CS-541
Wireless Sensor Networks

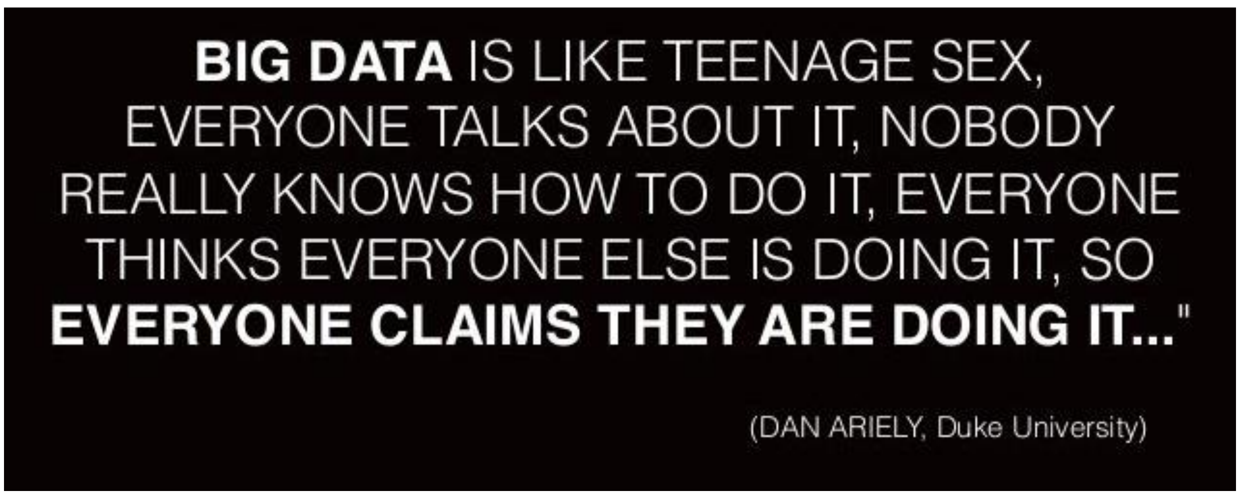
Lecture 14: Big Sensor Data

Spring Semester 2017-2018

Prof Panagiotis Tsakalides, Dr Athanasia Panousopoulou, Dr Gregory Tsagkatakis

Overview

- Big Data
- Big Sensor Data



Material adapted from: Recent Advances in Distributed Machine Learning Tie-Yan Liu, Wei Chen, Taifeng Wang Microsoft Research, AAAI 2017 Tutorial

Computing trends

Big Data → Big Computations → Big Models

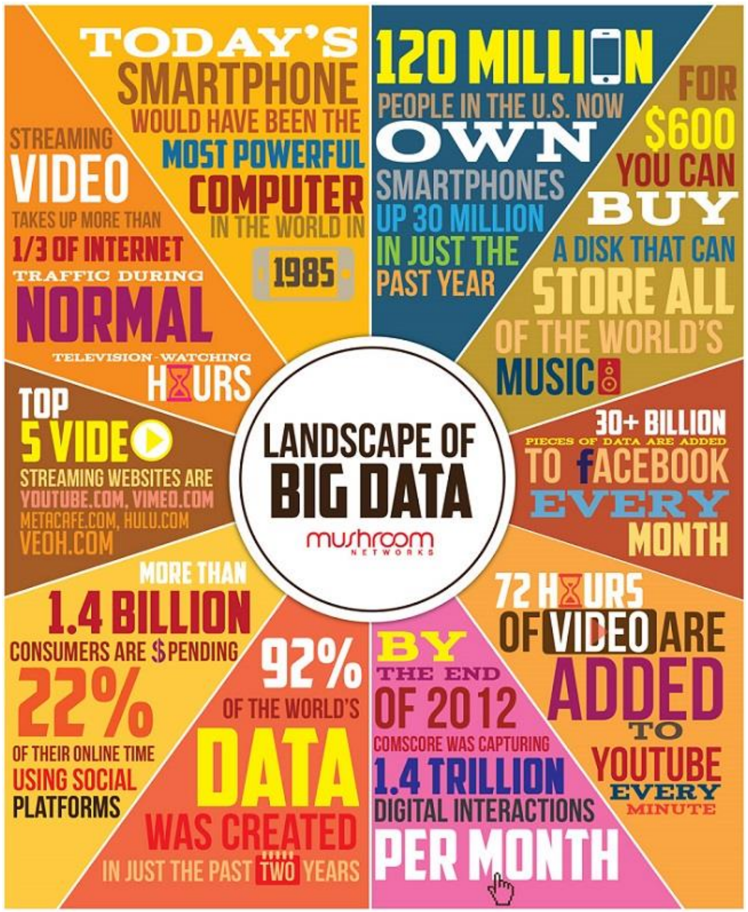
Big Data
- Signals, Information & Knowledge
- Digital Representation of the World

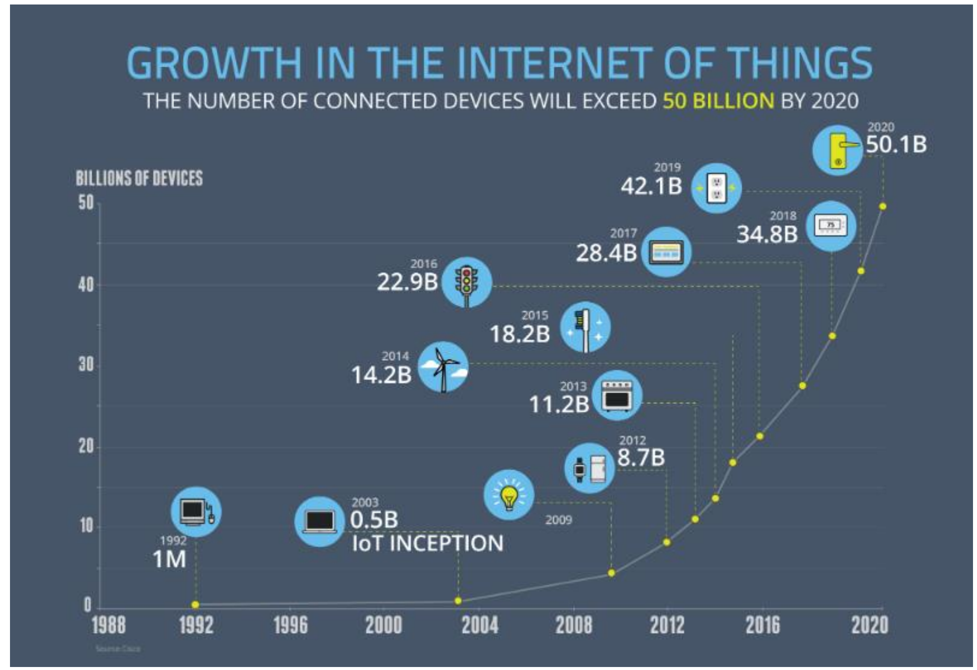
Big Computations
- Cloud Computing
- Internet of Things
- CPU/GPU/FPGA

Big Models
- Statistical machine Learning (e.g., DNN)
- >2^9 parameters

n/g AI
- Digital Life/Work
- New Form of HCI
- Reinvent Productivity & Business Process
- Personal Agent

Big Data & WSNs (IoT)

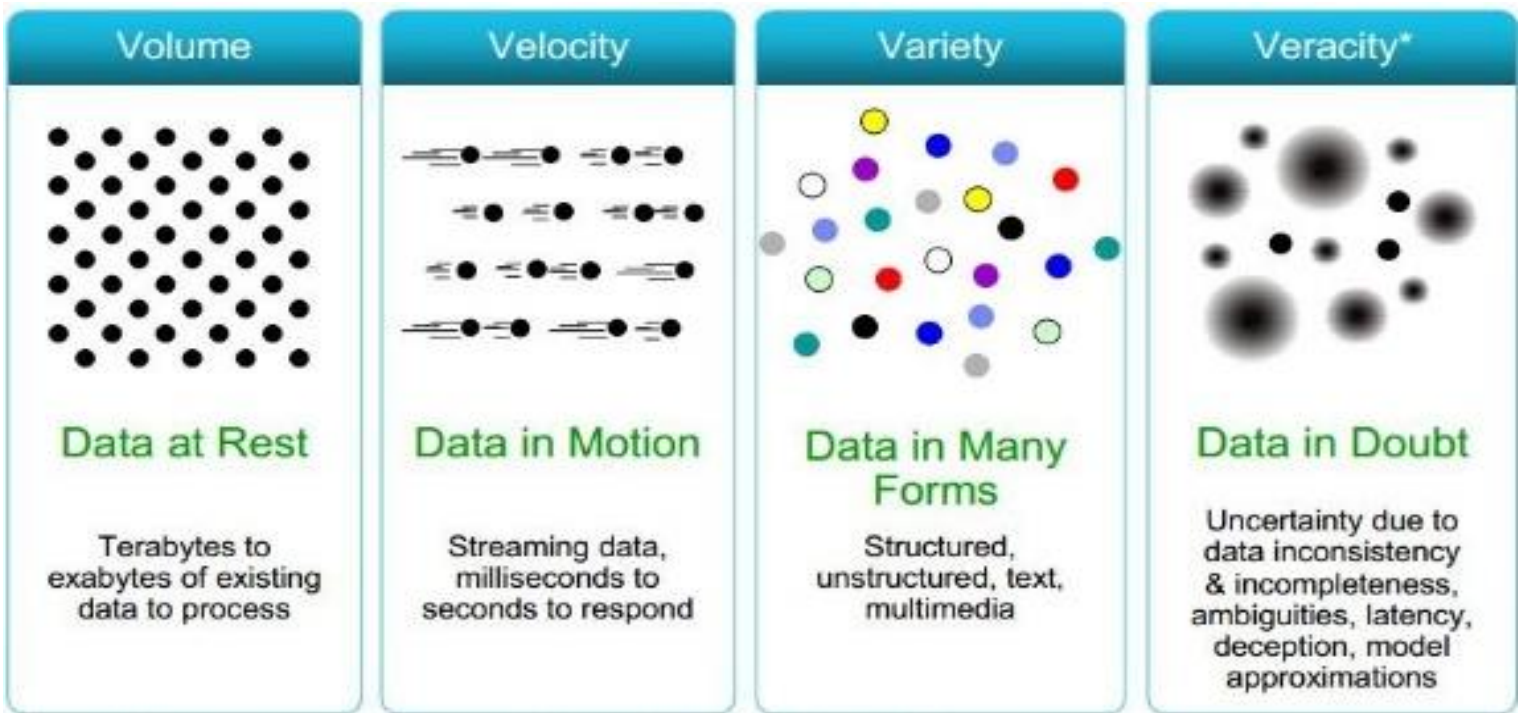




Big Data forms

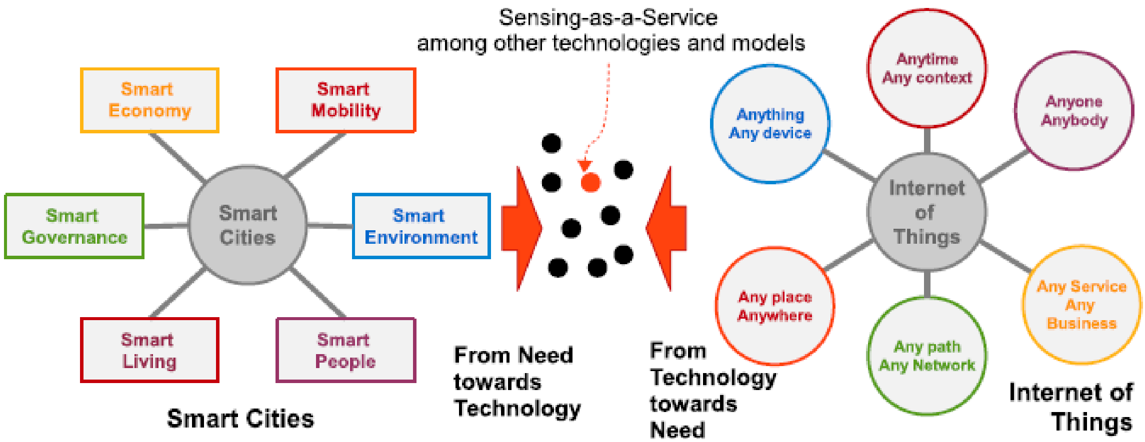
- "Big" data arises in many forms:
  - Physical Measurements: from science (physics, astronomy)
  - Medical data: genetic sequences, detailed time series
  - Activity data: GPS location, social network activity
  - Business data: customer behavior tracking at fine detail
- Common themes:
  - Data is large, and growing
  - There are important patterns and trends in the data
  - We don't fully know where to look or how to find them

