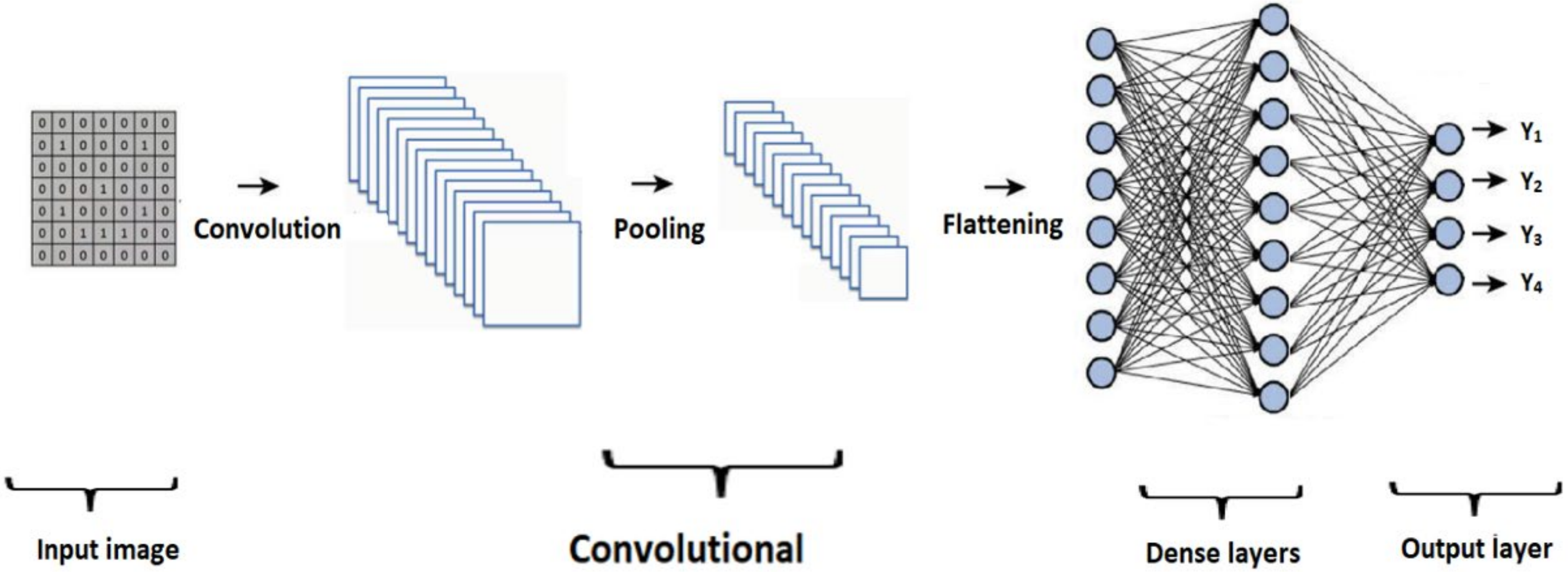
Convolutional layer & GPUs

THE UNIVERSITY OF
SOUTHERN MISSISSIPPI

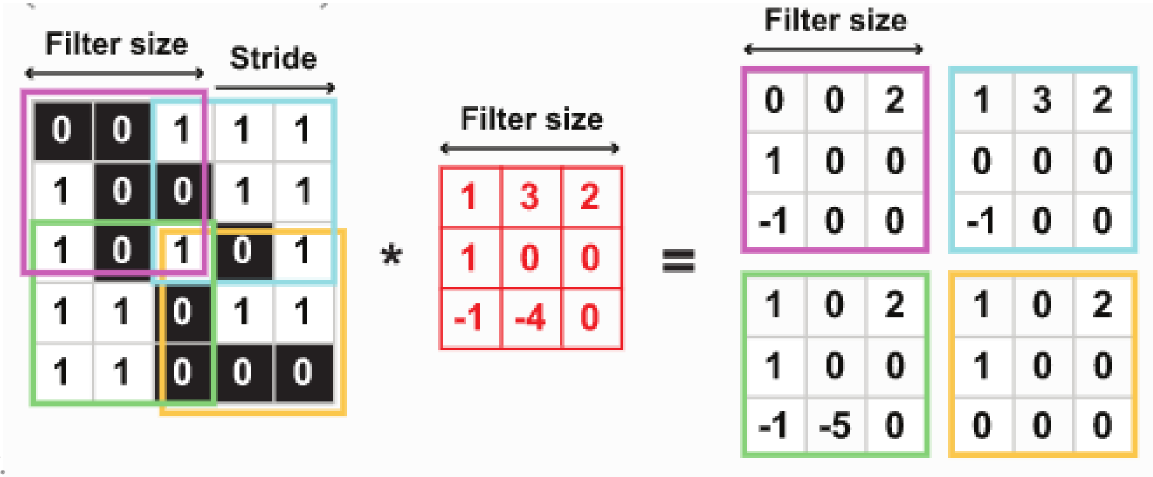
Convolutional layer

- The convolutional layer prepares the input data for the fully connected neurons (hidden layers)
- Procedures happening the convolutional layer:
  - Filtering
  - Pooling
  - Flatten & D



