Why GPUs for Deep Learning?

Julien Nyambal

Capitec Bank

February 23, 2023

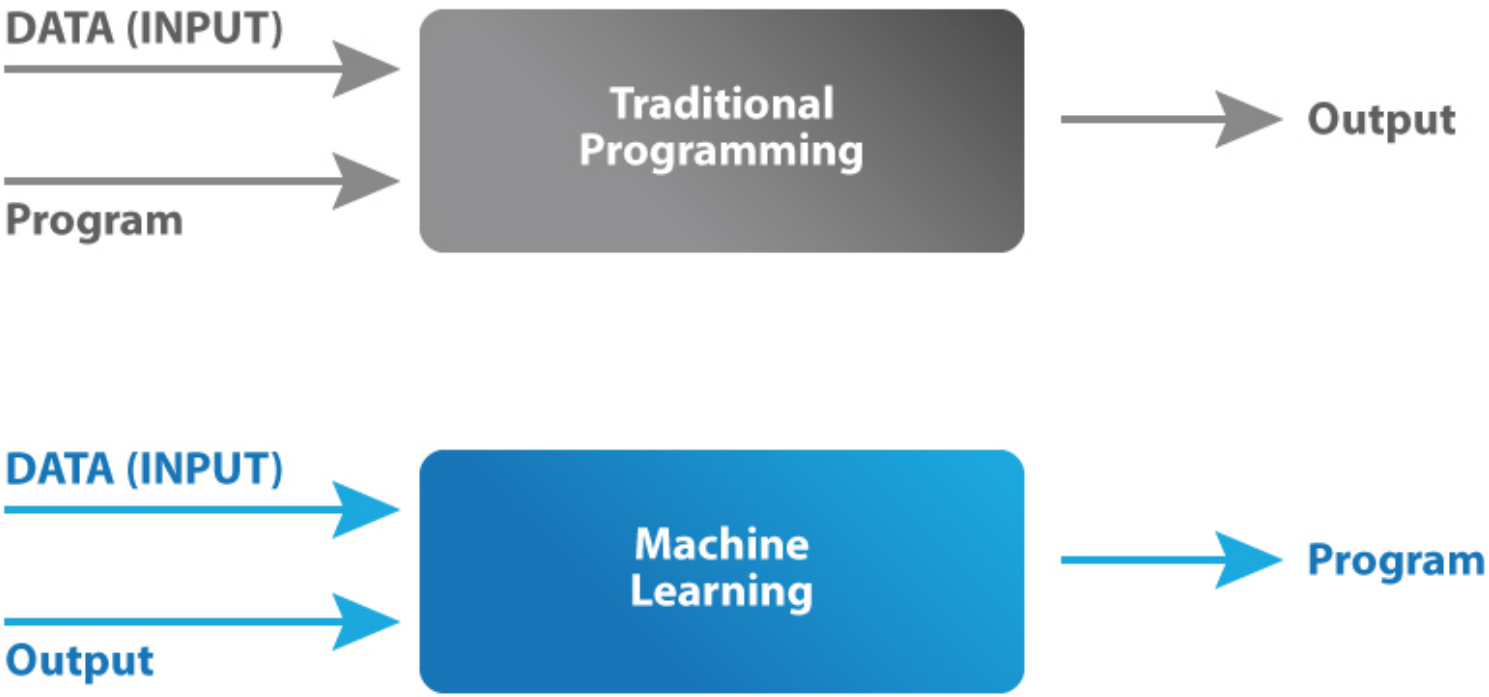
While a CPU is the brains of a computer, GPU is its soul ... Nvidia

What will be covered ...

1. What is Machine Learning?
2. RAM, CPU, GPU, TPU
   - Description
   - TPU: Tensor Processing Unit
3. Hardware Comparison CPU vs GPU
   - Hardware Comparison CPU vs GPU
4. NVidia GPUs - CUDA APsI
5. Tensor Operations on Hardware
   - Matrix Computation on GPU vs CPU
6. How does GPU works "faster" than the CPU?
7. Very short demo
8. References and Extended Reading

