Machine Learning on GPUs

Prof. Dr. Matthias S. Müller
Dr. Christian Terboven
Jannis Klinkenberg
Julian Miller

i12 High Performance Computing | RWTH AACHEN UNIVERSITY

What is This Chapter About?

- How to apply machine learning on GPUs
  - Overview
  - Supervised Learning
  - Unsupervised Learning
  - Reinforced Learning
  - Programming

Overview

What is Machine Learning?

- “Learning is any process by which a system improves performance from experience.” - Herbert Alexander Simon

- Traditional Program

- Machine Learning

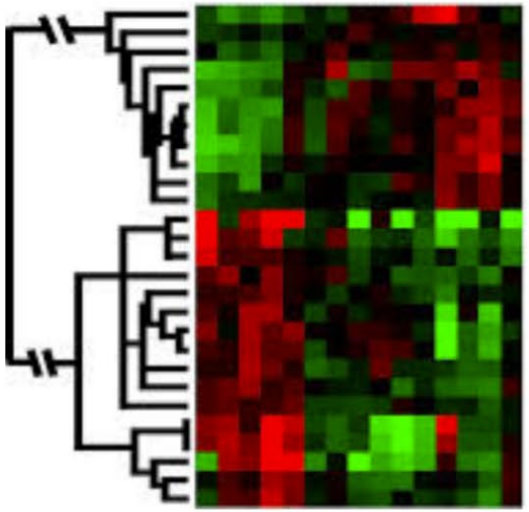
Herbert Alexander Simon
Turing Award 1975
Nobel Price in Economics 1978
Source: nobelprice.org

Based on a slide by Pedro Domingos

When to Use Machine Learning?

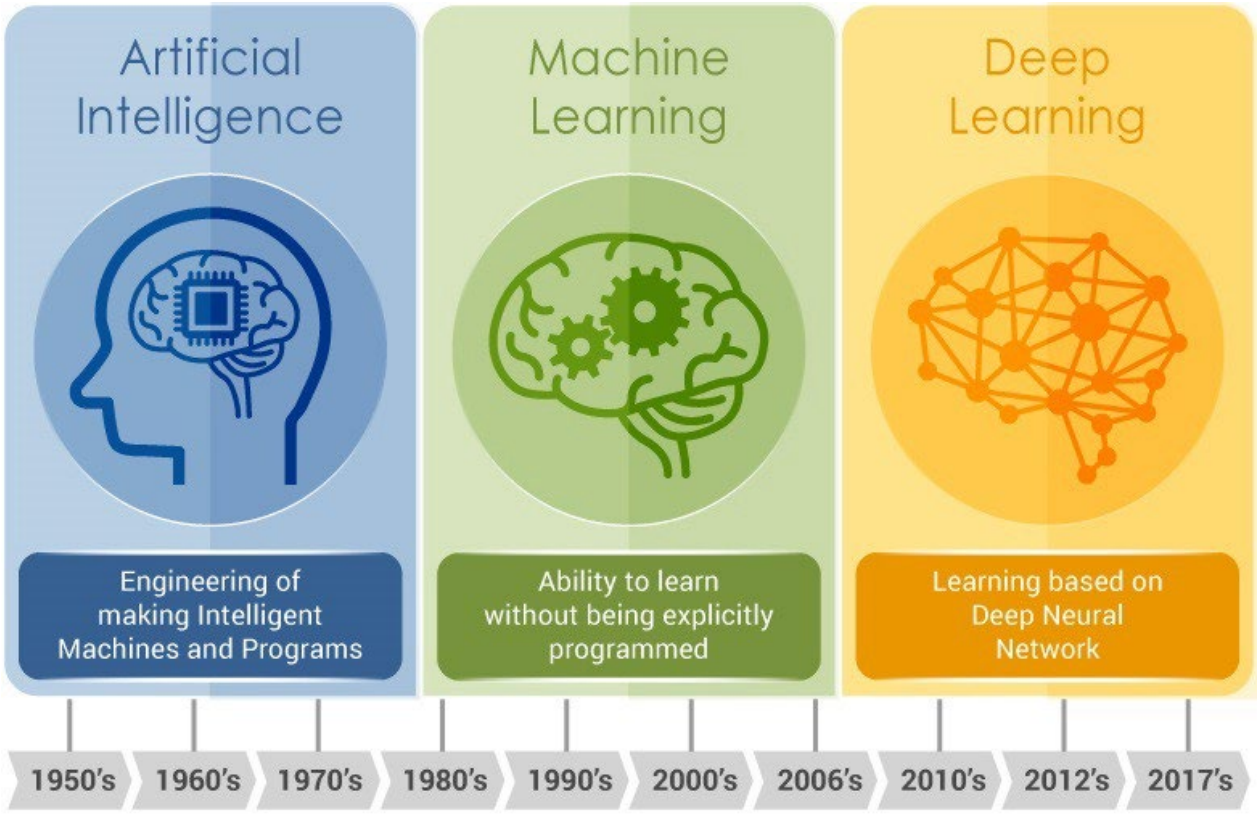
- When human expertise does not exist (e.g., navigating on Mars)
- When humans can’t explain their expertise (e.g., speech recognition)
- When models must be customized (e.g., personalized medicine)
- When models are based on huge amounts of data (e.g., genomics)