0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

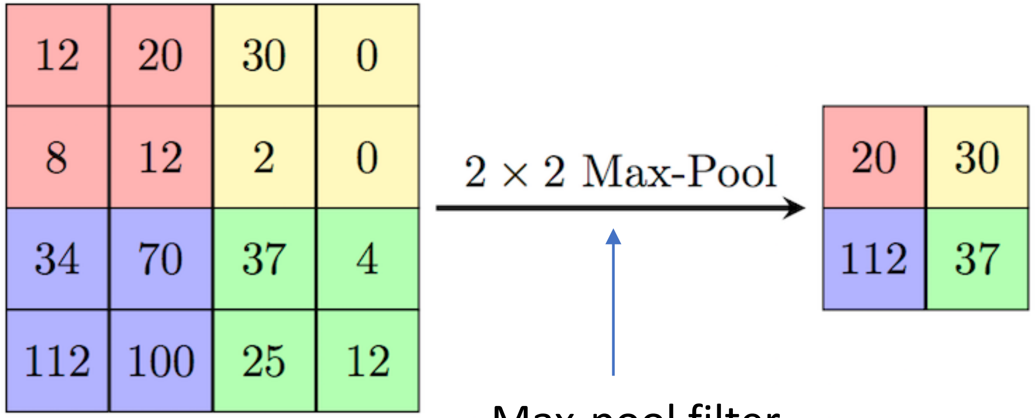
- Filters (kernel):
  - In charge to detect features
    - In images:
      - Can detect for example, lines, corners
      - Can also be specific: eyes, hair, etc.
  - Is a matrix of m * n. Size of the matrix depends on the application
    - Initialized randomly
    - Stride: how many “cells” the filter moves for a dot product
  - Operation:
    - A dot product operation is executed between the input data and the filter



$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 3 & 2 \\ 1 & 0 & 0 \\ -1 & -4 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 2 \\ 1 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 3 & 2 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 2 \\ 1 & 0 & 0 \\ -1 & -5 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 2 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- Pooling:
  - To reduce dimensionality
  - Eliminates noise from data

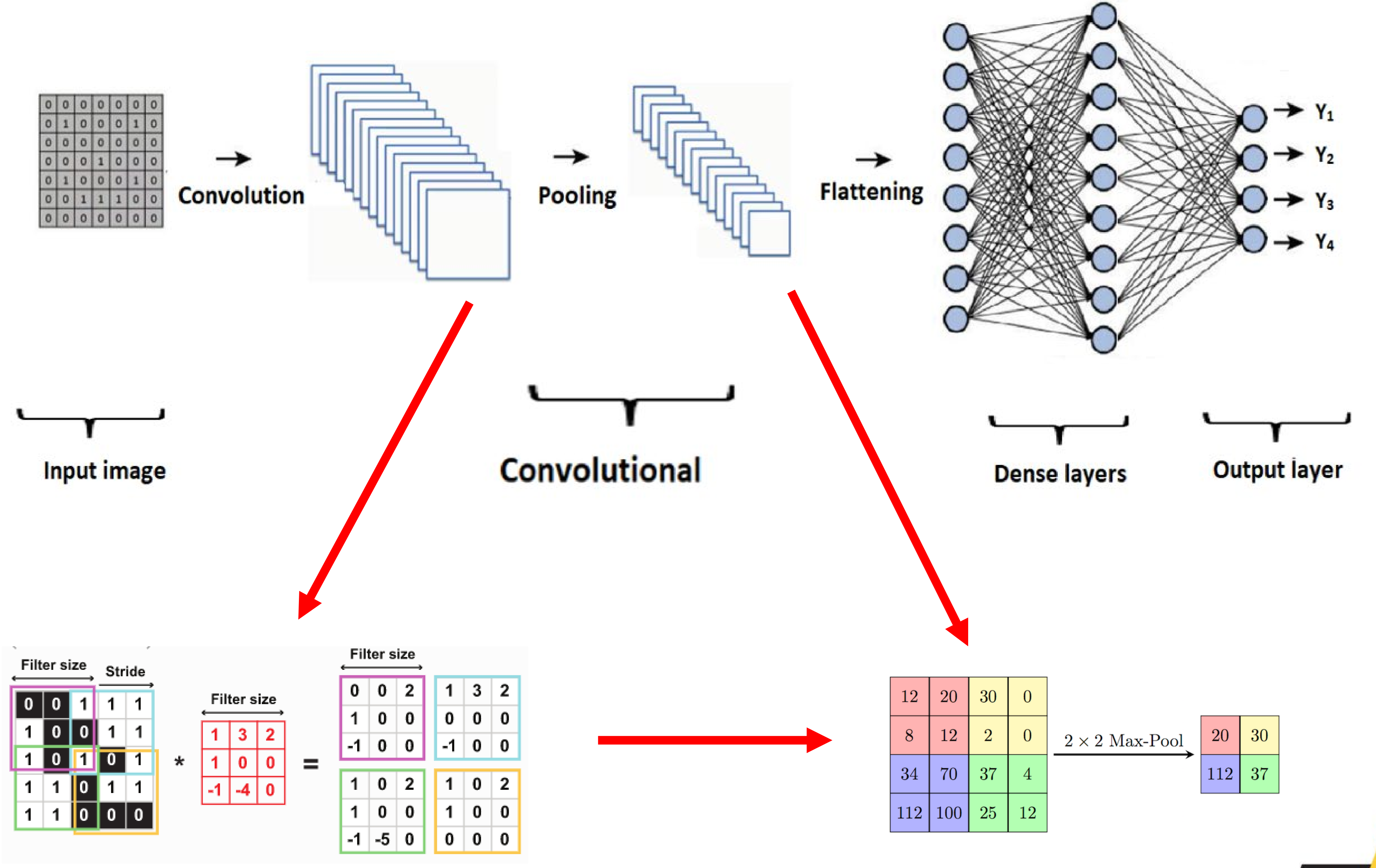
  - Max pooling: after the dot product of the filter and input is executed
    - A max pooling filter (of m*n) selects the max value and creates a new max pool.