CAPITEC

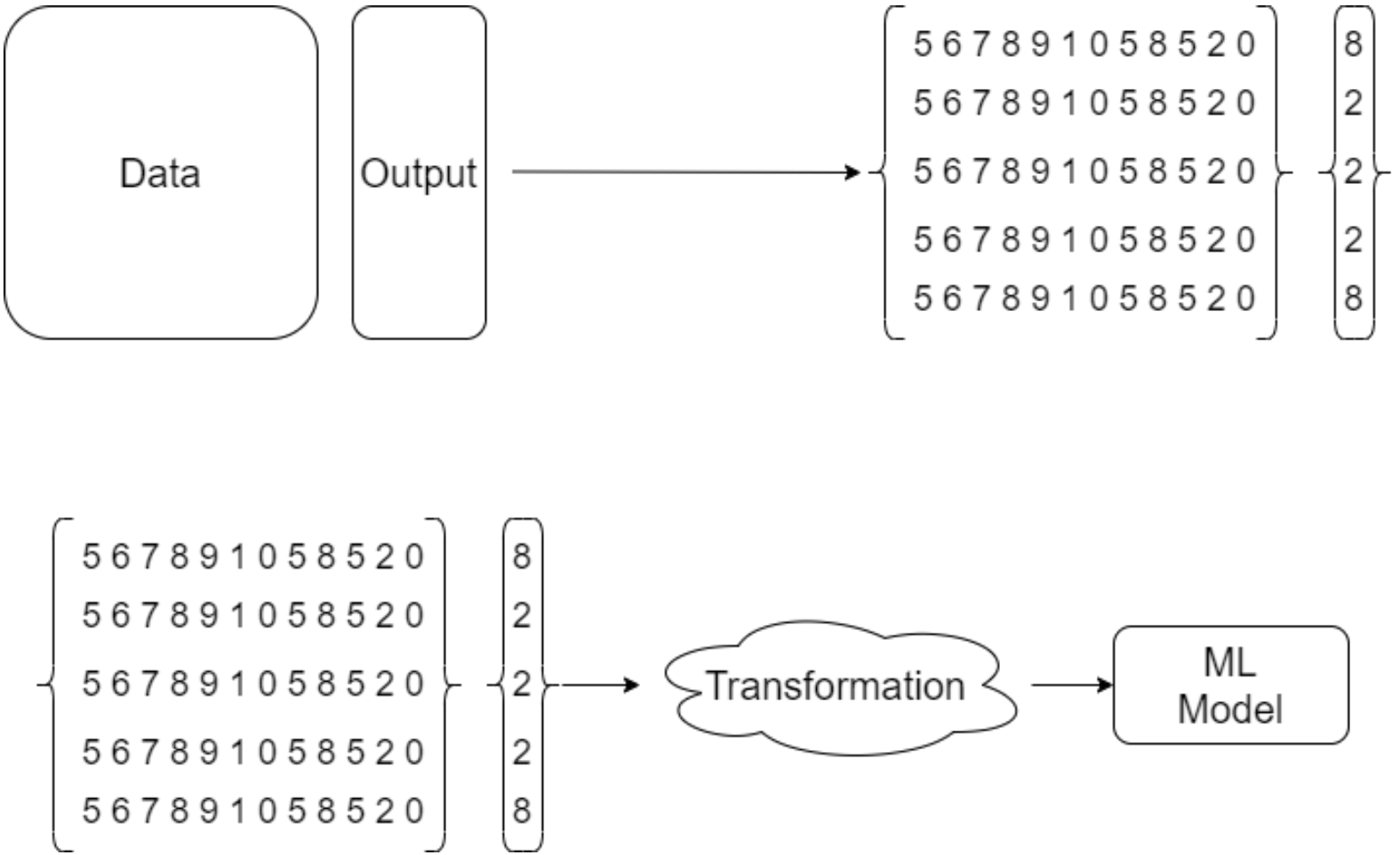
What is Machine Learning?



