ODTÜ METU

TÜBİTAK

EURO 4SEE

Optimizing Deep Learning Systems for Hardware

Assoc. Prof. Erdem AKAGÜNDÜZ, METU

ncc@ulakbim.gov.tr

Part II : Hardware & Memory Hierarchy

- PII.a : Existing Solutions: CPU, GPU, TPU, FPGA, ASIC basics
- PII.b : Memory hierarchy, bandwidth bottlenecks, movement costs
- PII.c : Precision

Existing Hardware Solutions:

- Now that we understand the fundamental computation types and bottlenecks from Part I, let’s look at
  - the major hardware solutions
  - and see how they map to those categories.

Existing Hardware Solutions:

- Now that we understand the fundamental computation types and bottlenecks from Part I, let’s look at
  - the major hardware solutions
  - and see how they map to those categories.

- Hardware types are designed to excel at specific clusters of computations.
- Choosing the right hardware depends on the dominant operations in workloads
- We will review CPUs, GPUs, TPUs, FPGAs, and ASICs in this context.

CPUs (Central Processing Units)

- CPU is a general-purpose processor designed for flexibility, capable of handling diverse tasks and complex control flow, but limited in parallel throughput.
    - Strengths: Flexible, good at logic + branching + control flow, memory access, OS-level tasks.
    - Limitations: Relatively few cores → less efficient for massive matrix multiplications.
    - Best at: Control-heavy tasks, preprocessing, small inference workloads.

MultiCore CPU Systems

- Modern CPUs include multiple cores, allowing parallel execution of tasks.
  - Strengths: Excellent for logic, branching, control flow, and sequential operations.
  - Limitations: Parallelism is limited compared to GPUs/TPUs; not as efficient for large-scale matrix multiplications.
  - Best at: very complex control flow, multiple interface systems

GPUs (Graphics Processing Units)

- GPU is a massively parallel processor optimized for matrix and tensor operations, offering very high throughput but less efficiency for branching-heavy tasks.
    - Strengths: Thousands of cores → excellent at parallel arithmetic (matrix/vector operations).
    - Limitations: Higher latency in single-threaded tasks, less efficient for branching.
    - Best at: Training deep networks, batch inference, dense linear algebra.

GPUs (Graphics Processing Units)

- Is GPU only an NVIDIA thing?
  - No, AMD, Intel, and others also produce GPUs.
  - GPUs are a class of hardware (parallel processors optimized for graphics/matrix math), not a brand.
- Why Deep Learning Platforms need only NVIDIA GPU then?
  - CUDA Ecosystem. NVIDIA developed CUDA, a mature programming framework tightly integrated with deep learning libraries.
  - Popular DL frameworks (TensorFlow, PyTorch, JAX, etc.) are built and optimized first for CUDA.
  - NVIDIA has maintained a strong lead in ML hardware adoption, making it the de facto standard.

TPUs (Tensor Processing Units)

- TPU is a domain-specific processor tailored for deep learning workloads, highly efficient for matrix multiplications and convolutions but less versatile outside ML.
    - Strengths: Specialized for tensor/matrix multiplications common in deep learning.
    - Limitations: Less flexible than GPUs, tailored to DL workloads only.
    - Best at: Large-scale training and inference in tensor-heavy models.

FPGA

- FPGA is a reconfigurable processor that allows custom hardware pipelines, offering low latency and energy efficiency at the cost of programming complexity.
    - Strengths: Reconfigurable hardware, can be optimized for specific arithmetic or logic patterns, low power requirements
    - Limitations: Programming complexity, slower development cycle.
    - Best at: Custom accelerations (low-latency inference, energy-efficient specialized operations).

ASICs (Application-Specific Integrated Circuits)

- ASIC is a fixed-function processor designed for maximum efficiency in a specific workload, achieving the best performance-per-watt but with no flexibility once fabricated.
    - Strengths: Fully customized for one set of operations, highest efficiency and energy savings.
    - Limitations: No flexibility — “locked” to one task.
    - Best at: Dedicated inference accelerators, production environments with fixed workloads.

Comparison

EURO 4SEE

HW	Key Strengths	Limitations	Typical Computation Type	Precision Trends
CPU	General-purpose, flexible, strong control flow, large ecosystem	Low parallelism, lower FLOPs/Watt	Scalar/vector ops, branching-heavy tasks	FP32 / FP64
GPU	High parallelism, excellent for matrix/tensor ops, optimized libraries (CUDA, ROCm)	High power use, less efficient for branching	Matrix multiplications, tensor ops, data-parallel workloads	FP32, FP16, BF16, INT8
TPU	Optimized for deep learning (matmul + conv), high throughput for inference & training	Limited to ML workloads, less general	Tensor ops (matmul, conv), low precision inference	FP32, BF16, INT8
FPGA	Customizable, low latency, energy efficient	Harder to program, long development cycle	Pipelined ops, fixed-function acceleration	Any (user-defined), often INT8 or lower
ASIC	Maximum efficiency for a fixed workload, very low energy cost	Inflexible, huge upfront design cost	Highly repetitive, domain-specific ops	Any (user-defined), often INT8 or lower

Application-specific Compatibility

EURO 4SEE

Case Study	Best Hardware	Why
Edge Devices	ASIC / FPGA / Small GPU	Low energy, compact form factor, efficient for small/medium matmul, inference focus
Cloud / Data Centers	GPU / TPU	High throughput, flexible for multiple tenants, strong software ecosystem
Supercomputers / HPC	CPU + GPU hybrid	HPC tasks often require control flow (CPU) + massive tensor ops (GPU), also support for FP64 precision in scientific computing

Next: Part II.b

- PII.a : Existing Solution: CPU, GPU, TPU, FPGA, ASIC basics
- PII.b : Memory hierarchy, bandwidth bottlenecks, movement costs
- PII.c : Precision (FP32 →INT8) and energy efficiency implications

EURO 4SEE

Thanks!

Co-funded by
the European Union

EuroHPC
Joint Undertaking

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101191697. The JU receives support from the Digital Europe Programme and Germany, Türkiye, Republic of North Macedonia, Montenegro, Serbia, Bosnia and Herzegovina.