12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

20	30
112	37

Max-pool filter



0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

0	0	1	1	1
1	0	0	1	1
1	0	1	0	1
1	1	0	1	1
1	1	0	0	0

1	3	2
1	0	0
-1	-4	0

0	0	2
1	0	0
-1	0	0

1	3	2
0	0	0
-1	0	0

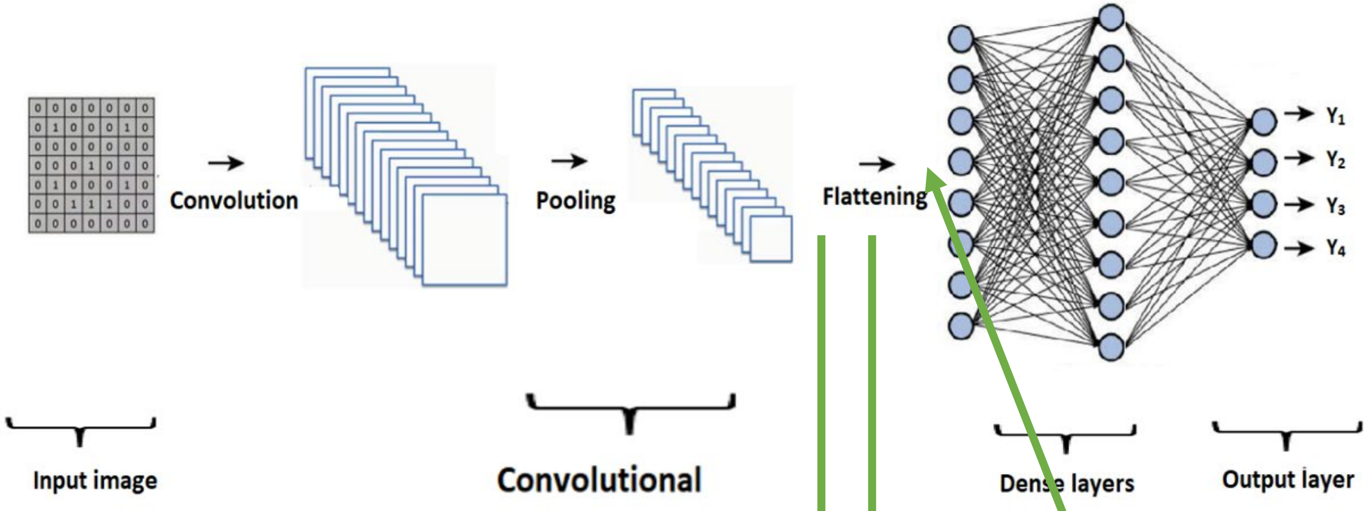
1	0	2
1	0	0
-1	-5	0

1	0	2
1	0	0
0	0	0

12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

20	30
112	37

- Flatten & Dense:
  - Flatten: For the neurons to perform their own operations, the input data needs to be an appropriate format:



0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

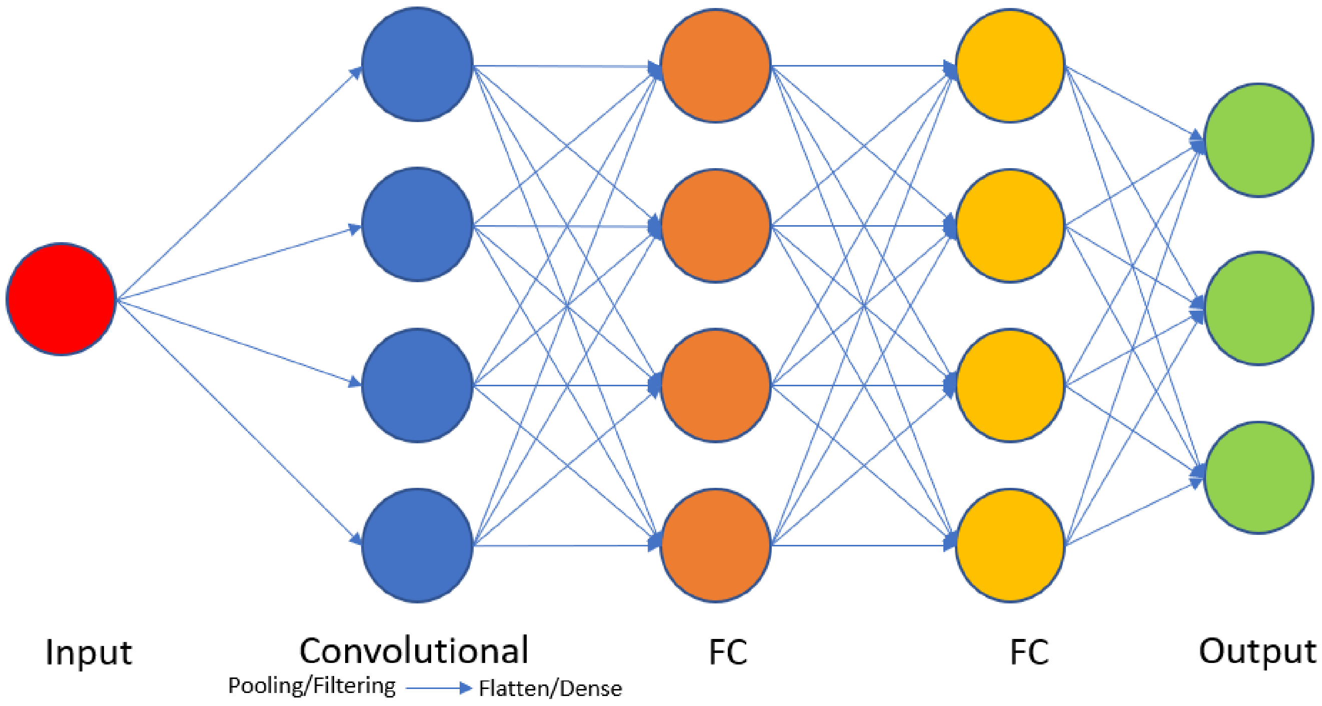
The output will be the classification
based on the features extracted
through the previous layers
and their different filters
(classification: outputs are
mapped to labels)