A general definition of a ML

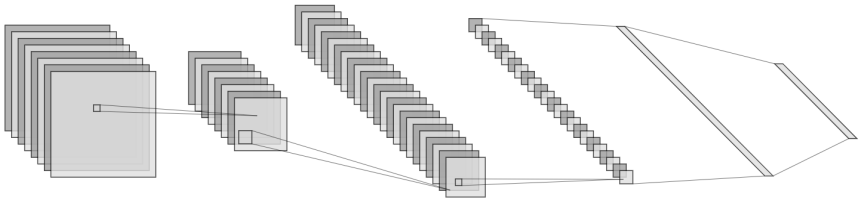
CAPITEC

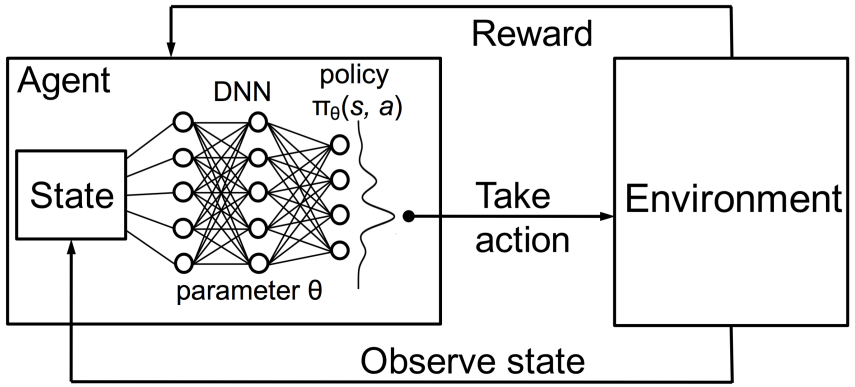
What is Machine Learning?: Conceptual Overview