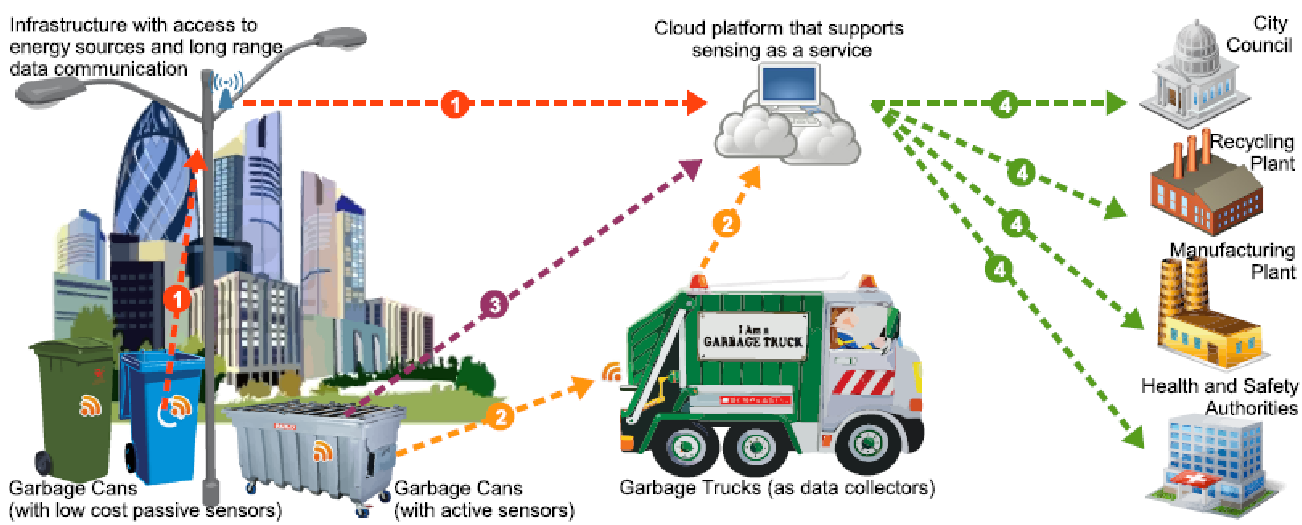
Big Data: The 4+1Vs



VALUE

Big Data in WSN: smart cities





Big Data in WSN: wearables

Sensor	Function
Blood-pressure sensor	Measures human blood pressure
Camera pill	Measures gastrointestinal tracts
Carbon dioxide sensor	Measures carbon dioxide gas
ECG/EEG/EMG sensor	Measures the electrical and muscular functions of the heart
Humidity sensor	Measures humidity changes
Blood oxygen saturation	Measures blood oxygen saturation
Pressure sensor	Measures pressure value
Respiration sensor	Measures human respiration values
Temperature sensor	Measures human body temperature

Table 2 Commonly Used Sensors in Body Area Networks or Body Sensor Networks

Big Sensor Data

Table 1 Common Sensors Integrated in Smartphones and Tablets

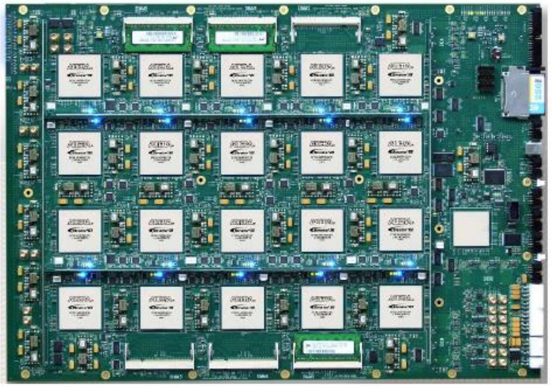
Sensors on Smartphones	Function
Microphone	The real-world sound and vibration are converted to digital audio
Camera	Senses visible light or electromagnetic radiation and converts them to digital image or video
Gyroscope	Provides orientation information
Accelerometer	Measures the linear acceleration
Compass or magnetometer	Works as a traditional compass. Provides orientation in relation to the magnetic field of Earth
Proximity sensor	Finds proximity of the phone from the user
Ambient light sensor	Optimizes the display brightness
GPS	Global Positioning System, tracks the target location or "navigates" the things by map with the help of GPS satellites
Barometer	Measures atmospheric pressure
Fingerprint sensor	Captures the digital image of fingerprint pattern

Big Computations

- Large computer clusters and highly parallel computational architectures

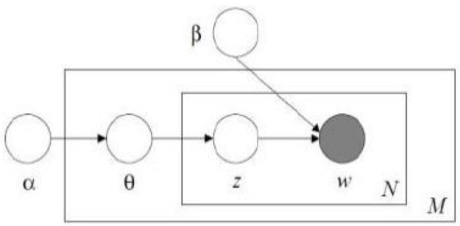
Cloud Computing

GPU Cluster

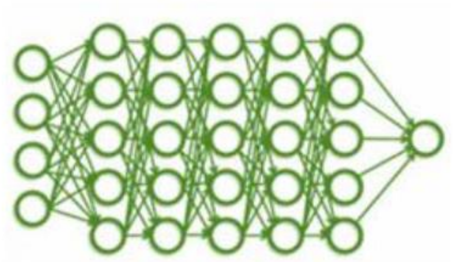


FPGA Farm

Big Models



LightLDA: LDA with 10^6 topics (10^{11} parameters); More topics → better performance in ad selection and click predictions



DistBelief: DNN with 10^{10} weights; Deeper and larger networks → better performance in image classification.

Human brain: 10^{11} neurons and 10^{15} connections, much larger than any existing ML model.

Machine learning & Big Data

Supervised

- Support Vector Machines
- Logistic Regression

- Convolutional Neural Net
- Recurrent Neural Net

Shallow **Deep**

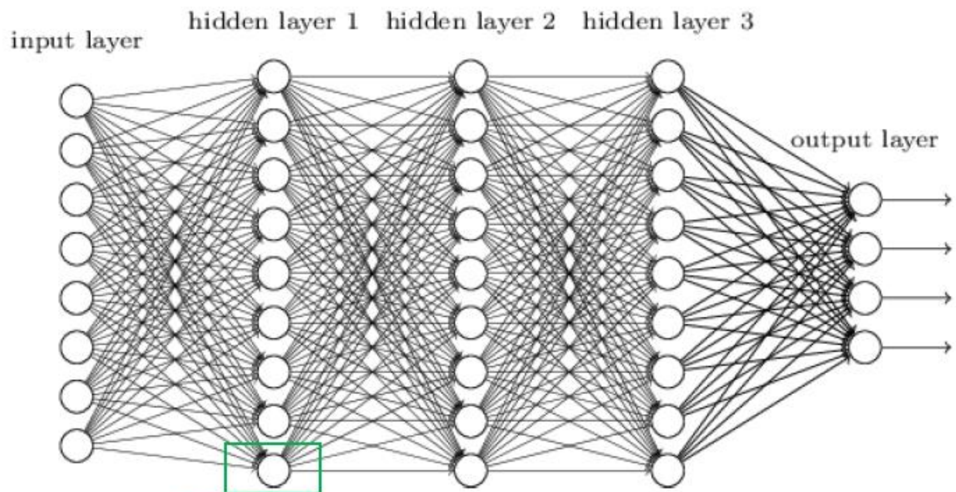
- Sparse coding
- Autoencoders

- Stacked Autoencoders
- Deep Belief Nets
- Hierarchical Sparse Coding

?

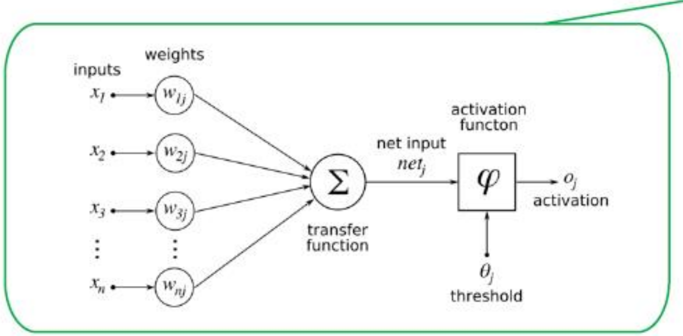
Unsupervised

Fully Connected Neural Networks

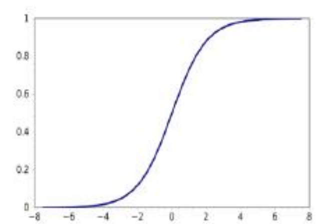


Cost functions:

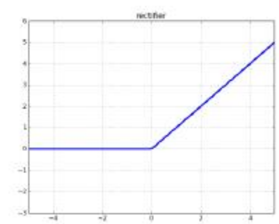
- Squared error
- Hinge loss
- Ranking loss



Activation functions



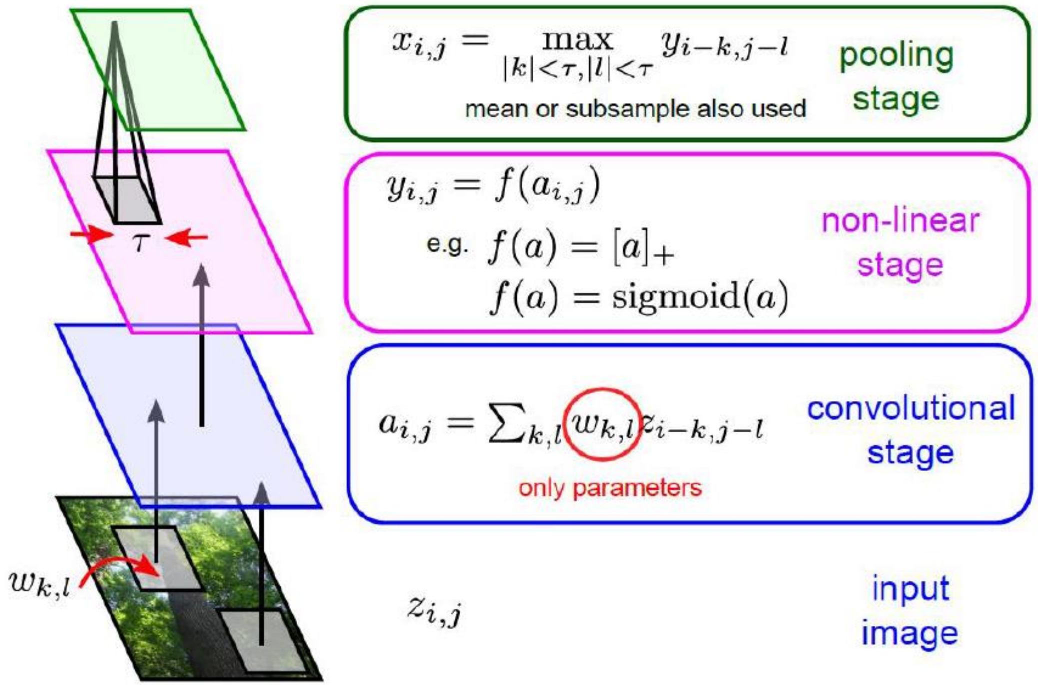
Sigmoid



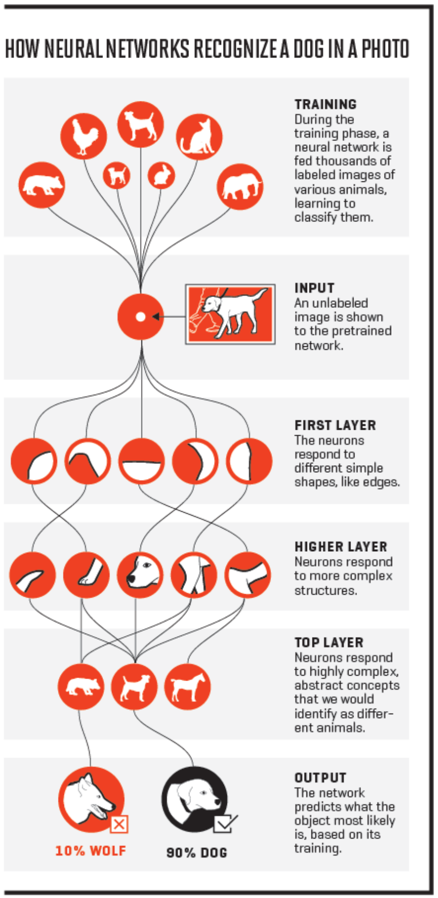
ReLU

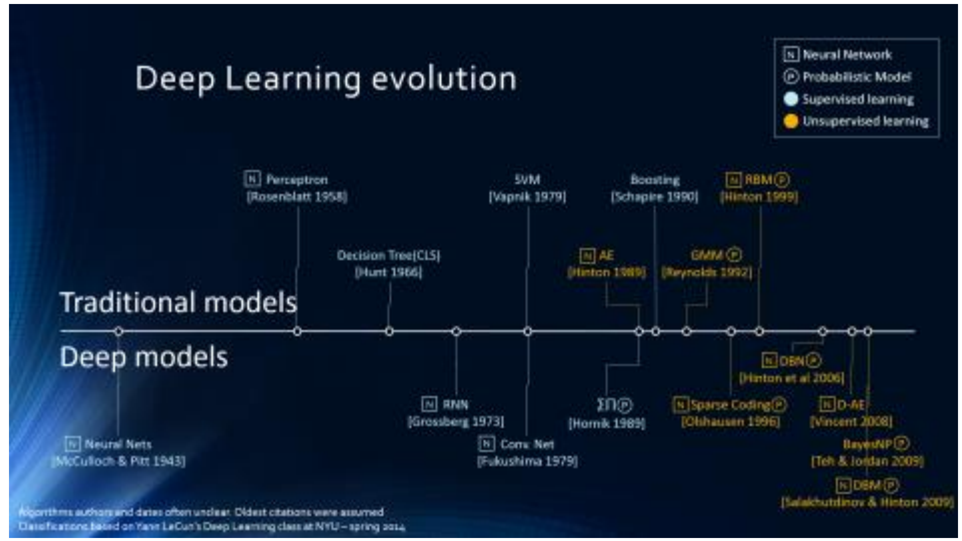
Convolutional Neural Networks

- Local connectivity
- Sharing weights
- Pooling (translation invariance)



Deep Learning

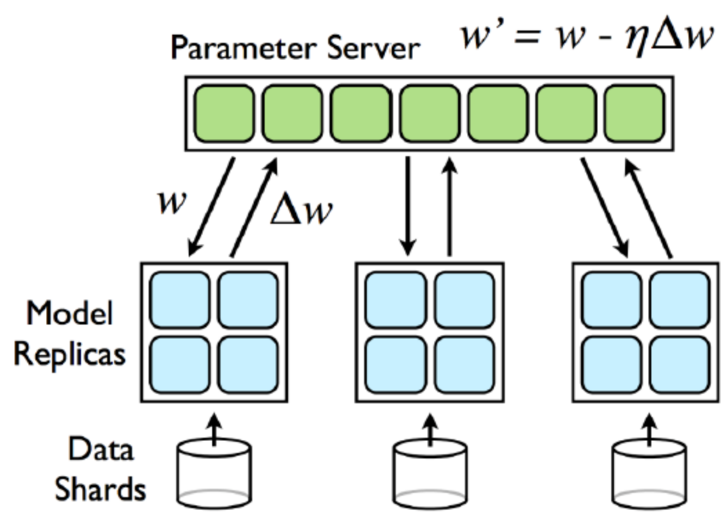




Distributed Machine Learning

Data Parallel Models

1. Partition the training data
2. Parallel training on different machines
3. Synchronize the local updates
4. Refresh local model with new parameters, then go to 2.