The vector (tensor) is passed as input of the FC

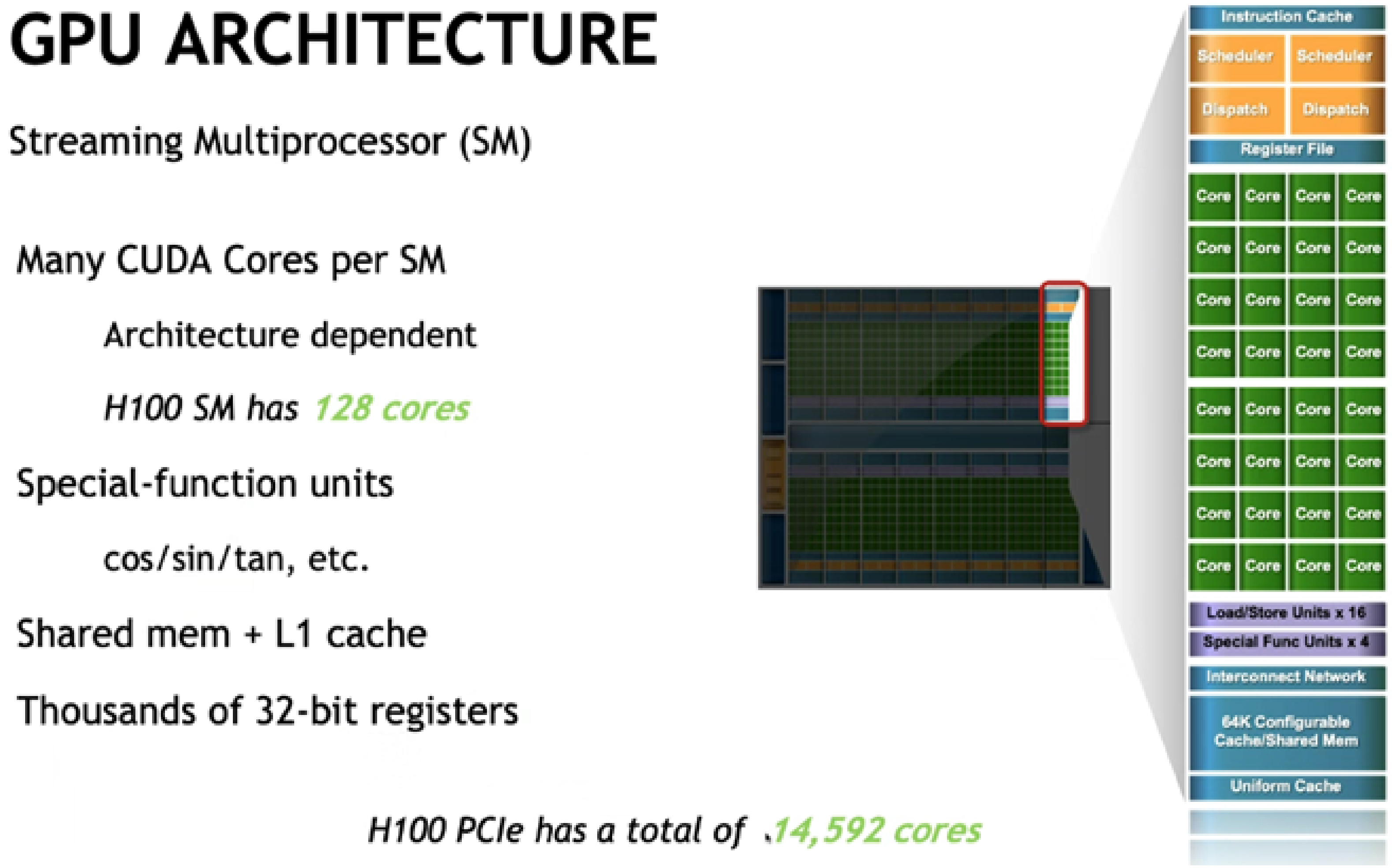
The output if the max-pool is a matrix
Can be of n-dimensions

Flattening: “Converts” n-dimension into a “readable” vector (tensor)

- Dense:
  - Method to create a fully connected neural network
  - Specifies the number of neurons per layer and activation method



- How is a CNN processed when GPUs are available?