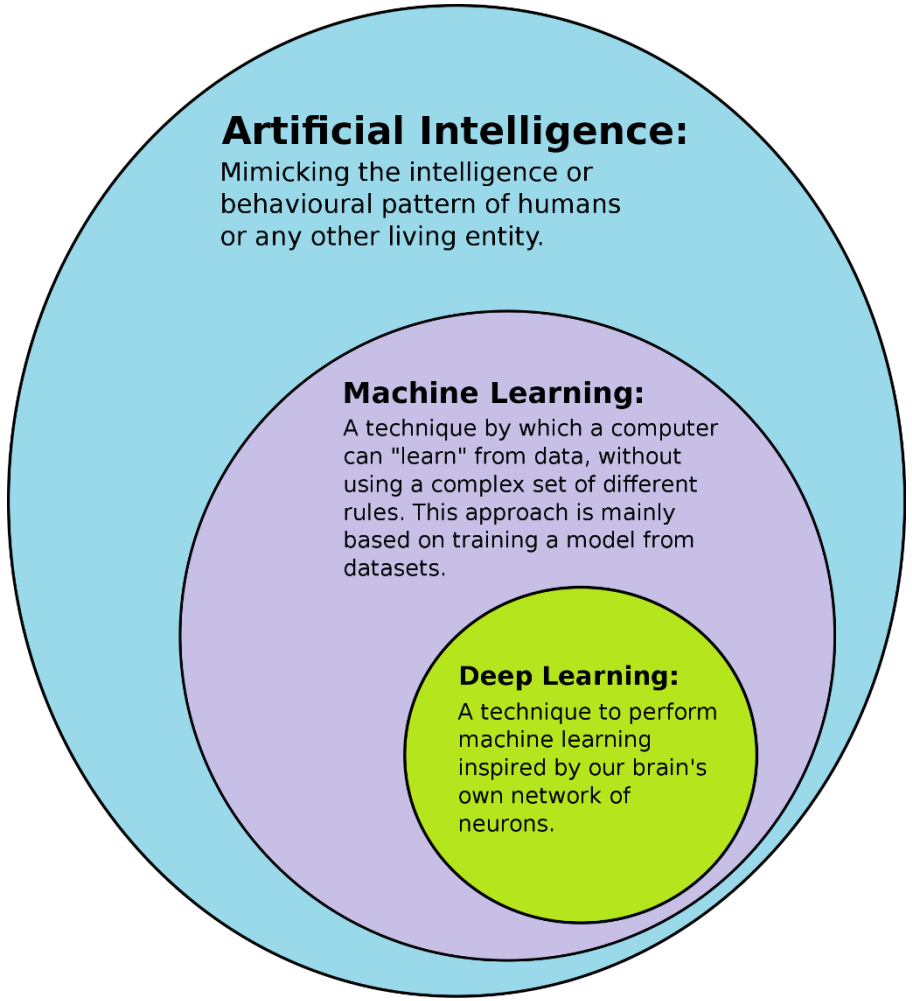



Based on a slide by Ethem Alpaydin

AI vs. ML vs. DL



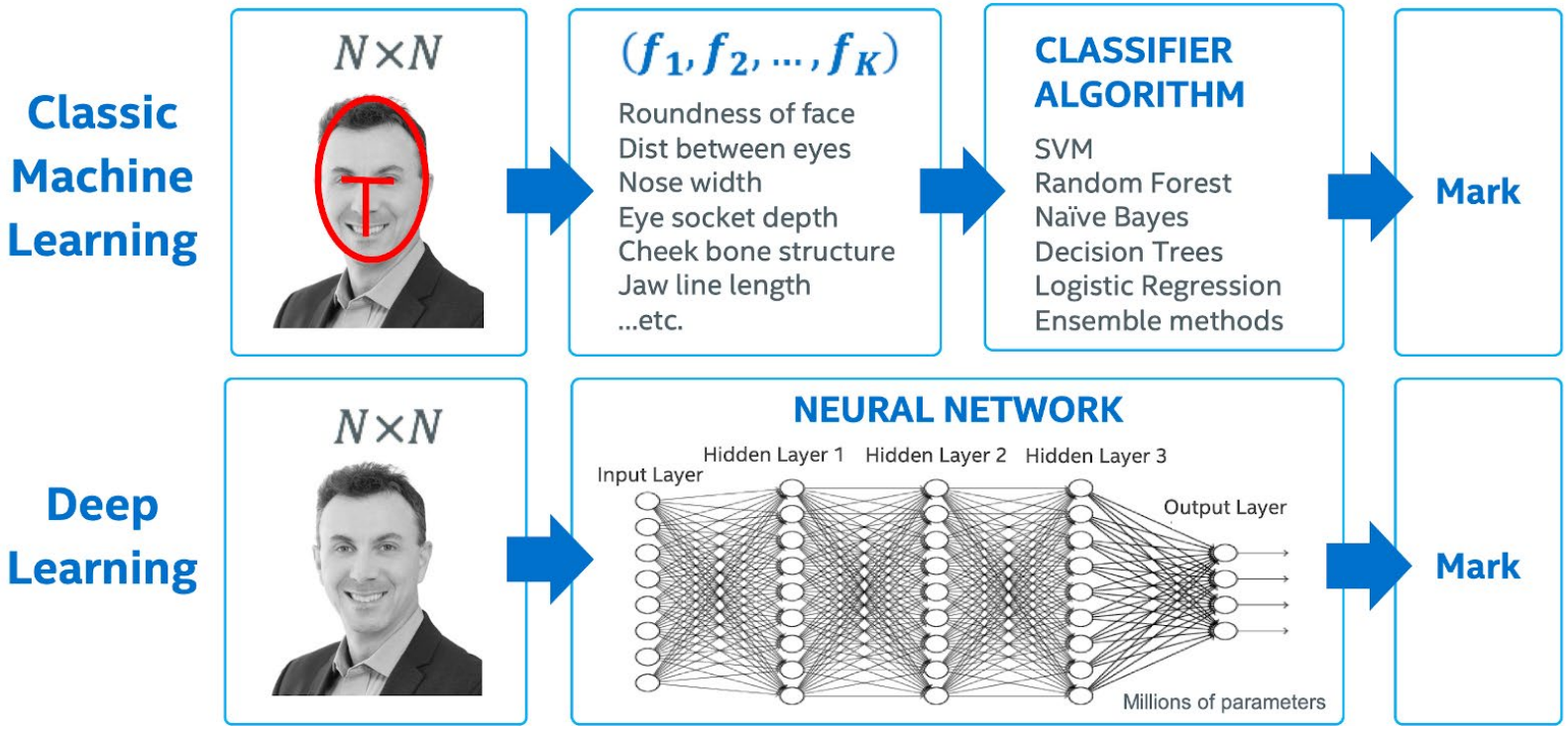
Source: Google



Source: Wikipedia

Deep Learning

- Datasets are increasing rapidly (more samples and higher dimension of features)
- Hardware gets more performant and cheaper
- New algorithms, typically openly accessible



Source: Intel

Supervised Learning

Supervised Learning

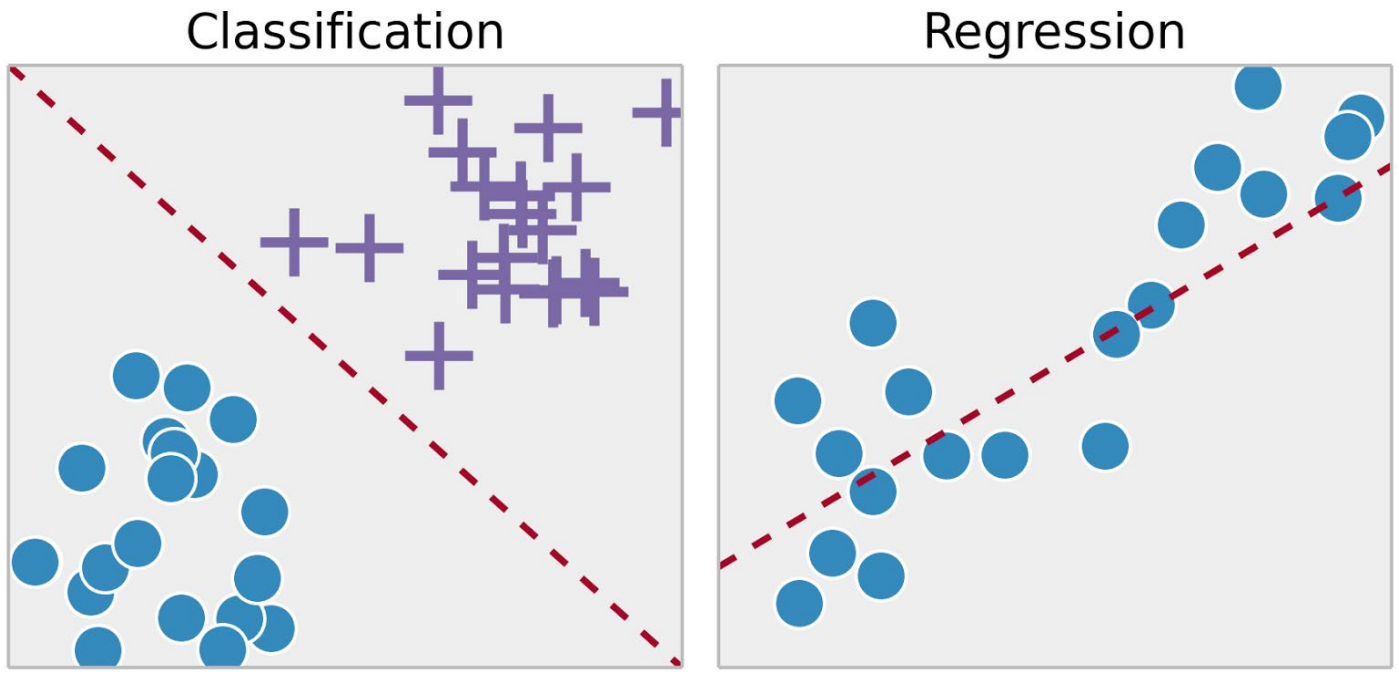
- Learn from examples for which the output is known (labeled data)
- Each example is a pair of an input object (typically a vector) and a desired output signal (supervisory signal)

- Typical steps
  1. Determine the type of training examples
  2. Gather a training set.
  3. Determine the input feature representation of the learned function.
  4. Determine the structure of the learned function and corresponding learning algorithm.
  5. Train the learning algorithm with data from the training set.
  6. Evaluate the accuracy of the learned function. (usually with a separate test set)

Supervised Learning Algorithms