Machine Learning Methods

Shallow Models

- Linear models

$$f(x) = \sum_{j=1}^{d} w_j x_j$$

- Kernel methods (see SVM)

$$f(x) = \sum_{i=1}^{n} w_i k(x, x_i)$$

- Regularizions

$$F(w) := \frac{1}{n} \sum f_i(w) + \lambda R(w)$$

Deep Models

- Fully connected Neural Networks
- Convolutional Neural Networks
- Recurrent Neural Networks

$$f \in \mathcal{F}_A^L(\sigma, n_1, \ldots n_{L-1}, K)$$

Optimization framework (shallow)

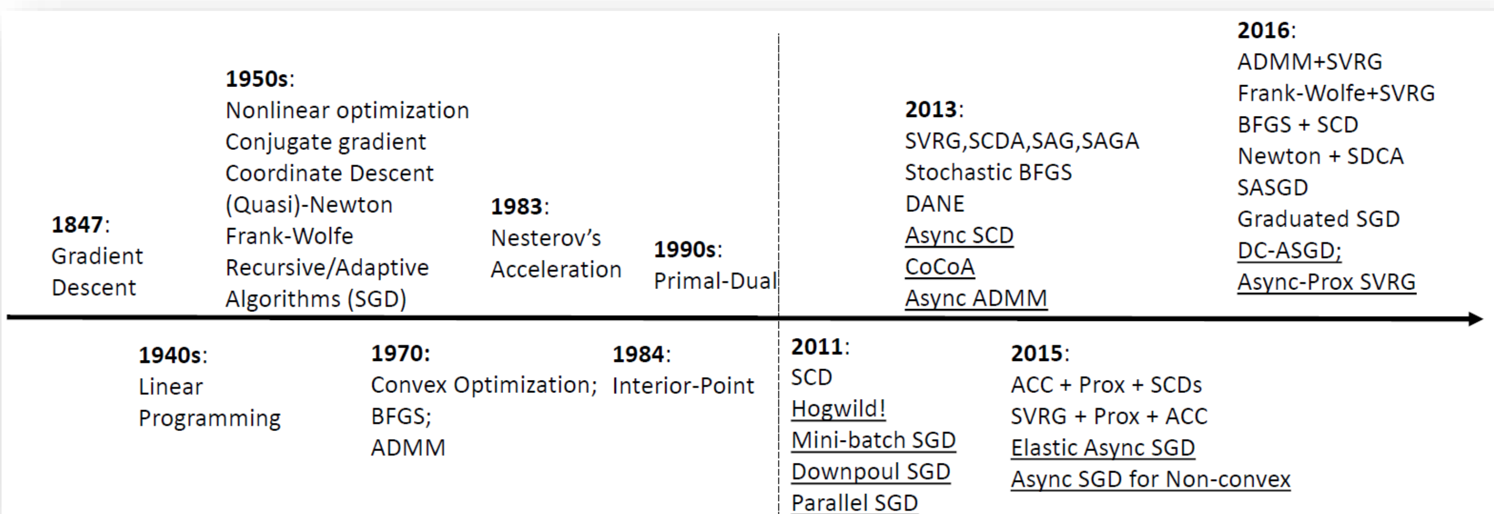
Problem: Empirical Risk Minimization $F(w) := \frac{1}{n}\sum f_i(w) + \lambda R(w)$

- Loss function

$$f_i(w) = L(w; x_i, y_i)$$

- Training Data

$$\{x_i, y_i; i = 1, \dots, n\}$$



Gradient Descent

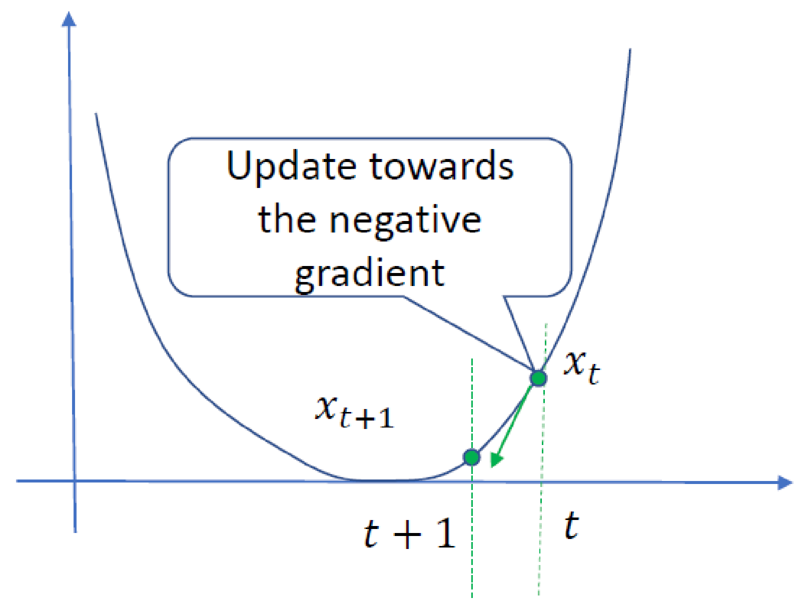
- Motivation: minimize first-order Taylor expansion of f at x

$$\min_x f(x) \approx \min_x f(x_t) + \nabla f(x_t)^\tau (x - x_t)$$

- Update rule

$$x_{t+1} = x_t - \eta \nabla f(x_t)$$

$\eta > 0$ is a fixed step-size



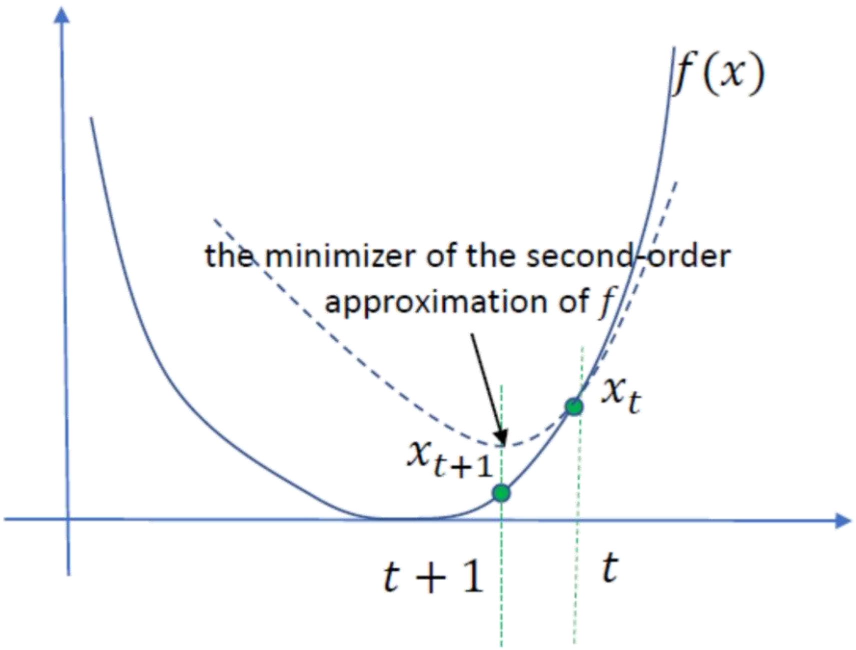
Newton's Method

- Motivation: minimize second-order Taylor expansion of f at x

$$\min_x f(x) \approx \min_x f(x_t) + \nabla f(x_t)^\tau (x - x_t) + \frac{1}{2}(x - x_t)^\tau \nabla^2 f(x_t)(x - x_t)$$

- Update rule

$$x_{t+1} = x_t - [\nabla^2 f(x_t)]^{-1} \nabla f(x_t)$$



Alternating Directions Method of Multipliers (ADMM)

- Separable objective with constraint

$$\min_{x,z} f(x) + g(z)$$
$$s.t.\ Ax + Bz = c$$

- Augmented Lagrangian: p>0

$$L_\rho(x, y, z) = f(x) + g(z) + y^T(Ax + Bz - c) + \left(\frac{\rho}{2}\right)||Ax + Bz - c||^2$$

- Update rule

$x^{t+1} = argmin_x L_\rho(x, z^t, y^t)$ -------x minimization

$z^{t+1} = argmin_z L_\rho(x^{t+1}, z, y^t)$ -------y minimization

$y^{t+1} = y^t + \rho(Ax^{t+1} + Bz^{t+1} - c)$ -------dual ascent update

Stochastic Optimization

Linear regression

- Objective

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x) = \frac{1}{n}\sum_{i=1}^{n}(a_i x - b_i)^2, x \in R^d$$

- Update rule

$$x_{t+1} = x_t - \eta \nabla f(x_t) = x_t - \frac{2\eta}{n}\sum_{i=1}^{n} a_i(a_i x - b_i)$$

Complexity

Linear increase with data size n

Linear increase with feature size d

Stochastic Gradient Descent (SGD)

- Data sampling (i: example index)

$$x_{t+1} = x_t - \eta_t \nabla f_i(x_t), \text{ where } \mathbb{E}_i \nabla f_i(x_t) = \nabla f(x_t)$$

	Convergence	Complexity (iter)	Complexity (overall)
GD	$O\left(\frac{\beta}{t}\right)$	$O(n \cdot d)$	$O\left(nd \cdot \beta \cdot \left(\frac{1}{\epsilon}\right)\right)$
SGD	$O\left(\frac{1}{\sqrt{t}}\right)$	$O(d)$	$O\left(\frac{d}{\epsilon^2}\right)$

Data Parallelism

- Optimization under different parallelization mechanisms
  - Synchronous vs Asynchronous
- Aggregation method
  - Consensus based on model averaging
- Data allocation
  - Shuffling + partitioning
  - Sampling

Distributed optimization with ADMM

- Problem formulation

$$\min_{\boldsymbol{w}} \sum_{k=1}^{K} L_k(w)$$

$$\text{s.t. } w_k - z = \mathbf{0}, k = 1, \dots, K$$

- Local updates

$$w_k^{t+1} = arg\,min_{w_k} \sum_k \left(L_k(w_k) + (\lambda_k^t)^T (w_k - z^t) + \frac{\rho}{2} \|w_k - z^t\|_2^2 \right)$$

