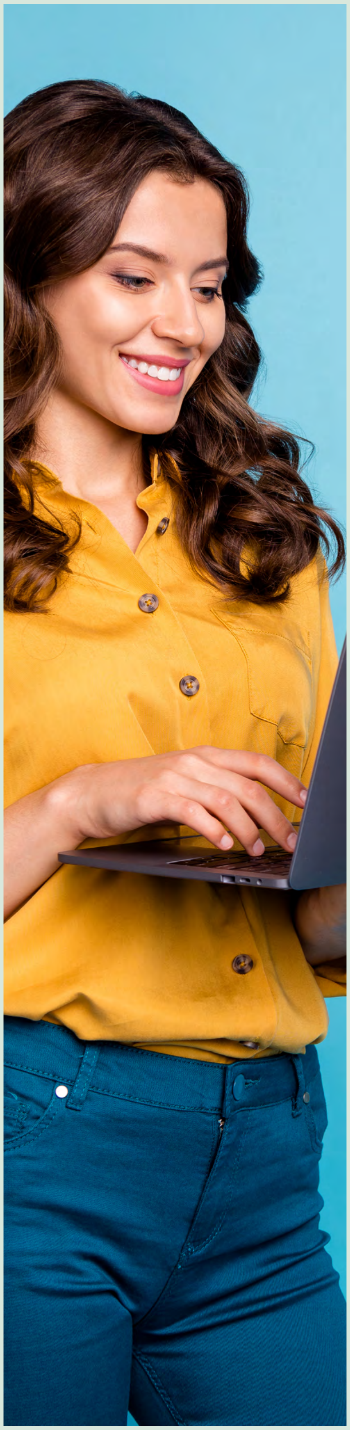
WHITE PAPER



BEYOND GPUs: A HARD LOOK AT TAKING AI EVERYWHERE

Abstract

Thanks to advances in AI, the world is at an inflection point in history where humans and computers share a symbiotic relationship instead of a human-tool relationship. Powering this era and the transformation of the human experience is a host of AI hardware. This paper looks at some of the recent advances in AI hardware, analyzes the latest trends, and examines the energy efficiency of AI architecture.

Navigate your next

Table of Contents

Introduction

Business processes that rely largely on unstructured data – such as documents, email, video, and voice – are evolving thanks to solutions like chatbots, automated voice agents, intelligent document processing, and video processing. Further, there is a new class of AI hardware that is emerging to cater to new customer experiences. In 2017, Google published a revolutionary deep learning transformer architecture. This snowballed into many organisations releasing large transformer-based architectures that generate language, image, and video, promises to drive AI-led transformation for most businesses and their customers.