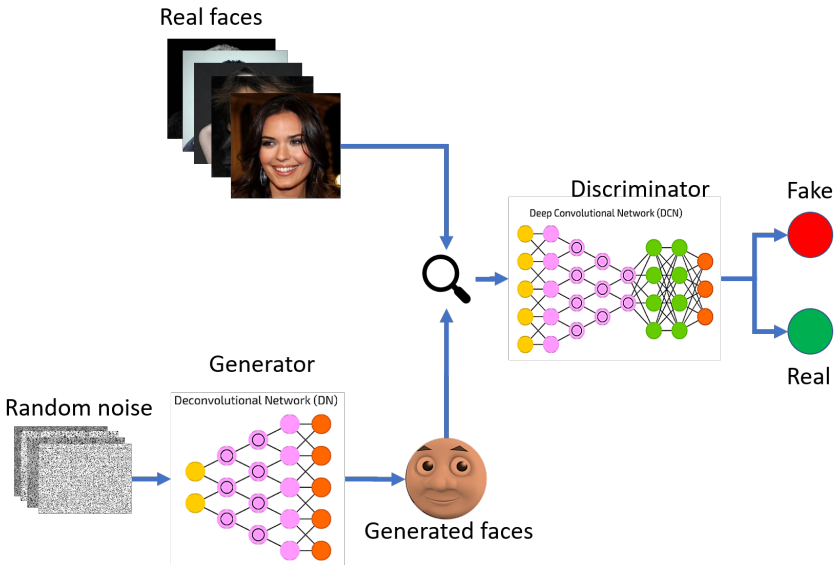


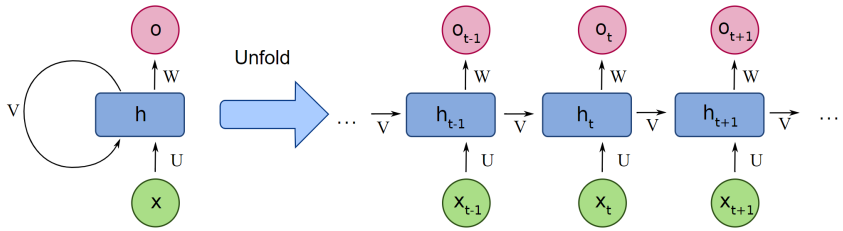
ML - Conceptual

What is Machine Learning?: Deep Learning







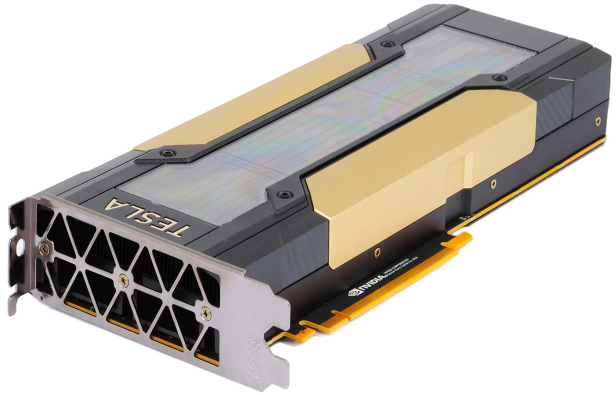


CAPITEC

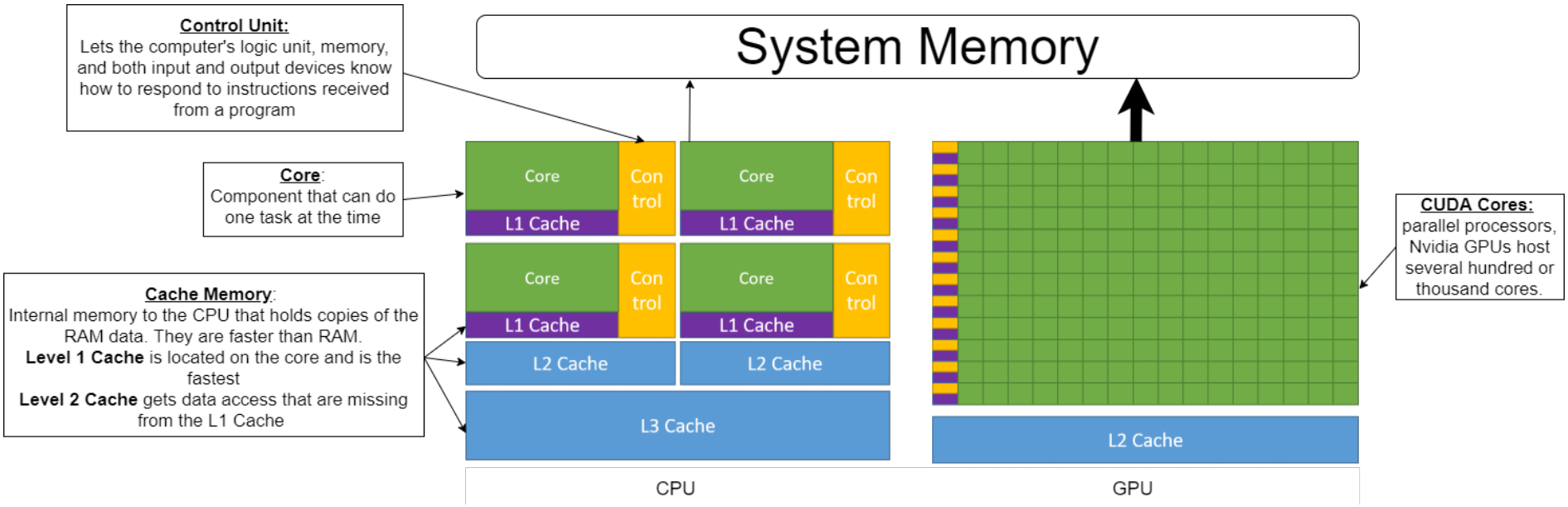
RAM, GPU, CPU, TPU

Acronym	Central Processing Unit	Graphical Processing Unit	Tensor Processing Unit
Description	Performs basic logic, arithmetic, and I/O operations, and allocate commands to other components and subsystems running in a computer. It is the orchestrator of the other hardware components of the computer system, including the GPU. Most systems have **a shared GPU resource for basic display**.	**Dedicated** device designed specifically to accelerate computer graphics workloads, particularly for 3D graphics. While they are still used for their original purpose of accelerating graphics rendering, GPU parallel computing is now used in a wide range of applications, including graphics and video rendering.	Google's custom-developed application-specific integrated circuits used to accelerate **machine learning** workloads, by accelerating the performance of linear algebra computations. Like a dedicated ***Machine Learning CPU***(link)
Processing style	Sequential (But multitask)	Distributed (Given a single heavy-duty load)	Distributed
Graphic Rendering API	OpenCL	CUDA (Nvidia)	TPU Machine Instructions
Specialty	Orchestrating several tasks performed by the OS	• Graphically intense video game Image Rendering • Distributed Complex matrix operation: **Machine Learning** and **HPC** • Video Editing	Handles very specific mathematical operations: • Matrix Multiply Unit (MXU) which runs matrix multiplications • Vector Processing Unit (VPU) for all other tasks such as activations, softmax, ...

