PARALLEL & SCALABLE MACHINE & DEEP LEARNING WITH APPLICATIONS

PROF. DR. – ING. MORRIS RIEDEL, UNIVERSITY OF ICELAND / JUELICH SUPERCOMPUTING CENTRE (JSC)

18TH FEBRUARY, 25TH INTERNATIONAL INFORMATION TECHNOLOGY CONFERENCE, MONTENEGRO, ONLINE

@ProfDrMorrisRiedel @Morris Riedel @MorrisRiedel @MorrisRiedel https://www.youtube.com/channel/UCWC4VKHmL4NZgFfKoHtANKg

INFORMATION TECHNOLOGY CONFERENCE

EuroHPC Joint Undertaking

EOSC NORDIC

RAISE Center of Excellence

ADMIRE

UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

EURO

HELMHOLTZAI ARTIFICIAL INTELLIGENCE COOPERATION UNIT

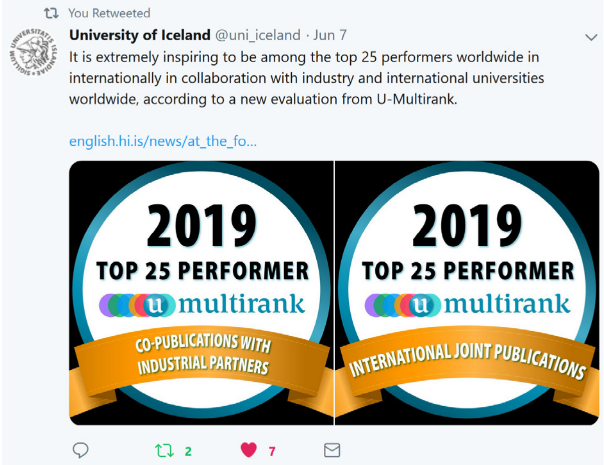
DEEP Projects

JÜLICH Forschungszentrum JÜLICH SUPERCOMPUTING CENTRE

UNIVERSITY OF ICELAND

School of Engineering and Natural Sciences (SENS)

- Selected Facts
  - *Ranked among the top 300 universities in the world (by Times Higher Education)*
  - ~2900 students at the SENS school
  - Long collaboration with Forschungszentrum Juelich
  - *~350 MS students and ~150 doctoral students.*
  - *Many foreign & Erasmus students; english courses*



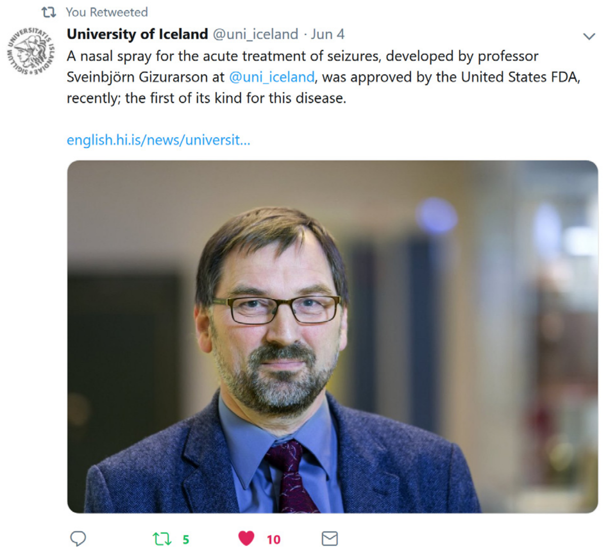
You Retweeted

University of Iceland @uni_iceland · Jun 7

It is extremely inspiring to be among the top 25 performers worldwide in internationally in collaboration with industry and international universities worldwide, according to a new evaluation from U-Multirank.

english.hi.is/news/at_the_fo...

2019
TOP 25 PERFORMER
multirank
CO-PUBLICATIONS WITH INDUSTRIAL PARTNERS
2019
TOP 25 PERFORMER
multirank
INTERNATIONAL JOINT PUBLICATIONS
2
7



You Retweeted

University of Iceland @uni_iceland · Jun 4

A nasal spray for the acute treatment of seizures, developed by professor Sveinbjörn Gizurarson at @uni_iceland, was approved by the United States FDA, recently; the first of its kind for this disease.

english.hi.is/news/universit...

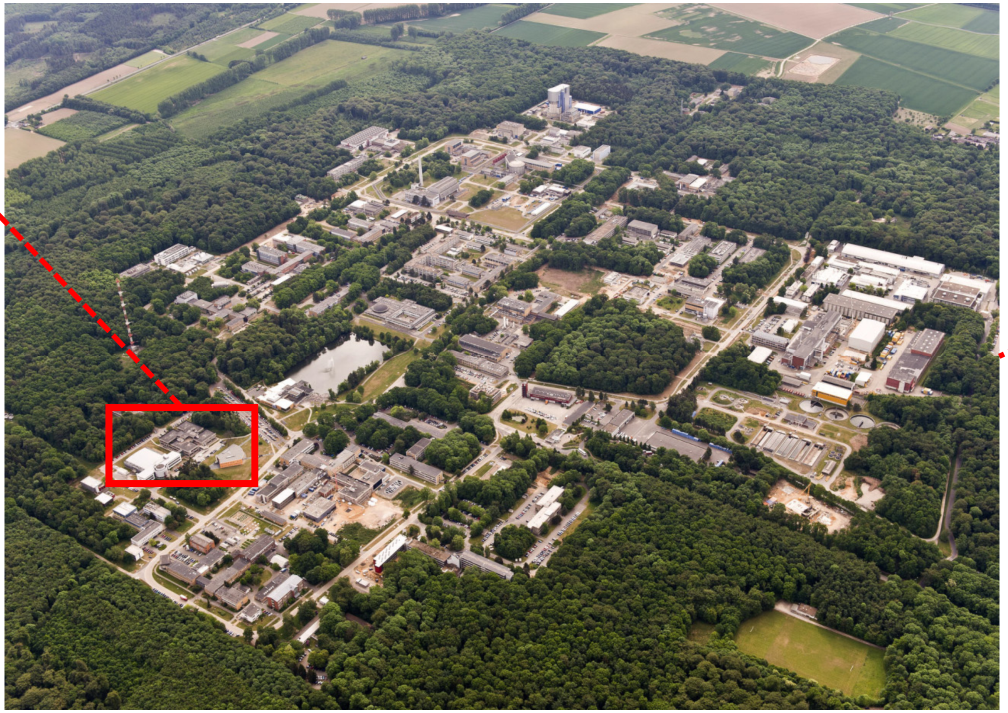
UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING,
MECHANICAL ENGINEERING AND COMPUTER SCIENCE

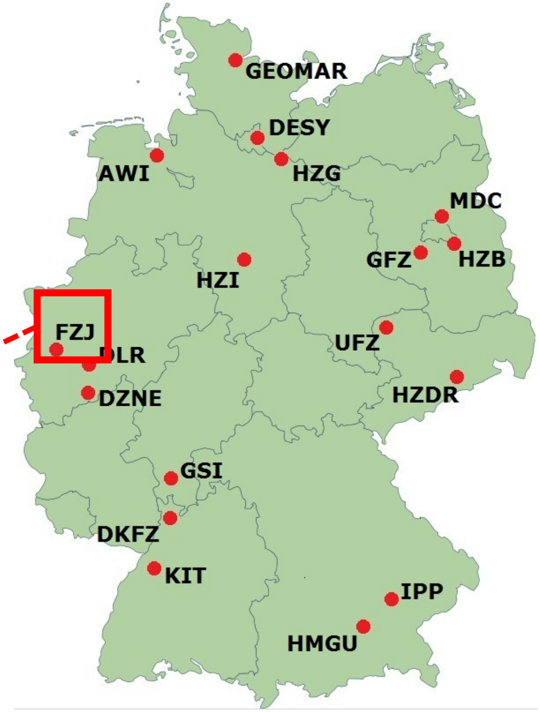
[2] University of Iceland Web page

JUELICH SUPERCOMPUTING CENTRE (JSC)

JÜLICH Forschungszentrum

Institute of Multi-Disciplinary Research Centre Forschungszentrum Juelich of the Helmholtz Association





HELMHOLTZ
RESEARCH FOR GRAND CHALLENGES

[1] Holmholtz Association Web Page

- Selected Facts
  - One of EU largest inter-disciplinary research centres (~5000 employees)
  - Special expertise in physics, materials science, nanotechnology, neuroscience and medicine & information technology (HPC & Data)

JUELICH SUPERCOMPUTING CENTRE (JSC) OF FZJ

Simulation & Data Labs (SDL) using High Performance Computing (HPC)

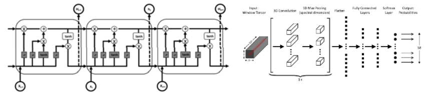
Research Group High Productivity Data Processing

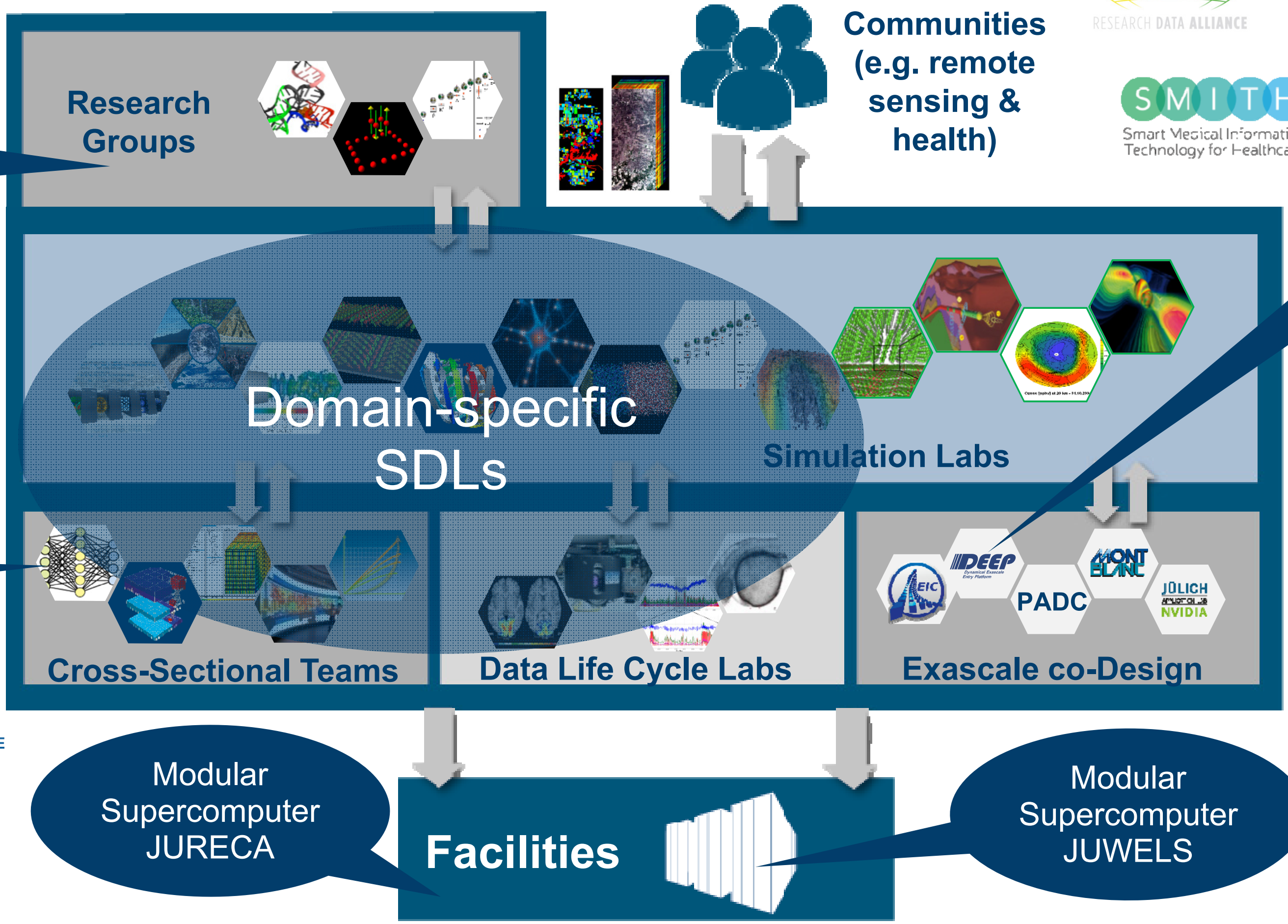
UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

Cross-Sectional Team Deep Learning

HELMHOLTZAI | ARTIFICIAL INTELLIGENCE COOPERATION UNIT





Modular Supercomputer JURECA

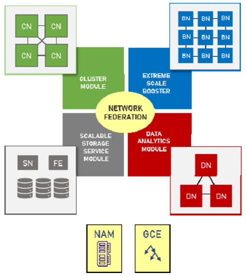
Modular Supercomputer JUWELS

RDA RESEARCH DATA ALLIANCE

Smart Data Innovation Lab

SMITH Smart Medical Information Technology for Healthcare

DEEP-EST EU PROJECT



ON 4 OFF

Landsvirkjun National Power Company of Iceland

SOCCERWATCH BETA

Industry Relations Team

18th February 2021 Page 4

INTRODUCTION TO HIGH PERFORMANCE COMPUTING

Selected Basics of HPC and Relevance in the European & International Landscape

HPC & DATA SCIENCE: A FIELD OF CONSTANT EVOLUTION

Perspective: Floating Point Operations per one second (FLOPS or FLOP/s)

1.000.000 FLOP/s

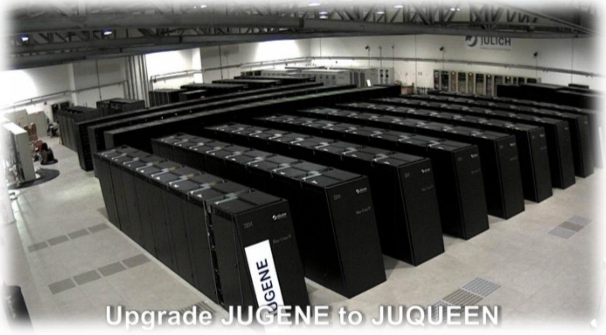
~1984

© Photograph by Rama, Wikimedia Commons

- 1 GigaFlop/s = 10^9 FLOPS
- 1 TeraFlop/s = 10^{12} FLOPS
- 1 PetaFlop/s = 10^{15} FLOPS
- 1 ExaFlop/s = 10^{18} FLOPS

1.000.000.000.000.000 FLOP/s

~295.000 cores~2009 (JUGENE)



>5.900.000.000.000.000 FLOP/s

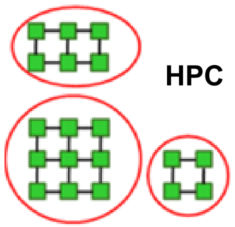
~ 500.000 cores

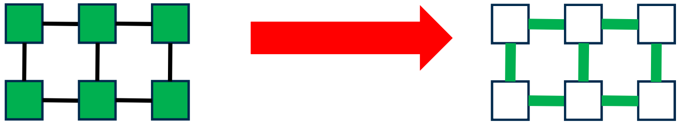
~2013 → end of service in 2018

HIGH PERFORMANCE COMPUTING (HPC)

In Comparison to High Throughput Computing (HTC)

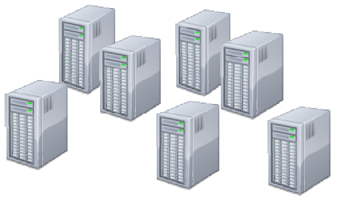
- **High Performance Computing (HPC) is based on computing resources that enable the efficient use of parallel computing techniques through specific support with dedicated hardware such as high performance cpu/core interconnections.**

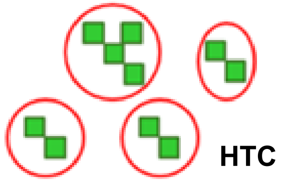




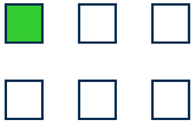
(network connection very important & costly)

- **High Throughput Computing (HTC) is based on commonly available computing resources such as commodity PCs and small clusters that enable the execution of ‘farming jobs’ without providing a high performance interconnection between the cpu/cores.**





(network connection less important)