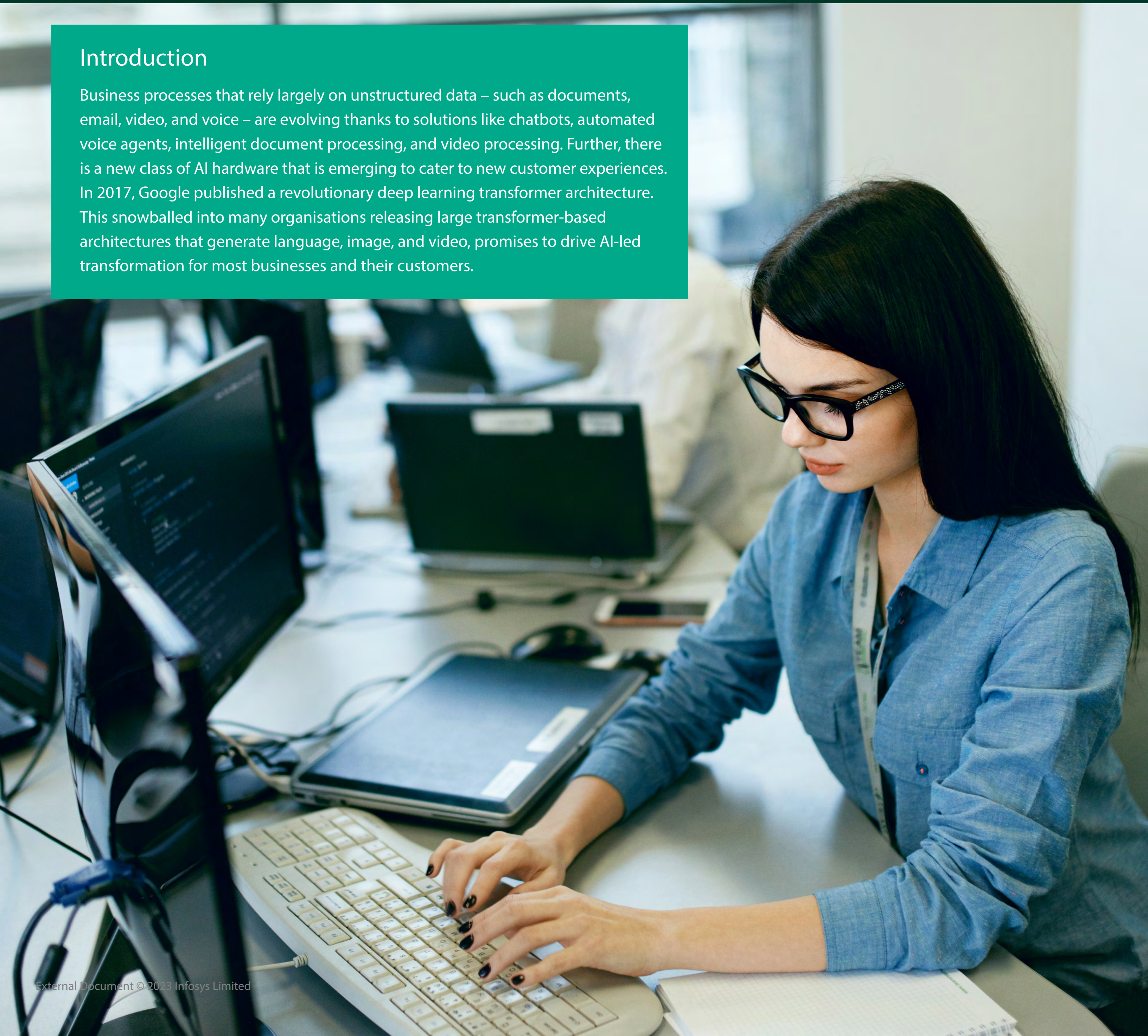
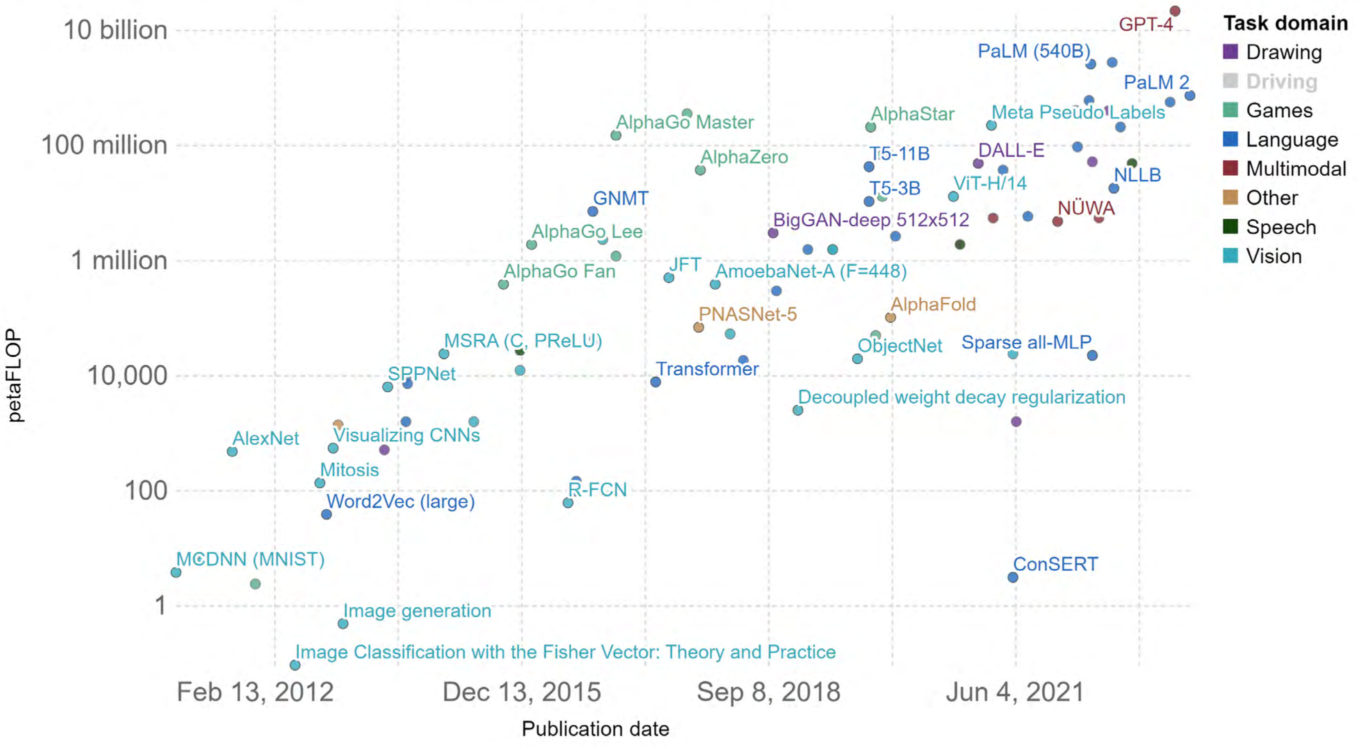


Figure 1: Computation used to train notable AI systems

Computation used to train notable artificial intelligence systems

Computation is measured in total petaFLOP, which is 10^{15} floating-point operations[1] .