- Different types of Supervised Learning

- Examples:
  - Support-vector machines
  - Linear regression
  - Logistic regression (Classification!!)
  - Naive Bayes
  - Linear discriminant analysis
  - Decision trees
  - K-nearest neighbor algorithm
  - Neural networks (e.g. Multilayer perceptron)
  - Similarity learning
  - …



Source: Devin Soni

MNIST Example

- Large database of handwritten digits
- Often used for training and testing
- 60,000 training and 10,000 testing images

- Accuracy: Correctly recognized digits

Sample images from MNIST test dataset
Source: wikipedia.org

MNIST Example: Accuracy Over Time

Type of classifier	Publication	Error rate (%)
Linear classifier	LeCun et. al, IEEE 1998	7.6
Non-Linear Classifier	LeCun et. al, IEEE 1998	3.3
Boosted Stumps	Kégl et. al, ICML 2009	0.87
Support vector machines	DeCoste & Schölkopf, MLJ 2002	0.56
K-Nearest Neighbors	Keysers et. al, IEEE PAMI 2007	0.52
Neural network	Ciresan et. al, Neural Comput 2010	0.35
Convolutonal neural network	Ciresan et. al, CVPR 2012	0.23
Random Multimodel Deep Learning	Kawsari et. Al, ACM ICISDM 2018	0.18