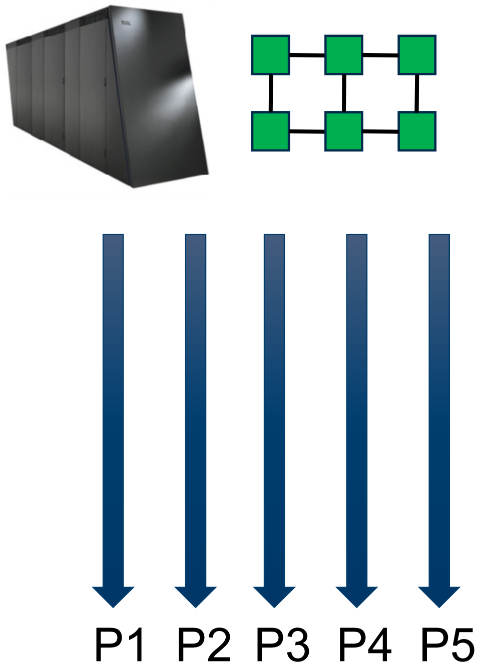
USING PARALLEL COMPUTING ON HPC MACHINES

Concurrency & Computation

- All modern supercomputers depend heavily on parallelism
  - Parallelism can be achieved with many different approaches

- **We speak of parallel computing whenever a number of 'compute elements' (e.g. cores) solve a problem in a cooperative way**

[5] Introduction to High Performance Computing for Scientists and Engineers

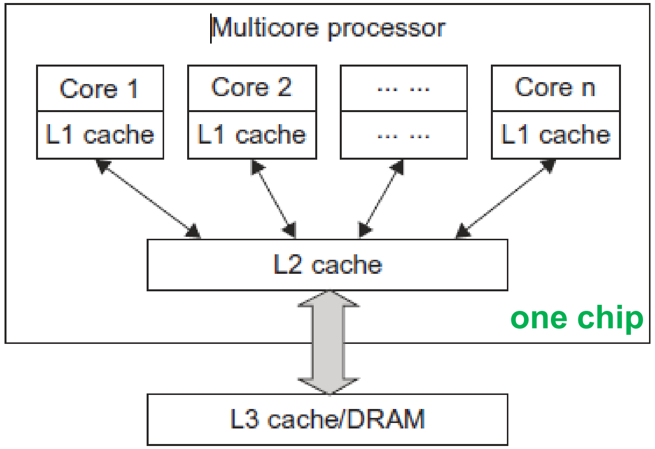


- Often known as 'parallel processing' of some problem space
  - Tackle problems in parallel to enable the 'best performance' possible
  - Includes not only parallel computing, but also parallel input/output (I/O)
- 'The measure of speed' in High Performance Computing matters
  - Common measure for parallel computers established by TOP500 list
  - Based on benchmark for ranking the best 500 computers worldwide

BUILDING BLOCKS OF HPC SYSTEMS

Multi-core CPU Processors

- Significant advances in CPU (or microprocessor chips)
  - Multi-core architecture with dual, quad, six, or n processing cores
  - Processing cores are all on one chip
- Multi-core CPU chip architecture
  - Hierarchy of caches (on/off chip)
  - L1 cache is private to each core; on-chip
  - L2 cache is shared; on-chip
  - L3 cache or Dynamic random access memory (DRAM); off-chip



[22] Distributed & Cloud Computing Book

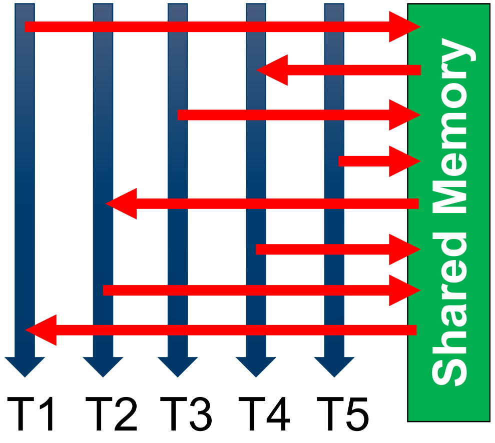
- **Clock-rate for single processors increased from 10 MHz (Intel 286) to 4 GHz (Pentium 4) in 30 years**
- **Clock rate increase with higher 5 GHz unfortunately reached a limit due to power limitations / heat**
- **Multi-core CPU chips have quad, six, or n processing cores on one chip and use cache hierarchies**

SHARED MEMORY PROGRAMMING MODEL

Using OpenMP

- Two varieties of shared-memory systems:
  - Unified Memory Access (UMA)
  - Cache-coherent Nonuniform Memory Access (ccNUMA)
- The Problem of 'Cache Coherence' (in UMA/ccNUMA)
  - Different CPUs use Cache to 'modify same cache values'
  - Consistency between cached data & data in memory must be guaranteed
  - 'Cache coherence protocols' ensure a consistent view of memory

[25] OpenMP API Specification



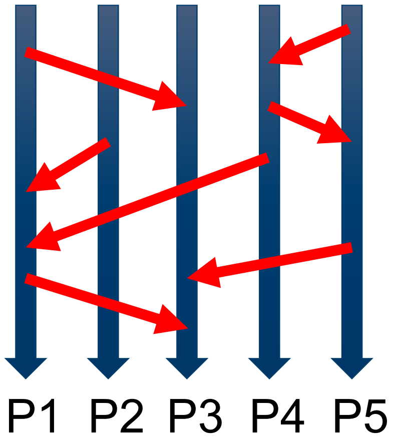
- A shared-memory parallel computer is a system in which a number of CPUs work on a common, shared physical address space
- Shared-memory programming enables immediate access to all data from all processors without explicit communication
- OpenMP is dominant shared-memory programming standard today (v3)
- OpenMP is a set of compiler directives to 'mark parallel regions'

DISTRIBUTED MEMORY PROGRAMMING MODEL

Using Message Passing Interface (MPI)

- Approach
  - No remote memory access on distributed-memory systems
  - Require to 'send messages' back and forth between processes PX
  - Many free Message Passing Interface (MPI) libraries available
  - Programming is tedious & complicated, but most flexible method
- Hybrid Programming
  - Combine Shared memory with Distributed Memory often in practice
  - Harder to program, but enables often more performance (if programmed well)

[26] MPI Standard

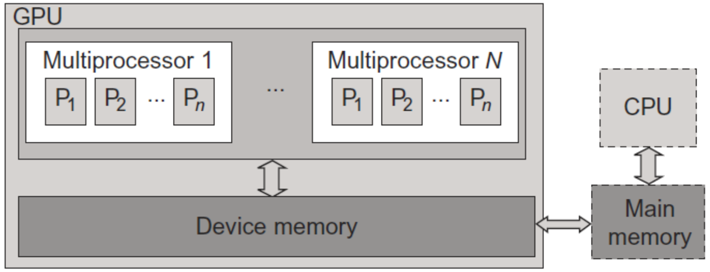


- **A distributed-memory parallel computer establishes a 'system view' where no process can access another process' memory directly**
- **Distributed-memory programming enables explicit message passing as communication between processors**
- **Message Passing Interface (MPI) is dominant distributed-memory programming standard today (available in many different version)**
- **MPI is a standard defined and developed by the MPI Forum**

BUILDING BLOCKS OF HPC SYSTEMS

Many-core GPGPUs

- Use of very many simple cores
  - High throughput computing-oriented architecture
  - Use massive parallelism by executing a lot of concurrent threads slowly
  - Handle an ever increasing amount of multiple instruction threads
  - CPUs instead typically execute a single long thread as fast as possible
- Many-core GPUs are used in large clusters and within massively parallel supercomputers today
  - Named General-Purpose Computing on GPUs (GPGPU)
  - Different programming models emerge

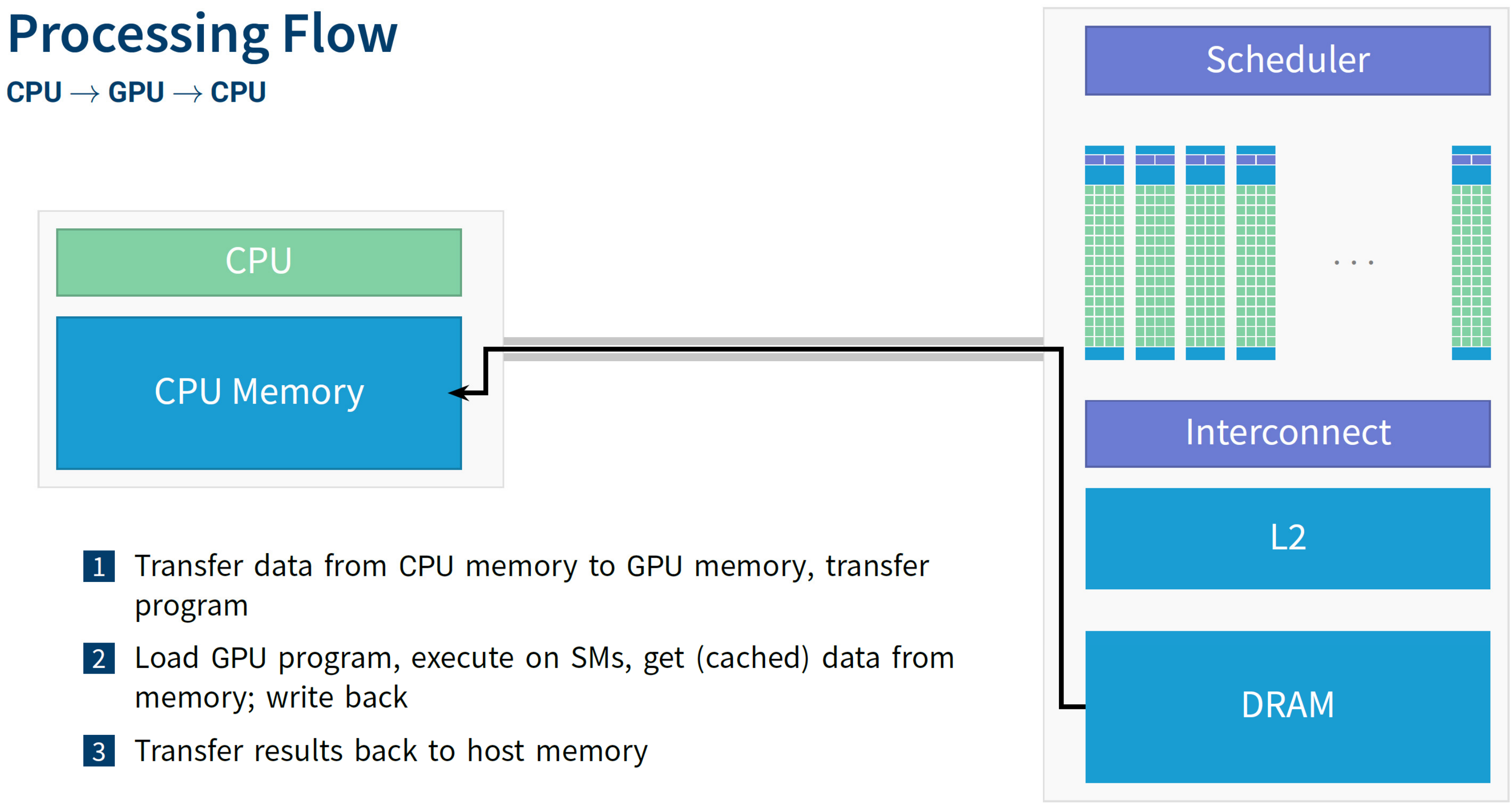


[22] Distributed & Cloud Computing Book

- **Graphics Processing Unit (GPU) is great for data parallelism and task parallelism**
- **Compared to multi-core CPUs, GPUs consist of a many-core architecture with hundreds to even thousands of very simple cores executing threads rather slowly**

GPGPU PROGRAMMING MODEL

Using Host & Device Memory



Processing Flow

CPU → GPU → CPU

1 Transfer data from CPU memory to GPU memory, transfer program

2 Load GPU program, execute on SMs, get (cached) data from memory; write back

3 Transfer results back to host memory

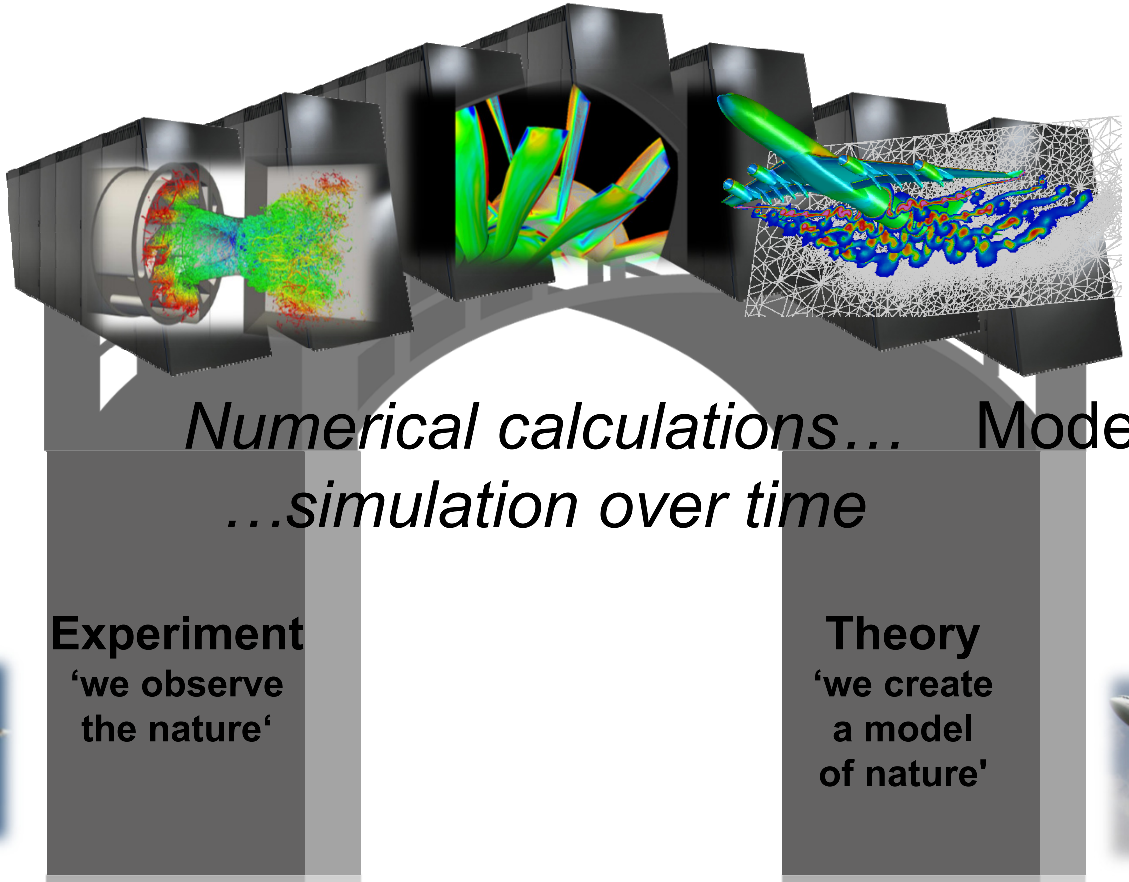
- CPU acceleration means that GPUs accelerate computing due to a massive parallelism with thousands of threads compared to only a few threads used by conventional CPUs
- GPUs are designed to compute large numbers of floating point operations in parallel
- The Processing flow is (a) transfer data from CPU memory to GPU memory; (b) Load GPU program and execute on GPU device using device memory; (c) transfer results back to host memory

[27] JSC GPU Course

SIMULATION SCIENCES APPLICATIONS

Traditional Supercomputing and HPC Impact in Scientific Computing

- Known physical laws
- Numerical methods
- Parallel Computing



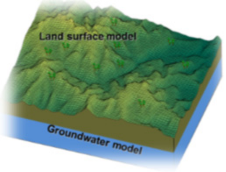
Numerical calculations…

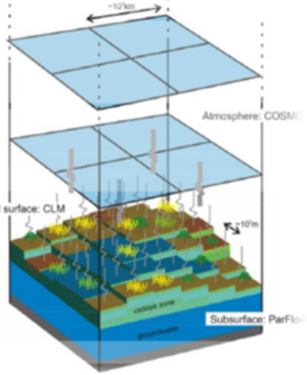
…simulation over time

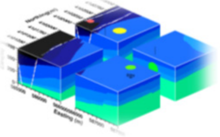
Experiment
‘we observe the nature‘

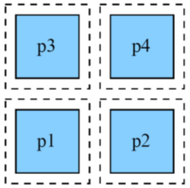
Theory
‘we create a model of nature'

Model









WORLDWIDE HPC ROADMAP TO EXASCALE

Coordinated Activities

🇯🇵

- **Flagship 2020**: Post-K
  - 2020
  - Fujitsu+ARM

🇨🇳

- **TaihuLight**
  - 2020
  - Lenovo+ShenWei/FeiTeng CPU

🇺🇸

- **CORAL**: 2 Exascale machines
  - 2023
  - Intel+Cray and IBM+NVIDIA

🇪🇺

- **H2020 + IPCEI + EuroHPC + EU Cloud initiative**
  - 2022
  - Technology and design not fixed yet

