

WHAT IS CUDA?

**CUDA (Compute Unified Device Architecture)** is a parallel computing platform and programming model developed by NVIDIA. It enables developers to harness the computational power of NVIDIA GPUs (Graphics Processing Units) for general-purpose processing tasks, in addition to their traditional graphics rendering capabilities.

*Here are some key aspects of CUDA:*

1. **Parallel Computing Platform:** CUDA provides a platform for parallel computing, allowing developers to write programs that can execute tasks concurrently across multiple GPU cores. This enables high-performance computing (HPC) applications, scientific simulations, machine learning algorithms, and other computationally intensive tasks to be accelerated on GPU hardware.

2. **Programming Model:** CUDA introduces a programming model that extends the C/C++ programming languages with GPU-specific constructs, such as kernel functions and thread synchronization mechanisms. Developers write GPU-accelerated code by defining kernel functions, which are executed in parallel by multiple threads on the GPU.

3. **GPU Architecture:** CUDA takes advantage of the massively parallel architecture of modern GPUs, which consist of hundreds or thousands of cores capable of executing computations simultaneously. These cores are organized into streaming multiprocessors (SMs) that manage the execution of threads and coordinate memory access.

4. **Memory Hierarchy:** CUDA provides a hierarchical memory model that includes various types of memory accessible to GPU threads, such as global memory, shared memory, and registers. Efficient memory management is crucial for maximizing GPU performance and minimizing data transfer overhead.

5. **CUDA Toolkit:** NVIDIA provides the CUDA Toolkit, which includes libraries, compiler tools, and development environments for building and optimizing CUDA applications. The toolkit includes libraries such as cuDNN (CUDA Deep Neural Network library) for deep learning, cuBLAS for linear algebra operations, and cuFFT for fast Fourier transforms.

6. **GPU Accelerated Libraries:** In addition to the CUDA Toolkit, NVIDIA offers GPU-accelerated libraries that provide optimized implementations of common algorithms and mathematical functions. These libraries leverage the parallel processing capabilities of GPUs to accelerate computation in various domains, including image processing, signal processing, and numerical simulation.

Overall, CUDA has become a widely adopted platform for GPU computing, enabling developers to unlock the full potential of NVIDIA GPUs for a wide range of applications and industries. Its flexibility,

performance, and extensive ecosystem of tools and libraries make it a popular choice for GPU-accelerated computing tasks.

<https://www.tensorflow.org/install/pip#windows-native> <sup>1</sup> helps to install tensorflow gpu and cpu versions in windows.

*nvidia-smi* command verifies whether nvidia GPU driver present in the machine or not.

<https://pytorch.org/get-started/locally/> helps to install pytorch gpu and cpu versions in windows.

**It is necessary to install anaconda before pytorch and tf.**

WHAT IS CUDA?