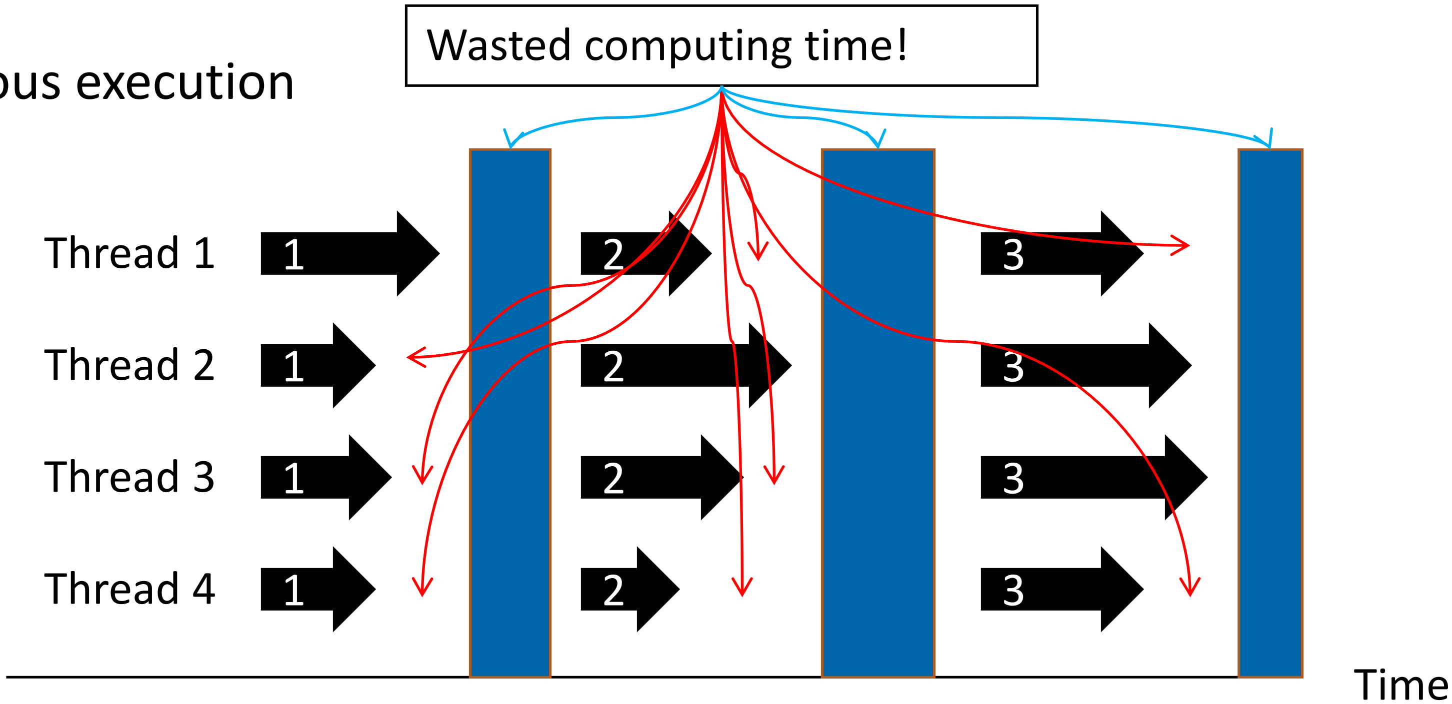
- Global consensus

$$z^{t+1} = \frac{1}{K} \sum_k (w_k^{t+1} + \frac{1}{\rho} \lambda_k^t)$$

$$\lambda_k^{t+1} = \lambda_k^t + \rho(w_k^{t+1} - z^{t+1})$$

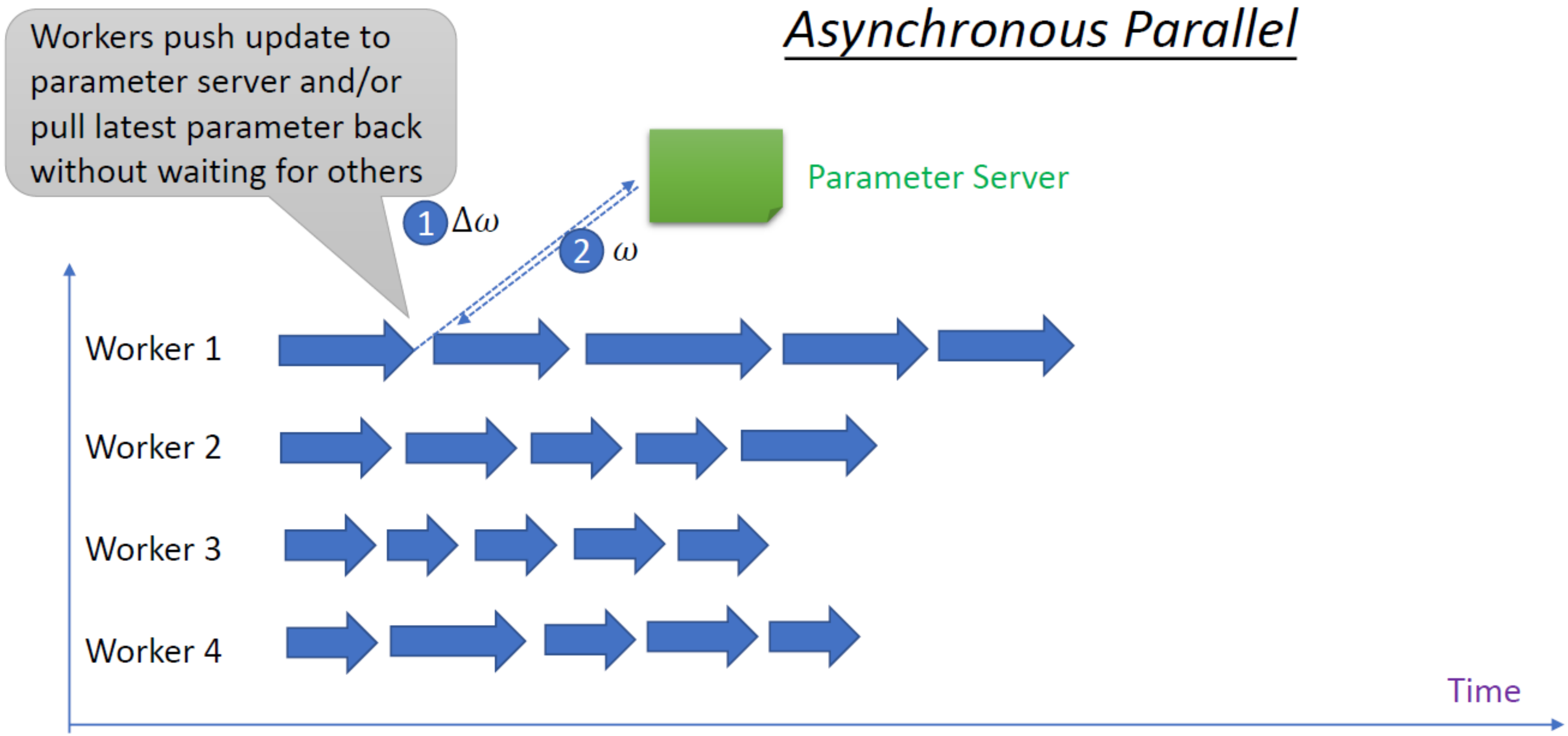
Distributed optimization

- Synchronous execution



- Exchange ALL updates at END of each iteration
  - Frequent, bursty communication
- Synchronize ALL threads each iteration
  - Straggler problem: stuck waiting for slowest

Asynchronous Parallel Processing



Encompassing model

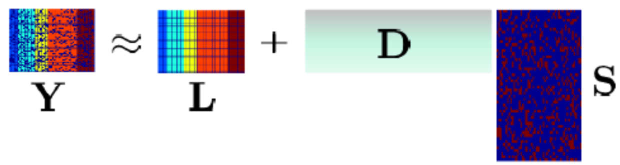
Background (low rank)

Patterns, innovations, (co-)clusters, outliers

Observed data $\in \mathbb{R}^{D\times T}$ → $\mathbf{Y} = \mathbf{L} + \mathbf{DS} + \mathbf{V}$ ← Noise

Dictionary $\in \mathbb{R}^{D\times Q}$

Sparse matrix $\in \mathbb{R}^{Q\times T}$



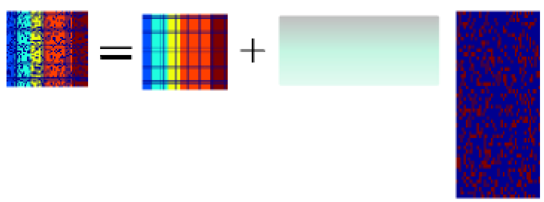
- $\boldsymbol{L} = \mathbf{0}, \boldsymbol{D}$ known $\Rightarrow$ Compressive sampling (CS) [Candes-Tao '05]
- $\boldsymbol{L} = \mathbf{0} \Rightarrow$ Dictionary learning (DL) [Olshausen-Field '97]
- $\boldsymbol{L} = \mathbf{0}, [\boldsymbol{D}]_{ij} \geq 0, [\boldsymbol{S}]_{ij} \geq 0 \Rightarrow$ Non-negative matrix factorization (NMF) [Lee-Seung '99]
- $\boldsymbol{D} = \boldsymbol{I}_D \Rightarrow$ Principal component pursuit (PCP) [Candes etal '11]
- $\boldsymbol{S} = \mathbf{0}, \mathrm{rank}(\boldsymbol{L}) \leq \rho \Rightarrow$ Principal component analysis (PCA) [Pearson 1901]

G. B. Giannakis, K. Slavakis, and G. Mateos , Signal Processing Tools for Big Data Analytics Nice, France August 31, 2015, ICASSP2015

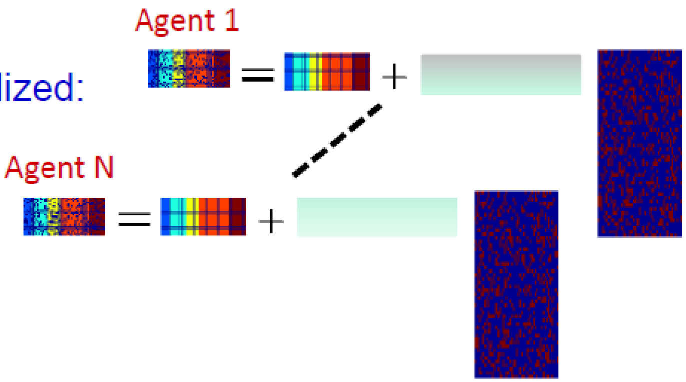
In-network decentralized processing

- Network anomaly detection: Spatially-distributed link count data

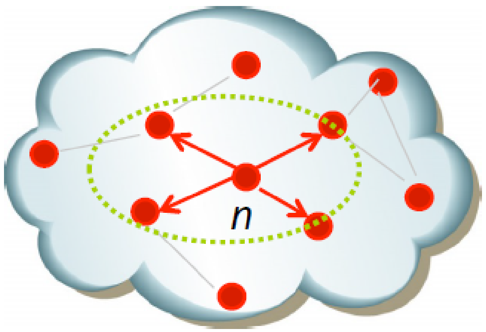
Centralized:



Decentralized:



In-network processing model:

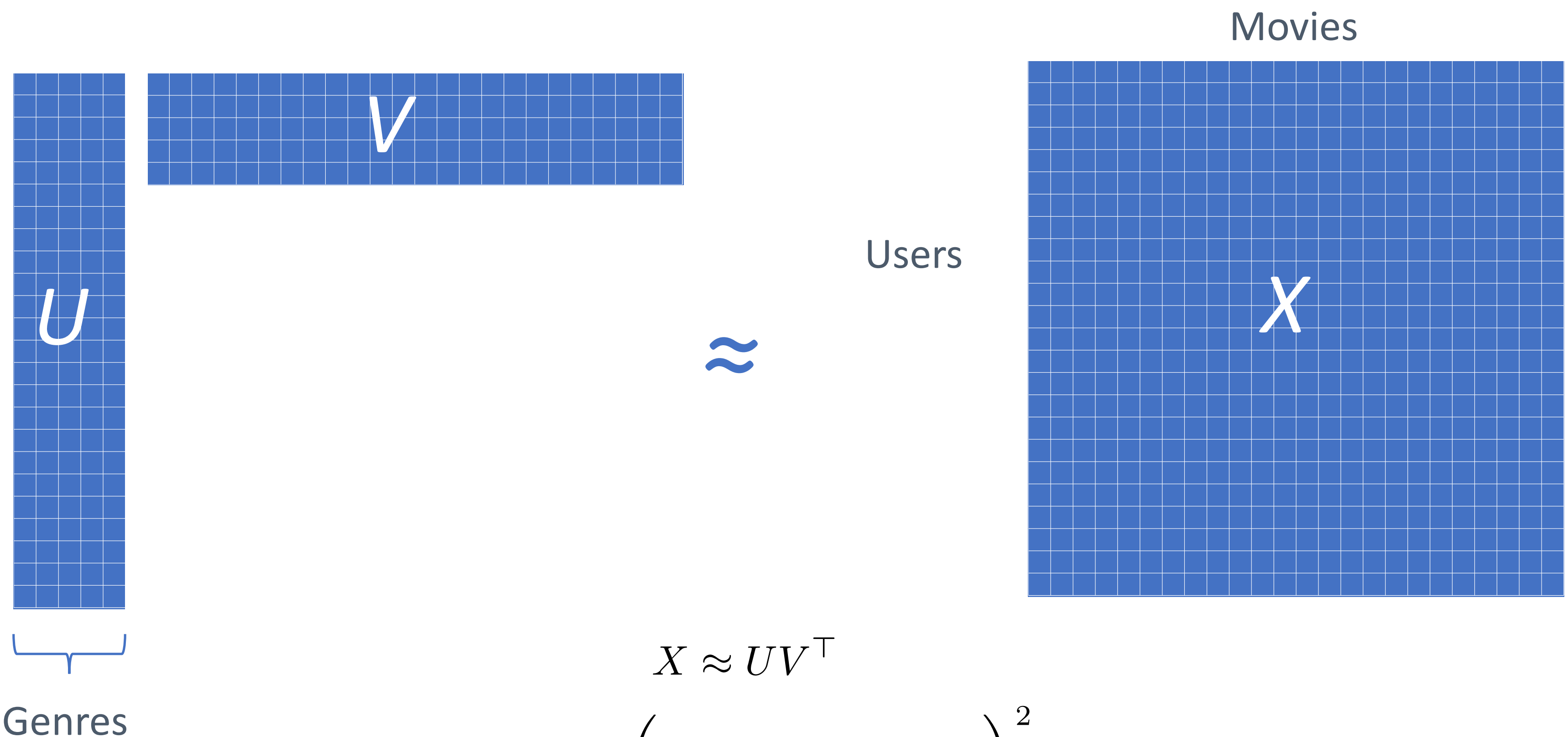


$$\|\mathbf{X}\|_* := \min_{\{\mathbf{U},\mathbf{\Psi}\}} \frac{1}{2}\left[\|\mathbf{U}\|_F^2 + \|\mathbf{\Psi}\|_F^2\right], \quad \text{s.to } \mathbf{X} = \mathbf{U}\mathbf{\Psi}$$