RAM, GPU, CPU, TPU



Hardware Comparison CPU vs GPU



Hardware Comparison CPU vs GPU

Processor Type	CPU	GPU
Processor Model	i9 7980XE	Tesla V100
Manufacturer	Intel	Nvidia
Processor Speed (Max)	**4.3 GHz**	1.380 GHz
Memory (up to)	**128 GB (RAM)**	32 GB (GRAM)
Number of Cores	18	**5120 (CUDA cores)**
Single Precision GFLOPS	13.1	**148.992**
Memory Bandwidth	45.8 GB/s	**900 GB/s**
Price	~**R 16 000**	~R 200 000

CAPITEC

NVidia GPUs - CUDA API

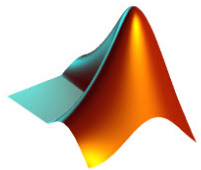
- CUDA is a parallel computing platform and programming model developed by NVIDIA for general computing on graphical processing units (GPUs). With CUDA, developers are able to dramatically speed up computing applications by harnessing the power of GPUs.
- CUDA is a programming language that uses the Graphical Processing Unit (GPU).
- Its is an extension of C++, to provide basic instructions to GPU CUDA cores. The developer needs to be very much aware of C++ Memory Management for best performance.

CAPITEC

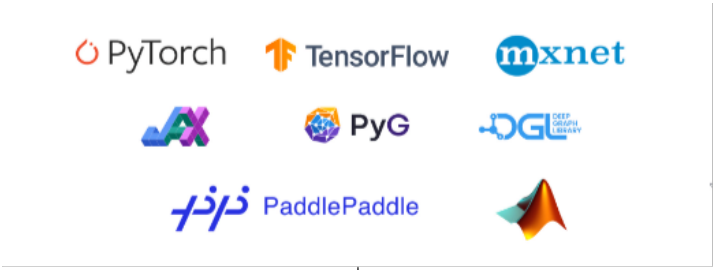
NVidia GPUs - CUDnn

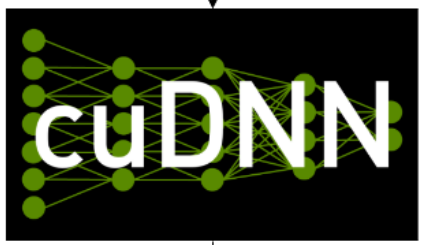
Deep Learning specific CUDA GPU-accelerated library for deep neural network routines likes:

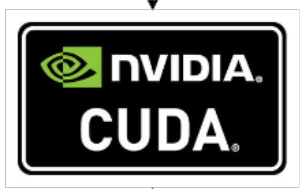
- Back-and-forth Cross-correlation and/or convolution, neuron activations, pooling ,batch normalization
- Tensor transformations
- Equivalent: BLAS, cuBLAS, Eigen