EUROPEAN HPC STRATEGY

Coordinated Activities

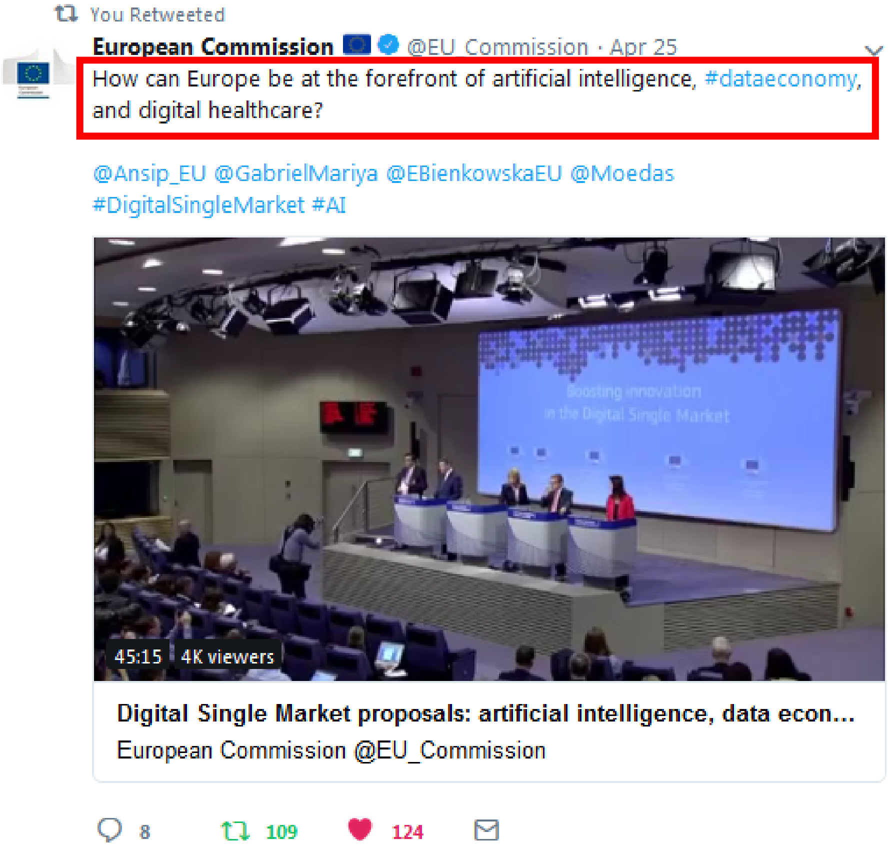
- **PRACE**
  - Computing infrastructures for European Users
  - Operation, support and training
- **ETP4HPC**
  - Industry-driven Roadmap (SRA)
  - Pushing for Extreme Scale Demonstrators
- **EuroHPC**
  - EU-based technology development, eg. processor
  - Pushing for EU-made machine by 2022
- **H2020**
  - Technology (HW+SW) development in Co-design
  - FETHPC + Flagships + Quantum Computing

EUROPEAN UNION & COMMISSION PLANS

Supporting Artificial Intelligence & Supercomputers – Relevance of HPC & AI in Europe

"By supporting strategic projects in frontline areas such as artificial intelligence, supercomputers, cybersecurity or industrial digitisation, and investing in digital skills, the new programme will help to complete the Digital Single Market, a key priority of the Union."

[9] COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE EUROPEAN COUNCIL, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS, EC, 2018, 2nd May 2018



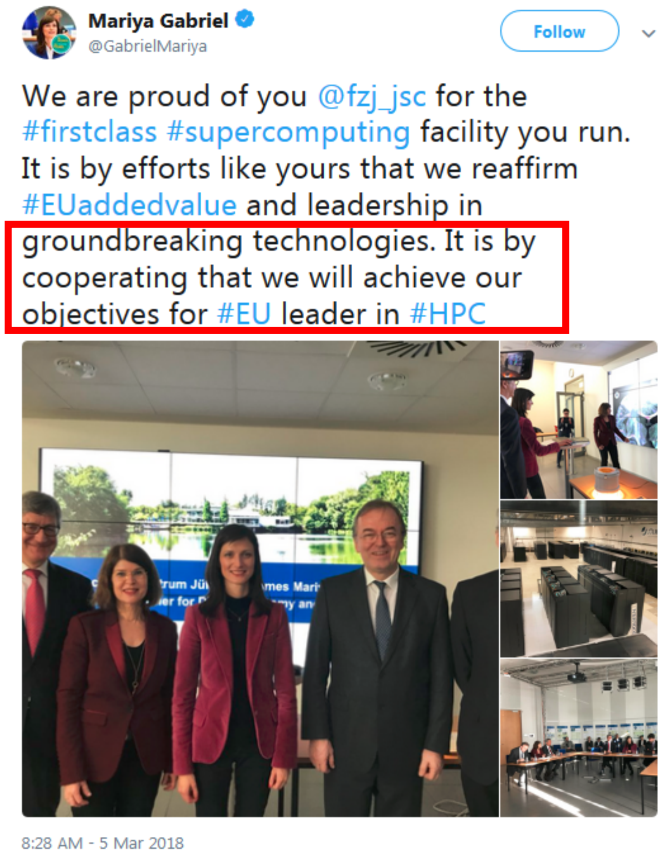
You Retweeted

European Commission @EU_Commission · Apr 25

How can Europe be at the forefront of artificial intelligence, #dataeconomy, and digital healthcare?

@Ansip_EU @GabrielMariya @EBienkowskaEU @Moedas #DigitalSingleMarket #AI

Digital Single Market proposals: artificial intelligence, data econ...
European Commission @EU_Commission



Mariya Gabriel @GabrielMariya

We are proud of you @fzj_jsc for the #firstclass #supercomputing facility you run. It is by efforts like yours that we reaffirm #EUaddedvalue and leadership in groundbreaking technologies. It is by cooperating that we will achieve our objectives for #EU leader in #HPC

8:28 AM - 5 Mar 2018

IMPACTS OF ARTIFICIAL INTELLIGENCE IN HPC

HPC System Design Influence

ARTIFICIAL INTELLIGENCE OVERVIEW

Terminology & Methods

Artificial Intelligence (AI)

A wide area of techniques and tools that enable computers to mimic human behaviour (+ robotics)

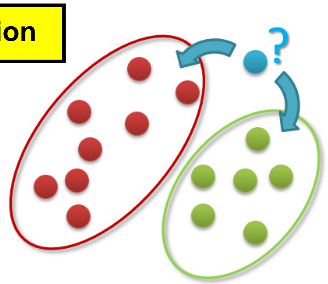
Machine Learning (ML)

Learning from data without explicitly being programmed with common programming languages

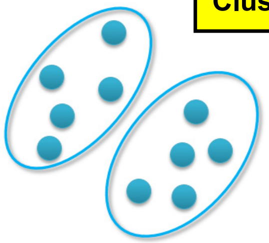
Deep Learning (DL)

Systems with the ability to learn underlying features in data using large neural networks

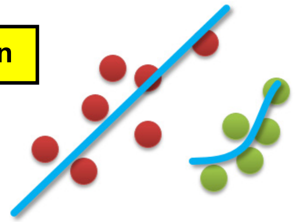
Classification



Clustering

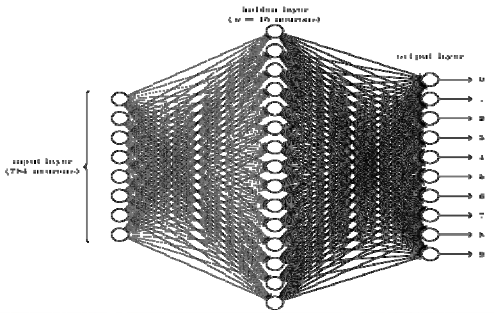


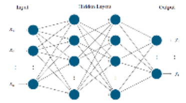
Regression

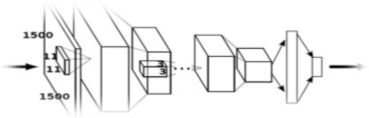


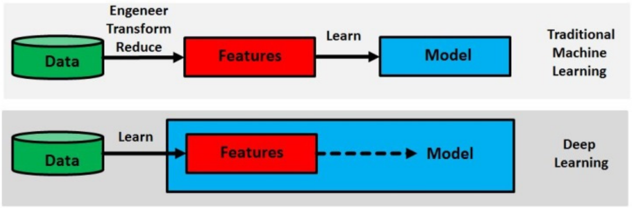
INNOVATIVE DEEP LEARNING TECHNOLOGIES

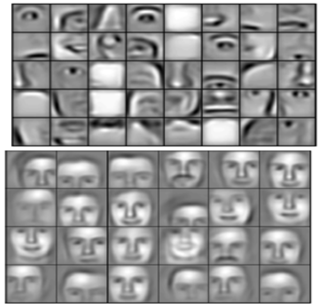
Short Introduction & Role of Cross-Sectional Team Deep Learning @ JSC











[3] M. Riedel, 'Deep Learning - Using a Convolutional Neural Network', Invited YouTube Lecture, six lectures, University of Ghent, 2017

[4] M. Riedel et al., 'Introduction to Deep Learning Models', JSC Tutorial, three days, JSC, 2019

JÜLICH
Forschungszentrum

JÜLICH
SUPERCOMPUTING
CENTRE

[5] H. Lee et al., 'Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations'

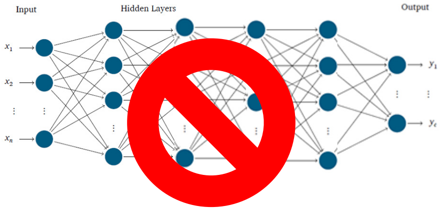
Cross-Sectional Team Deep Learning

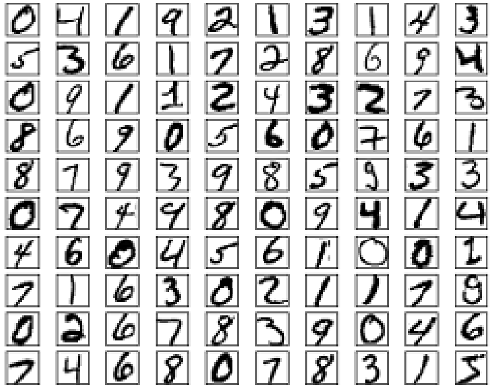
- **Provide deep learning tools that work with HPC machines (e.g. Python/Keras/Tensorflow)**
- **Advance deep learning applications and research on HPC prototypes (e.g. DEEP-EST, SMITH, etc.)**
- **Engage with industry (industrial relations team) & support SMEs (e.g. Soccerwatch, ON4OFF)**
- **Offer tutorials & application enabling support for commercial & scientific users (e.g. YouTube)**
- **Cooperate in a artificial intelligence network across Helmholtz Association (e.g. HAICU)**

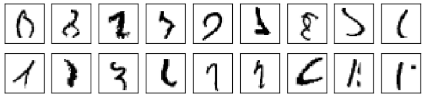
DEEP LEARNING TECHNIQUE EXAMPLE

Convolutional Neural Network (CNN) for Image Analysis

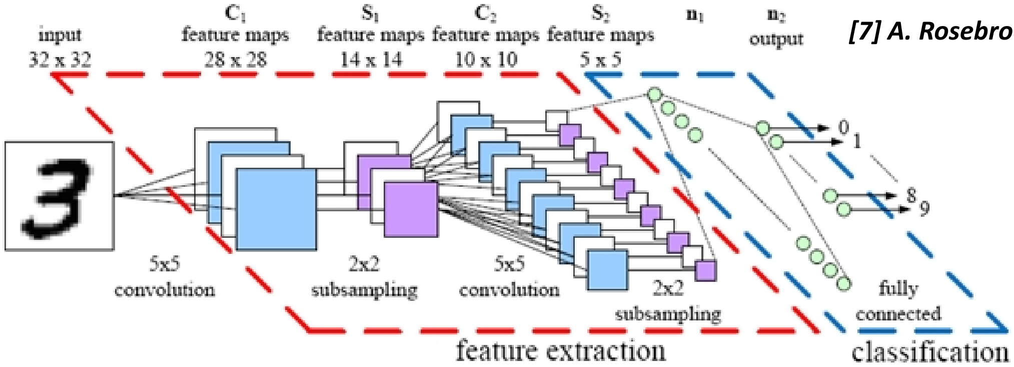
[6] Neural Network 3D Simulation







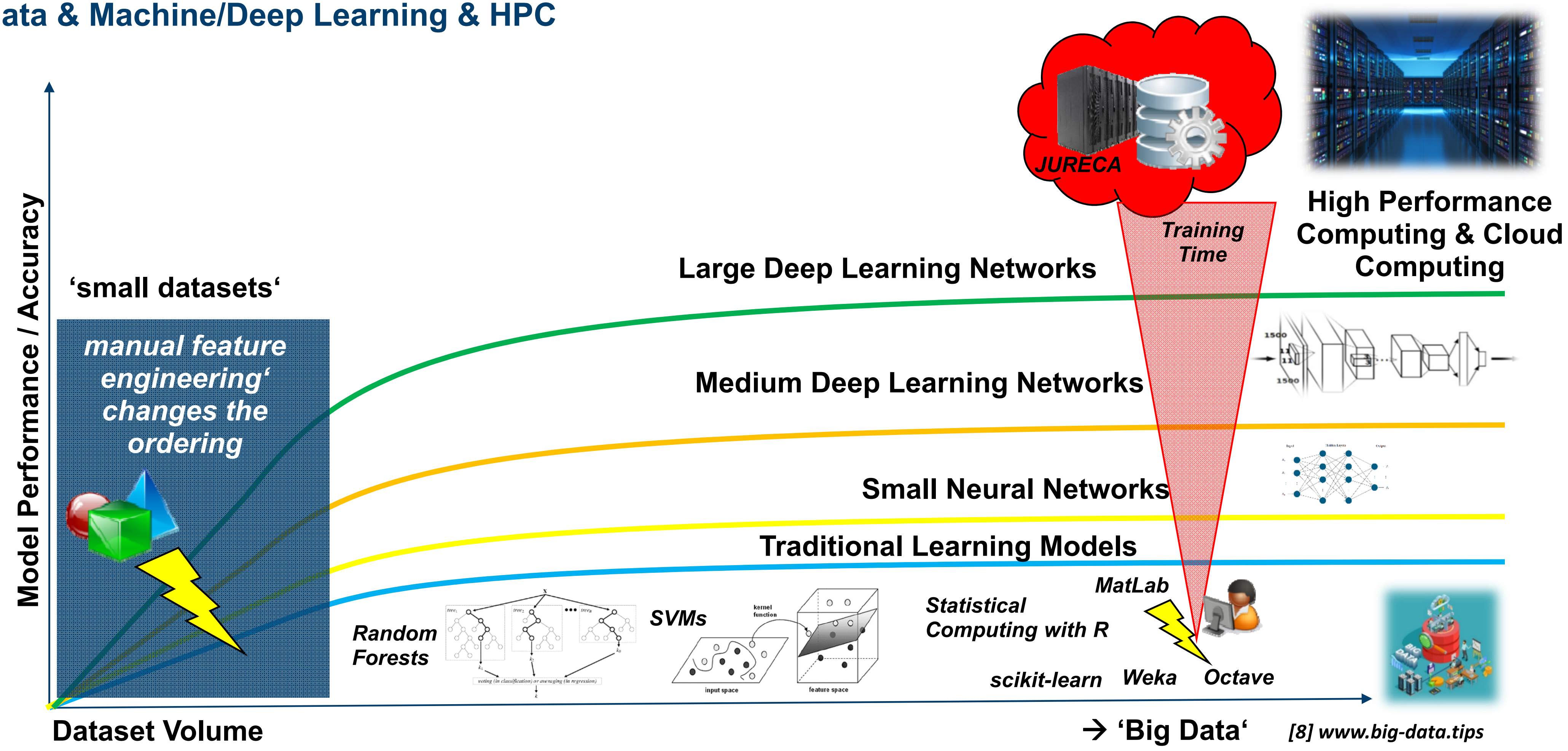
- **Innovation via specific layers and architecture types**



[7] A. Rosebrock

ARTIFICIAL INTELLIGENCE – COMPLEX RELATIONSHIPS

Big Data & Machine/Deep Learning & HPC


JURECA
High Performance Computing & Cloud Computing
Training Time
Model Performance / Accuracy
'small datasets'
manual feature engineering' changes the ordering
Large Deep Learning Networks
Medium Deep Learning Networks
Small Neural Networks
Traditional Learning Models
Random Forests
SVMs
kernel function
input space
feature space
Statistical Computing with R
MatLab
scikit-learn
Weka
Octave
Dataset Volume
→ 'Big Data'