GPU ARCHITECTURE

Streaming Multiprocessor (SM)

Many CUDA Cores per SM

Architecture dependent

H100 SM has 128 cores

Special-function units

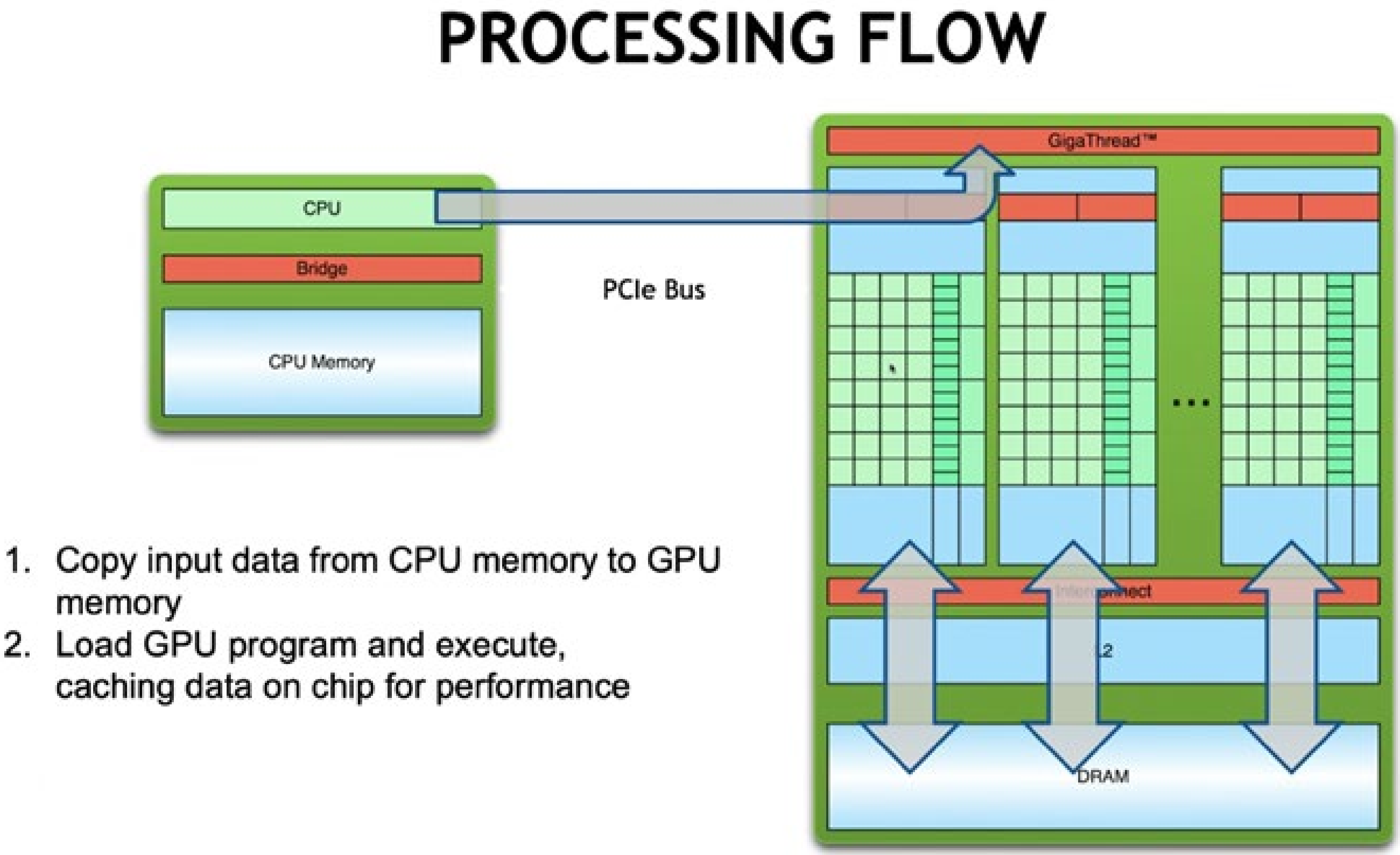
cos/sin/tan, etc.