Source: Sevilla et al. (2023)

OurWorldInData.org/artificial-intelligence • CC BY

Note: Computation is estimated based on published results in the AI literature and comes with some uncertainty. The authors expect the estimates to be correct within a factor of 2.

[1] **Floating-point operation**: A floating-point operation (FLOP) is a type of computer operation. One FLOP is equivalent to one addition, subtraction, multiplication, or division of two decimal numbers.

AI/ML models are consistently growing bigger. Last year, Microsoft and NVIDIA introduced a model called Megatron-Turing NLG, trained on 530 billion parameters, which is one of the largest models in the world. To put this in perspective, a model of 530 billion parameters can perform approximately 113 teraFLOPs (floating-point operations per second) in a single prediction. This raises the question: How can enterprises adopt AI and what devices are suitable for various AI experiences?

Let us look at various computing devices to understand how established industry players are tackling this question.

Progress in AI models is largely attributed to different deep learning architectures, BERT being a foundational architecture. Linear algebra is the central to all mathematics behind all machine learning and deep learning architectures. According to a research paper titled 'Data Movement is All You Need', 99% of operations involve tensor contractions (or MMM, i.e., matrix-matrix multiplication) in a BERT encoder layer, as shown in the table below.

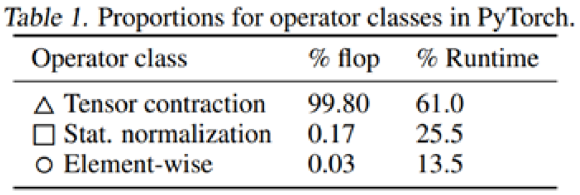


Table 1. Proportions for operator classes in PyTorch.

Operator class	% flop	% Runtime
△ Tensor contraction	99.80	61.0
□ Stat. normalization	0.17	25.5
○ Element-wise	0.03	13.5

To predict outcomes, a simple BERT model requires one sub-word to be represented by 768 vector dimensions (or embedding vectors) followed by multiple matrix multiplications of the similar dimension of learned weights. While traditional CPUs can also execute these calculations, architectures based on graphical processing units (GPUs) are far more efficient due to their multi-core parallel thread executions (SIMD/SIMT architecture). To give an analogy, consider human hands that can perform many tasks.

Yet, these are typically faster at adapting to new tasks with the limitation of doing one task at a time. In other words, it a flexible design that is optimal for performing several unknown tasks. But, when they are compared with robotic arms in a car assembly line, human hands are not as fast or powerful at doing repetitive tasks like fixing or lifting the body parts of a car. Further, given a similar task, there could be multiple robotic hands working in parallel on a production floor to increase the production output. Thus, robotic arms are superior at doing the same task in a massively concurrent fashion, thereby achieving increased production throughput.

Due to heavy data requirements, numerous MMMs, and SIMD architectures, GPUs are the more popular option.

Overview of AI Hardware Products

The following section describes some of the AI hardware products offered by established players. It also explains hardware terms such as GPU, ASSP, ASIC, and FPGA, as well as their possible applications.

1. GPUs – A de facto AI hardware device

GPU or graphical processing unit is a term popularized by NVIDIA. Earlier, these were called 'math co-processors' and were, essentially, a chip on a motherboard other than CPUs. The early iterations of GPUs were adept at floating point operations (FPUs). Today, GPUs are throughput processors. As explained by NVIDIA, they are called general purpose GPUs (GPGPUs).

Driven by relentless innovation, NVIDIA dominates the GPU market with 80% share of global revenue for cloud and data center revenues in 2020, according to a report by Omdia. NVIDIA has various GPU architectures including Tesla, Volta, Ampere, and their very latest, Hopper. The latter two provide unique multi-instance GPU (MIG) capabilities to partition GPUs. They allow workloads of various sizes to run on logically-partitioned GPU slices. Both of these architectures are the most popular within this space. NVIDIA's proprietary CUDA, cuBLAS, and cuDNN are used by most deep learning frameworks like PyTorch, TensorFlow, and MXNet, so that GPUs can be utilized to take advantage of massively parallel calculations.

The recent Hopper architecture launched by NVIDIA has introduced confidential computing and dynamic programming (DPX) instructions along with a specialized transformer engine. It has also created a chip that addresses the specific need for AI transformer architecture. Hopper's Transformer Engine can adapt from FP16 to FP8 during training. It requires minimum precision without losing accuracy. This unique capability reduces the number of calculations to be performed, thereby speeding up the overall training time.