[8] www.big-data.tips

DEEP LEARNING STARTUP EXAMPLE

Understanding the Different Factors that all Combined Provide new Chances – NOW

1952 Stochastic Gradient Descent
- Solving optimization problems

1958 Perceptron Learning Model
- Learning weights

⋮

1985 'Backpropagation of Error' approch in learning
- Artificial Neural Networks

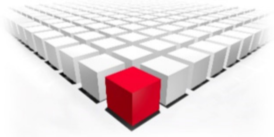
1995 Deep Convolutional Neural Networks
- Significant improvements in image analysis

⋮

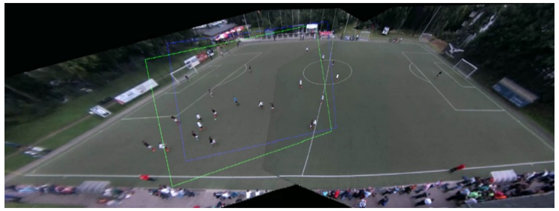
Impact in AI & HPC in industry & science

Big Data
- Large datasets
- Easy access
- More storage for less cost

Hardware
- More memory
- Graphical Processing Units (GPUs)
- HPC & parallel systems



Software
- Scalable data science tools
- New learning models
- Open Source & free software packages



SOCCERWATCH™ K2

DIE VOLLAUTOMATISCHE KAMERA FÜR DEINEN VEREIN

[10] NVIDIA

Keras

[11] Keras

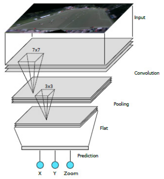
TensorFlow

[12] TensorFlow

UNIVERSITÄT DUISBURG ESSEN

eXIST Existenzgründungen aus der Wissenschaft

[13] soccerwatch.tv

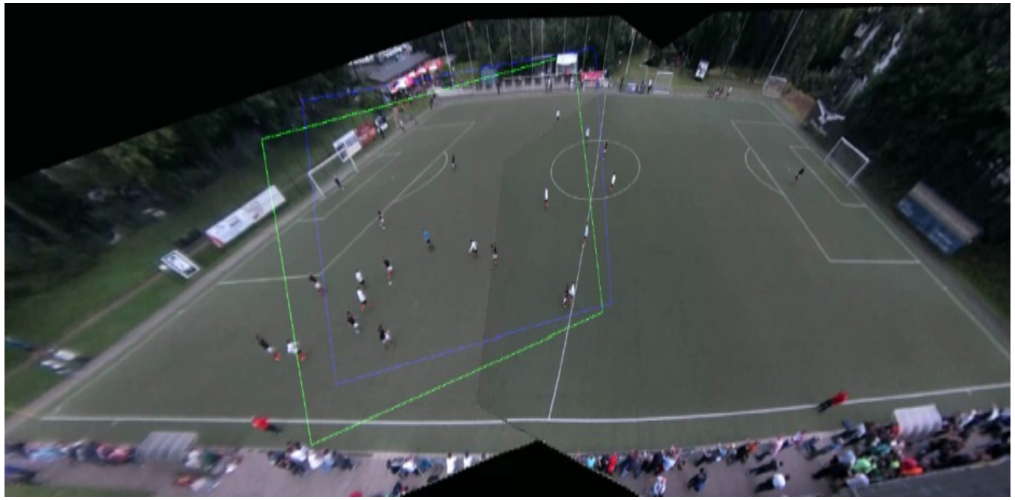


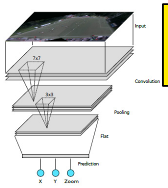
Combination: Start-up Example of my research group

[14] C. Bodenstein & M. Riedel et al., Automated Soccer Scence Tracking using Deep Neural Networks

IMPACTS OF ARTIFICIAL INTELLIGENCE IN APPLICATIONS

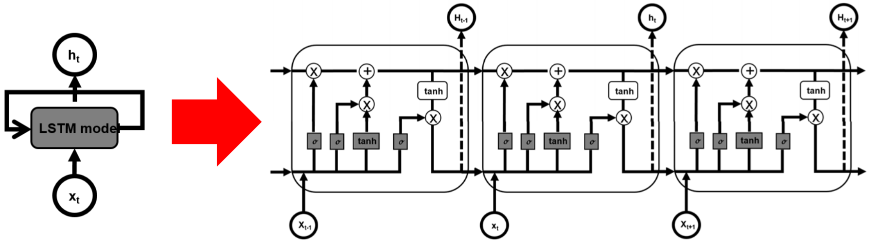
Success in Image & Time Series Analysis Examples





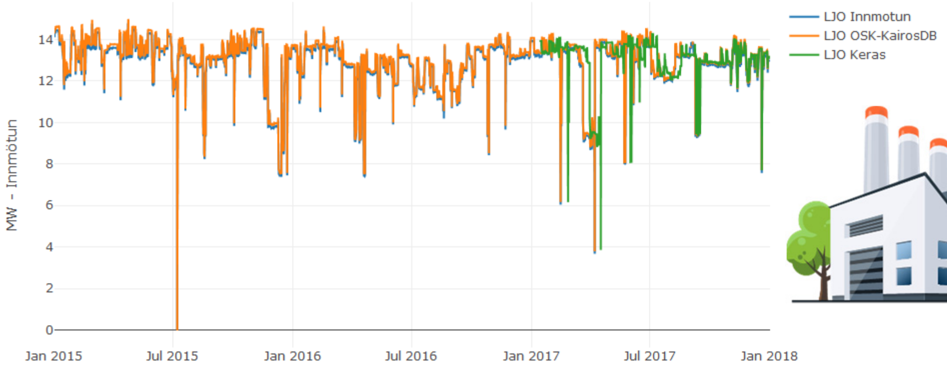
Using Deep Learning to enable automatic camera tracking of soccer

[14] *C. Bodenstein, M. Goetz and M. Riedel et al., NIC Symposium, 2016*



Using Long Short-Term Memory (LSTMs) with electric power production time series data

Landsvirkjun
National Power Company of Iceland

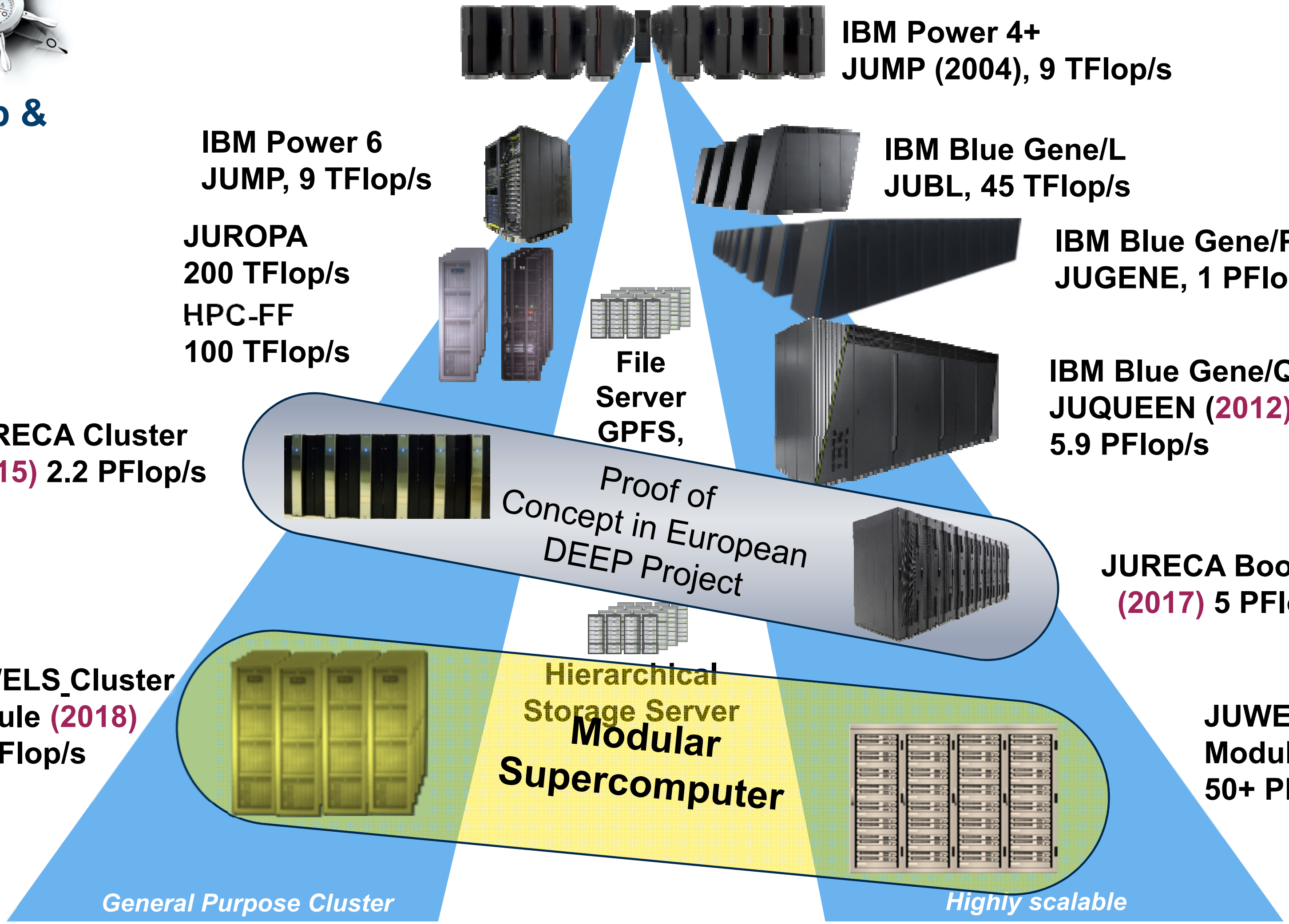


MODULAR SUPERCOMPUTING ARCHITECTURE CO-DESIGN

Shape the HPC Systems of the Future & towards Exascale

JSC

HPC Roadmap & Key Vendors



IBM Power 4+
JUMP (2004), 9 TFlop/s

IBM Power 6
JUMP, 9 TFlop/s

IBM Blue Gene/L
JUBL, 45 TFlop/s

IBM Blue Gene/P
JUGENE, 1 PFlop/s

JUROPA
200 TFlop/s
HPC-FF
100 TFlop/s

File
Server
GPFS,

IBM Blue Gene/Q
JUQUEEN (2012)
5.9 PFlop/s

JURECA Cluster
(2015) 2.2 PFlop/s

Proof of
Concept in European
DEEP Project

JURECA Booster
(2017) 5 PFlop/s

JUWELS Cluster
Module (2018)
12 PFlop/s

Hierarchical
Storage Server

Modular
Supercomputer

JUWELS Scalable
Module (2019/20)
50+ PFlop/s

General Purpose Cluster

Highly scalable

DEEP SERIES OF PROJECTS

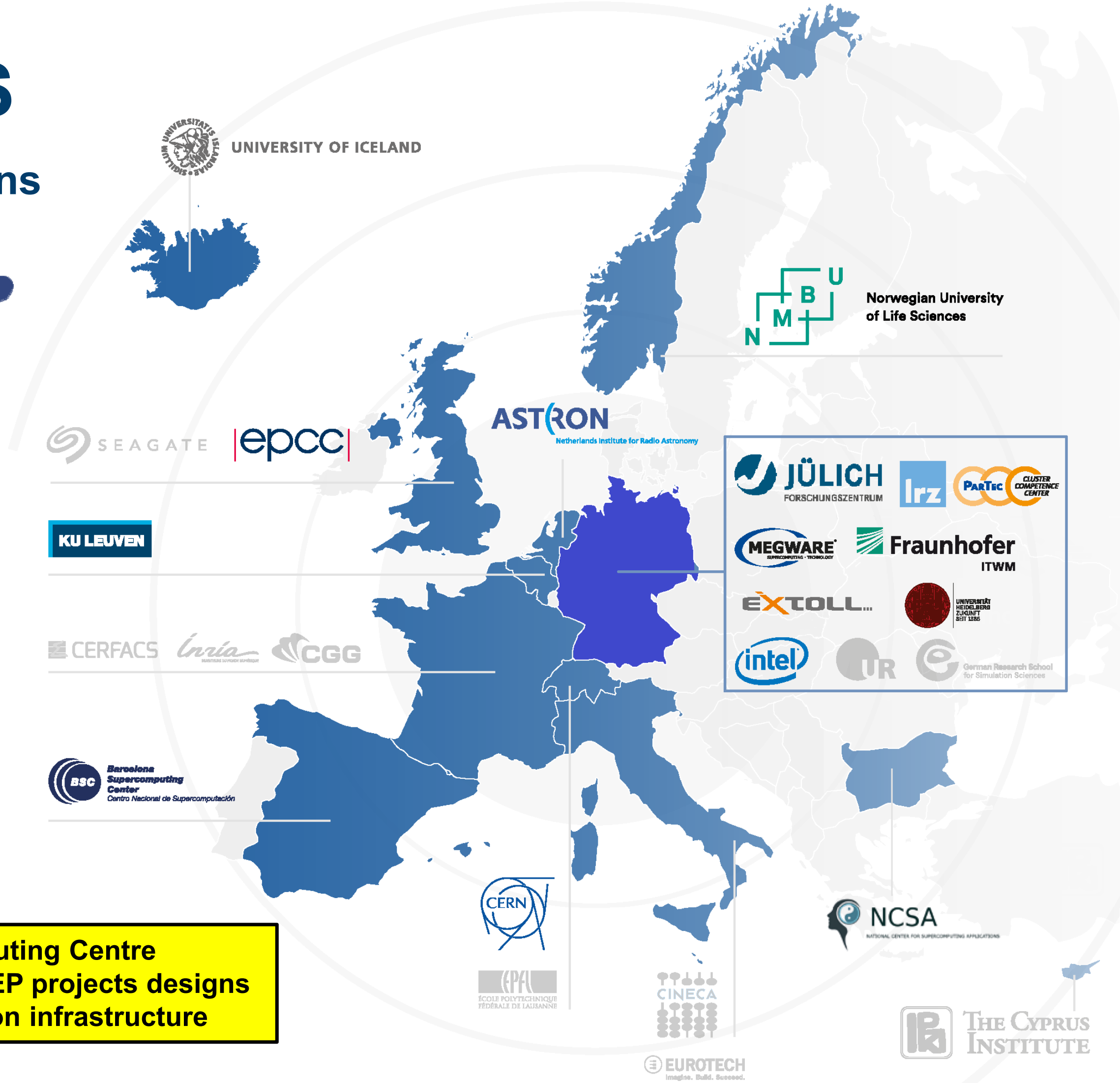
EU Projects Driven by Co-Design of HPC Applications

DEEP Projects

- 3 EU Exascale projects
  DEEP, DEEP-ER, DEEP-EST
- 27 partners
  Coordinated by JSC
- EU-funding: 30 M€
  JSC-part > 5,3 M€
- Nov 2011 – Dec 2020

Strong collaboration with our industry partners Intel, Extoll & Megware

Juelich Supercomputing Centre implements the DEEP projects designs in its HPC production infrastructure

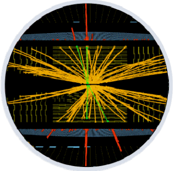


UNIVERSITY OF ICELAND
NMBU Norwegian University of Life Sciences
SEAGATE
epcc
ASTRON Netherlands Institute for Radio Astronomy
KU LEUVEN
JÜLICH FORSCHUNGSZENTRUM
lrz
ParTec Cluster Competence Center
MEGWARE
Fraunhofer ITWM
EXTOLL
Universität Heidelberg
intel
UR
German Research School for Simulation Sciences
CERFACS
Inria
CGG
Barcelona Supercomputing Center Centro Nacional de Supercomputación
CERN
EPFL École Polytechnique Fédérale de Lausanne
CINECA
EUROTECH
NCSA
The Cyprus Institute

[15] DEEP Projects Web Page

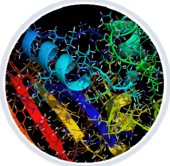
IMPACTS OF ARTIFICIAL INTELLIGENCE IN HPC DESIGN

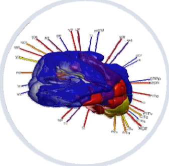
Co-Design via Requirements from Machine/Deep Learning Applications & Innovative Simulation Sciences