Shared mem + L1 cache

Thousands of 32-bit registers

H100 PCIe has a total of .14,592 cores

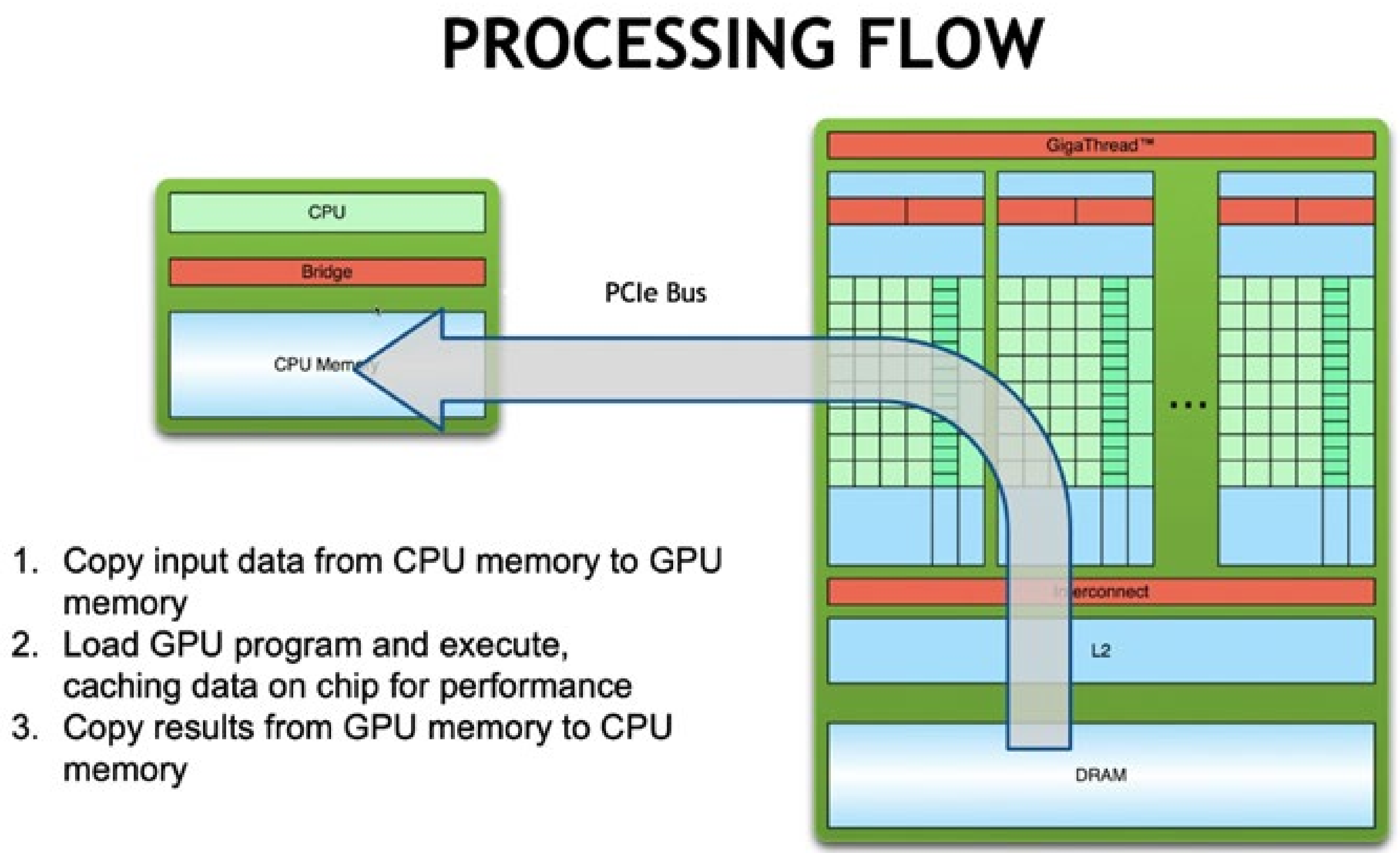
- How is a CNN processed when GPUs are available?



PROCESSING FLOW

1. Copy input data from CPU memory to GPU memory
2. Load GPU program and execute, caching data on chip for performance

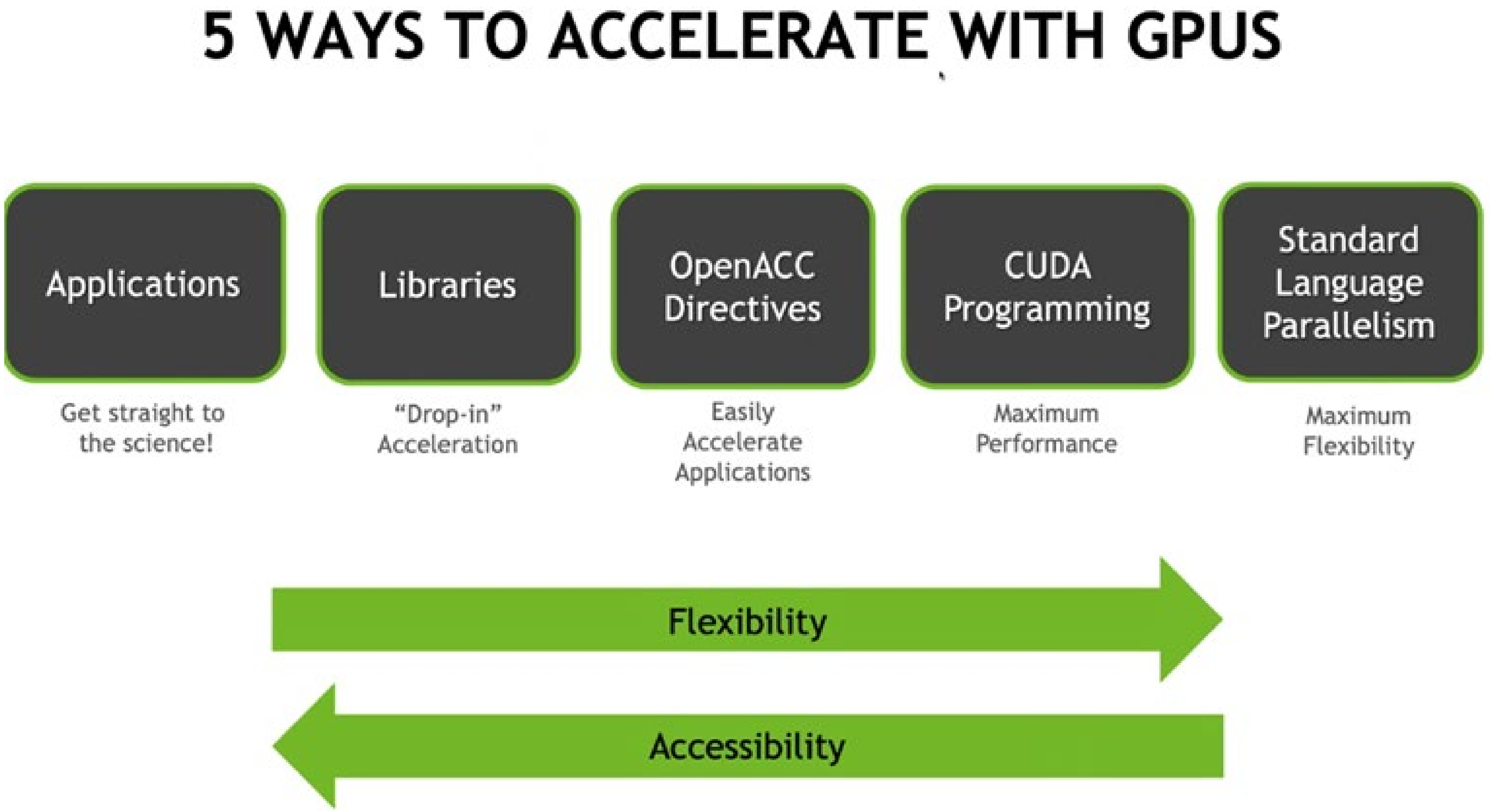
- How is a CNN processed when GPUs are available?