NVidia GPUs - CUDnn







PITEC

Tensors: Operations - Addition

Original Image

CAPITEC

Tensors: Operations - Addition

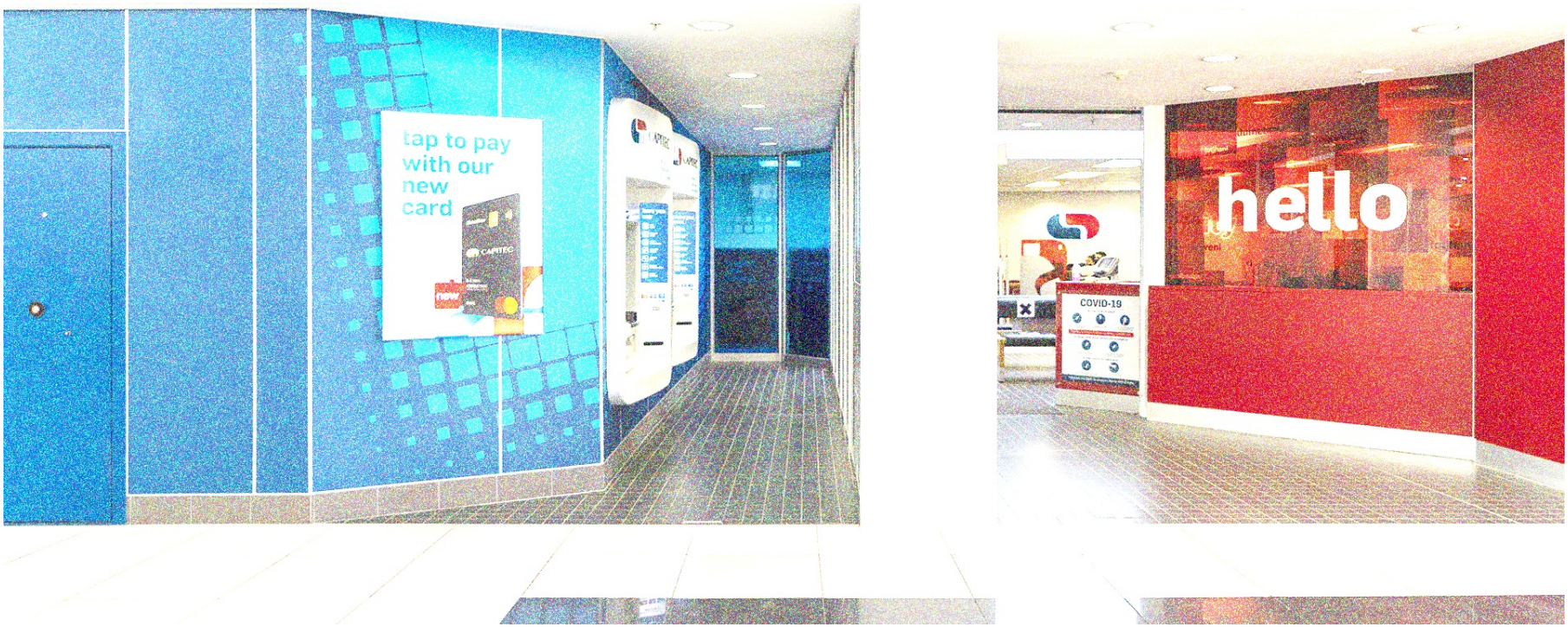
Noise

Tensors: Operations - Addition

Image + Noise

CAPITEC

Tensors: Operations - Addition



(Image + Noise) + Image

CAPITEC

Tensors: Operations - Matrix Multiplication

- The most used operation in Machine Learning/Deep Learning
- There are many types of Matrix Multiplication of ***MatMul***. There 3 are the most common matrix multiplication:
  - Brute Force Multiplication
  - Column-Wise Multiplication
  - Block Multiplication

CAPITEC

Tensors: Operations - Matrix Multiplication

Image

CAPITEC

Tensors: Operations - Matrix Multiplication

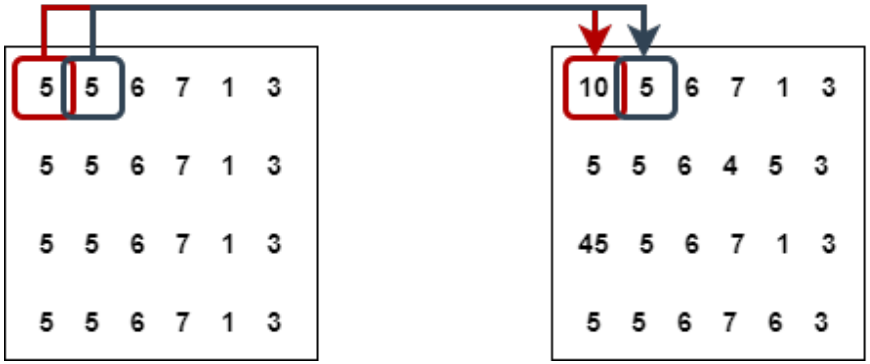
Convolution 1

CAPITEC

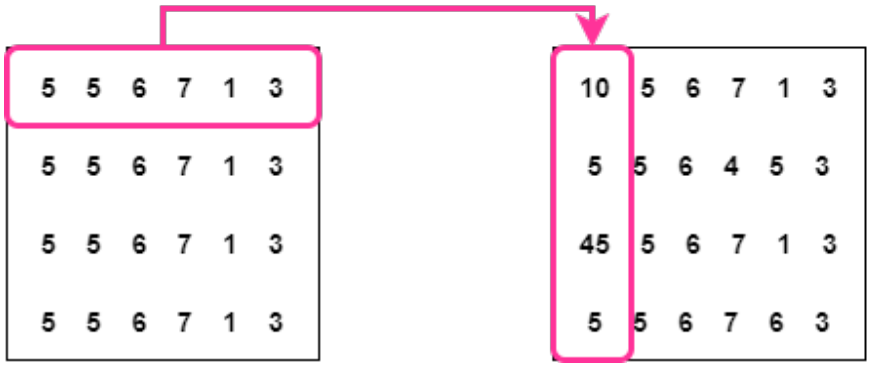
Tensors: Operations - Matrix Multiplication

