TECHNOLOGY WHITEPAPER

Enabling AI & Machine Learning: The Role of Tensor Cores

CURTISS-WRIGHT

DEFENSE SOLUTIONS

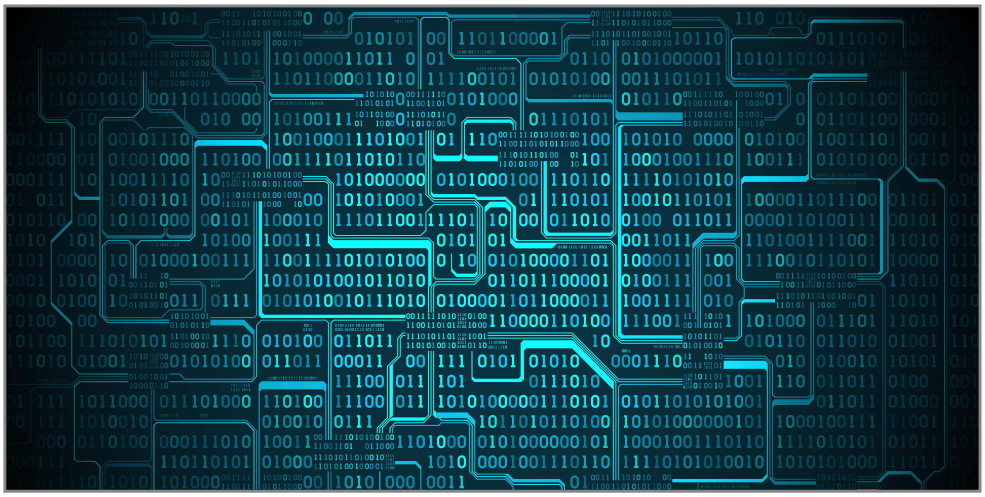
Read About

- What a tensor core is
- How tensor cores enable machine learning and artificial intelligence
- Rugged applications for tensor core technology

What is a Tensor and Why Should I Care?

Over time, the definition of a tensor has varied across different communities, from mathematics to quantum physics. Lately, the term has been embraced by the machine learning community. If you search the web for the definition of a tensor, you will likely be overwhelmed by all the different definitions, explanations, and even heated discussions.

In 1900, Gregorio Ricci Curbastro and his student Tullio Levi-Civita first published their theory of tensor calculus, which is also known as absolute differential calculus. The importance of tensor calculus became abundantly apparent in 1915 when Einstein revealed that he had found it indispensable for the gravitational field equations used in his theory of general relativity. During March and May of 1915, Einstein and Levi-Civita wrote to each other, their correspondence filled with complex mathematical equations, proofs, and counterproofs. All eleven letters that Einstein wrote to Levi-Civita have survived, while only one of Levi-Civita's letters still exists. To honor Levi-Civita, the mathematical permutation symbol,εijk, used in tensor calculus is known today as the Levi-Civita symbol.



Artificial Intelligence and Machine Learning

One way of understanding the importance of tensor calculus is to think about the problem of drawing right angles. If you are developing a system that uses the flat earth model, you can draw right angles using the Pythagorean Theorem. The limits of the Pythagorean Theorem become clear when you try to draw a right angle on a spherical surface. In this case, you will discover that the Pythagoras Theorem no longer works.

It's here that the metric tensor comes to the rescue. It generalizes coordinates and geometries so that distance can be measured in any given space. The magic of tensors comes from their special transformational properties that enable them to describe the same physics in all reference frames. Think of a tensor as a multi-linear map. Given a set of coordinates (or expand out to functions or other objects), each of these coordinates can be transformed according to a set of rules (linear transformations) into a new set of coordinates. The key here is that each coordinate can have a unique transformation. For example, you can stretch or distort different coordinates in different ways. Imagine if we take a rectangular piece of bubble gum with edges on the x-, y-, and z-axes, and then squeeze the bubble gum on the x-axis (one dimension input). The x dimension will compress a certain amount, while the y and z dimensions will expand a given amount, resulting in output changes in three dimensions but maintaining a constant volume. Assuming a linearity of the squeezing reaction, the behavior can also be calculated using a metric tensor if the gum is squeezed off-axis.