High Energy Physics

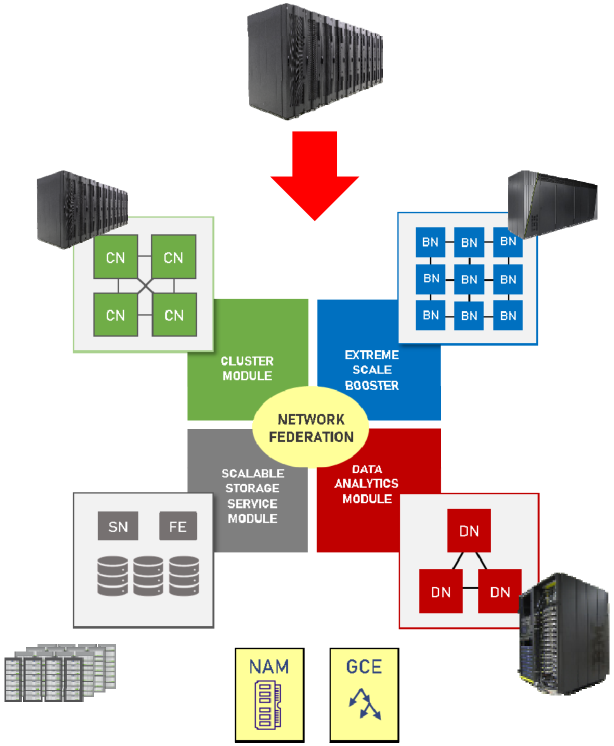
Earth Science

Space Weather

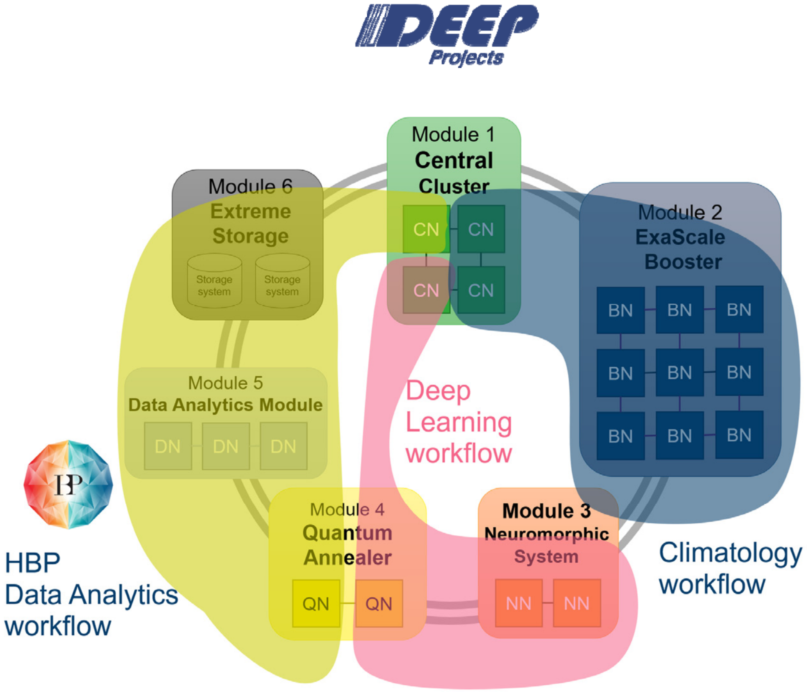

Molecular Dynamics


Neuroscience

Radio Astronomy



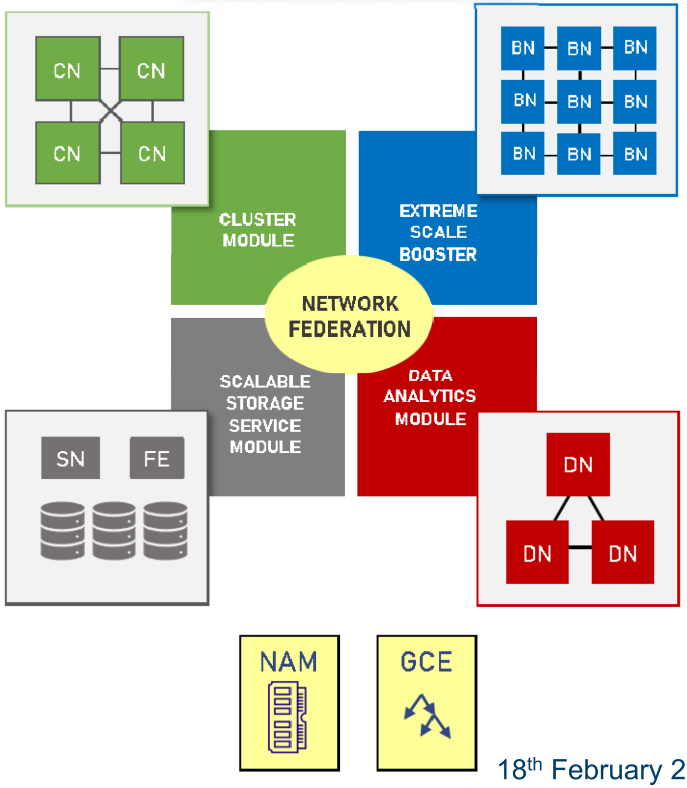
The modular supercomputing architecture (MSA) enables a flexible HPC system design co-designed by the need of different application workloads



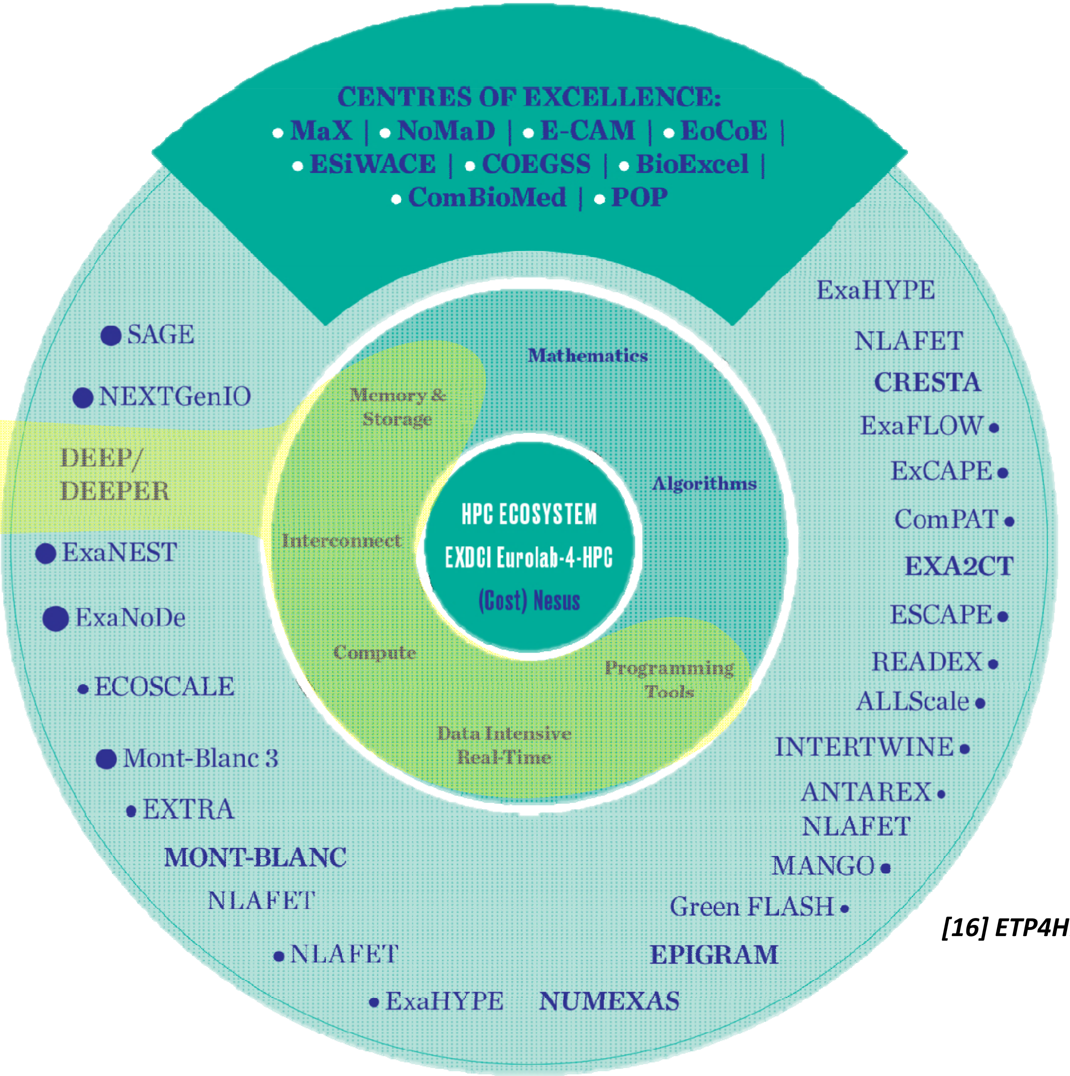
[15] DEEP Projects Web Page

EU HPC PROJECTS OVERVIEW

DEEP-EST Modular Supercomputing Architecture



DEEP Extreme Scale Technologies



CENTRES OF EXCELLENCE:
• MaX | • NoMaD | • E-CAM | • EoCoE |
• ESiWACE | • COEGSS | • BioExcel |
• ComBioMed | • POP

HPC ECOSYSTEM
EXDCI Eurolab-4-HPC
(Cost) Nesus

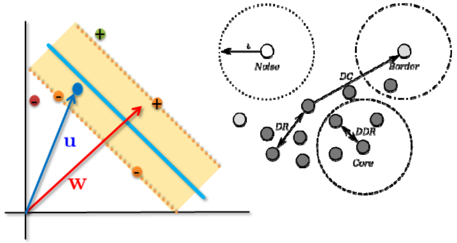
● SAGE
● NEXTGenIO
DEEP/ DEEPER
● ExaNEST
● ExaNoDe
• ECOSCALE
● Mont-Blanc 3
• EXTRA
MONT-BLANC
NLAFET
• NLAFET
• ExaHYPE
NUMEXAS
EPIGRAM
Green FLASH •
MANGO •
NLAFET
ANTAREX •
INTERTWINE •
ALLScale •
READEX •
ESCAPE •
EXA2CT
ComPAT •
ExCAPE •
ExaFLOW •
CRESTA
NLAFET
ExaHYPE

[16] ETP4HPC

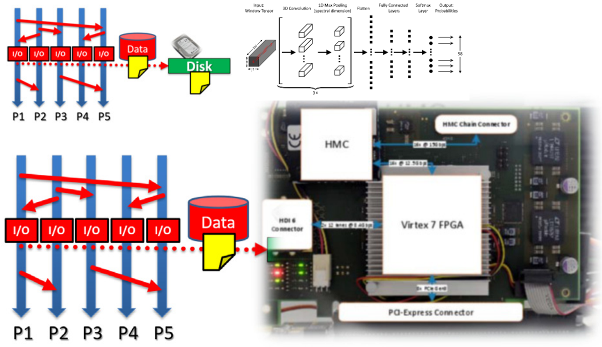
INNOVATIVE HPC HARDWARE VIA CO-DESIGN FOR AI

Co-Design of Innovative HPC Memory Designs and GPU/CPU Communications in Modular Supercomputing

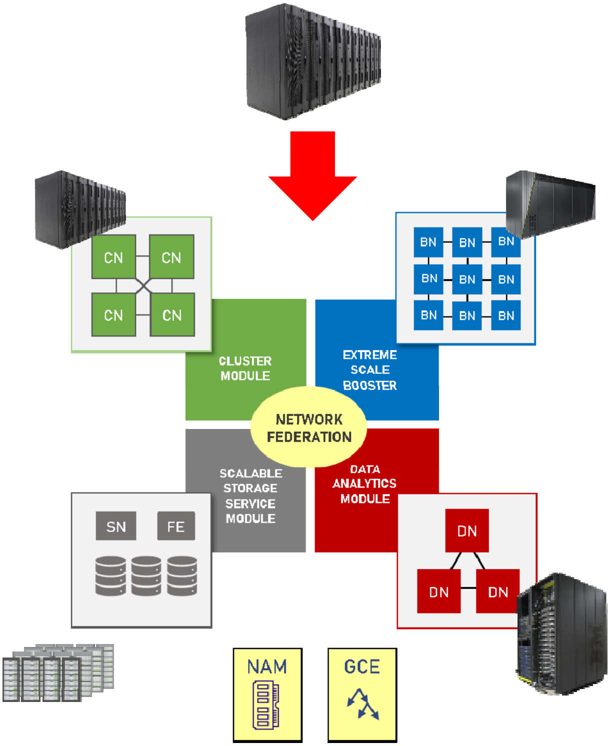
DEEP Projects



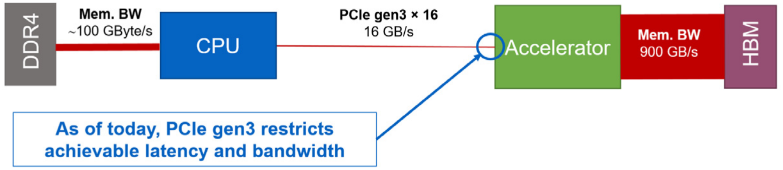
Explore Network Attached Memory (NAM)



[17] E. Erlingsson, M. Riedel et al., IEEE MIPRO Conference, 2018

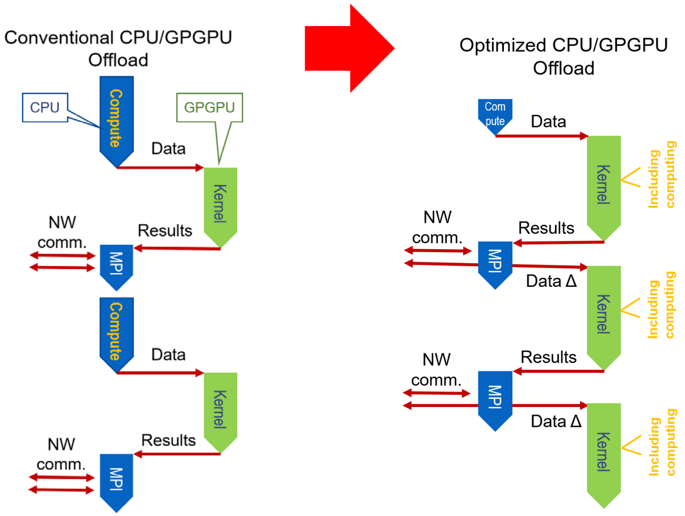


The modular supercomputing architecture (MSA) enables a flexible HPC system design co-designed by the need of different application workloads



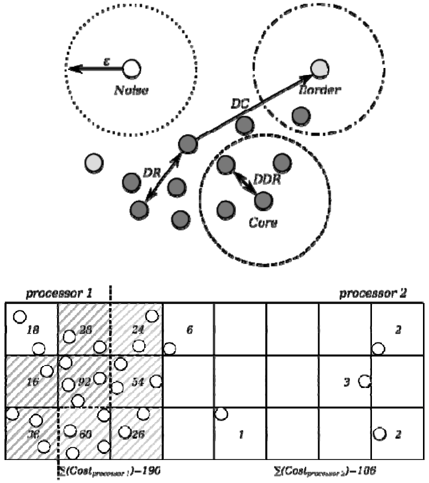
As of today, PCIe gen3 restricts achievable latency and bandwidth

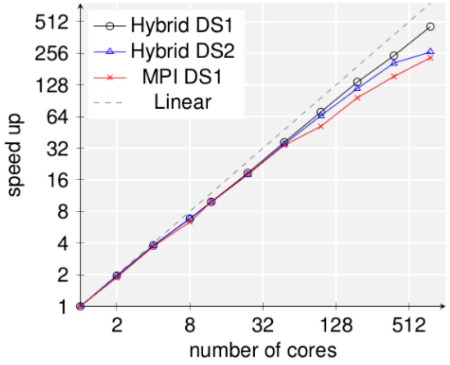
Explore more scalability with NVIDIA GPUDirect beyond one node compared to NVIDIA NVLink/NVSwitch 'islands'

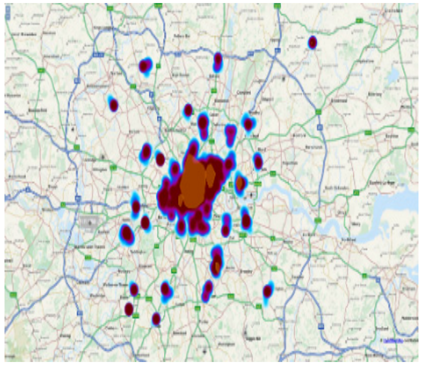


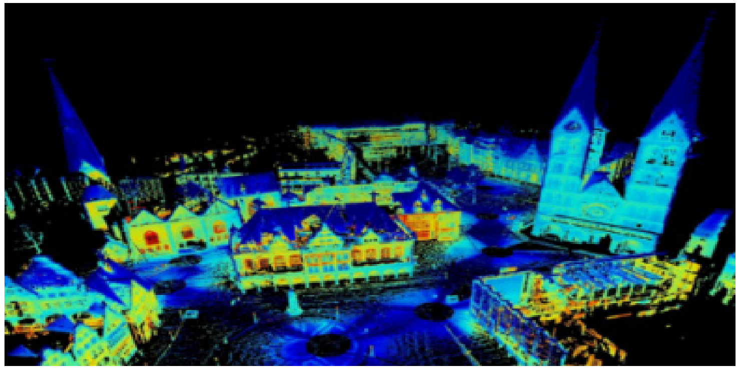
PARALLEL & SCALABLE ALGORITHM DEVELOPMENT