Convolution 2

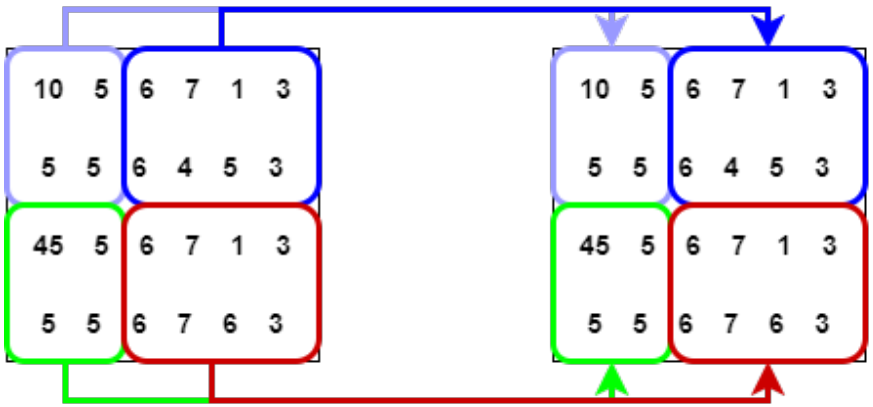
Tensors: Operations - Matrix Multiplication



Brute Force Matrix Multiplication



Column-Wise Matrix Multiplication



Block Matrix Multiplication

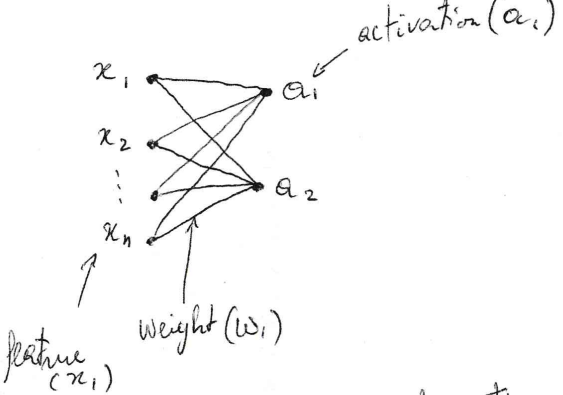
CAPITEC

Tensors: Operations - Matrix Multiplication

Forward Propagation

$f(w_i, x_i) = W \cdot X + b \quad (1)$

Assuming:



To calculate the activations

$a_1 = w_{1,1} \times x_1 + w_{1,2} \times x_2 + \cdots + w_{1,n} \times x_n + b$

The above applies for all activations

By hand, and that way this is tedious!

Using Matrix notation:

$$\begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,n} \\ \vdots & & & \\ w_{n,1} & w_{n,2} & \cdots & w_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