With DPX addressing recursion and temporary results storage, NVIDIA is now targeting niche workloads such as route planning, robotics, and computational biology in its GPUs.

Similarly, AMD has also launched an MI series of AI accelerators, supported by AMD ROCm, as a competitor to the Ampere series of architecture. ROCm is an open-source platform for GPU accelerated computing. It enables OpenCL, equivalent to NVIDIA's propriety CUDA, and supports popular deep learning frameworks.

GPU versus CPU inferencing

Before large language models (LLMs) emerged, there was an ongoing debate on whether to use GPUs for inferencing. But the true issue lies in deciphering the scenarios in which GPUs can be used. For example, consider an application that requires low latency inferencing on the server side, such as an LLM-based chatbot or a video analytics application based on the YOLO (You Only Look Once) algorithm. Here, throughput and latency are important, making it sensible to use GPU-like devices for inferencing.

GPU inferencing is important, considering the need to serve large models in real-time for multiple use-cases. However, there are many scenarios where one might use object detection techniques to detect tables and titles in a document extraction pipeline. For instance, a relatively smaller model like Detectron works well on CPU inferencing. While there are enough use cases where CPUs are used for inferencing, companies remain keen to innovate with GPU inferencing, as mentioned below:

- NVIDIA has released A30, a specialized GPU for inferencing. Another product, T4, and L4 GPUs are also very popular for GPU inferencing. There are also low-powered GPUs by A2 (40-60W range) which support hardware accelerated transcoding for video inferencing applications
- Intel DL Boost using AVX-512 instruction set as well as future (4th Gen) AMX (advanced matrix extensions) to accelerate deep learning workloads enables Intel Xeon – traditionally a top-notch data center CPUs to enable use-cases like model inferencing traditional CPUs, where the existing architecture of CPU chips starts accommodating more AI-specific workload/ instruction sets

Are GPUs the Answer to Everything? A Nano-peek into Hardware Architecture

Calculators run fixed programs require direct instructions. Conversely, modern CPUs that run multiple programs such as word processors, browsers, editors, etc., must load different instructions and different data, and execute these as a series of instructions such as 'load', 'multiply', and 'add'. This means that for every instruction executed inside an arithmetic logical unit (ALU) the results must be stored in L1/L2 caches. The chip architecture that stores both instructions and data is called 'Von Neumann architecture'.

Figure 2 attempts to indicate a very high level logical view of how a single core CPU multiplies two matrices.

The first part indicates the multiplication of two matrices [[1,2], [3,4]], [[5,6], [7,8]] in a single core across different timesteps (T1, T2, etc.). It also indicates how every element of the matrix and every single operation (add and multiply) accesses memory through different color codes.

Figure 2: A single core CPU versus GPU performing multiplication on two matrices

T1	1	*	5	=	5
T2	2	*	7	=	14
T3	5	+	14	=	19
T4	1	*	6	=	6
T5	2	*	8	=	16
T6	6	+	16	=	22
T7	3	*	5	=	15
T8	4	*	7	=	28
T9	15	+	28	=	53
T10	3	*	6	=	18
T11	4	*	8	=	32
T12	18	+	32	=	50

T1	1	*	5	=	5	1	*	6	=	6	3	*	5	=	15	3	*	6	=	18
T2	2	*	7	=	14	2	*	8	=	16	4	*	7	=	28	4	*	8	=	32
T3	5	+	14	=	19	6	+	16	=	22	15	+	28	=	53	18	+	32	=	50

CPU - RISC based instructions

GPU - CISC based instructions, a 4 core indicative display

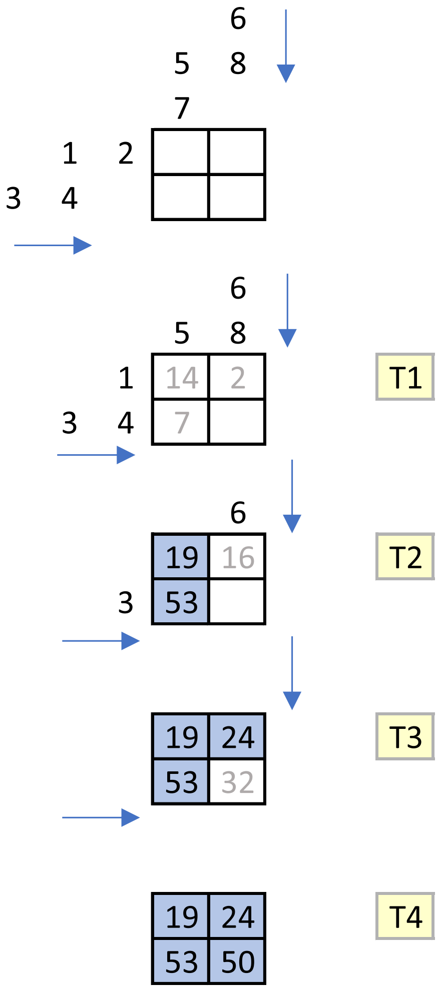
🟨 Cells indicate memory access/storage
🟦 Final Values

How do GPUs work?