$L \times \rho$

$\geq \text{rank}[\boldsymbol{X}]$

SGD for Matrix Factorization

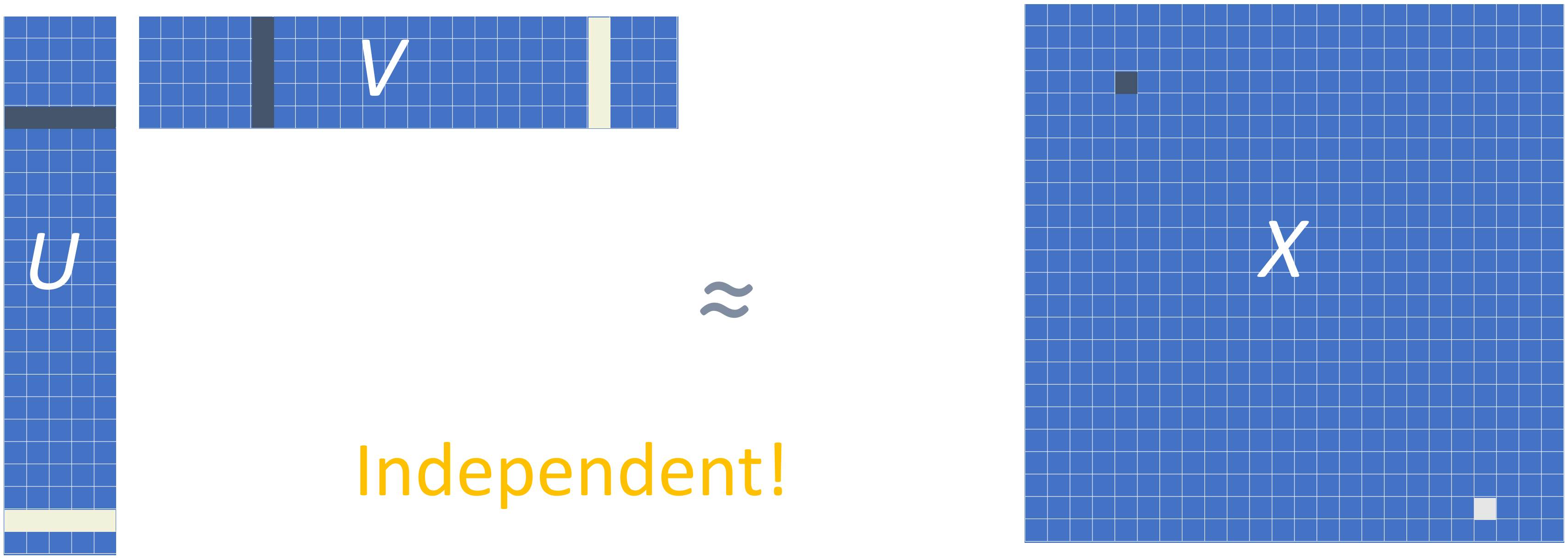


$$X \approx UV^\top$$

$$\min_{U,V} \|X - UV^\top\|_F^2 = \min_{U,V} \sum_{(i,j)\in X} \left(X_{i,j} - \sum_r U_{i,r} V_{j,r} \right)^2 = \min_{U,V} \sum_{(i,j)\in X} L_{i,j}(U,V)$$

$$L_{i,j}(U,V) = \left(X_{i,j} - \sum_r U_{i,r} V_{j,r} \right)^2$$

SGD for Matrix Factorization

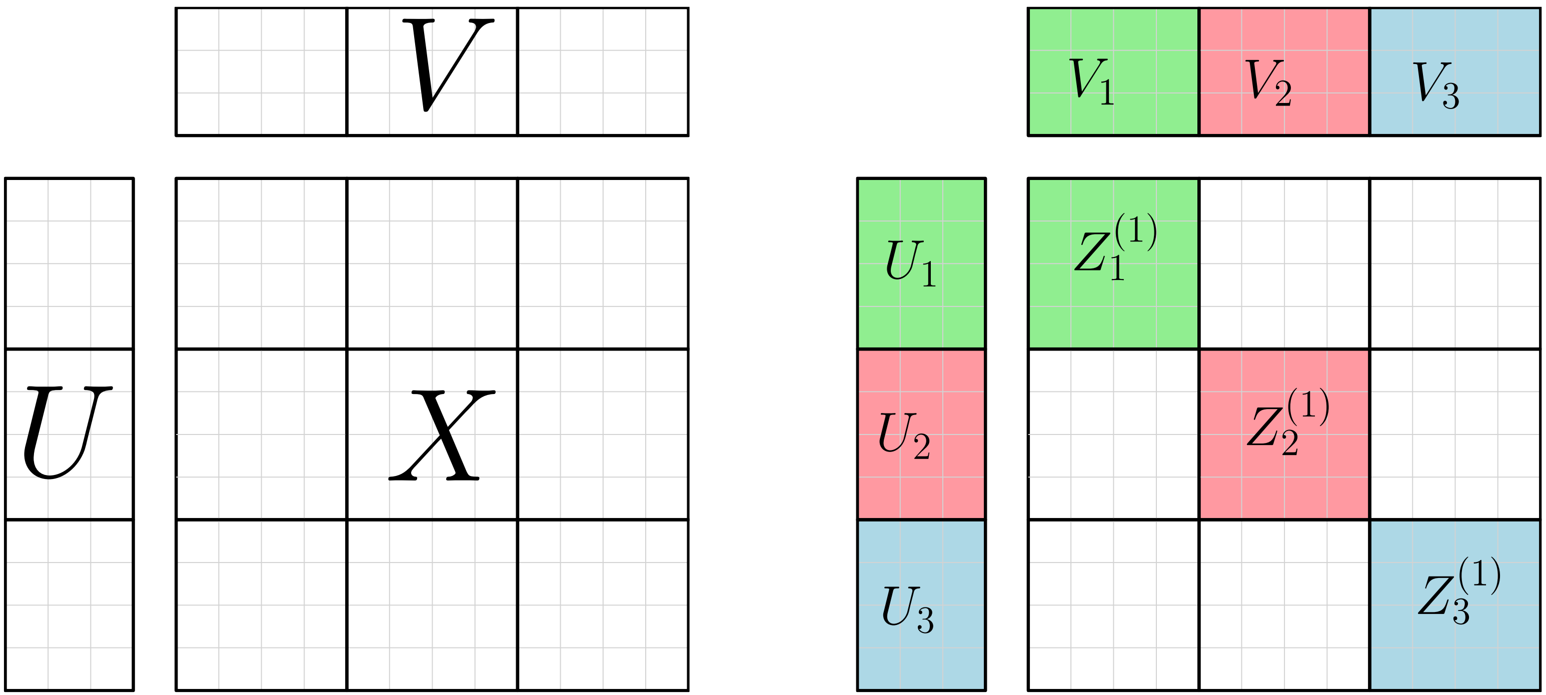


$$U_i = U_i - \eta \frac{\partial L_{i,j}(U,V)}{\partial U_i} \qquad \frac{\partial L_{i,j}(U,V)}{\partial U_i} = -2(X_{i,j} - \sum_r U_{i,r} V_{j,r}) V_j$$

$$V_j = V_j - \eta \frac{\partial L_{i,j}(U,V)}{\partial V_j} \qquad \frac{\partial L_{i,j}(U,V)}{\partial V_j} = -2(X_{i,j} - \sum_r U_{i,r} V_{j,r}) U_i$$

Material from: N. Sidiropoulos (UMN), E. Papalexakis (CMU), Tutorial ICASSP 2014, Florence, Italy

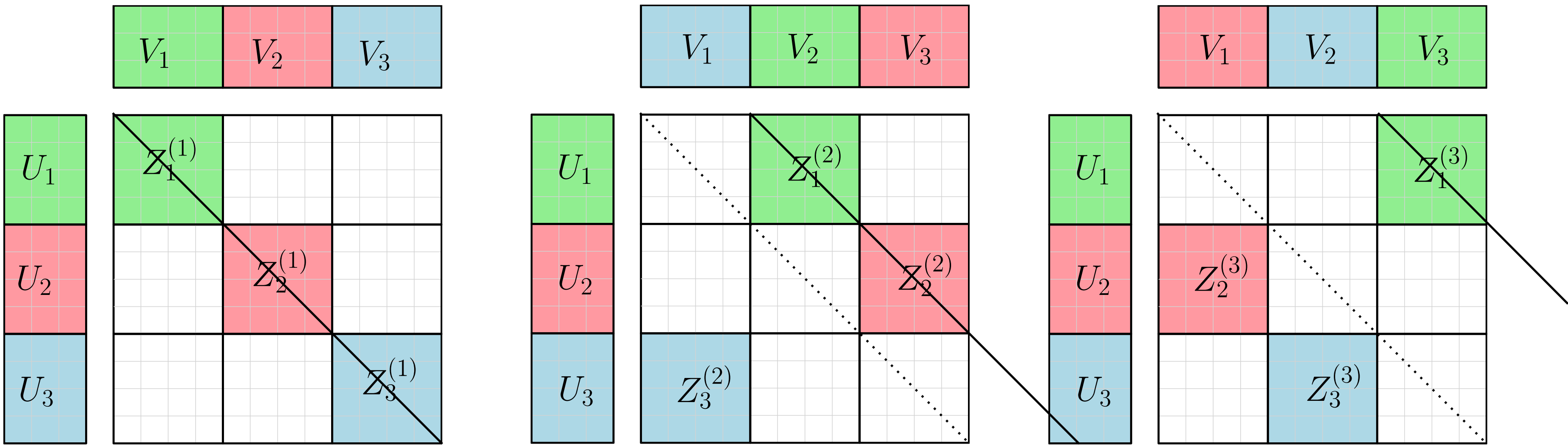
DSGD for Matrix Factorization



Material from: N. Sidiropoulos (UMN), E. Papalexakis (CMU), Tutorial ICASSP 2014, Florence, Italy

DSGD for Matrix Factorization

Partition your data & model into *d* × *d* blocks

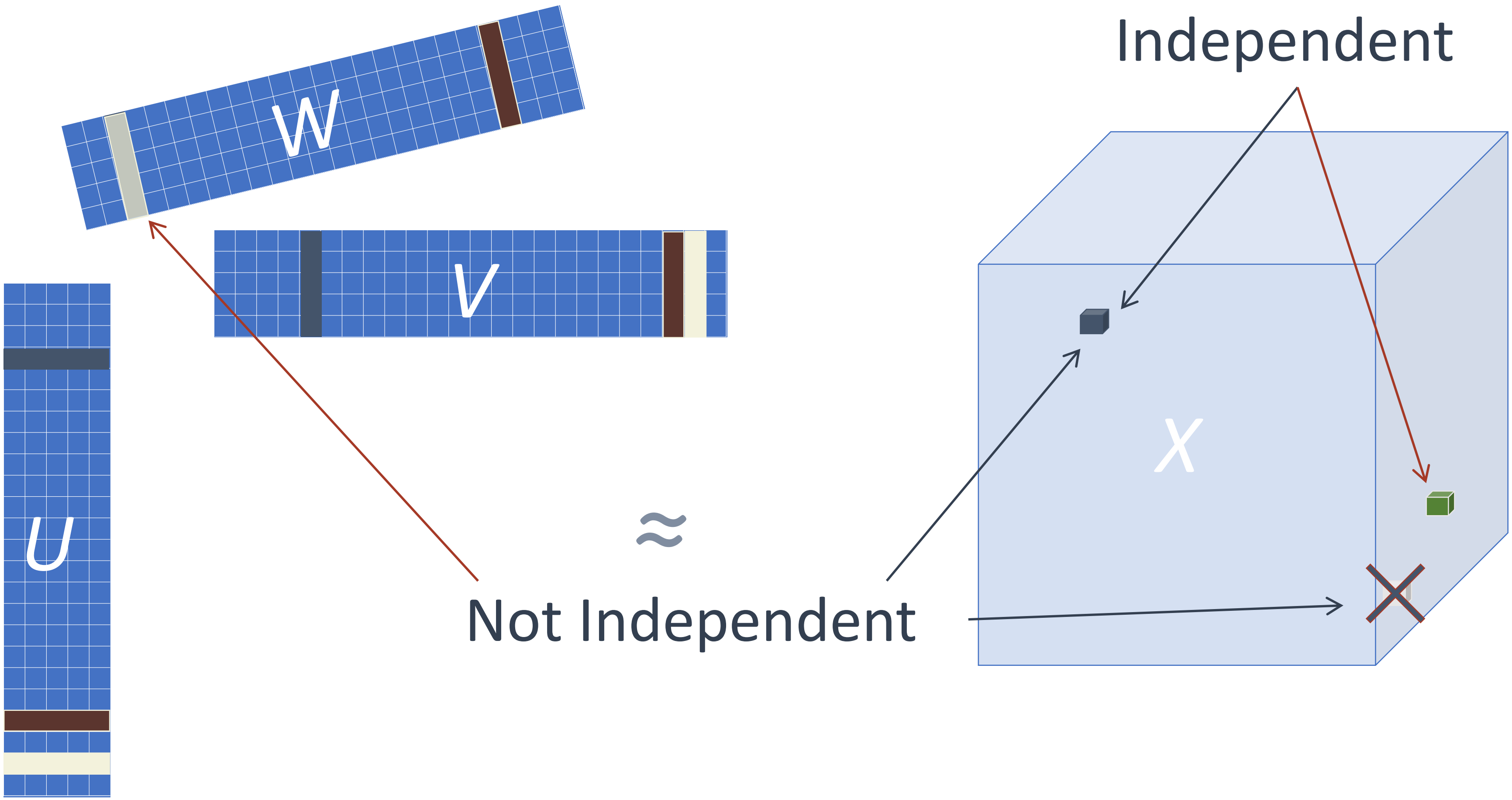


Results in *d=3* strata

Process strata sequentially,

process blocks in each stratum in parallel

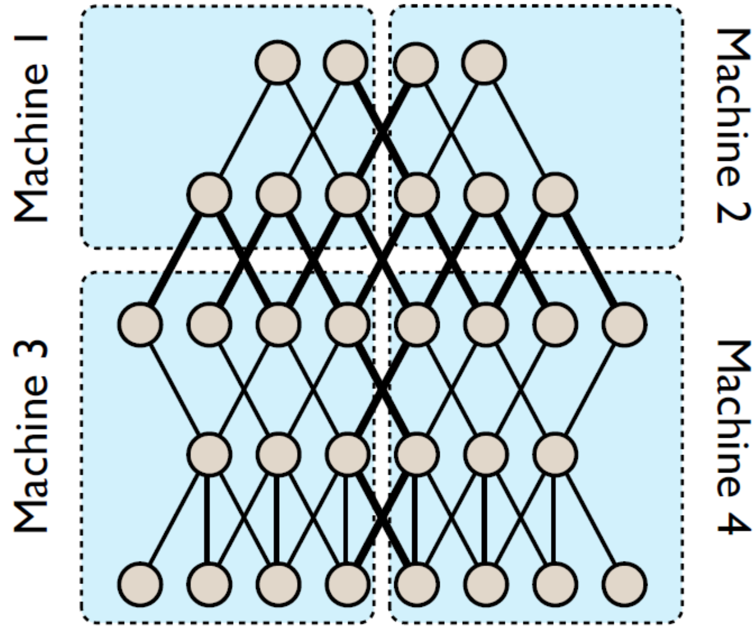
Tensor Decomposition



Distributed Machine Learning

Model Parallel Models

1. Partition the model into multiple local workers
2. For every sample, local workers collaborate to perform optimization