Unsupervised Learning

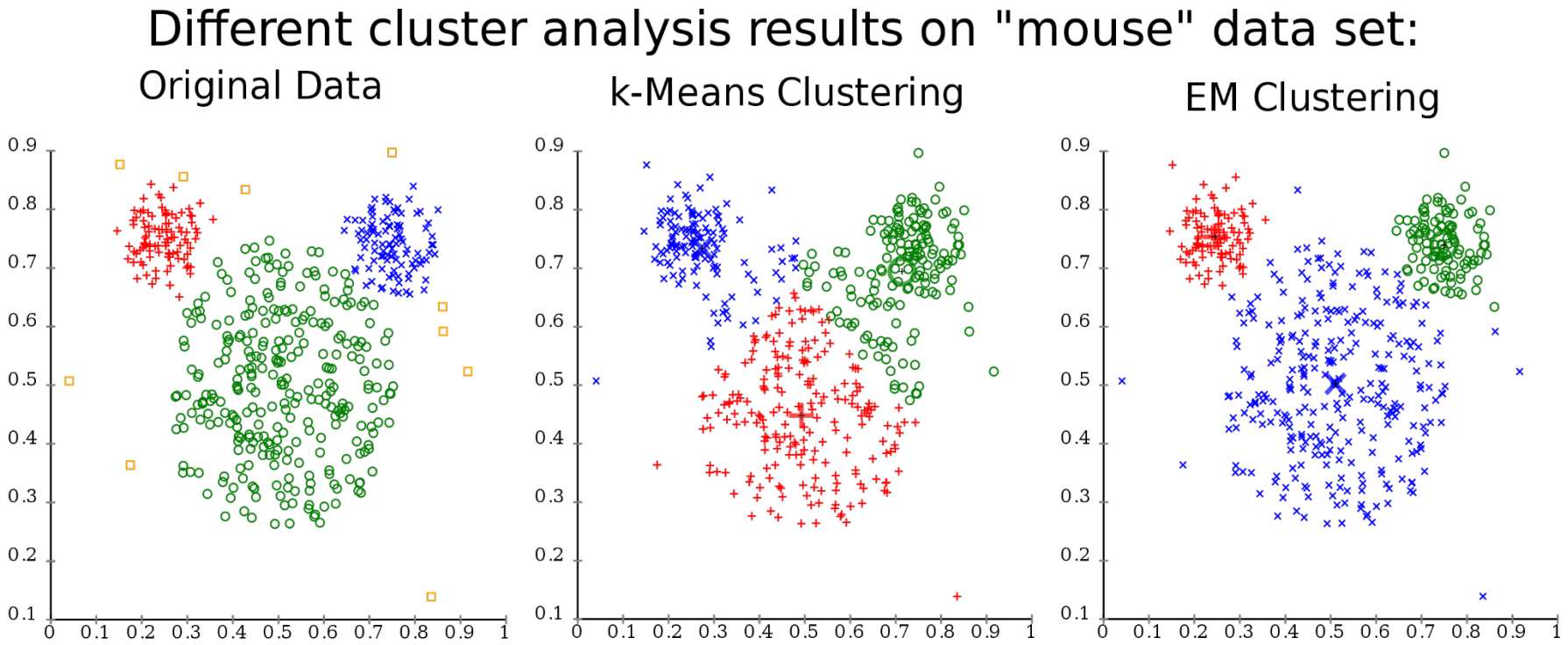
Unsupervised Learning

- Learn from examples from which the output is unknown
- Machine builds its own internal representation of its world
- Accuracy typically unknown since there are no labels

- Typical application areas
  - Exploratory analysis: identify structure in data
  - Dimensionality reduction: transform data from high to low dimensionality while retaining (most) meaningful properties
    - Feature selection
    - Feature projection (e.g., Principal component analysis)

Unsupervised Learning Algorithms

- Clustering methods
  - hierarchical clustering
  - k-means
  - mixture models
  - DBSCAN
  - OPTICS algorithm
- Anomaly detection
  - Local Outlier Factor
  - Isolation Forest
- Latent variable models
  - Expectation–maximization algorithm
  - Principal component analysis
- …