Tensors in Artificial Intelligence

Tensors are the most fundamental data type used in all major artificial intelligence (AI) frameworks. A tensor is a container shaped to fit our data perfectly while defining the maximum size of the tensor. Take the example of a processor that's temperature is 45 degrees. First, we need to ask if we are talking about degrees Centigrade or Fahrenheit. The complete answer will require a "denominate" number, which is a number with a name. In this case, the number 45 should be "named" as representing a certain number of degrees Centigrade. This is what is known as a scalar (or 0D) tensor, which is a tensor with 0 dimensions. In contrast, a 1D tensor, better known as a vector, is implemented by an array that stores data in a single row or column. The best-known definition of a vector is an object that has both magnitude and direction.

A more detailed description of a vector is a member of a space where

+ an operation exists that maps two elements in that space to another element in the space, and
+ an operation exists that maps an element in the space and a scalar to another point in that space.

The next step in the hierarchy is a matrix, which represents a two to an n-dimensional tensor. An example of a 3D tensor (or cube) is time series data used with radar processing, which has three parameters (time, frequency, and channel). Described by width, height, and depth (color), a two-dimensional JPG image can be expressed with a 3D tensor. Adding the number of pictures to process grows it to a 4D tensor. A collection of videos would be stored as a 5D tensor (number of videos, number of frames per video, width, height, and depth). As these images process through the deep learning layers, they can be broken down into hundreds of features, thus expanding the number of dimensions.

But beware: even though tensors and matrices appear similar, they are not the same. The following TensorFlow snippet will highlight the difference. Note that NumPy is a general-purpose array-processing package for N-dimensional arrays in Python.

First, we create a 2x2 matrix and initialize its elements to powers of two. Likewise, we create a tensor to perform the same operation. While the output of the matrix is as expected, the tensor output, in contrast, creates a computational graph, which serves as a roadmap to the final answer. Evaluation of the resulting graph produces the expected answer.

Table 1

	MATRIX	TENSOR

Input:	import numpy as np
	np.matrix([[1,2],[3,4]])**2
		import tensorflow as tf
		tt=tf.constant([[1,2],[3,4]])
		t = t**2
		t

Output:	array([[1, 4], [9, 16]]), dtype=int32	<tf.Tensor 'pow_1:0' shape=(2,2) dtype = int32>

Evaluate:		tgo=tf.Session()
		tgo.run(tt)

Output:		array([[1, 4], [9, 16]]), dtype=int32

A tensor decomposition is unique whenever components are linearly independent. In contrast, a matrix decomposition is unique only when components are orthogonal. Compared to traditional matrix-based code, tensor-based modeling is faster and requires less memory space.

Tensor functions fall into one of four main categories: reshaping, element-wise operations, reduction, and access. Some of the tensor reshaping operations include squeeze, unsqueeze, flatten, and reshape. Combining with another tensor will also reshape a tensor. Think back to our earlier chewing gum analogy – reshaping the tensors in a deep learning model can be visualized the same way.

Depending on your mathematical background, your definition and understanding of a tensor may vary. If you're looking for deeper math equations, we recommend reading the NASA paper, An Introduction to Tensors for Students of Physics and Engineering, by Joseph C. Koleck.

Residing where the compute engines, the algorithms and the data intersect, tensors are at the heart of deep learning, and as demonstrated, they easily represent high order relationships. Tensors will often discover hidden relationships that a human did not see in the data and could not program as a feature, and, like linear algebra, tensor algebra is parallelizable. Which brings to mind Albert Einstein's advice: "Everything should be made as simple as possible, but not simpler."

Bringing Tensor Cores to Rugged Applications

As modern developers build AI into the various systems on the battlefield, the need for even more powerful computers at the edge will remain paramount. For some, that will mean the integration of single board computers (SBC), FPGA modules, and/or GPGPU modules housed in OpenVPX™ systems. For others, platforms will integrate small form factor (SFF) standalone line replaceable unit (LRU) mission computers. There are different tools for every problem, depending on the computational performance requirements, SWaP restrictions, and cost and thermal management considerations.

Since traditional CPU-based computers may not keep up with the increasing amount of data flowing into a particular system, the embedded industry is creating more and more GPU-accelerated computing architectures to manage multiple streams of sensor data. GPGPU processing is enabling better intelligence by more efficiently managing the higher data throughput and computing operations that deep learning requires.