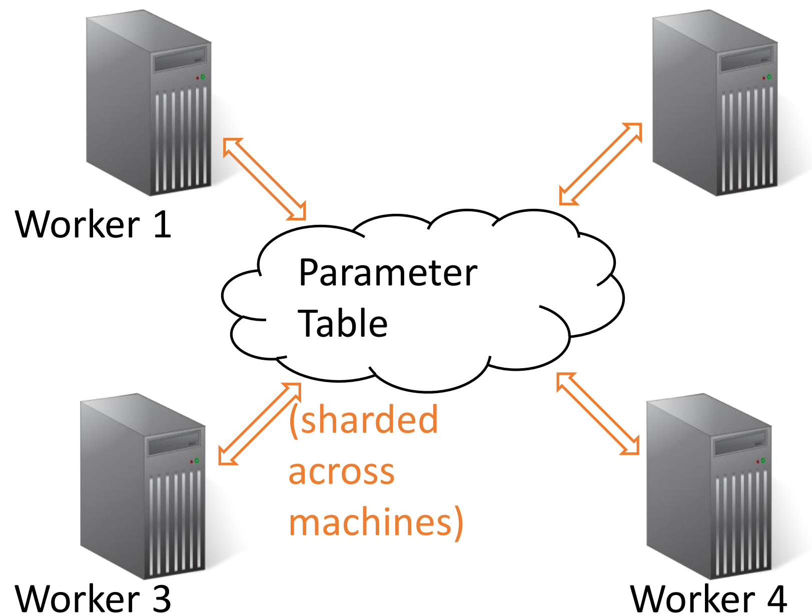
Parameter Server

- Single Machine Parallel

```
UpdateVar(i) {
  old = y[i]
  delta = f(old)
  y[i] += delta }
```

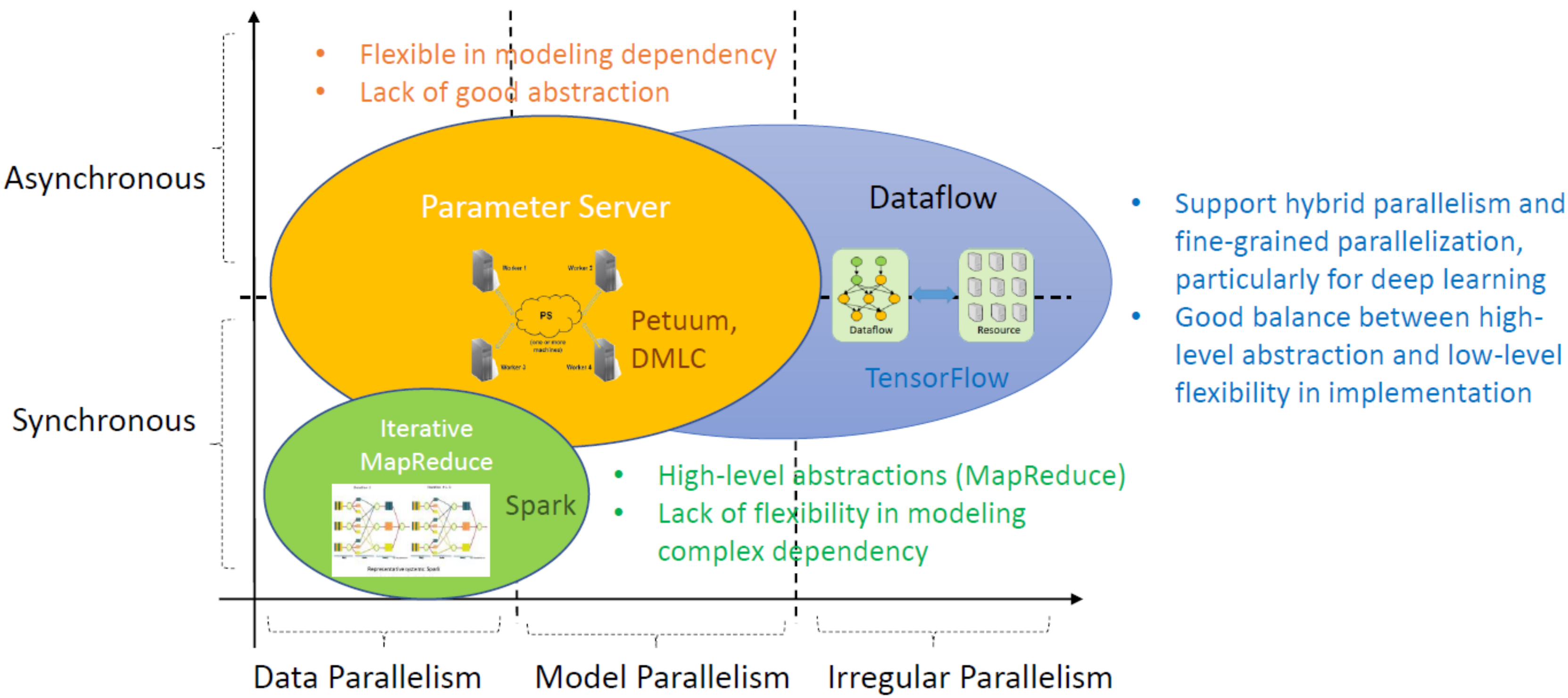
- Distributed with PS

```
UpdateVar(i) {
  old = PS.read(y,i)
  delta = f(old)
  PS.inc(y,i,delta) }
```



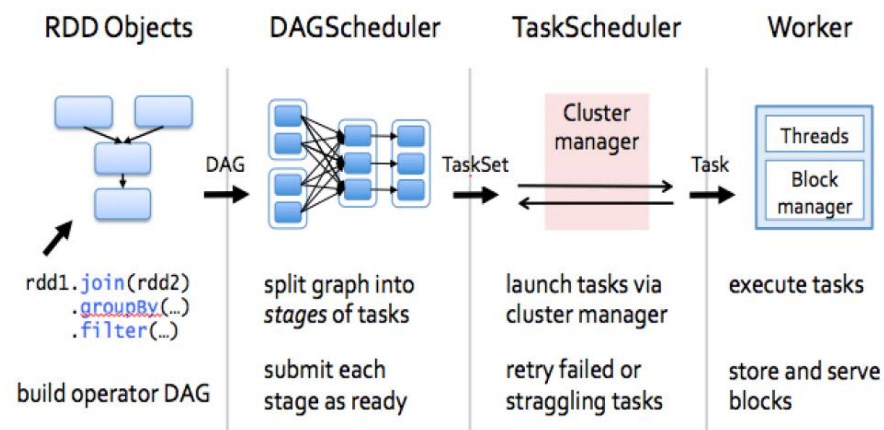
- Examples: Petuum, MXNet, TensorFlow, etc

Distributed Machine Learning Architectures

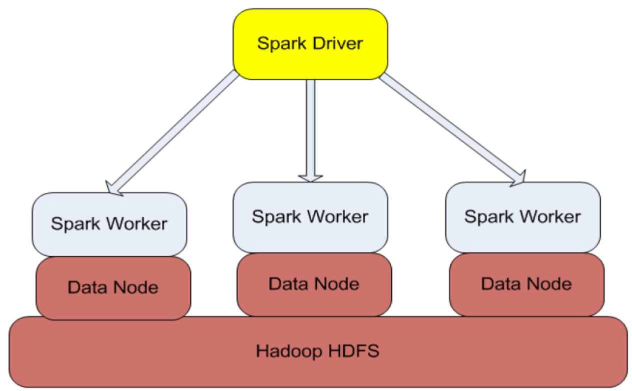


Spark

Resilient distributed datasets (RDD)
- Programming language with distributed collection data-structure



Distributed learning on Spark



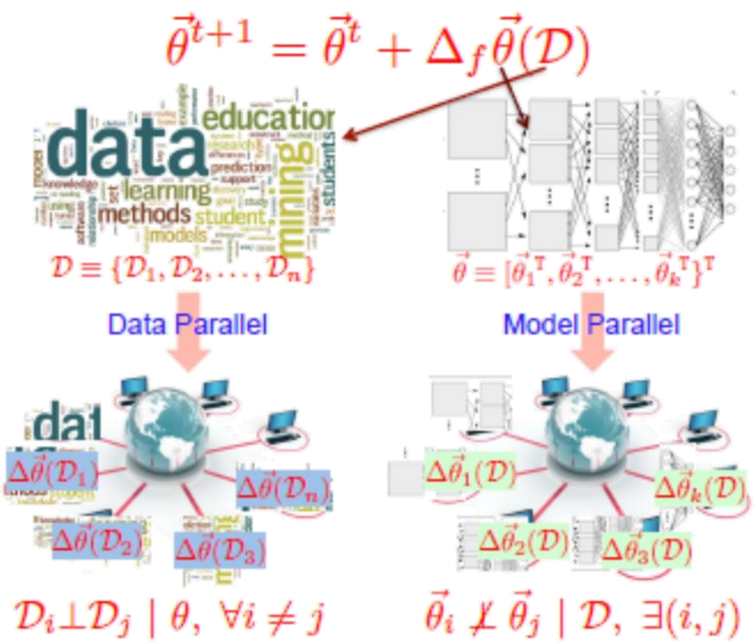
MLlib

- classification: logistic regression, linear SVM, naïve Bayes, least squares, classification tree
- regression: generalized linear models (GLMs), regression tree
- collaborative filtering: alternating least squares (ALS), non-negative matrix factorization (NMF)
- clustering: k-means|| decomposition: SVD, PCA optimization: stochastic gradient descent, L-BFGS

Petuum

The difference between data and model parallelism:

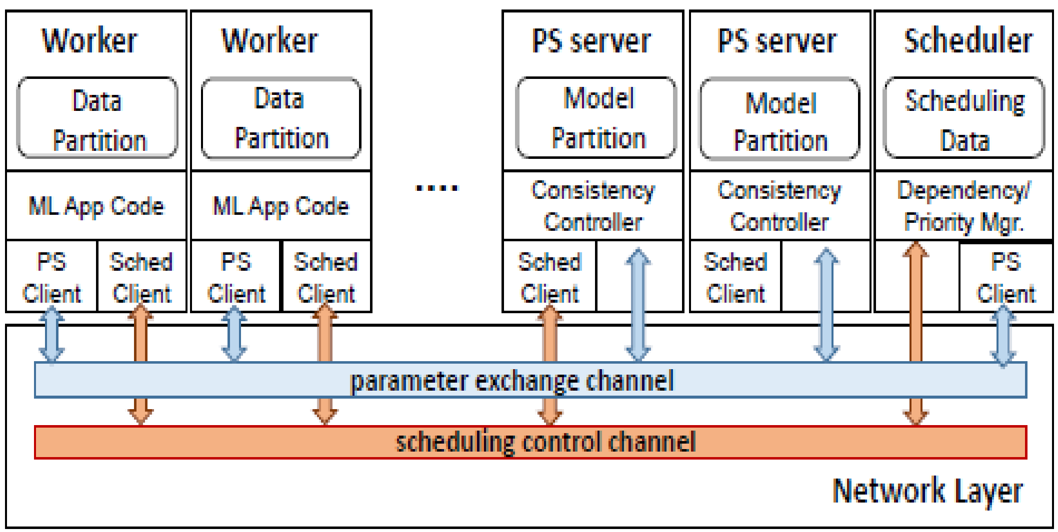
- data samples are always conditionally independent given the model
- Some model parameters that are not independent of each other.



Petuum

A parameter server: allows access to global model state from any machine via distributed shared-memory interface

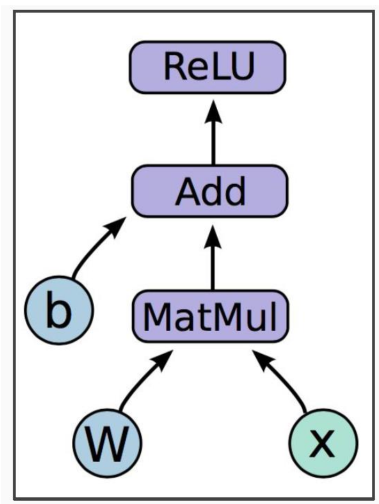
A scheduler allows fine-grained control over the parallel ordering of model-parallel updates