The GPU instructions in Figure 2 are a sample four ALUs. Now imagine having thousands of ALUs in a single processor. This means one can execute thousands of multiplications in parallel. However, general purpose GPUs (GPGPUs) must support thousands of instructions for the different applications and software that run on them. It leads to a Von Neuman bottleneck since the thousands of ALUs that perform matrix multiplications must also store the intermediate calculation results in a shared memory and then read from it [yellow cells - in diagram above]. Since GPUs perform more parallel calculations, these require much greater power and consume more energy when accessing the shared memory, subsequently leading to a higher energy footprint.

Figure 3 shows a systolic array with highly optimized high-level instructions. These arrays have a specialized computing unit. It can multiply and add, store a number in place, and pass the original input to either the right or bottom of a connected array, which does a similar operation. Systolic arrays remove the need for shared memory to store interim results. However, the algorithm is not generic enough and, hence, has limited usage. Nevertheless, it still addresses the Von Neumann bottleneck and is very efficient at matrix multiplications.

Figure 3: Systolic Array



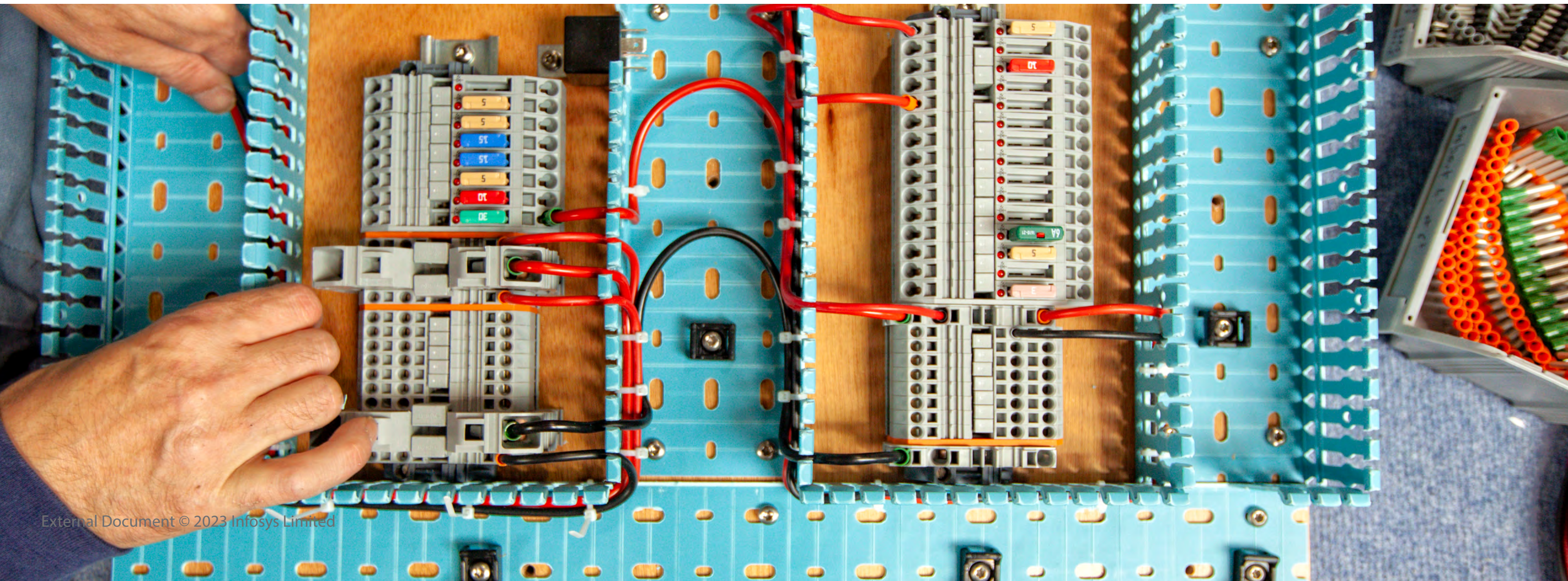
Grey values are temporary

GPGPUs as well as systolic array-based chips are co-processors that cannot work on their own. Thus, they must be paired with CPUs to even read data from a disk or network. Due to this fundamental limitation, CPU-to-GPU communication for data transfers is a challenge during large training jobs and low-latency inferencing use cases.

NVIDIA GPUs have CUDA cores which are more generic GPU cores and Tensor cores which reflect design similar to systolic array specially included for deep learning framework so in that sense it offers best of both worlds.