Embedded Computing Modules

Curtiss-Wright, in partnership with Wolf Advanced Technology, offers both 3U and 6U high-performance OpenVPX™ GPGPU cards featuring NVIDIA Quadro® Turing™ GPUs. The Turing architecture's tensor cores accelerate the tensor/matrix computation used for deep learning neural network training and inference operations, answering the growing demand for artificial intelligence and high-performance processing in deployed EW and ISR applications.

For 3U VPX systems, the VPX3-4935 delivers incredible processing power from its NVIDIA Quadro Turing TU104 GPU, providing an impressive 3072 CUDA® cores for parallel processing and a 50 percent improvement in performance per CUDA core compared to the previous Pascal generation. Designed to eliminate interoperability challenges between COTS modules, the VPX3-4935 offers a variant developed in alignment with The Open Group Sensor Open Systems Architecture™ (SOSA) Technical Standard. For 6U VPX systems, the VPX6-4955 features dual TU104 GPUs with a remarkable 6144 CUDA cores for parallel processing.

The VPX3-4935 and VPX6-4955 feature NVIDIA Turing GPU technology and are designed for intense processing and AI in 3U HPEC systems

Small Form Factor Systems

Volta, the GPU microarchitecture that succeeded Pascal™, was NVIDIA's first chip to feature tensor cores. In fact, Volta is the GPU microarchitecture within the Xavier generation focused on self-driving cars.

Xavier contains 7 billion transistors and 8 custom ARMv8 cores, a Volta GPU with 512 CUDA cores, and an open sourced tensor processing unit (TPU) called the Deep Learning Accelerator (DLA). With high-efficiency and low power consumption, the Xavier system on a chip (SOC) is the most advanced AI computing solution for SFF embedded systems.

The highly capable NVIDIA Jetson AGX-Xavier SoM can support AI applications for use on the battlefield with 20 times the performance and 10 time the energy efficiency of its predecessor, the NVIDIA Jetson TX2. This module is supported by NVIDIA JetPack and DeepStream SDKs, as well as CUDA, the CUDA Deep Neural Network library (cuDNN), and TensorRT software libraries.

Curtiss-Wright has developed a rugged, embedded compute system based on the NVIDIA Jetson AGX Xavier module called the Parvus® DuraCOR® AGX-Xavier. This robust system is an SFF modular mission computer that integrates NVIDIA CUDA-core accelerated graphics processing, AI and deep learning inference, and edge computing capabilities of the embedded Jetson AGX Xavier SoM in an ultra-rugged chassis optimized for military and aerospace vehicle and aircraft platforms.

Parvus DuraCOR AGX-Xavier

For many, this class of performance resides at a sweet spot for network edge computing. It is more affordable than full-sized VPX solutions, but still offers high FLOPS per watt performance. For very SWaP-constrained deployments, users may scale down to the DuraCOR 312, which features the NVIDIA Jetson TX2i.

With a high degree of modularity, the DuraCOR AGX-Xavier also has I/O flexibility to support interfacing with a variety of different sensors and cameras that might be needed for your AI solution. System I/O expandability supports high-speed NVMe Flash data storage, 10 Gigabit network interfaces, and integration of avionics/vetronics (i.e. MIL-STD-1553, ARINC-429, video capture) and other cards for various sensor payloads. This LRU enables system integrators to harness the supercomputer-class capabilities of the AGX Xavier module and deploy in SWaP-constrained extended temperature, high shock/vibration, humidity, altitude, and noisy EMI environments.

Author

Tammy Carter
Senior Product Line Manager
Curtiss-Wright Defense Solutions

Learn More

White Papers

- How Can I Teach My Machine to Learn?
- Comparing CPUs and GPUs for Deep Learning and Artificial Intelligence
- Enabling AI at the Network Edge of the Battlefield
- Machine Learning and Artificial Intelligence in Defense and Aerospace Applications - What You Need to Know

Products

- VPX3-4935 GPGPU Processor with NVIDIA Quadro Turing TU104/ RTX5000E
- VPX6-4955 GPGPU Processor with Dual NVIDIA Quadro Turing TU104/ RTX5000E GPUs
- Parvus DuraCOR AGX-Xavier

TRUSTED PROVEN LEADER

CURTISSWRIGHTDS.COM