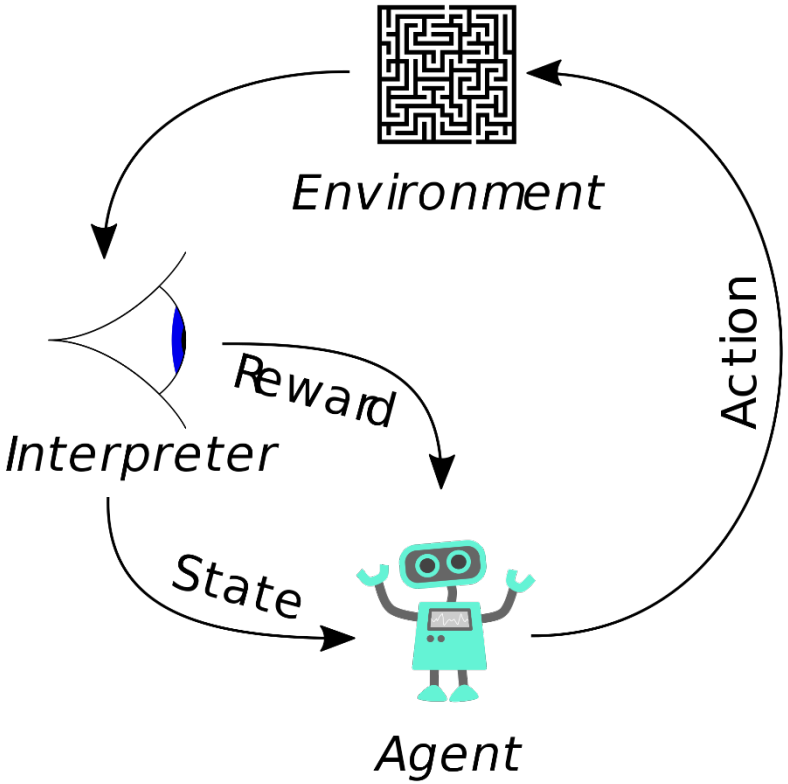
Source: wikipedia.org

Reinforced Learning

Reinforced Learning

- Agent learns through maximizing rewards of trial-and-error actions
- Labelled data not required but reward function
- Environment is typically a Markov decision process
- Programming typically via dynamic programming methods

- Balance between exploration and exploitation of knowledge



Source: wikipedia.org

Reinforced Learning Algorithms

- Discrete action space
  - Monte Carlo
  - Q-learning: State–action–reward–state
  - SARSA: State–action–reward–state–action
  - Combination with eligibility traces
  - …
- Continuous action space
  - Deep Deterministic Policy Gradient
  - Asynchronous Advantage Actor-Critic Algorithm
  - …

Initialized

Q-Table		Actions					
		South (0)	North (1)	East (2)	West (3)	Pickup (4)	Dropoff (5)
States	0	0	0	0	0	0	0
	. . .	. . .	. . .	. . .	. . .	. . .	. . .
	327	0	0	0	0	0	0
	. . .	. . .	. . .	. . .	. . .	. . .	. . .
	499	0	0	0	0	0	0

Training

Q-Table		Actions					
		South (0)	North (1)	East (2)	West (3)	Pickup (4)	Dropoff (5)
States	0	0	0	0	0	0	0
	. . .	. . .	. . .	. . .	. . .	. . .	. . .
	328	-2.30108105	-1.97092096	-2.30357004	-2.20591839	-10.3607344	-8.5583017
	. . .	. . .	. . .	. . .	. . .	. . .	. . .
	499	9.96984239	4.02706992	12.96022777	29	3.32877873	3.38230603

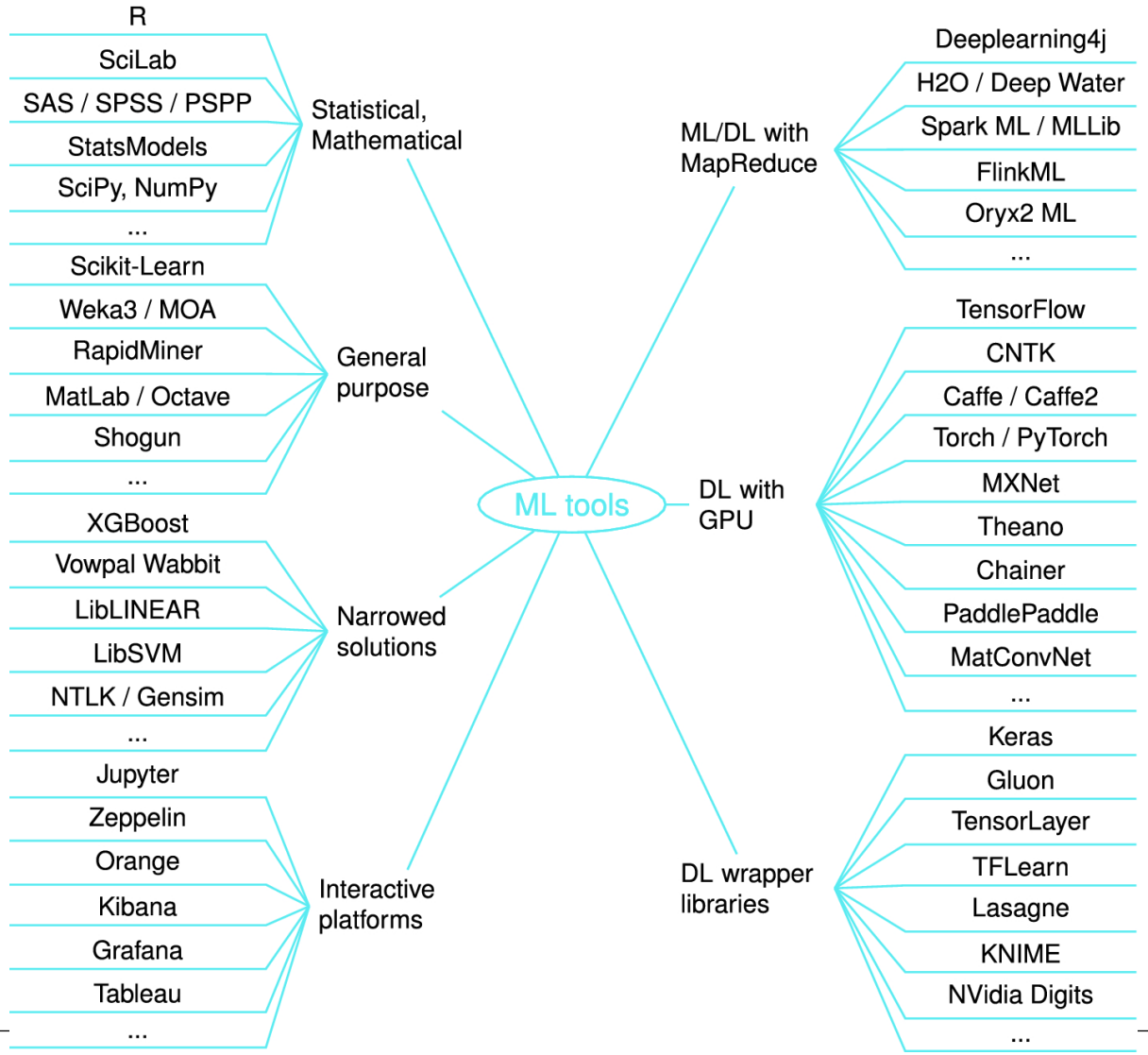
Q-Learning table of states by actions that is initialized to zero, then each cell is updated through training.
Source: wikipedia.org