A degree of this communication challenge can be addressed by using faster PCIe interfaces. Other approaches include NVIDIA's DALI, where pre- and post-inference processing can take place on the GPU that will reduce the communication between CPU and GPU. A recent and very innovative attempt made by NVIDIA has been in Grace Hopper Superchip using Nvlink-C2C to deliver a CPU+GPU coherent memory model which delivers fastest bandwidth whilst optimizing for energy.

There are other hardware approaches via certain devices (mentioned in later sections) that aim to overcome this.



GPUs and energy efficiency

One of the key issues in traditional GPU-based architecture is power consumption and its environmental impact. For example, NVIDIA's latest H100 has a maximum thermal design power (TDP) of 700W, a whopping increase from A100's 400W. These carbon emissions will be a growing concern. According to research in 2019, the average carbon emission to train a base transformer model with 65m parameters with natural language processing (NLP) pipeline is 78,468 lbs, or the equivalent of a human's carbon footprint over nearly 7.2 years.

One of the key factors to determine AI system performance is memory bandwidth. Data is initially loaded on dynamic random access memory (DRAM) and then on the GPU cores. This influences the overall throughput of GPU processes. One way to reduce the power consumption is to use a lower precision (FP16, INT8), which, however, may compromise the accuracy of the models.

Therefore, due to these massive power requirements, deploying and training AI models at the Edge is difficult. Approaches like federated learning is limited to smaller models. While low-powered sensors may support AI deployment, these will still be a challenge.

Recent domain-specific chips can address these energy concerns and are addressed in later sections.

2. ASICs and ASSPs

Application-specific integrated circuit (ASIC) is a customized/specialized GPU that is designed for small computations such as matrix multiplications. They are more effective at these computations compared to GPGPUs, which support more kernels (instructions).

For example, Google tensor processing units (TPUs) are an example of ASICs. These optimize matrix multiplications by using a specialized systolic array chip, where thousands of multipliers and adders are connected directly to form a large physical matrix of operators. There is no need to store intermediate results in a shared memory/cache. Instead, they are stored within the systolic array, resulting in higher throughput and lesser energy expenditure since no shared memory is accessed for every instruction. However, performing dynamic programming (as mentioned in H100 architecture) is challenging with TPU because the instruction support and processor architecture are not generic.

ASIC-based approaches can bolster new products by bringing in a hardware/software co-design era and innovation. Some companies are already doing this, as described below:

- In November 2019, Amazon Web Services (AWS) released Inferentia, a chip that delivers high-performance inference at a lower cost. It supports different deployment patterns such as multi-device inferencing of complex models. AWS Trainium is a purpose-built, deep learning training chip. AWS Neuron SDK supports ML frameworks such as PyTorch and TensorFlow. While there is insufficient information on AWS's chip architecture, they seem to closely follow Google TPUs with similar claims.
- Intel has announced Habana Gaudi 2 and Habana Greco (scheduled for H1 2023) for training and inferencing processors with Synapse AI software suits. These enable integration of popular deep learning frameworks and the writing of custom kernels for expert programmers. In the MLCommons 2.1 training benchmark, Intel Habana Gaudi 2 claims to beat the BERT and RestNet MLPerf benchmark of NVIDIA A100-80G AI processors. It is available on AWS EC2, making it a contender in this space.
- Innovations like on-chip RoCE v2 RDMA allow for Intel-Gaudi processor communication via direct routing or ethernet. This is useful during distributed training for faster communication. More importantly, for vision applications, it offers media decoders (HEVCm, H.264, etc.) as well as post-decode image transformations needed to prepare data for AI pipelines. These are very useful capabilities for video processing applications. It is interesting to note that Habana architecture is classified as Application-specific Standard Parts (ASSP) partly because the architecture seems to be designed for video processing.
- Devices like accelerated processing units (APUs) support neural network-based document search using custom hardware from GSI technology. There are various other examples such as IPU, NPU (NVIDIA Xavier), DPU, XPU, and VPUs, which are different but fall into the similar class of hardware that is designed to be application specific.
- Other organizations like Qualcomm with Mobile AI and some exciting start-ups like GraphCore, SambaNova, Cerebras, and Wave Computing are introducing innovative products by combining AI hardware, and software.
- It is also obvious that AWS and Google both have opted for ASIC-based design since they can offer their cloud data centers. Intel, on the other hand, has invested in ASSP because they can offer the same device to multiple vendors.

ASICs/ASSPs and energy efficiency

Essentially, all ASIC-based chips have internal cores where certain algorithms (matrix multiplication or systolic arrays) are frozen in silicon. This allows higher performance at relatively lower energy use. However, since ASICs/ASSPs are also co-processors, they still have issues with latency.

To summarize, ASIC and ASSP are very similar. While the latter is a little more generic than the former, both are optimized for performance and power when compared to general purpose GPUs.