PROCESSING FLOW

1. Copy input data from CPU memory to GPU memory
2. Load GPU program and execute, caching data on chip for performance
3. Copy results from GPU memory to CPU memory

- How is a CNN processed when GPUs are available?



5 WAYS TO ACCELERATE WITH GPUS

Applications

Get straight to the science!

Libraries

"Drop-in" Acceleration

OpenACC Directives

Easily Accelerate Applications

CUDA Programming

Maximum Performance

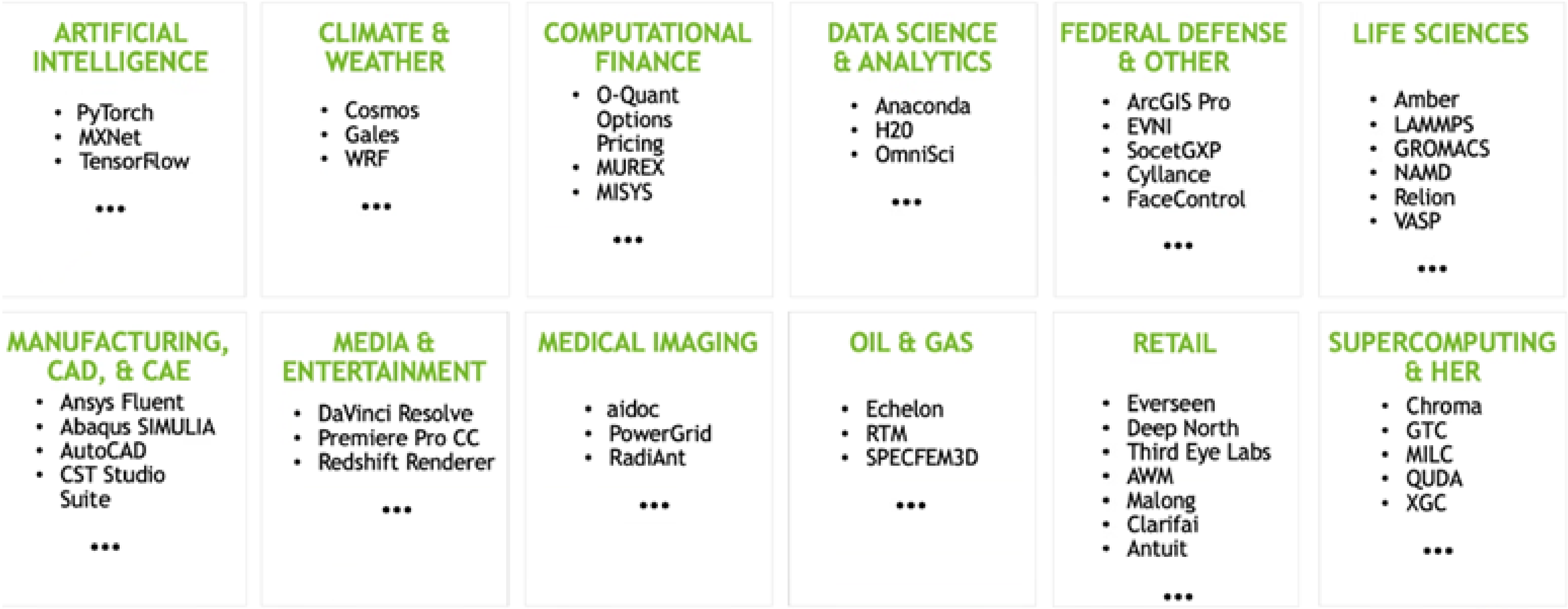
Standard Language Parallelism

Maximum Flexibility

Flexibility

Accessibility

- How is a CNN processed when GPUs are available?



ARTIFICIAL INTELLIGENCE

- PyTorch
- MXNet
- TensorFlow

...

CLIMATE & WEATHER

- Cosmos
- Gales
- WRF

...

COMPUTATIONAL FINANCE

- O-Quant Options Pricing
- MUREX
- MISYS

...

DATA SCIENCE & ANALYTICS

- Anaconda
- H20
- OmniSci

...

FEDERAL DEFENSE & OTHER

- ArcGIS Pro
- EVNI
- SocetGXP
- Cyllance
- FaceControl

...

LIFE SCIENCES

- Amber
- LAMMPS
- GROMACS
- NAMD
- Relion
- VASP

...