Programming

Machine Learning Tools

- Typically regular/structured programs
- Limited set of computational and algorithmic patterns
- Often implemented in high-level libraries
- Interfaces in many programming models

Machine Learning Tools



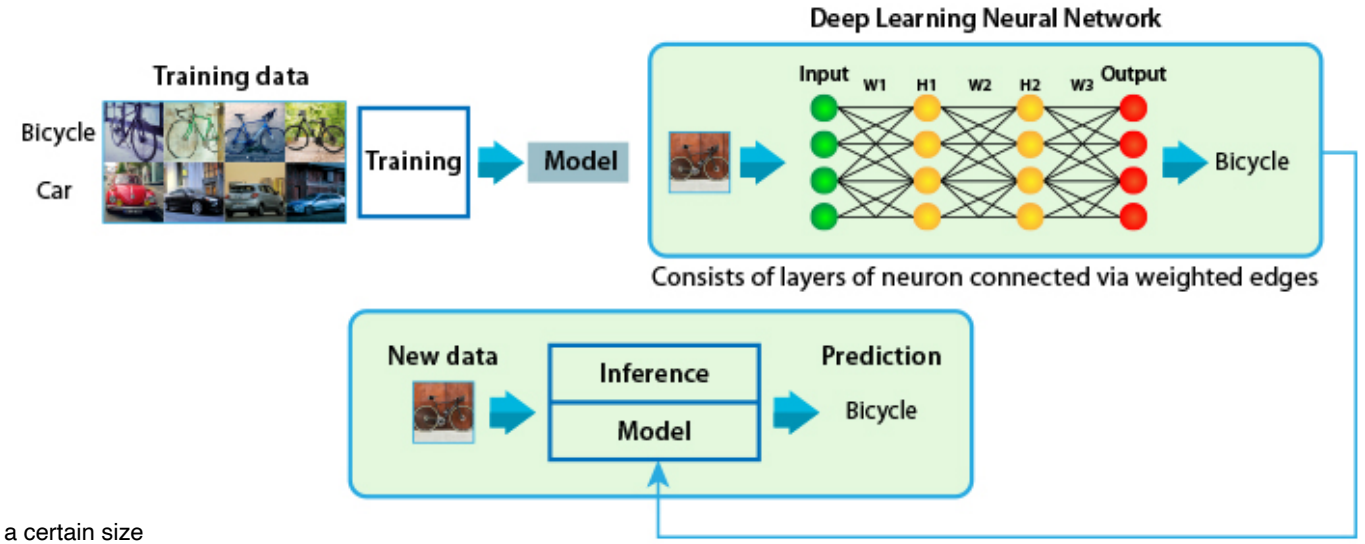
Source: Nguyen et. al, Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey, Artificial Intelligence Review, 2019

Machine Learning on GPUs

- Many ML algorithms are highly data-parallel or even embarrassingly parallel

- Machine vision
  - Lots of matrix computations
  - GPU is designed for high-throughput image manipulations

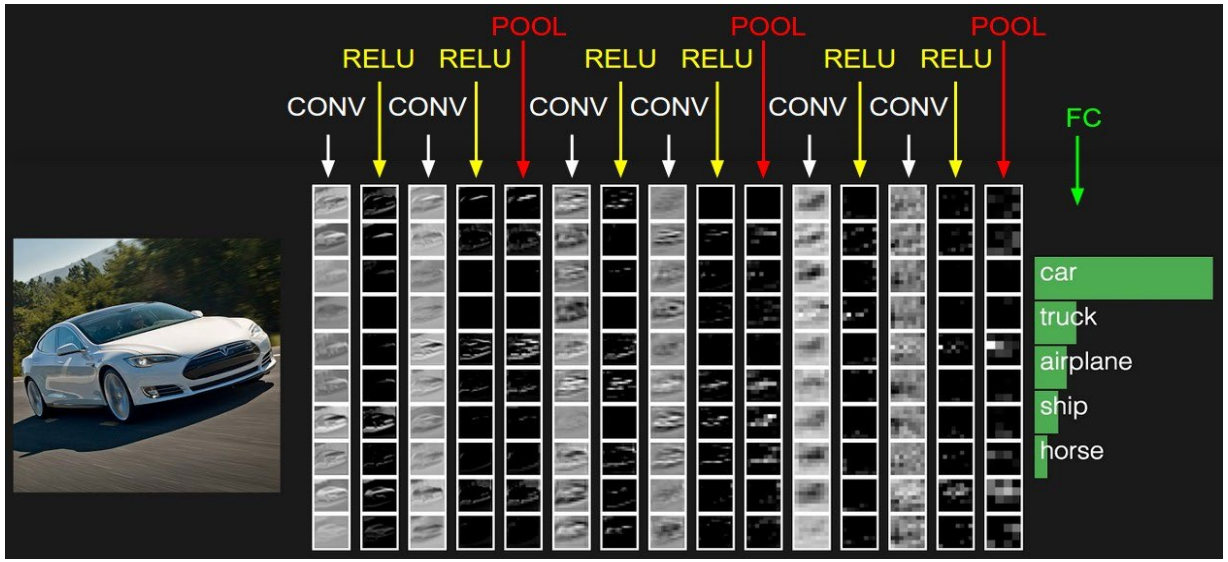
- Neural networks
  - Lots of convolutions for multiplying the input data by the weights
  - Modern GPUs utilize specialized tensor Cores to speedup convolutions further
  - Lower & mixed precision is typically supported which further speeds up the computation