TensorFlow

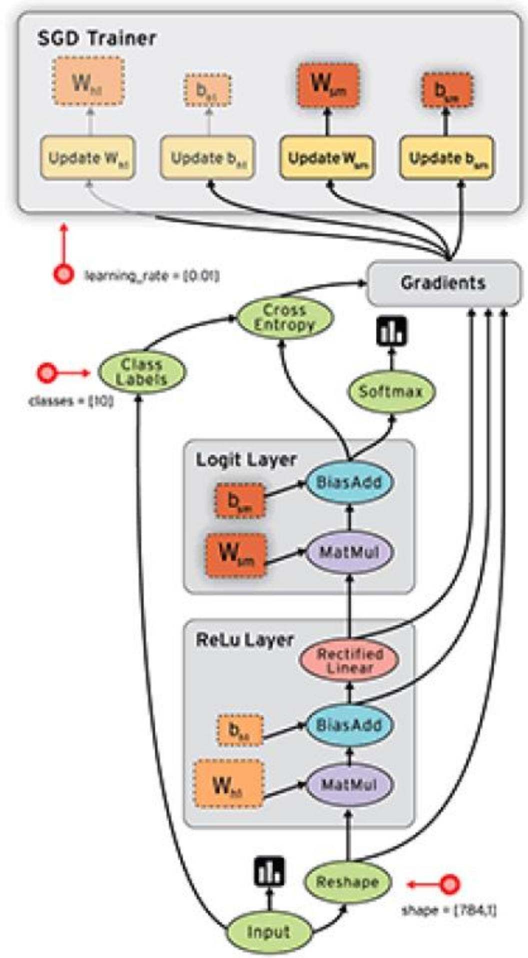
- TensorFlow is a deep learning library recently open-sourced by Google.
- But what does it actually do?
    - TensorFlow provides primitives for defining functions on tensors and automatically computing their derivatives.
- Computation graph

$$h = ReLU(Wx + b)$$

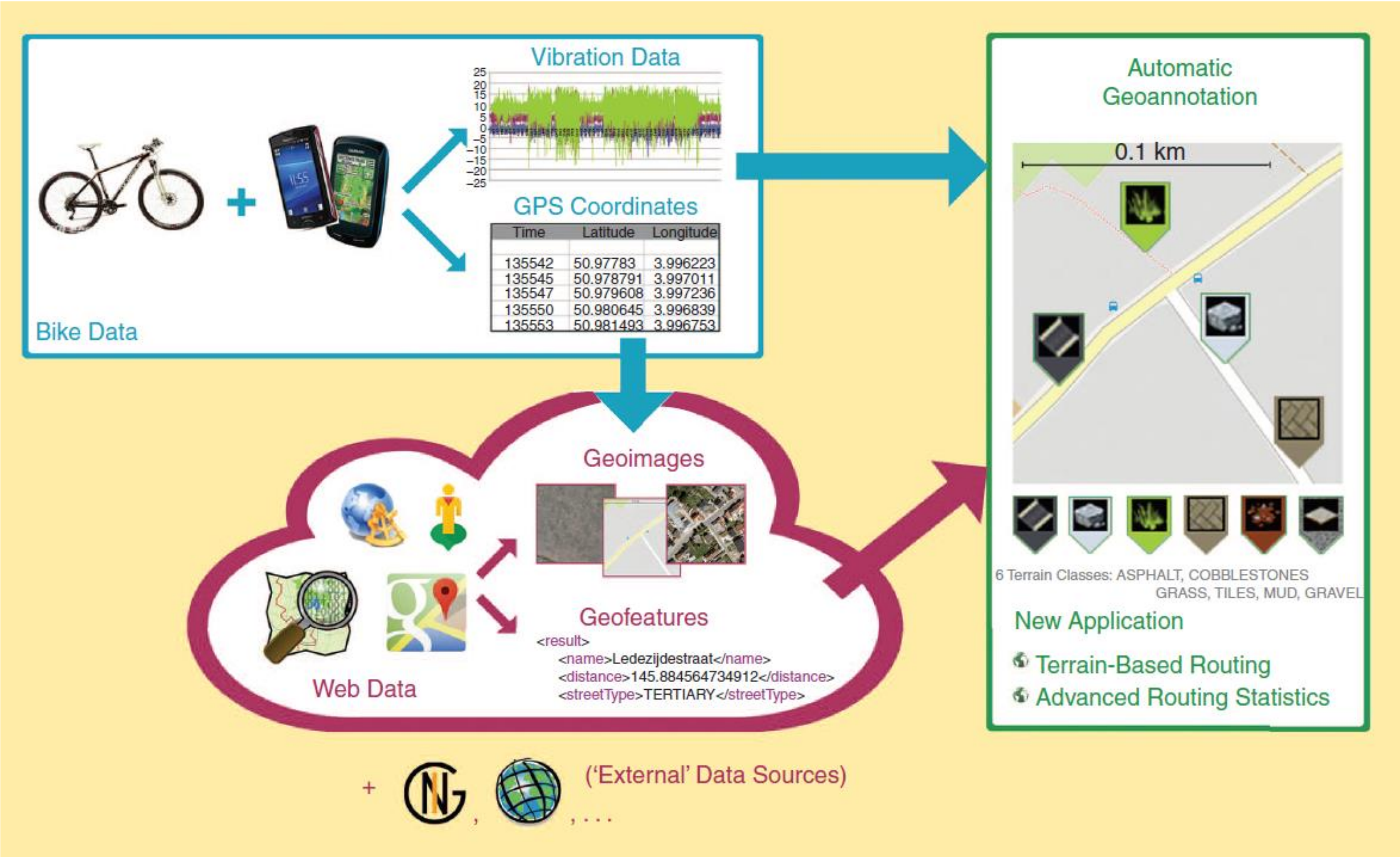


TensorFlow

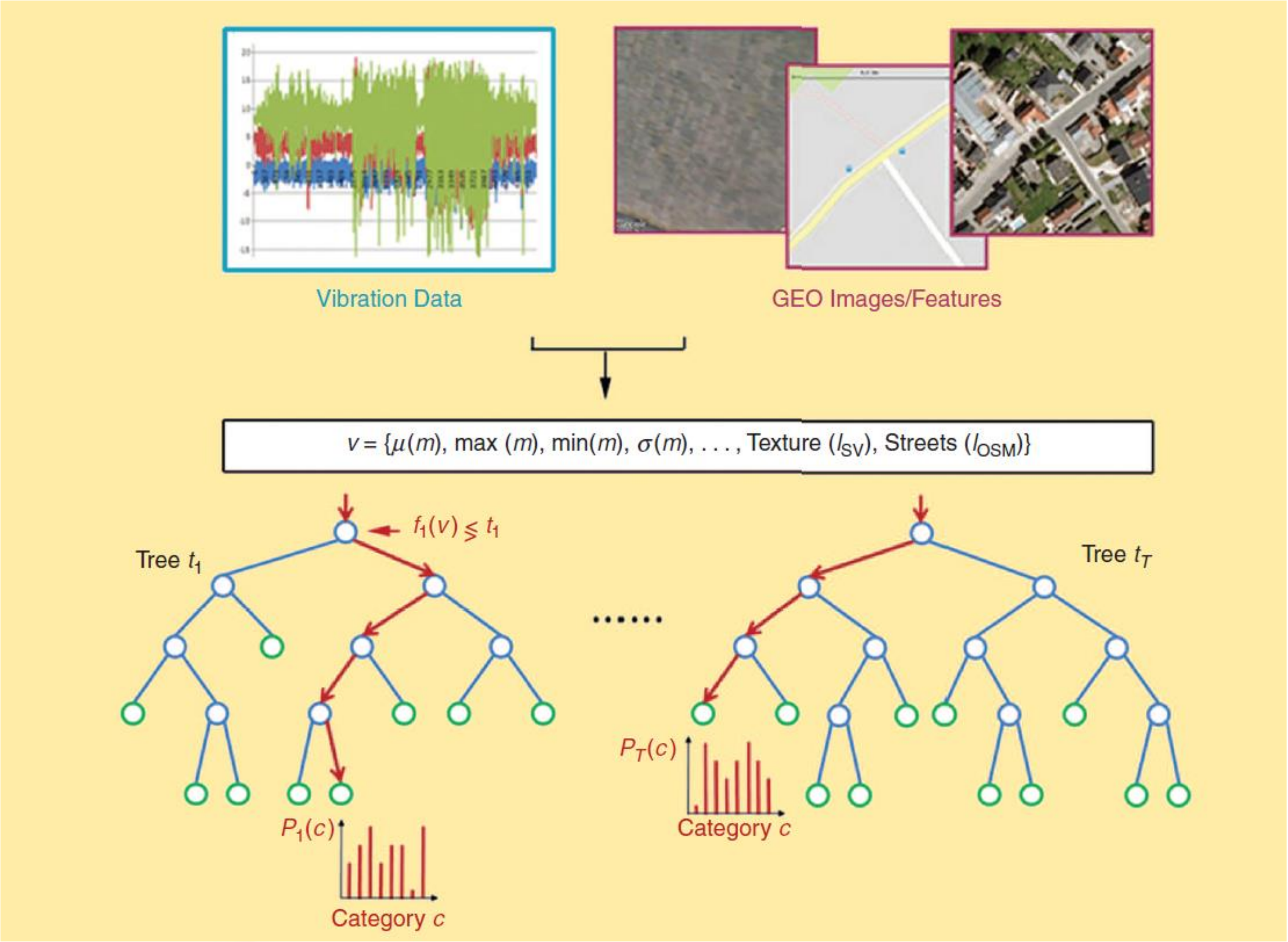
- In TensorFlow computation <-> Graphs.
  - Each node is an operation (op).
- Data is represented a Tensors.
  - Op takes Tensors and returns Tensors.
- Variables maintain state across executions of the graph.
- Two phases in the program:
  - Construct the computation graph.
  - Executes a graph in the context of a Session.



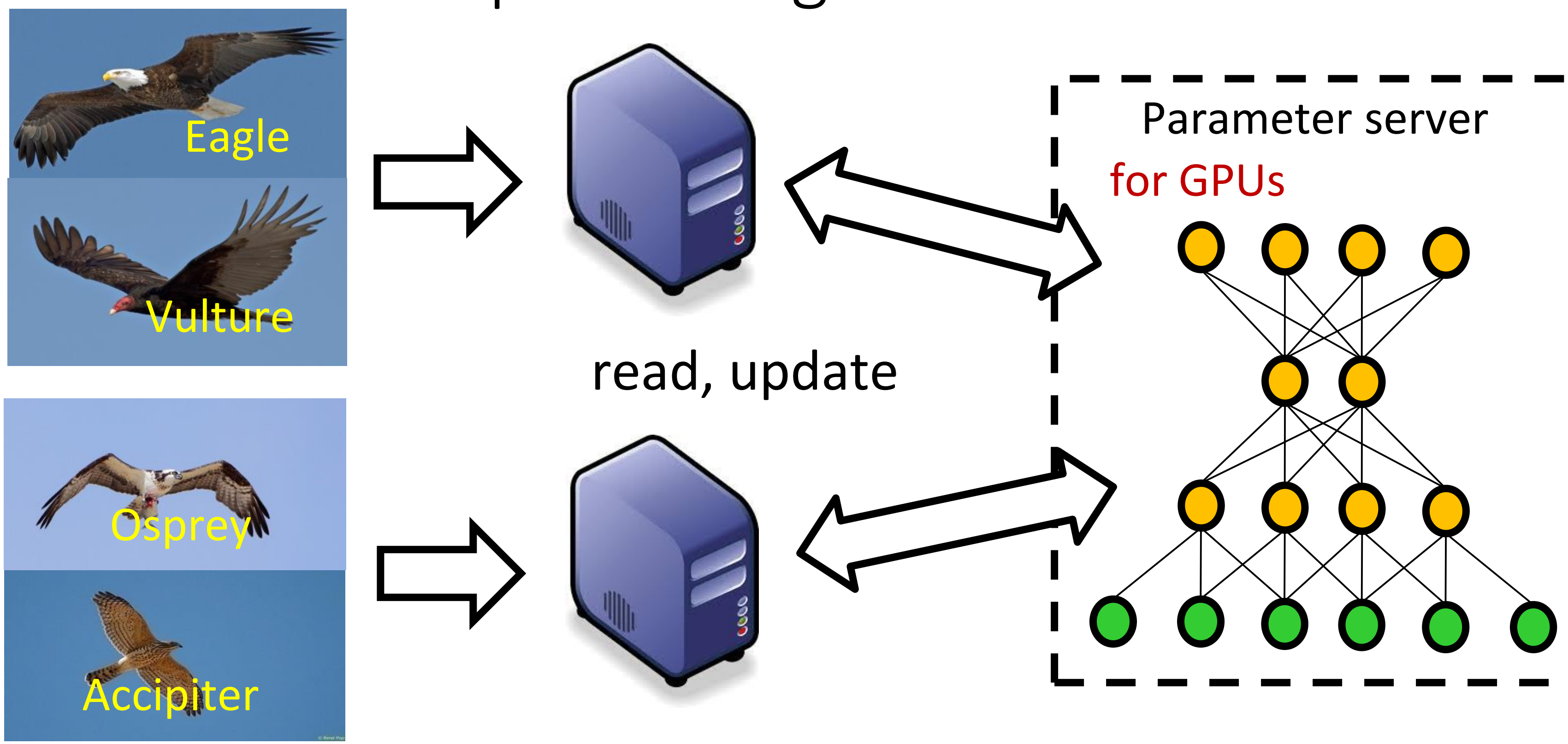
A multimodal bike-sensing setup for automatic geo-annotation of terrain types



