CS 181AI
Lecture 12

GPU Programming Pt. 1

Arthi Padmanabhan

Feb 27, 2023

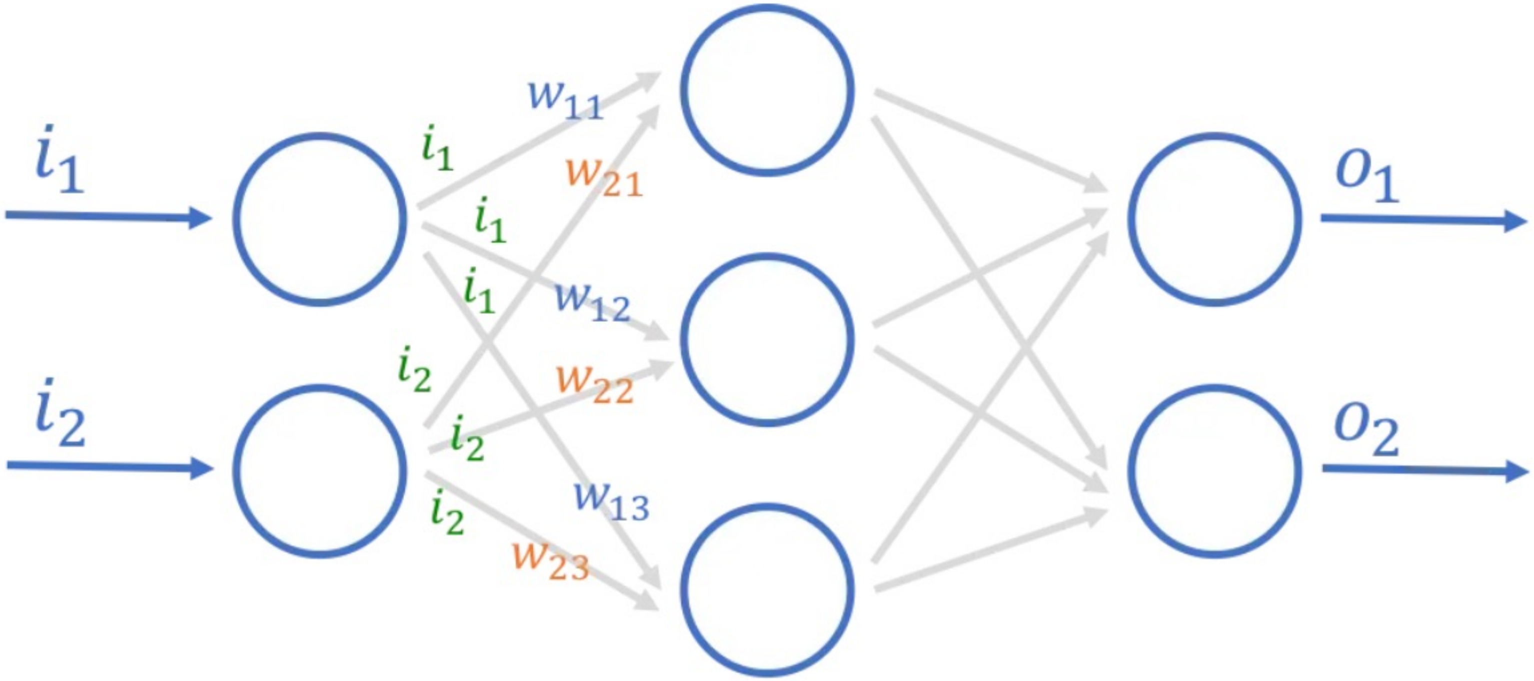
Last Time

- Review of CPU architecture
- GPU architecture
- GPUs consists of hundreds or thousands of ALUs
- ALUs are divided into groups, and a whole group follows the same instruction but each on a different section of the data (Single-Instruction, Multiple Data)

Important Concepts

- What is PCIe bus? Is it fast or slow?
- When you write code for the GPU, do you initialize your data, model, etc. on the CPU or GPU?

Review of ML Operations



$$\begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \\ w_{13} & w_{23} \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} (w_{11} \times i_1) + (w_{21} \times i_2) \\ (w_{12} \times i_1) + (w_{22} \times i_2) \\ (w_{13} \times i_1) + (w_{23} \times i_2) \end{bmatrix}$$

Matrix Multiplication

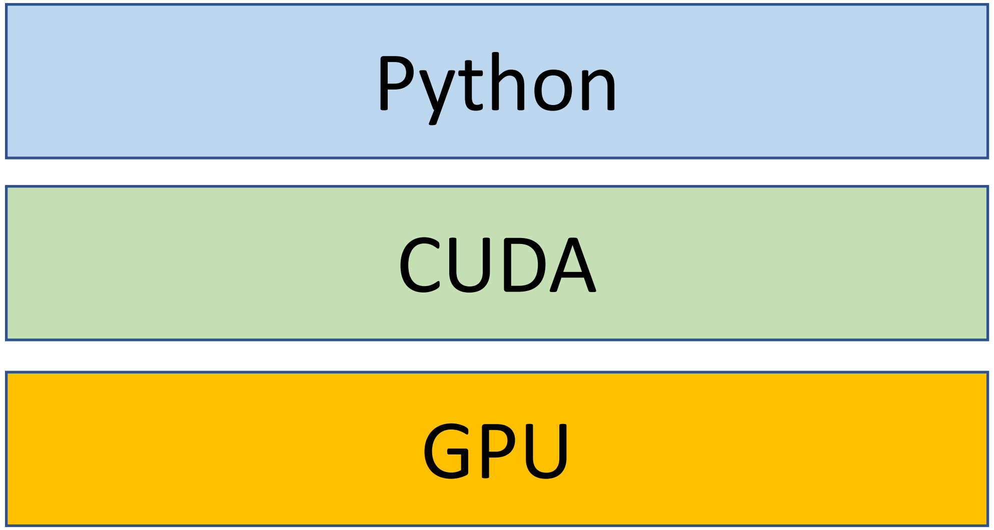
- Write a function to multiply two matrices using Python loops and time the function