weight of size 2×4

data $n \times 1$

bias 2×1

activation 2×1

In the CPU: Individual operation are done at a time

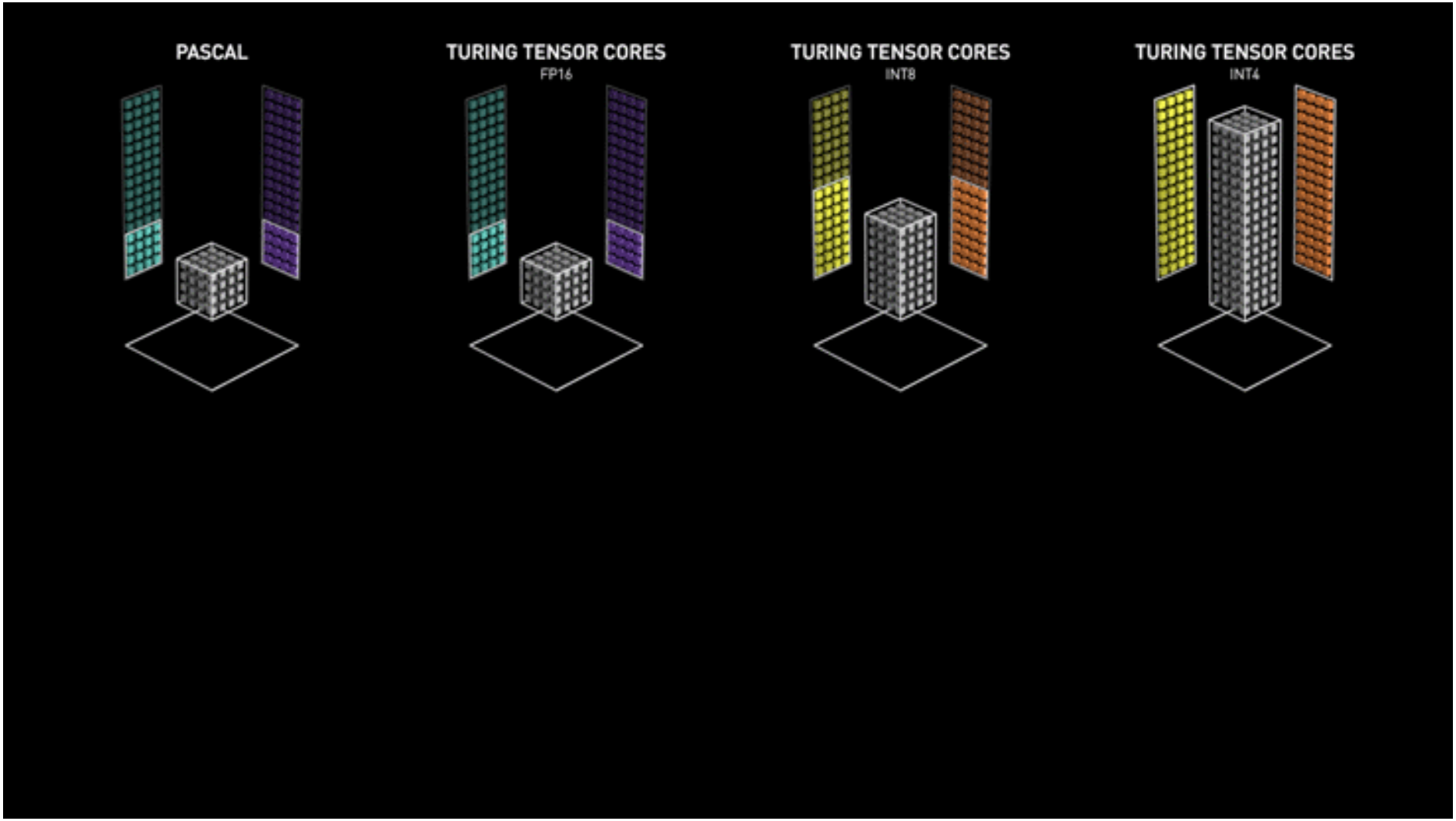
$= w_{1,1} \times x_1 + w_{1,2} \times x_2 \cdots$

The CPU is pretty fast but if the are many rows and columns it, will take time.

In the GPU Batch operations:

Instead of a single value computed, the GPU takes in a whole row multiplied by a column.

Matrix Computation on GPU vs CPU



CAPITEC

How does GPU works "faster" than the CPU?

- **Larger memory bandwidth**: More data can get in and out the component at a time,
- **Parallelization**: Those many CUDA cores work well in parallel for one given task
- **Fast Memory access**: Multiple L1 and L2 Cache memory,
- CUDA API allows you not to worry about memory allocation or deallocation, type of Matrix Multplication to use or how to orchestrate the parallelism of the cores. Some popular frameworks using CUDA:

CAPITEC

Demo

We will run short 2 experiments on both the CPU and the GPU:

- Multiplication of 2 scalar: 8.8 x 8.9
- Multiplication of 2 relatively big matrices

Demo Link: Colab Experiment.

CAPITEC

References and Extended Reading (Links)

- GPU-Parallel-Computing
- CPU-vs-GPU 1
- CPU-vs-GPU 2
- TPU
- Keras
- TPU
- CUDA
- Intro to CUDA
- CUDnn
- Fundamental DL: Pytorch
- NVidia Titan V
- i9

CAPITEC