3. Field programmable gate arrays (FPGA)

FPGAs consist of an array of programmable circuits that can individually do a small amount of computation as well as a programmable interconnect that connects these circuits. FPGAs are classified as spatial architecture based on the translation of high-level language programs directly into hardware structures. Unlike CPUs and GPUs that have a fixed instruction set, FPGA code compiles into specialized circuits. The compilation of FPGA code takes a long time and goes through multiple rounds of performance optimization. Thus, it is a fundamental challenge to program FPGAs.

One of the benefits of FPGAs is that these can be hooked to any data source. They have their own input/output pins and, hence, do not require a computer for hosting. As CPU/GPU data does not have to go through standardized buses, it enables the low-latency application to run much faster than before. It has several applications in radar, astronomy, and military where low-latency processing requirements are a must. Here is an interesting case study on how Microsoft Bing uses FPGAs in Project Catapult.

FPGAs support the operations of intrusion prevention systems (IPS) as well as intrusion detection systems (IDS) by performing continuous network monitoring and tagging live traffic. IDS/IPS typically require 100 GBPS throughput with thousands of concurrent connections with live traffic. Thus, AI-optimized FPGAs are viable in Edge deployments of low-latency inferencing. Similar applications can be considered for database query engines, video processing at scale for thousands of cameras, etc. Some companies are already building use cases for FPGAs, as mentioned below:

- Solutions like Intel Stratix 10 NX FPGAs and Intel Agelix ensure high low-latency memory bandwidth and provide throughput of up to 10 Gbps per less than a watt
- In 2020, AMD (the first company to offer FPGA) acquired Xlinx, the second leading provider of overall AI cloud data center solutions
- AWS provides Amazon EC2 F1 instances
- Azure provides NP-series, powered by Xlinx FPGAs

FPGAs and energy efficiency

Traditionally, FPGA tops the list for its energy-efficient and fixed-precision computations. While it is beneficial for fixed rule-based programs, the same can be implemented in AI using quantized models. Intel Stratix supports floating point units, making them better at floating point computations (AI applications). However, the use of floating points calls for greater power and compromises energy efficiency.

An argument can be made on the grounds of the per watt efficiency. For instance, Intel Stratix can achieve 10 teraFLOPS where the power consumption is listed at 225 watts as compared with NVIDIA V100, which requires 15 teraFLOPS with 250 watts. Nevertheless, the latest Stratix FPGA NX includes 30 tensor cores. This is an important step in the journey of inferencing AI models at low latency and higher throughput, bringing AI applications to multiple use cases.

In conclusion, FPGA can be a self-hosted solution. It is best-suited for low-latency model inferencing in niche use-cases. Microsoft's Project Brainwave (along with Intel's API) will make FPGA applications more accessible to end users. However, it may be a mainstream skill due to its application and underlying hardware complexity.

We haven't covered NVIDIA DPUs and Intel IPU, which are specific devices or coprocessor sitting on top of NIC (network interface card) mainly used by Cloud providers to enable network as code, but they are very similar to FPGA category of cards.

Other Emerging Trends – Analog AI

IBM announced Analog AI, a phase-changed memory-based (PCM) hardware, similar to Memristor, where the calculation is done in-place, instead of fetching details from the DRAM. Fetching data from DRAM is more energy-intensive than performing in-place calculations. Thus, this eliminates the fundamental challenges posed by Von Neumann architecture.

Mythic.ai is a company that champions a similar cause for Edge deployment using compute-in-memory architecture. Here, a matrix multiplication and accumulation operation takes 0.5 pJ instead of 10pJ when deployed at scale. It results in far more energy-efficient throughput, which is particularly useful for battery-powered devices and Edge AI deployments.

Many companies (such Imec and GlobalFOUNDRIES and the Indian Institute of Science) are picking up research on Analog AI.

Recommendation for Application Developers

- How will the model be deployed and served in production, and what devices are possibly available should be kept in mind at the start of the project
- Size of the model – ensure appropriate size of model is small enough to fit into device(s) available for inferencing (as a thumb rule – keep quantization out of scope for estimation)
- It is perfectly okay to train and deploy a model on separate devices. Infact, choose the cheapest/most efficient training device available for training/fine-tuning
- Ensure the data type selected for model weights and training pipeline is supported on training as well as inferencing devices
- Consider throughput and latency demands for model deployment, for heavy data types like video push the model serving either at edge or near the edge
- Consider FPGA only for specific asks. They are not general purposes AI deployments
- Look for vendor specific toolkits and APIs to maximize the utilization of devices during training and inferencing

Conclusion and Takeaways

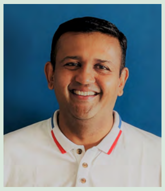
In essence, there are four categories of AI hardware:

- With AI enabling transformative experiences, the use of heterogeneous devices at different lifecycle stages is now a possibility. It will drive the adoption of ONNX and PMML standards. Companies can consider the following AI hardware for different functions:
  - Model training – NVIDIA H/A/V100, Intel's Habana Gudi, Habana Gaudi, AWS Tranium, and Intel TPUs
  - Model inferencing – A10, A30, Xeon 3rd and 4th generation, T4, AWS Inferential, and Habana at server-side/enterprise applications
  - Model inferencing at low latency for niche apps – FPGAs such as Intel Stratix and Xlinx NP series
  - Edge deployment – Low-powered devices is still in experimental space, so look at Edge TPU or device/sensor specific guidelines
- For low-powered Edge devices, compute devices like Coral.ai/Google Edge TPU have an efficiency of about 2 trillion operations per second (TOPS) per watt and represent a completely different set of AI architectures
- Addressing the Von Neumann bottleneck will spur the innovation and use of ASIC-based processors, ushering in an era of new domain-specific hardware and software combined in AI applications
- Fundamentally, new AI learning algorithms like forward-forward instead of back-propagation will enable low-powered/compute devices to come to the fore. This will make 'AI on Edge' for low-powered devices with federated learning a reality
- Hyperscalers like AWS, Google, and Microsoft will offer more ASIC-based hardware owing to higher energy efficiency and better workloads
- Apple's M1 chip (a hybrid chip for CPU and GPU architectures), NVIDIA Hopper superchip presents interesting choices for new chip architectures
- Different computing architectures, neuromorphic chips like IBM's TrueNorth, Intel's Loihi, and quantum computing will drive further energy efficiency and computing power to enable deeper AI applications

The success of AI algorithms in driving hardware innovation is a good indicator for future applications and makes "AI everywhere" a real possibility.

	Energy demand	**Deployment use cases**	**Compute offered**	**Model precision support**	**Known devices**	**Complexity to adopt multiple ML frameworks and other applications**
GPGPUs	Very high	ML training and inferencing	Highest	FP32	NVIDIA H100, NVIDIA A100, AMD M1000	Low
ASICs/ASSPs	High to medium	ML training	Very high	FP32/16	Google TPU, AWS Trainium, Intel Habana	Medium
		ML inferencing in the data center	Very high	FP32/16, BFLOAT16	AWS Infernia NVIDIA A2, T4, L4	Medium
FPGAs	Medium	ML inferencing near the Edge, at lowest latency	Medium	Mostly INT8, FP16 (Limited)	Intel Stratix	Very high
Micro devices	Low	ML inferencing	Low	INT8	Edge TPU	Experimental

Author



Kaushal Desai

Senior Principal Technology Architect, AI Infosys

✉

Additional References

https://www.cs.cornell.edu/courses/cs4787/2022fa/

https://www.anandtech.com/show/17327/nvidia-hopper-gpu-architecture-and-h100-accelerator-announced

https://www.intel.com/content/www/us/en/artificial-intelligence/programmable/fpga-gpu.html

https://towardsdatascience.com/how-fast-gpu-computation-can-be-41e8cff75974

https://thedatafrog.com/en/articles/cuda-kernel-python/

https://hai.stanford.edu/news/ais-carbon-footprint-problem

Energy and Policy Considerations for Deep Learning in NLP

https://cloud.google.com/blog/products/ai-machine-learning/what-makes-tpus-fine-tuned-for-deep-learning

https://blog.inten.to/hardware-for-deep-learning-part-4-asic-96a542fe6a81

https://www.karlrupp.net/2018/02/42-years-of-microprocessor-trend-data/

https://www.cs.cornell.edu/courses/cs4787/2022fa/lectures/lecture22-demo.html

https://towardsdatascience.com/a-complete-guide-to-ai-accelerators-for-deep-learning-inference-gpus-aws-inferentia-and-amazon-7a5d6804ef1c

https://towardsdatascience.com/choosing-the-right-gpu-for-deep-learning-on-aws-d69c157d8c86

https://www.cloudmanagementinsider.com/amazon-inferentia-for-machine-learning-and-artificial-intelligence/

https://developer.nvidia.com/blog/cuda-refresher-reviewing-the-origins-of-gpu-computing/

https://cloud.google.com/blog/products/gcp/quantifying-the-performance-of-the-tpu-our-first-machine-learning-chip/

https://cacm.acm.org/magazines/2020/7/245702-a-domain-specific-supercomputer-for-training-deep-neural-networks/fulltext

https://ebics.net/socfpga/#:~:text=SoC%20FPGAs%20combine%20processors%20and,improve%20the%20reliability%20of%20systems.

https://dl.acm.org/doi/10.1145/3154484

https://blog.esciencecenter.nl/why-use-an-fpga-instead-of-a-cpu-or-gpu-b234cd4f309c

https://www.cs.virginia.edu/~robins/The_Limits_of_Quantum_Computers.pdf

Tensorflow won’t support OpenCL

PyTorch support for OpenCL

For more information, contact askus@infosys.com

Infosys® Navigate your next