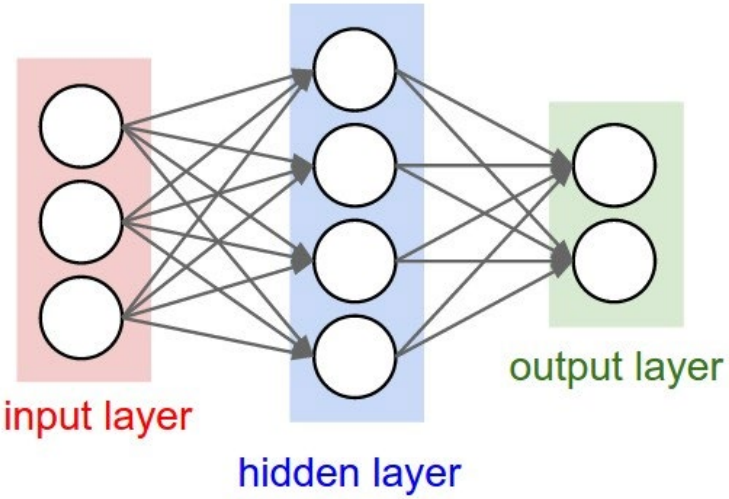


GPU helps convolutional network such that the input can be processed in a certain size
Position of the input data is not important thus it is sufficient to use lower and mixed precision data to imporve the porformance

Deep Learning on GPUs

- Limited set of often used kernels: convolutions, rectifiers, pooling, fully connected layers, etc.
- Typically implemented in low-level libraries: oneDNN (Intel), cuDNN (NVIDIA), etc.
- High-level interfaces to specify networks: PyTorch, TensorFlow, etc.



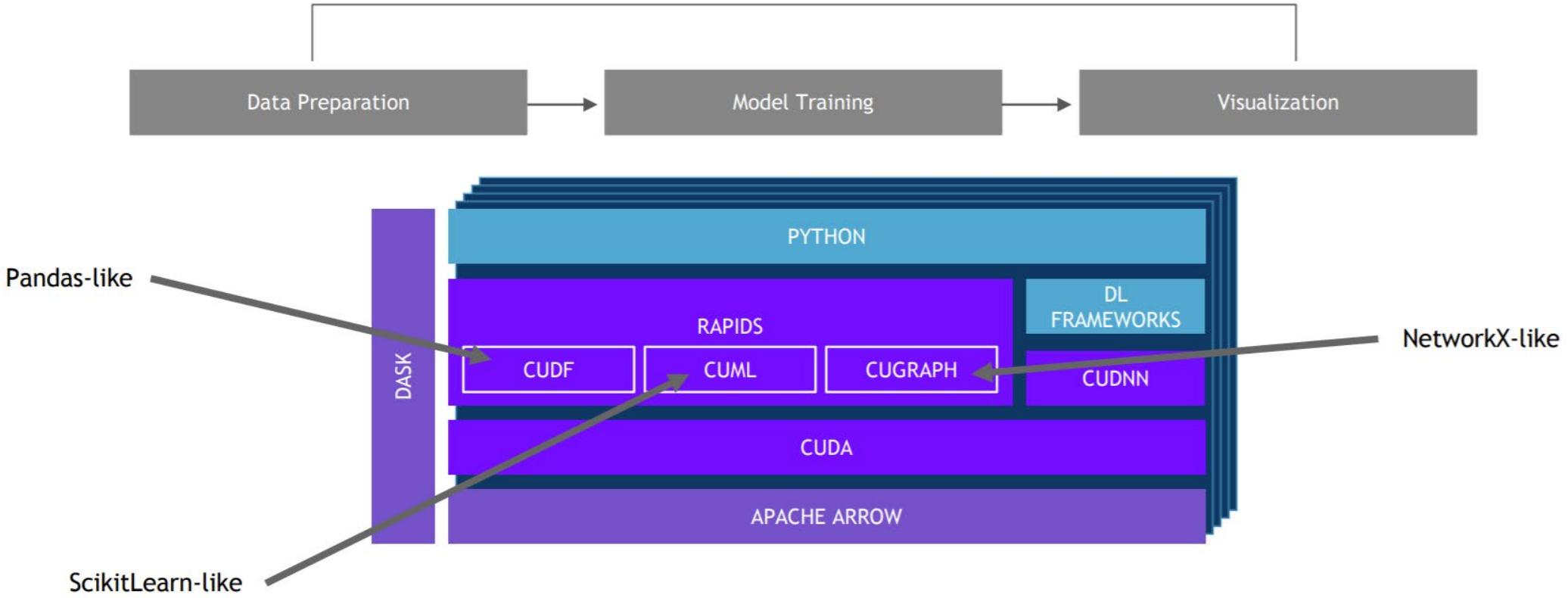


those conv, pool, Relu etc are computed on GPU and GPU enable to apply high level interfaces