A multimodal bike-sensing setup for automatic geo-annotation of terrain types



Distributed Deep Learning



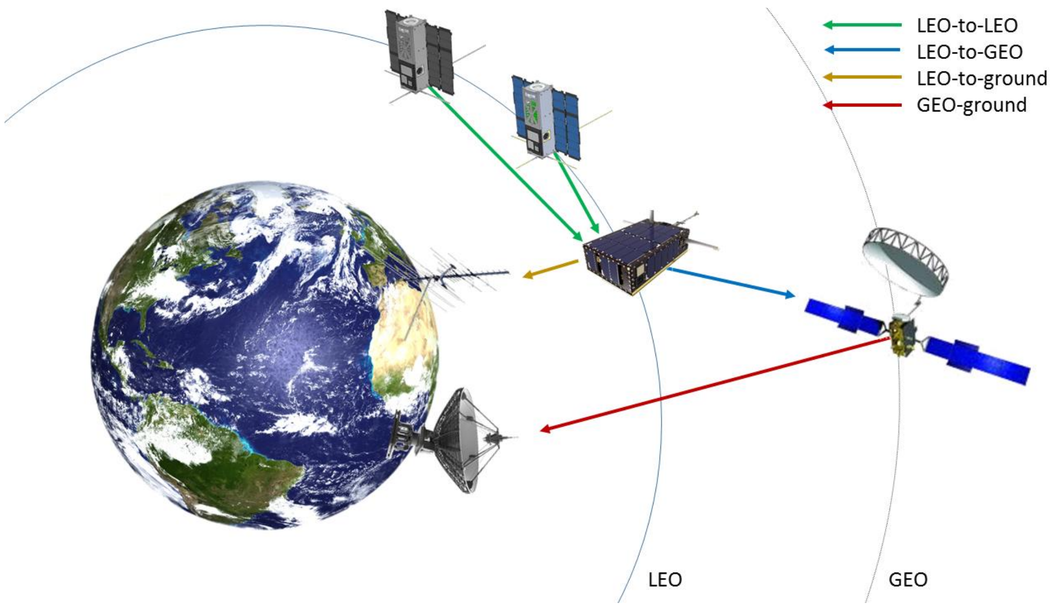
Partitioned training data

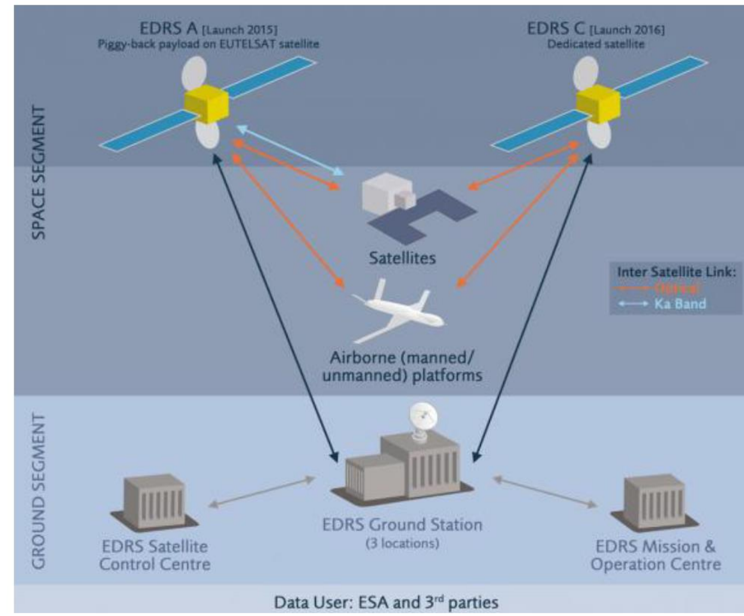
Distributed ML workers

Shared model parameters

WSN to space

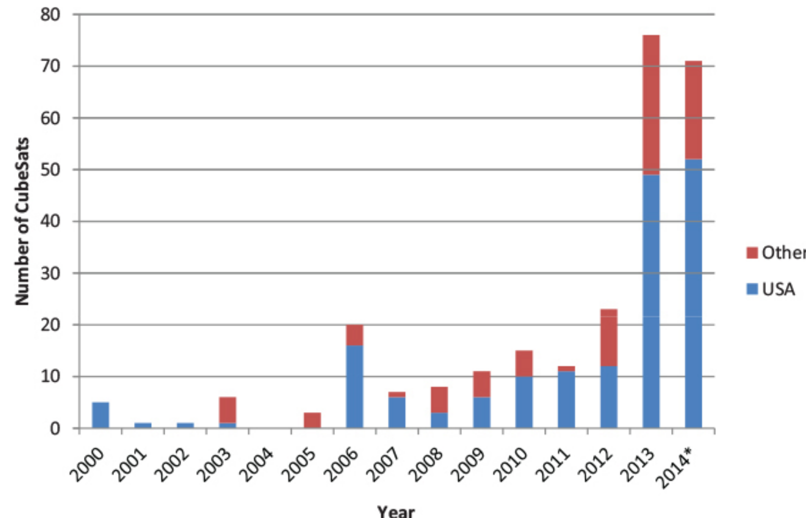
- Federated satellite architectures

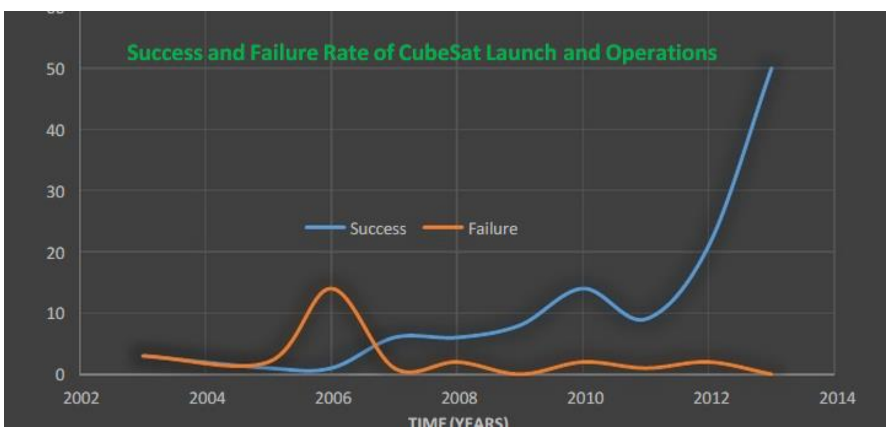


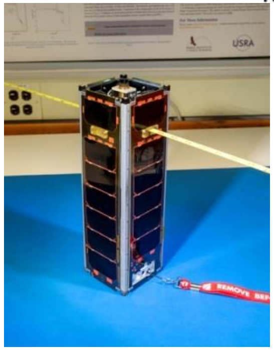


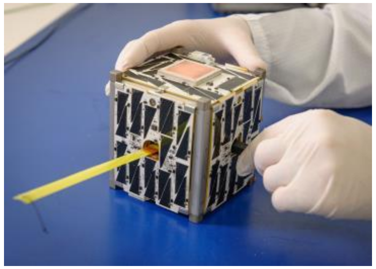
The CubeSat space platforms

Category	Mass (kg)	Cost (USD)
Large satellite	> 1000	0.1-2 B
Medium satellite	500-1000	50-100 M
Minisatellite	100-500	10-50 M
Microsatellite	10-100	2-10 M
Nanosatellite	1-10	0.2-2 M
Picosatellite	0.1-1	20-200 K
Femtosatellite	< 0.1	0.1-20 K



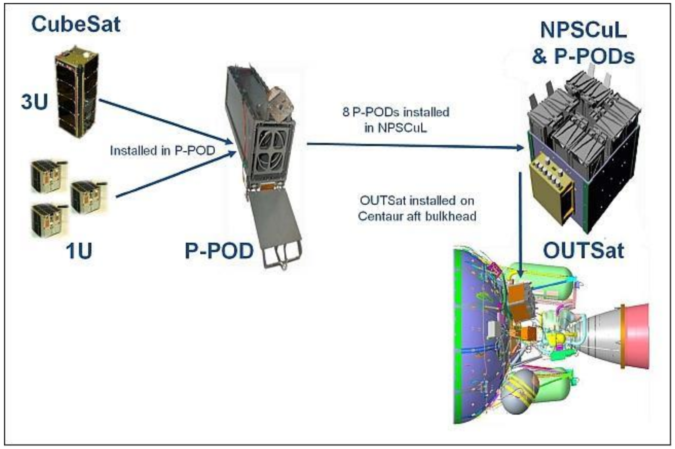




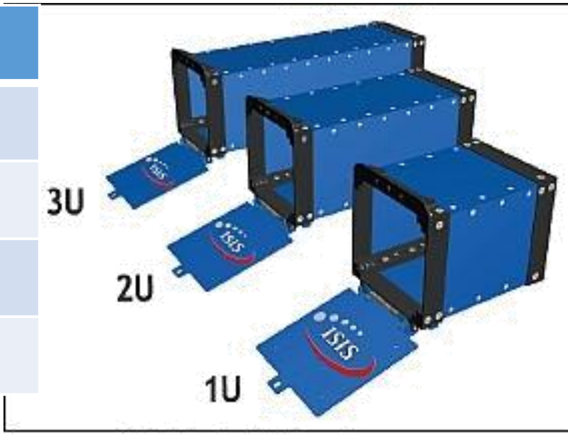


The CubeSat space platforms

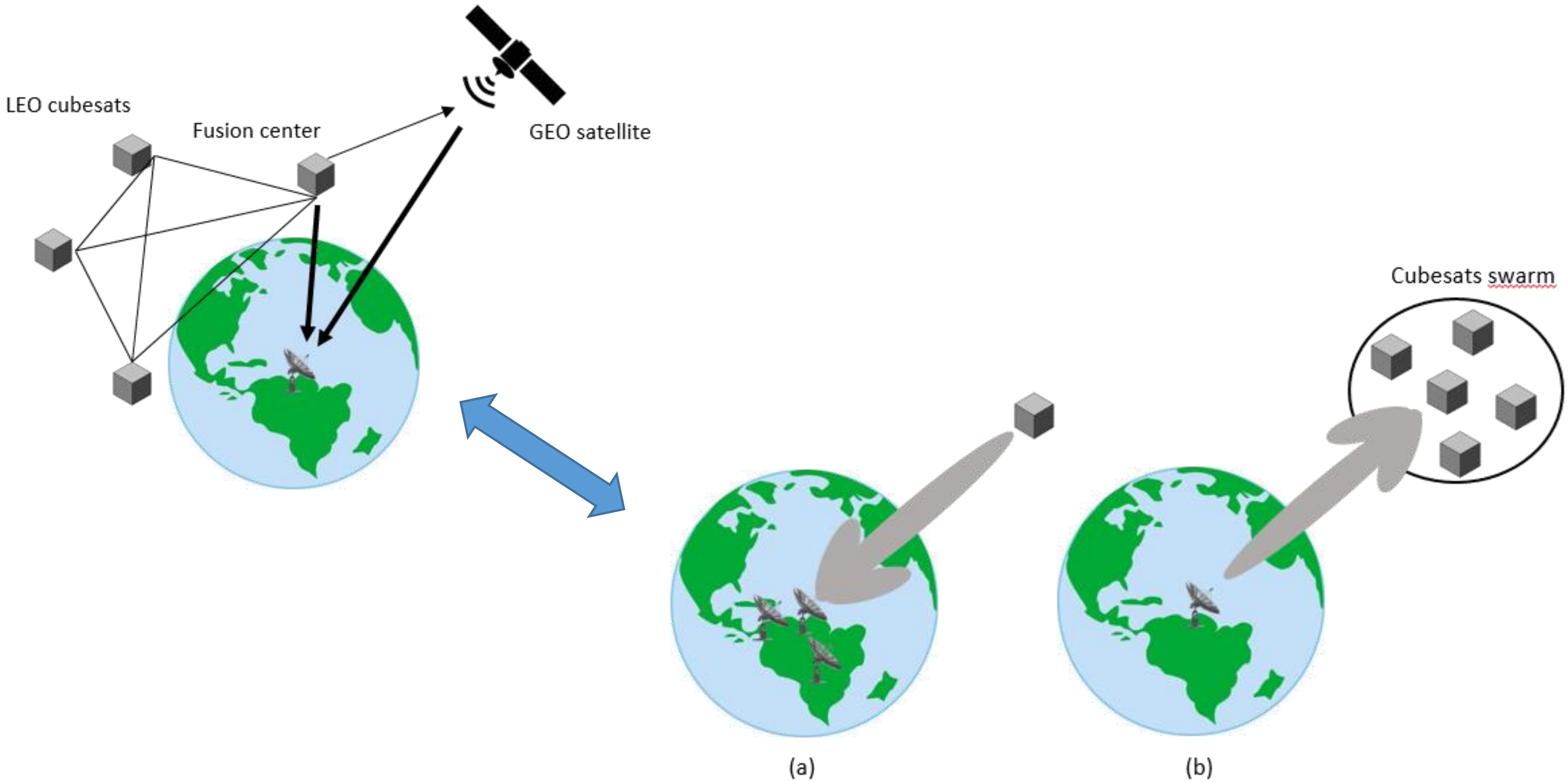
- Dimensioning
  - 1U: 10x10x10cm, 1Kg
  - 2U, 3U: 10x10x10:20/30, 2/3 Kg
- Applications



Name	Size	Organization	Mission	Launch
ExoCube	3U	CalPoly	Space weather	1/2015
GRIFEX	3U	U. Michigan & Nasa	Atmosphere	1/2015
AAU sat	1U	Aalborg University	Imaging	Failed
QuakeSat	3U	U. Stanford	Earthquakes	6/2003



Deployment architectures



Key objectives

- **Objective 1 – Computational remote sensing.**
  - minimize acquisition time, complexity of the sensor, removing mechanical components and replacing them with electronic ones, along with sophisticated computational methods.
- **Objective 2 – On-board payload data processing.**
  - optimally exploit and utilize heterogeneous processing units
  - low-level processing and high-level analysis of the acquired data.
- **Objective 3 – On-board compression and storage.**
  - high-capacity on-board COTS memory modules + mathematical tools -> maximizing the utilization of on-board storage
- **Objective 4 – Flexible high-rate communications.**
  - low-power high-rate space communication links.
  - low-latency direct and relay links for inter-satellite links and between satellites and ground stations.
- **Objective 5 – Distributed ground station networks.**
  - low-cost hardware components
  - reduce the complexity and latency of data reception and high-level data understanding

Case studies

Mission name	Number of small satellites	Mass of small satellites (Kg)	Inter-satellite links	Inter-satellite communication approach	Launched/Projected launch year
GRACE	2	480	Available	RF based (S-band)	2002
ESSAIM	2	120	Not available	Not available	2004
PRISMA	4	145, 50	Available	RF based (UHF-band)	2010
ELISA	4	130	Not available	Not available	2011
EDSN	8	1.7	Available	RF based (UHF-band)	2015
QB-50	50	2, 3	Available	RF based (S-band)	2016
PROBA-3	2	320, 180	Available	RF based (S-band)	2017
eLISA	3	To be determined	Available	Optical based (LASER)	2028
MAGNAS	28	210, 5	Available	RF based (UHF-band)	To be determined

Reading List

Zhang Y, Li W, Zhou P, Yang J, Shi X. Big Sensor Data: A Survey. In International Conference on Internet and Distributed Computing Systems 2016 Sep 28 (pp. 155-166). Springer International Publishing.