Example of a Co-Design Application using Modular Supercomputing Architecture Concepts

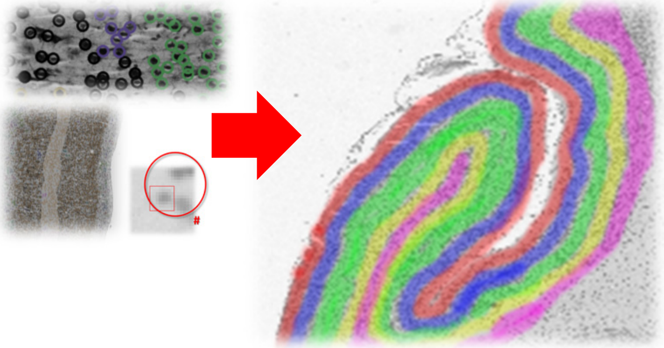


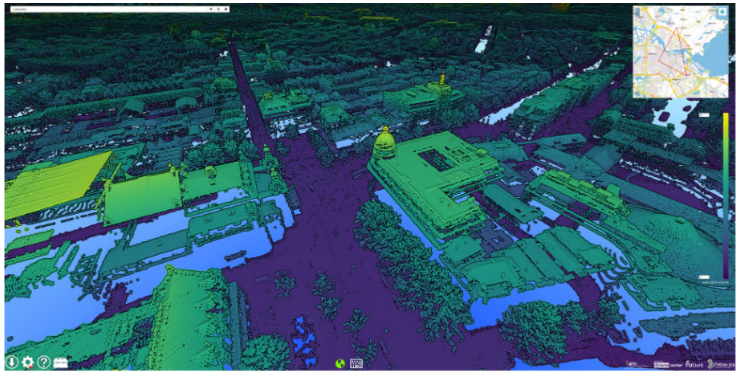






Parallel & Scalable Clustering with DBSCAN

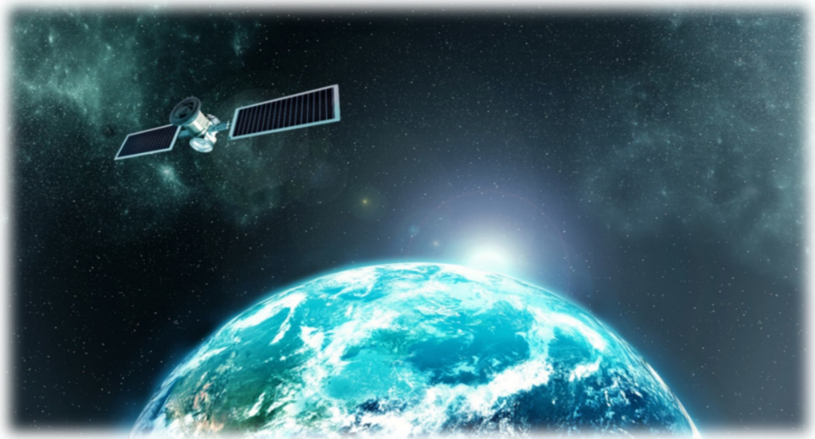


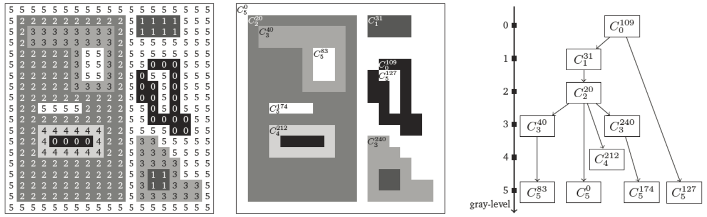


[18] M. Goetz and M. Riedel et al, Proceedings IEEE Supercomputing Conference, 2015

PARALLEL & SCALABLE ALGORITHM DEVELOPMENT

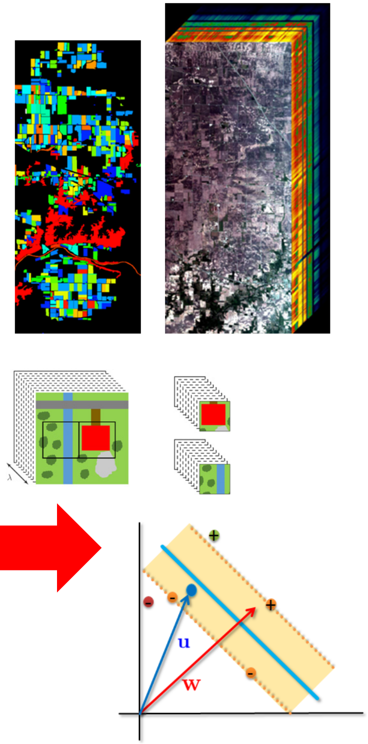
Parallelizing Feature Engineering & Machine Learning Algorithms in Remote Sensing Applications





Parallel & Scalable Feature Engineering with Component Trees

[19] M. Goetz and M. Riedel et al., Journal of Transactions on Parallel and Distributed Systems, 2018



Parallel & Scalable Classification with SVMs based on Message Passing Interface (MPI) using HPC resources

Scenario ‘pre-processed data‘, 10xCV **serial**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	48.90 (18.81)	65.01 (19.57)	73.21 (20.11)	75.55 (22.53)	74.42 (21.21)
4	57.53 (16.82)	70.74 (13.94)	75.94 (13.53)	76.04 (14.04)	74.06 (15.55)
8	64.18 (18.30)	74.45 (15.04)	**77.00** (14.41)	75.78 (14.65)	74.58 (14.92)
16	68.37 (23.21)	76.20 (21.88)	76.51 (20.69)	75.32 (19.60)	74.72 (19.66)
32	70.17 (34.45)	75.48 (34.76)	74.88 (34.05)	74.08 (34.03)	73.84 (38.78)

Scenario ‘pre-processed data‘, 10xCV **parallel**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	75.26 (1.02)	65.12 (1.03)	73.18 (1.33)	75.76 (2.35)	74.53 (4.40)
4	57.60 (1.03)	70.88 (1.02)	75.87 (1.03)	76.01 (1.33)	74.06 (2.35)
8	64.17 (1.02)	74.52 (1.03)	**77.02 (1.02)**	75.79 (1.04)	74.42 (1.34)
16	68.57 (1.33)	76.07 (1.33)	76.40 (1.34)	75.26 (1.05)	74.53 (1.34)
32	70.21 (1.33)	75.38 (1.34)	74.69 (1.34)	73.91 (1.47)	73.73 (1.33)

First Result: best parameter set from 14.41 min to 1.02 min
Second Result: all parameter sets from ~9 hours to ~35 min

[20] G. Cavallaro and M. Riedel et al., Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 2015

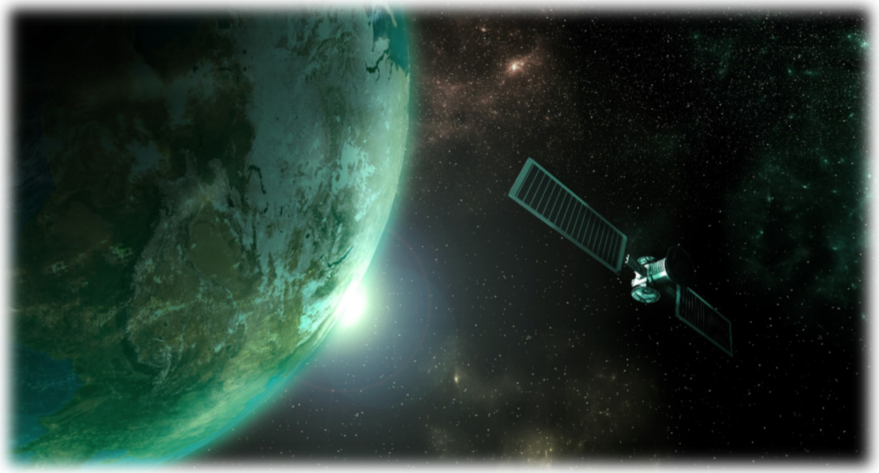
➢ **Appendix offers details on understanding Support Vector Machines (SVMs) & Kernel Methods with a geometric SVM interpretation**

DISTRIBUTED DEEP LEARNING

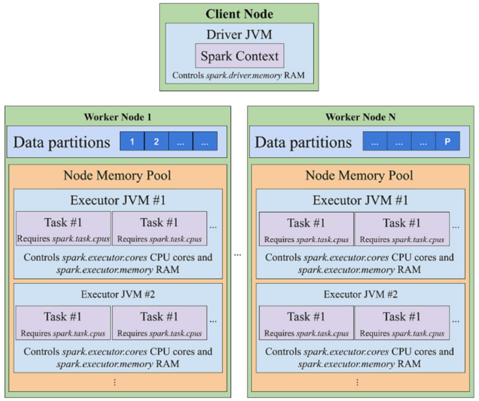
From Apache Spark to Horovod using the Message Passing Interface (MPI) on HPC

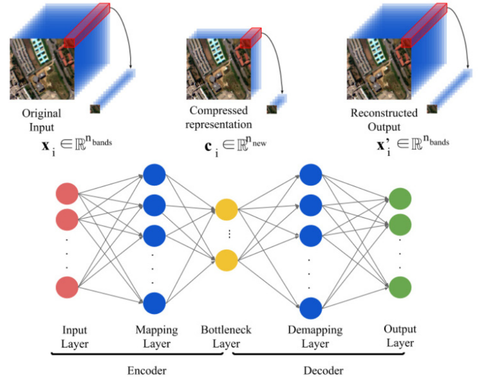
DISTRIBUTED DEEP LEARNING WITH AUTO-ENCODERS

Using Cloud Computing and Auto-Encoder Neural Networks for Remote Sensing Applications



Performing parallel computing with Apache Spark across different worker nodes

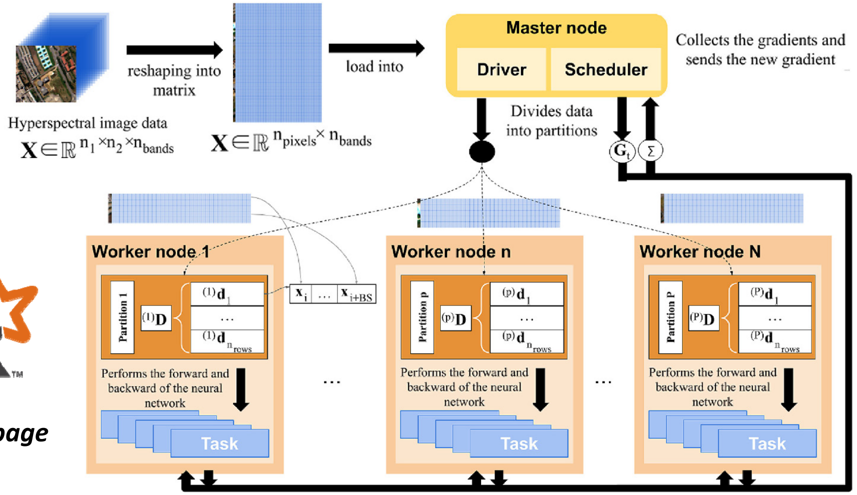

Client Node
Driver JVM
Spark Context
Controls *spark.driver.memory* RAM
Worker Node 1
Data partitions
1
2
...
...
Node Memory Pool
Executor JVM #1
Task #1
Requires *spark.task.cpus*
Task #1
Requires *spark.task.cpus*
...
Controls *spark.executor.cores* CPU cores and *spark.executor.memory* RAM
Executor JVM #2
Task #1
Requires *spark.task.cpus*
Task #1
Requires *spark.task.cpus*
...
Controls *spark.executor.cores* CPU cores and *spark.executor.memory* RAM
...
Worker Node N
Data partitions
...
...
...
P
Node Memory Pool
Executor JVM #1
Task #1
Requires *spark.task.cpus*
Task #1
Requires *spark.task.cpus*
...
Controls *spark.executor.cores* CPU cores and *spark.executor.memory* RAM
Executor JVM #2
Task #1
Requires *spark.task.cpus*
Task #1
Requires *spark.task.cpus*
...
Controls *spark.executor.cores* CPU cores and *spark.executor.memory* RAM


Original Input
$x_i \in \mathbb{R}^{n_{bands}}$
Compressed representation
$c_i \in \mathbb{R}^{n_{new}}$
Reconstructed Output
$x'_i \in \mathbb{R}^{n_{bands}}$
Input Layer
Mapping Layer
Bottleneck Layer
Demapping Layer
Output Layer
Encoder
Decoder

[24] J. Haut, G. Cavallaro and M. Riedel et al., IEEE Transactions on Geoscience and Remote Sensing, 2019

Using Autoencoder deep neural networks with Cloud computing

[23] Apache Spark Web page


reshaping into matrix
Hyperspectral image data
$X \in \mathbb{R}^{n_1 \times n_2 \times n_{bands}}$
load into
$X \in \mathbb{R}^{n_{pixels} \times n_{bands}}$
Master node
Driver
Scheduler
Collects the gradients and sends the new gradient
Divides data into partitions
G_t
Σ
Worker node 1
Partition 1
$^{(1)}D$
$^{(1)}d_1$
...
$^{(1)}d_{n_{rows}}$
x_i
...
x_{i+BS}
Performs the forward and backward of the neural network
Task
...
Worker node n
Partition p
$^{(p)}D$
$^{(p)}d_1$
...
$^{(p)}d_{n_{rows}}$
Performs the forward and backward of the neural network
Task
...
Worker node N
Partition P
$^{(P)}D$
$^{(P)}d_1$
...
$^{(P)}d_{n_{rows}}$
Performs the forward and backward of the neural network
Task

DISTRIBUTED DEEP LEARNING TRAINING ON IMAGENET

Using a Standard Deep Learning Architecture for Image Classification

- Dataset: ImageNet
  - Total number of images: 14.197.122
  - Images with bounding

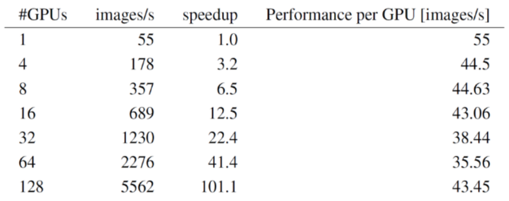
Numbers in brackets: (the number of synsets in the subtree).
ImageNet 2011 Fall Release (32326)
plant, flora, plant life (4486)
geological formation, formation (1
natural object (1112)
sport, athletics (176)
artifact, artefact (10504)
instrumentality, instrumentatior
device (2760)
musical instrument, inst
acoustic device (27)
Treemap Visualization
Images of the Synset
Downloads

(huge collection of images with high level categories)

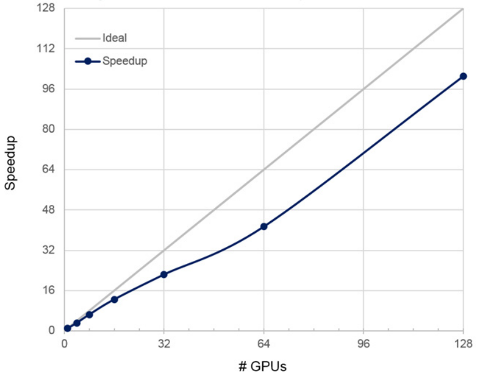
- **Open source tool Horovod enables distributed deep learning with TensorFlow / Keras**
- **Machine & Deep Learning: speed-up is just secondary goal after 1st goal accuracy**
- **Speed-up & parallelization good for faster hyperparameter tuning, training, inference**
- **Third goal is to avoid much feature engineering through ‘feature learning‘**

(ImageNet as a benchmark in deep learning community)

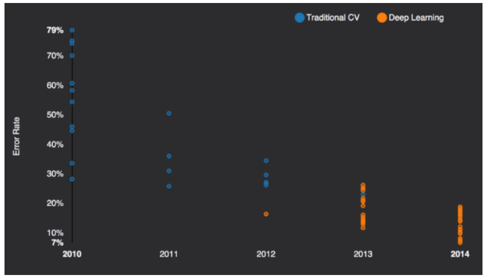
(setup 1.2 Mio Images 224x224 pixels: TensorFlow 1.4, Python 2.7, CUDA 8, cuDNN 6, Horovod 0.11.2, MVAPICH-2.2-GDR on JURECA K80 GPUs)