Images courtesy of CS Stanford University course cs231n

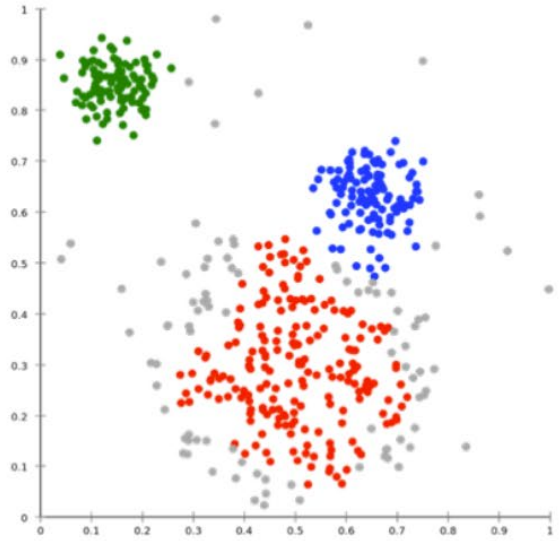
RAPIDS Framework for NVIDIA GPUs

- High-level framework with similar interface than Pandas and ScikitLearn for GPUs



Source: rapids.ai

RAPIDS: Supported ML Algorithms



Cross Validation

Hyper-parameter Tuning

Classification / Regression

Decision Trees / Random Forests
Linear Regression
Logistic Regression
K-Nearest Neighbors
Kalman Filtering
Bayesian Inference
Gaussian Mixture Models
Hidden Markov Models

Statistical Inference

Clustering

K-Means
DBSCAN
Spectral Clustering

Decomposition & Dimensionality Reduction

Principal Components
Singular Value Decomposition
UMAP
Spectral Embedding

Timeseries Forecasting

ARIMA
Holt-Winters

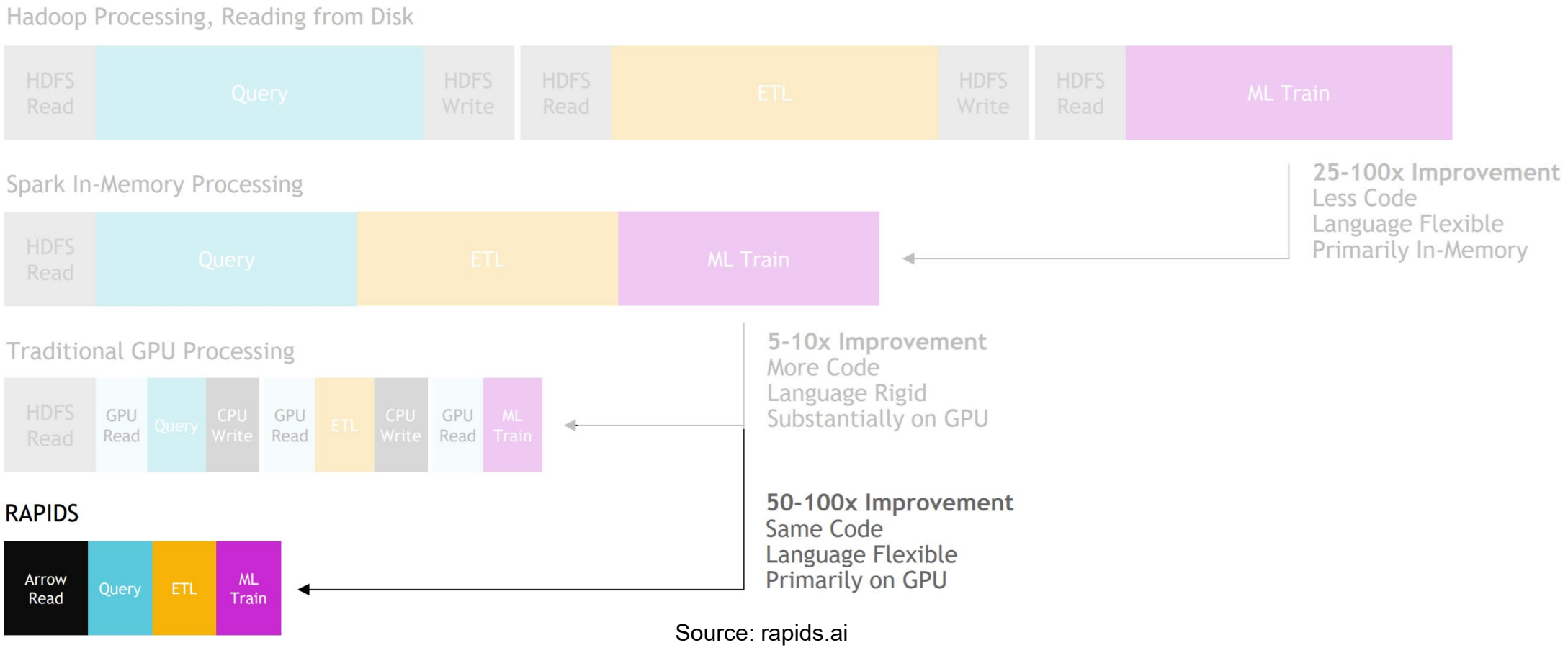
Recommendations

Implicit Matrix Factorization

Source: rapids.ai

RAPIDS: Data Management

- Keep all ML operations on GPU to minimize CPU-GPU data transfers



Source: rapids.ai

Machine Learning on GPUs Summary

- Machine learning becomes more ubiquitous
  - Access to larger datasets, more advanced hardware, and methods/algorithms

- ML algorithmic classes
  - Supervised learning
  - Unsupervised learning
  - Reinforced learning

- Structured programming in ML
  - Many regular and highly parallel problems
  - Many frameworks and interfaces available
  - Neural networks and machine vision are especially efficient on GPUs