAM. While RRAMs are two-terminal devices, ECRAMs are three-terminal devices. The voltage applied to the third terminal controls the intercalation of ions from a LiPON electrolyte layer into a WO_3 conductor. Resistance depends on a redox reaction that can be controlled precisely and repeatably, in both the on and off directions.

Beyond neural networks

While the convolutional neural network is the most common machine learning technique, it is not necessarily the best one. The nonlinear, probabilistic behavior of emerging memory devices is a challenge for some algorithms, but can be an advantage for others.

Generative adversarial networks, for example, use one neural network to generate test examples for another. The "discriminator" network is successful when it can distinguish between real data and the examples generated by the "generator" network. So, for instance, the discriminator network might learn to identify photos of puppies by being shown a set of not-puppy images created by the generator network. One challenge for generative adversarial network algorithms is to generate test examples that cover the full range of real world situations of interest. "Mode dropping," in which generated examples cluster around a limited number of categories, might be reduced by the randomness inherent in RRAM networks. The same nonlinear behavior that makes precise weights difficult to store might lead to more diverse test examples.

RRAM behavior is history-dependent. The probability that a given RESET pulse actually will reset the device drops as the number of previous SET pulses increases. A group at Imec used this behavior as the basis of a learning rule for temporal sequences—devices that are active at time t are used to predict the devices that will be active at time t + Δ. This prediction is compared to the actual data, then devices with correct predictions are strengthened by a SET pulse, while devices with incorrect predictions are weakened by a RESET pulse. After training, the resulting network topology was used as a model to generate new data sequences.

Finally, researchers at the University of Michigan used an RRAM crossbar array in combination with stochastic conductive bridge memory devices to solve "spin glass" optimization problems by simulated annealing. Derived from physics, but also applicable in many other fields, the spin glass problem seeks to find the lowest energy state for a random two-dimensional array of interacting spins. Simulated annealing randomly flips a set number of individual spins, retains those flips that reduce the overall energy of the system, then decreases the temperature of the system and repeats the process. The Michigan group used the stochastic switching probability of CBRAMs to reduce the risk of finding a local minimum state rather than the true lowest energy state.

In-memory computing looks to the future

Historically, electronic device research has come first, and then electrical engineers and software developers have learned how to take advantage of the new capabilities. Over the last few years, emerging memory devices have gone from laboratory curiosities, to disappointingly poor alternatives to flash memory, to enablers of novel machine learning approaches. The next few years will show whether the semiconductor industry can use these devices to help manage the data explosion that it is helping to create.

Related Stories

> In-Memory Vs. Near-Memory Computing

> Using Memory Differently

> Processing In Memory

> Embedded Phase-Change Memory Emerges

# 2.5D   # 5G   # 7nm   # advanced packaging   # AI   # ANSYS

# Apple   # Applied Materials   # ARM   # automotive   # business

# Cadence   # EDA   # eSilicon   # EUV   # finFETs   # GlobalFoundries

# Google   # IBM   # imec   # Intel   # IoT   # IP   # Lam Research

# machine learning   # memory   # Mentor   # Mentor Graphics   # MIT

# Moore's Law   # Nvidia   # NXP   # Qualcomm   # Rambus   # Samsung

# security   # SEMI   # Siemens   # Siemens EDA   # software   # Sonics