MANUFACTURING, CAD, & CAE

- Ansys Fluent
- Abaqus SIMULIA
- AutoCAD
- CST Studio Suite

...

MEDIA & ENTERTAINMENT

- DaVinci Resolve
- Premiere Pro CC
- Redshift Renderer

...

MEDICAL IMAGING

- aidoc
- PowerGrid
- RadiAnt

...

OIL & GAS

- Echelon
- RTM
- SPECFEM3D

...

RETAIL

- Everseen
- Deep North
- Third Eye Labs
- AWM
- Malong
- Clarifai
- Antuit

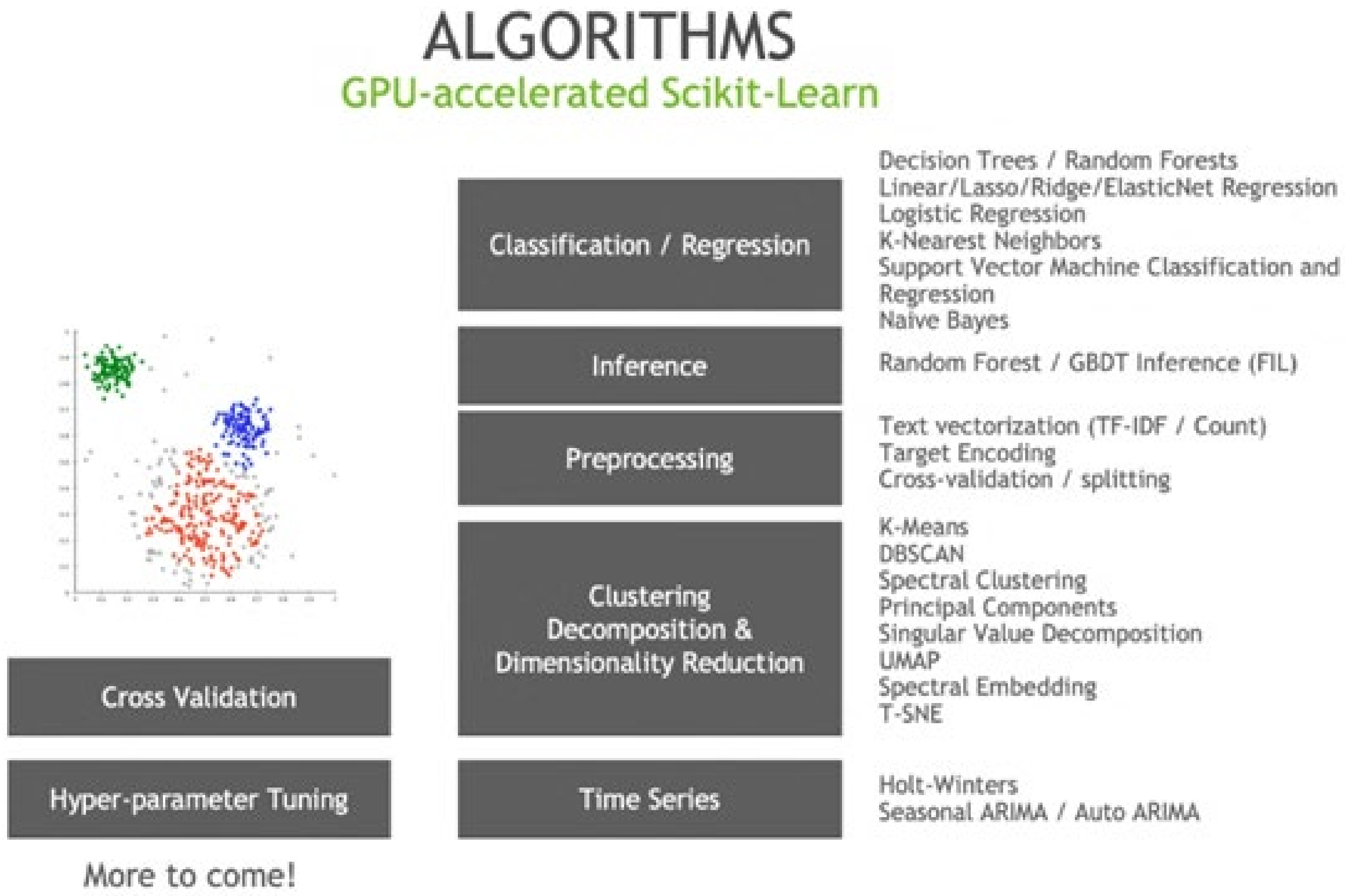
...

SUPERCOMPUTING & HER

- Chroma
- GTC
- MILC
- QUDA
- XGC

...

- How is a CNN processed when GPUs are available?



ALGORITHMS

GPU-accelerated Scikit-Learn

Classification / Regression

Decision Trees / Random Forests
Linear/Lasso/Ridge/ElasticNet Regression
Logistic Regression
K-Nearest Neighbors
Support Vector Machine Classification and Regression
Naive Bayes

Inference

Random Forest / GBDT Inference (FIL)

Preprocessing

Text vectorization (TF-IDF / Count)
Target Encoding
Cross-validation / splitting

Clustering Decomposition & Dimensionality Reduction

K-Means
DBSCAN
Spectral Clustering
Principal Components
Singular Value Decomposition
UMAP
Spectral Embedding
T-SNE

Time Series

Holt-Winters
Seasonal ARIMA / Auto ARIMA

Cross Validation

Hyper-parameter Tuning

More to come!

THE UNIVERSITY OF
SOUTHERN MISSISSIPPI®