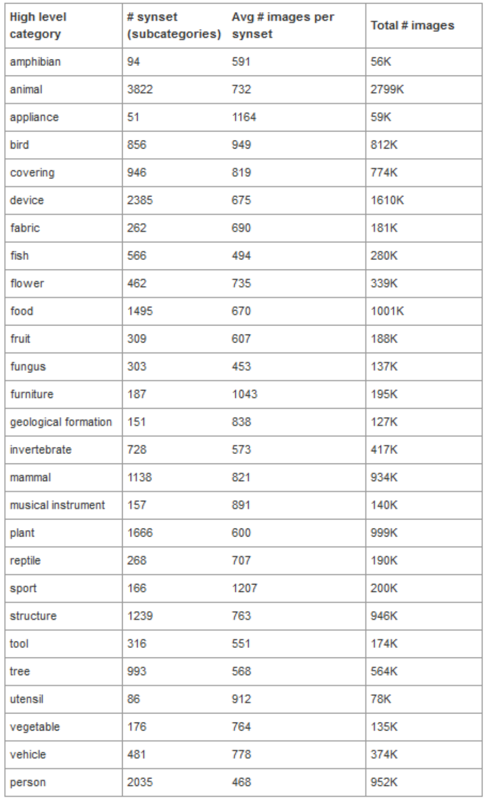
#GPUs	images/s	speedup	Performance per GPU [images/s]
1	55	1.0	55
4	178	3.2	44.5
8	357	6.5	44.63
16	689	12.5	43.06
32	1230	22.4	38.44
64	2276	41.4	35.56
128	5562	101.1	43.45



[30] Horovod



[34] J. Dean et al., ‘Large-Scale Deep Learning’

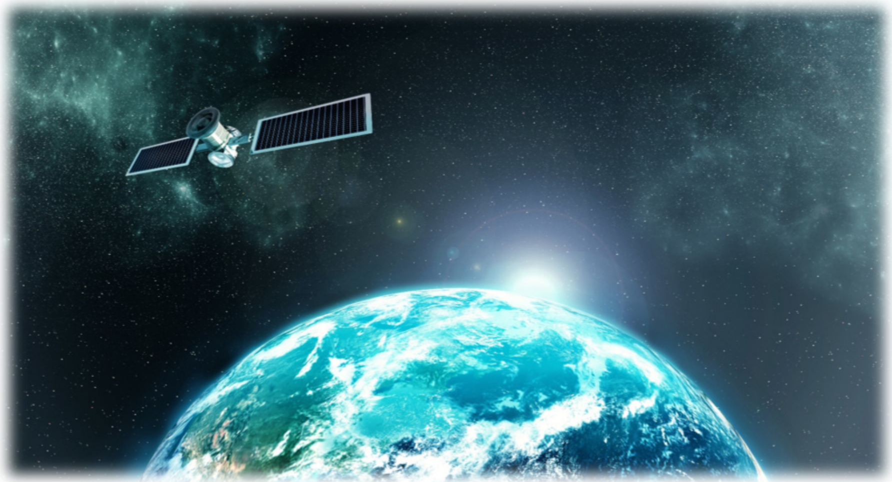


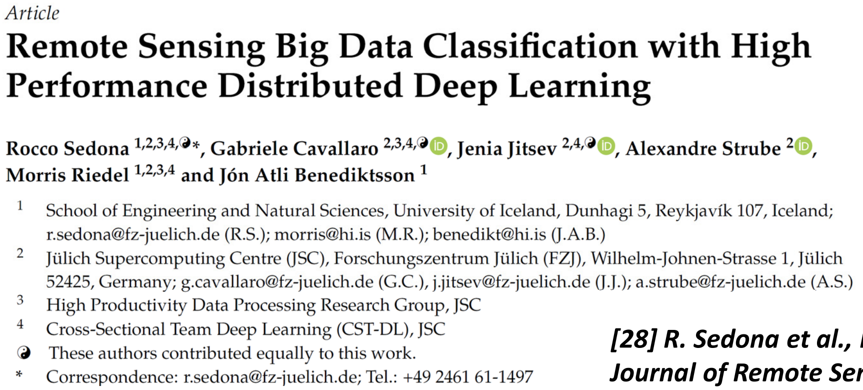
High level category	# synset (subcategories)	Avg # images per synset	Total # images
amphibian	94	591	56K
animal	3822	732	2799K
appliance	51	1164	59K
bird	856	949	812K
covering	946	819	774K
device	2385	675	1610K
fabric	262	690	181K
fish	566	494	280K
flower	462	735	339K
food	1495	670	1001K
fruit	309	607	188K
fungus	303	453	137K
furniture	187	1043	195K
geological formation	151	838	127K
invertebrate	728	573	417K
mammal	1138	821	934K
musical instrument	157	891	140K
plant	1666	600	999K
reptile	268	707	190K
sport	166	1207	200K
structure	1239	763	946K
tool	316	551	174K
tree	993	568	564K
utensil	86	912	78K
vegetable	176	764	135K
vehicle	481	778	374K
person	2035	468	952K

[35] ImageNet Web page

DISTRIBUTED DEEP LEARNING WITH RESNET-50

Tune a ‘standard architecture‘ for Remote Sensing Applications





Article

Remote Sensing Big Data Classification with High Performance Distributed Deep Learning

Rocco Sedona [1,2,3,4,◑,*], Gabriele Cavallaro [2,3,4,◑], Jenia Jitsev [2,4,◑], Alexandre Strube [2], Morris Riedel [1,2,3,4] and Jón Atli Benediktsson [1]

[1] School of Engineering and Natural Sciences, University of Iceland, Dunhagi 5, Reykjavík 107, Iceland; r.sedona@fz-juelich.de (R.S.); morris@hi.is (M.R.); benedikt@hi.is (J.A.B.)

[2] Jülich Supercomputing Centre (JSC), Forschungszentrum Jülich (FZJ), Wilhelm-Johnen-Strasse 1, Jülich 52425, Germany; g.cavallaro@fz-juelich.de (G.C.), j.jitsev@fz-juelich.de (J.J.); a.strube@fz-juelich.de (A.S.)

[3] High Productivity Data Processing Research Group, JSC

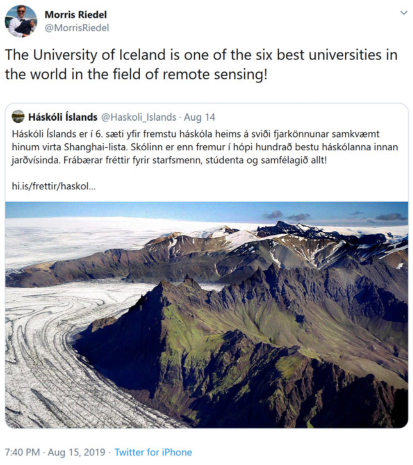
[4] Cross-Sectional Team Deep Learning (CST-DL), JSC

◑ These authors contributed equally to this work.

* Correspondence: r.sedona@fz-juelich.de; Tel.: +49 2461 61-1497

[28] R. Sedona et al., MDPI Journal of Remote Sensing

IMPACT FACTOR 4.118



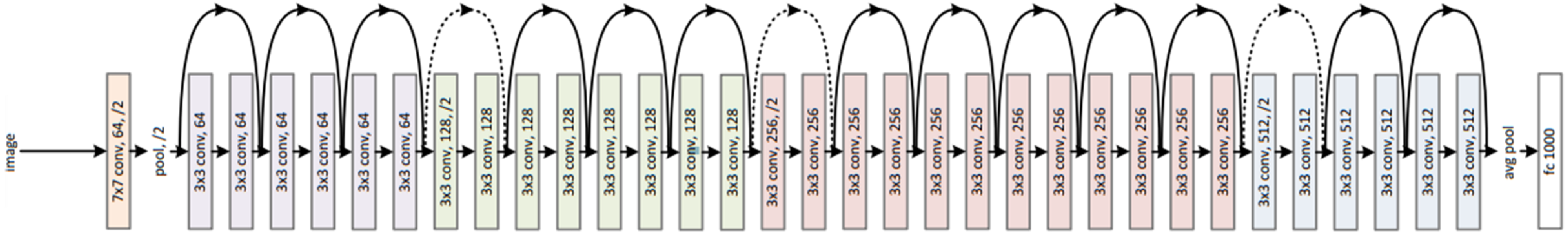
Morris Riedel
@MorrisRiedel

The University of Iceland is one of the six best universities in the world in the field of remote sensing!

Háskóli Íslands @Haskoli_Islands · Aug 14
Háskóli Íslands er í 6. sæti yfir fremstu háskóla heims á sviði fjarkönnunar samkvæmt hinum virta Shanghai-lista. Skólinn er enn fremur í hópi hundrað bestu háskólanna innan jarðvísinda. Frábærar fréttir fyrir starfsmenn, stúdenta og samfélagið allt!

hi.is/frettir/haskol...

7:40 PM · Aug 15, 2019 · Twitter for iPhone

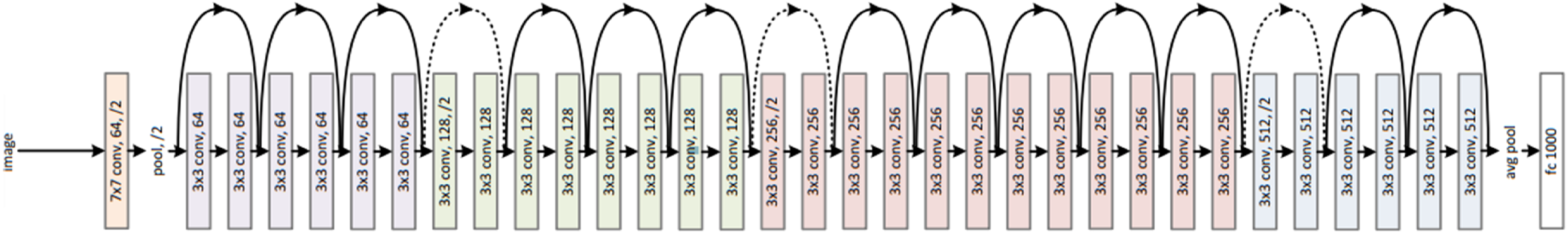


[29] RESNET

DEEP LEARNING VIA RESNET-50 ARCHITECTURE

Demand for Distributed Training because of Network Architecture Complexity

- Classification of land cover in scenes (cf. Invited Talk G. Cavallaro)
  - Very suitable for parallelization via distributed training on multi GPUs

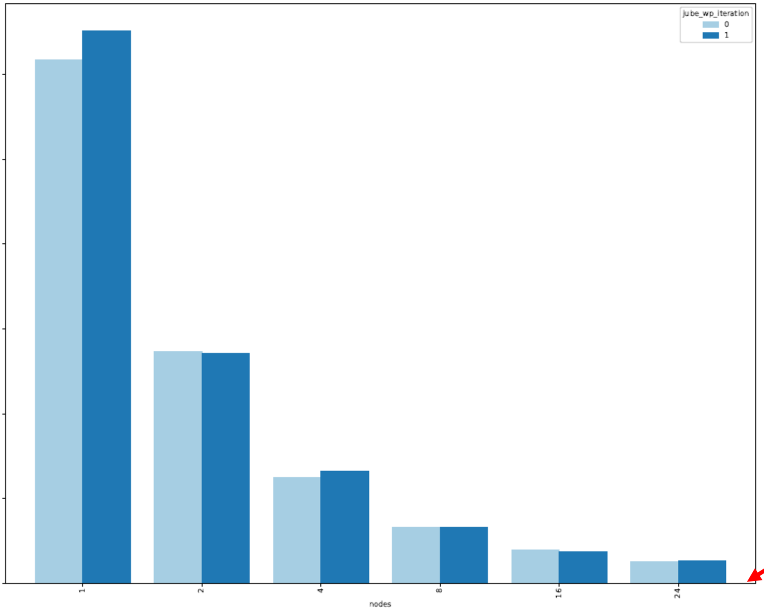


[29] RESNET

- **RESNET-50 is a known neural network architecture that has established a strong baseline in terms of accuracy**
- **The computational complexity of training the RESNET-50 architecture relies in the fact that is has ~ 25.6 millions of trainable parameters**
- **RESNET-50 still represents a good trade-off between accuracy, depth and number of parameters**
- **The setups of RESNET-50 makes it very suitable for parallelization via distributed training on multi GPUs**

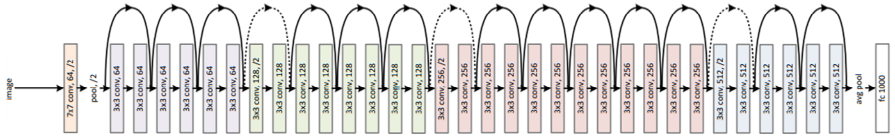
DISTRIBUTED DEEP LEARNING TRAINING VIA HOROVOD

Using MPI for Node Interactions in the Distributed Training Framework Horovod

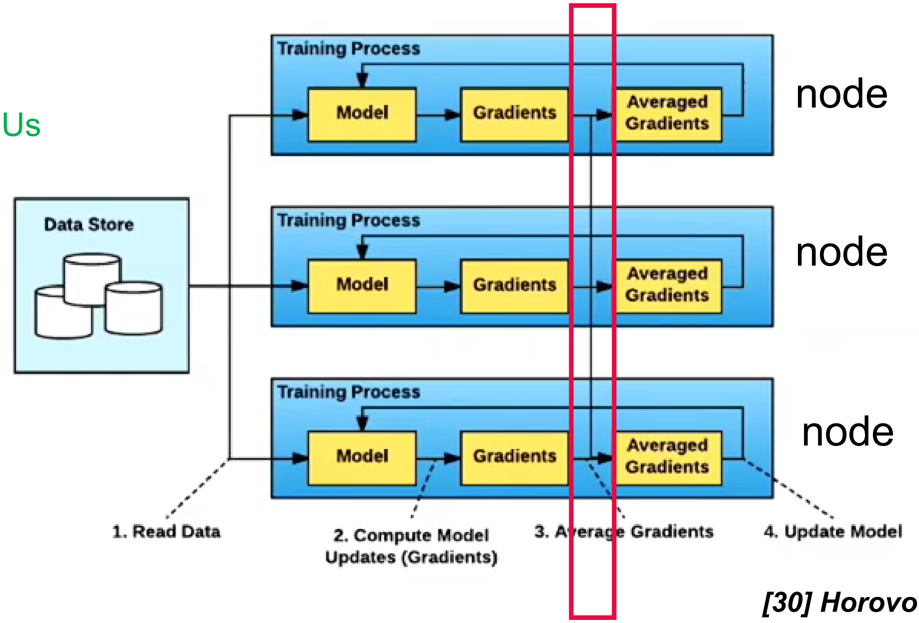


A partition of the JUWELS system
has 56 compute nodes,
each with 4 NVIDIA V100 GPUs
(equipped with 16 GB of memory)

24 nodes x 4 GPUs = 96 GPUs



[29] RESNET



[30] Horovod

MPI_Allreduce()

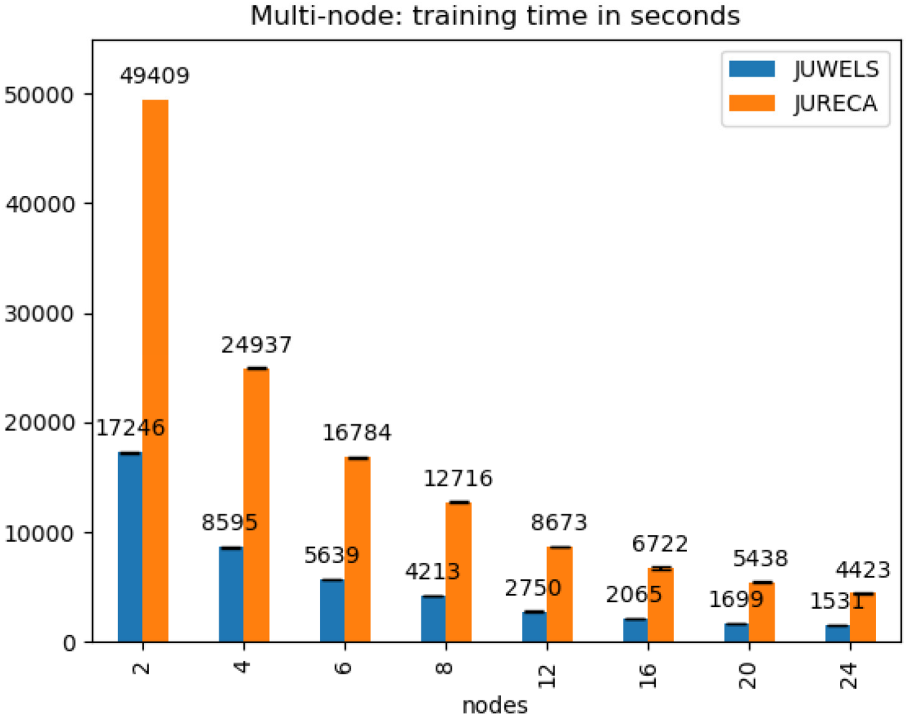
- **Horovod is a distributed training framework used in combination with low-level deep learning frameworks like Tensorflow**
- **Horovod uses MPI for inter-process communication, e.g., MPI_Allreduce()**
- **Distributed training using data parallelism approach means: (1) Gradients for different batches of data are calculated separately on each node; (2) But averaged across nodes to apply consistent updated to the deep learning model in each node**