Numpy

- import numpy as np
- Python wrappers around C code
- You can create numpy matrices of random numbers in any dimension using `np.random.randint`
  - ex/ `np.random.randint(1,10,size = (1,2))`
  - for a 1x2 (2D) structure
- Write a function to multiply two matrices using numpy and time the function

Intro to Using the GPU

- Nvidia has a software platform that pairs with their GPUs. This is called CUDA
- You can write CUDA code through Python (comes baked into PyTorch)



Python

CUDA

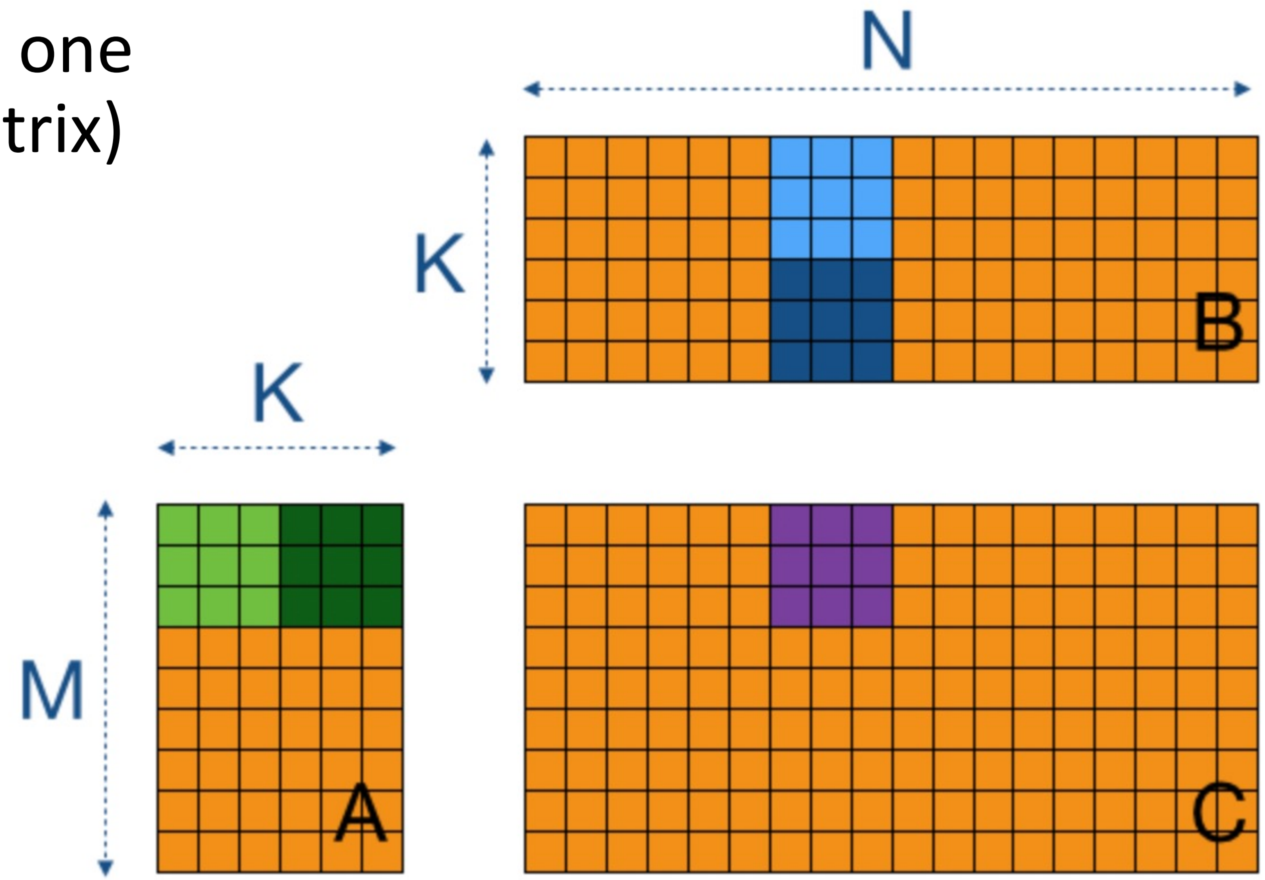
GPU

Torch tensors

- Tensor: an n-dimensional matrix
- To use the GPU, you must create a torch tensor and move it to the GPU
  - Let’s start with an example of moving a tensor back and forth
- You can create random tensors using torch.rand, ex/ `torch.rand([1,2])`
- You can use torch.matmul to multiply two torch tensors
- Create two tensors of random numbers and write a function to multiply them
  - Remember to control the flow to the GPU and back, i.e., your function must return a tensor on the CPU

Block Multiplication

- Each core is responsible for one block in C (the resulting matrix)



Next Time

- While CUDA takes care of assigning work to each core, we can still take a more fine-grained look at these assignments and even control them
- We'll look at GPU architecture in more details and how to control assignments with some new libraries (PyCUDA, Numba)