DISTRIBUTED DEEP LEARNING TRAINING VIA HOROVOD

Generation of GPUs Matter → Kepler → Pascal → Volta

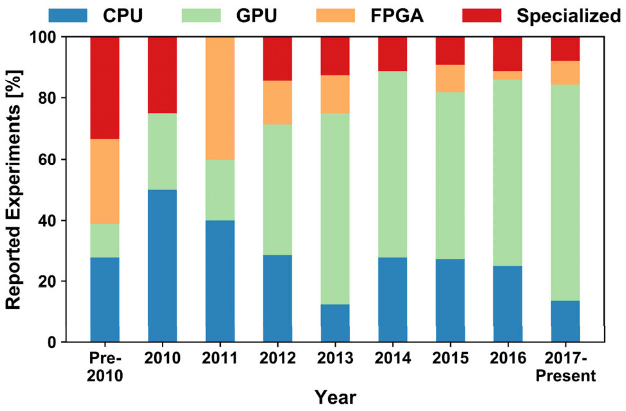
- JURECA
  - 75 compute nodes equipped with two NVIDIA K80 GPUs (four visible devices per node)

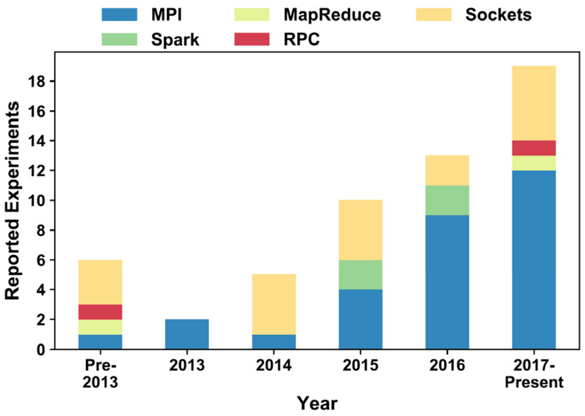
- JUWELS
  - 56 accelerated compute nodes dual core equipped with four NVIDIA V100 GPUs



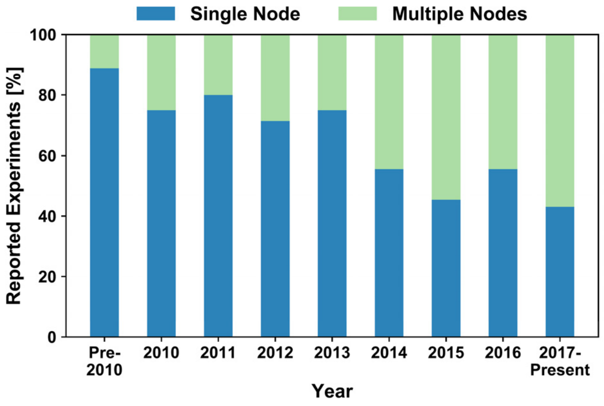
DISTRIBUTED DEEP LEARNING TRAINING EVOLUTION

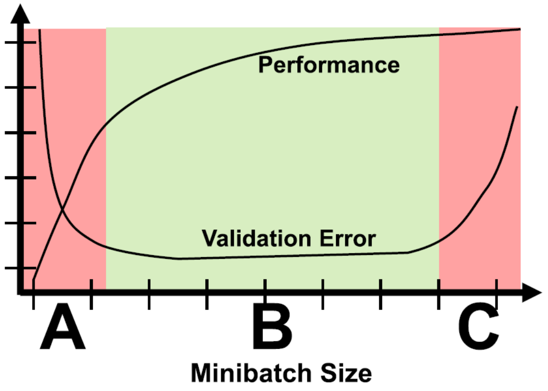
Selected Facts of using CPUs vs. GPUs and Communication Frameworks for Distribution

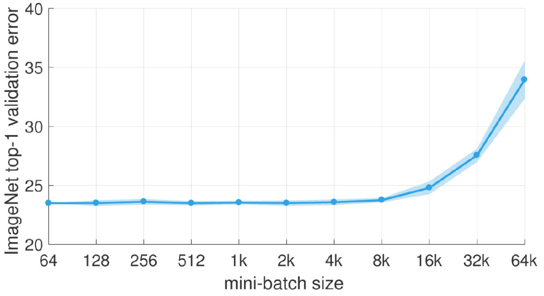




- Facts: GPUs are mostly used today for deep learning compared to CPUs, FPGA, and specialized hardware
- Facts: ~55% of all users that use deep learning use it with multiple nodes instead of just a single node
- Facts: The communication layer MPI is mostly used as communication layer for distributed training compared to Spark, Remote Procedure Calls, MapReduce, or traditional Sockets
- Most users use deep learning today with minibatches that are selected numbers of samples for performing the optimization (e.g. SGD on minibatches)
- Minibatches should be not too small to increate performance, but also not too large to increase validation error

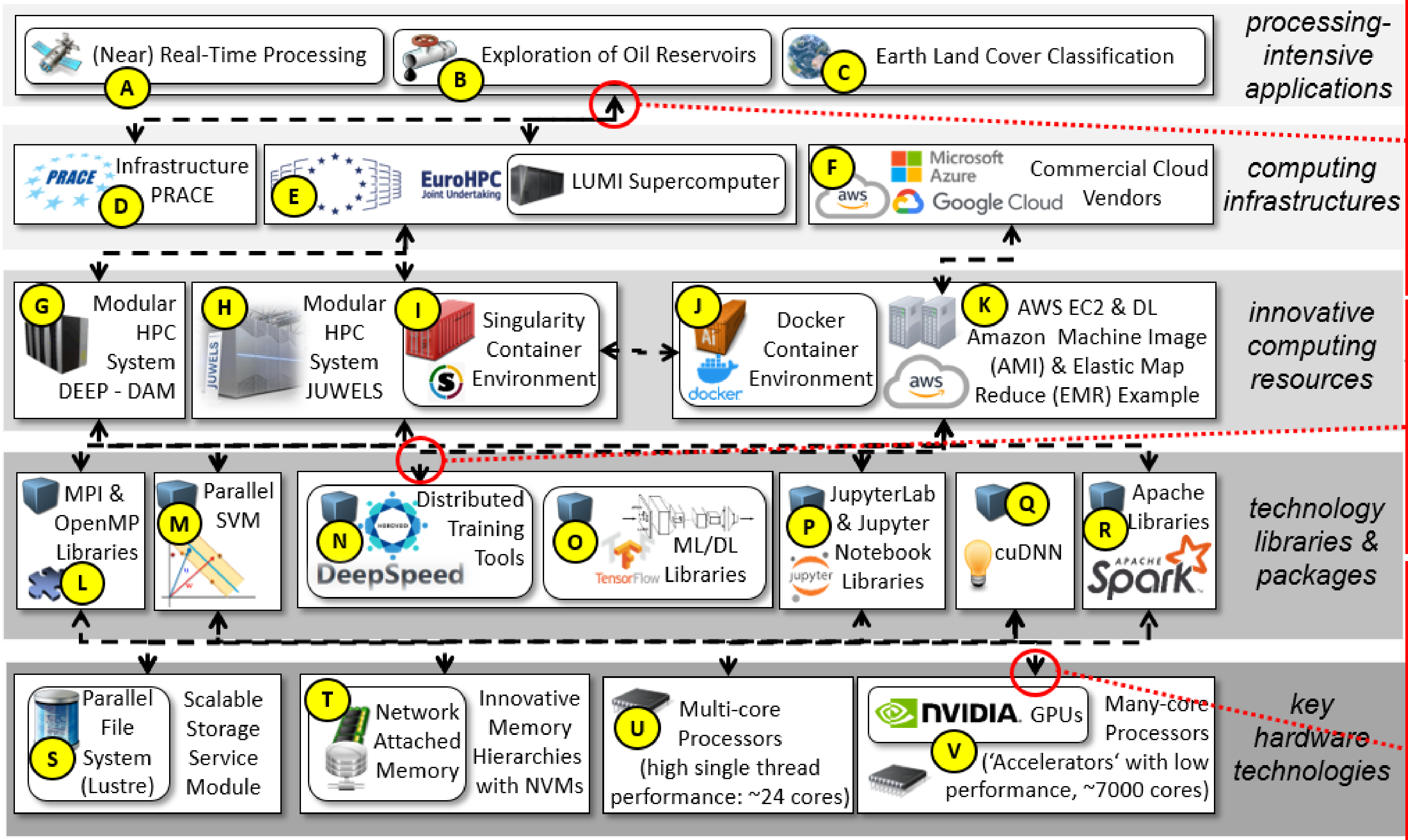






[49] T. Ben-Nun & T. Hoefler

REMOTE SENSING APPLICATIONS IN HPC & AI – SUMMARY



processing-intensive applications

A (Near) Real-Time Processing

B Exploration of Oil Reservoirs

C Earth Land Cover Classification

computing infrastructures

D Infrastructure PRACE

E EuroHPC Joint Undertaking LUMI Supercomputer

F Microsoft Azure Google Cloud aws Commercial Cloud Vendors

innovative computing resources

G Modular HPC System DEEP - DAM

H Modular HPC System JUWELS

I Singularity Container Environment

J Docker Container Environment

K AWS EC2 & DL Amazon Machine Image (AMI) & Elastic Map Reduce (EMR) Example

technology libraries & packages

L MPI & OpenMP Libraries

M Parallel SVM

N Distributed Training Tools DeepSpeed

O ML/DL Libraries TensorFlow

P JupyterLab & Jupyter Notebook Libraries

Q cuDNN

R Apache Libraries Apache Spark

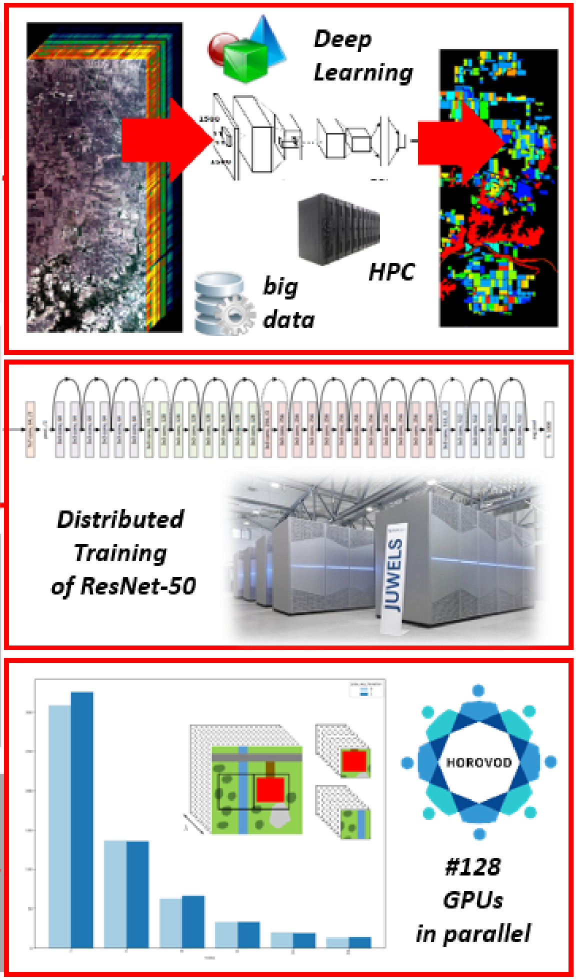
key hardware technologies

S Parallel File System (Lustre) Scalable Storage Service Module

T Network Attached Memory Innovative Memory Hierarchies with NVMs

U Multi-core Processors (high single thread performance: ~24 cores)

V NVIDIA GPUs Many-core Processors ('Accelerators' with low performance, ~7000 cores)



Distributed Training of ResNet-50

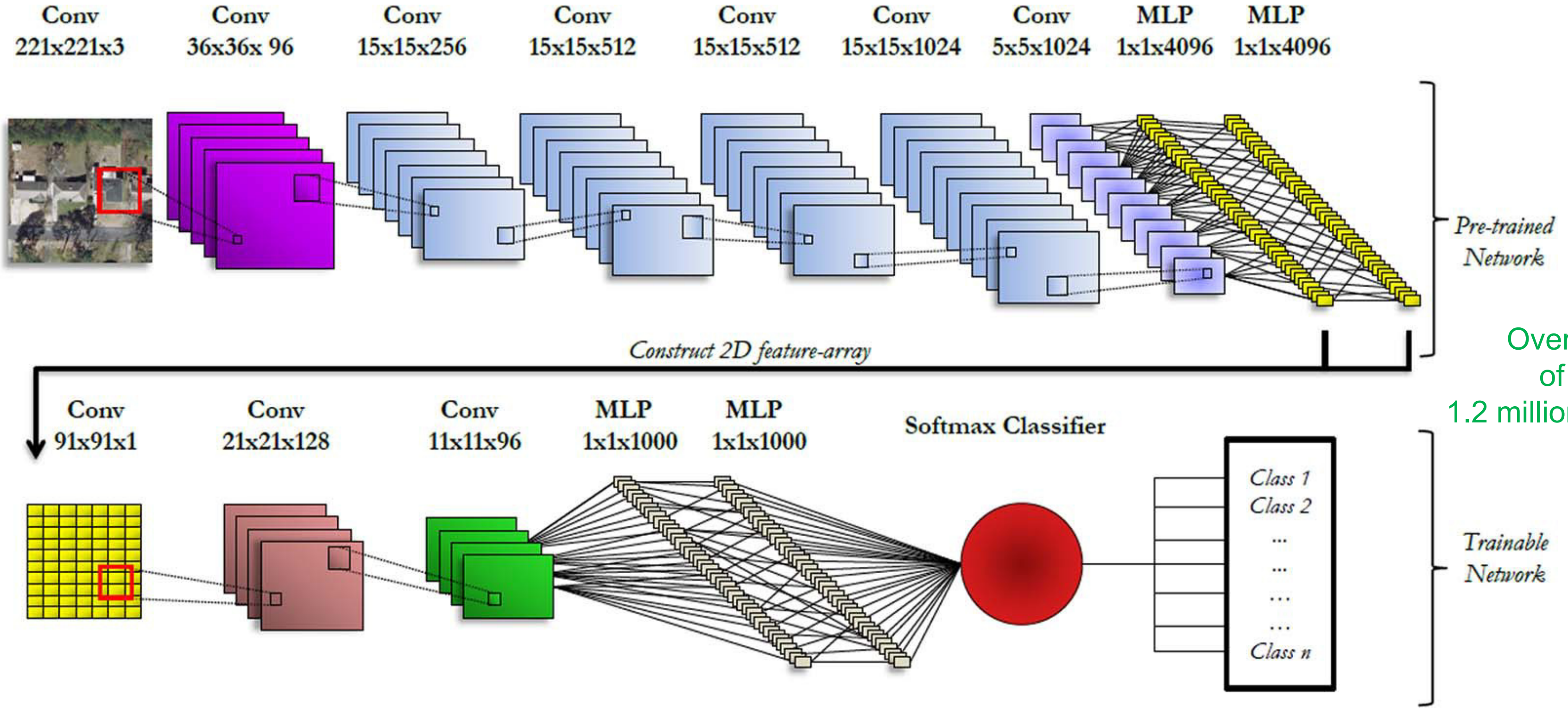
#128 GPUs in parallel

TRANSFER LEARNING APPROACHES

Selected Approaches when Facing Small Datasets

PRE-TRAINED CONVOLUTIONAL NEURAL NETWORKS

Example for ImageNet Application



Using available Overfeat as pre-trained network

Overfeat is an improved version of AlexNet and is trained on 1.2 million labeled images from ImageNet

[46] D. Marmanis et al., 'Deep Learning Earth Obervation Classification Using ImageNet Pretrained Networks', 2016

[47] P. Sermanet et al., 'OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks'

NEURAL ARCHITECTURE SEARCH

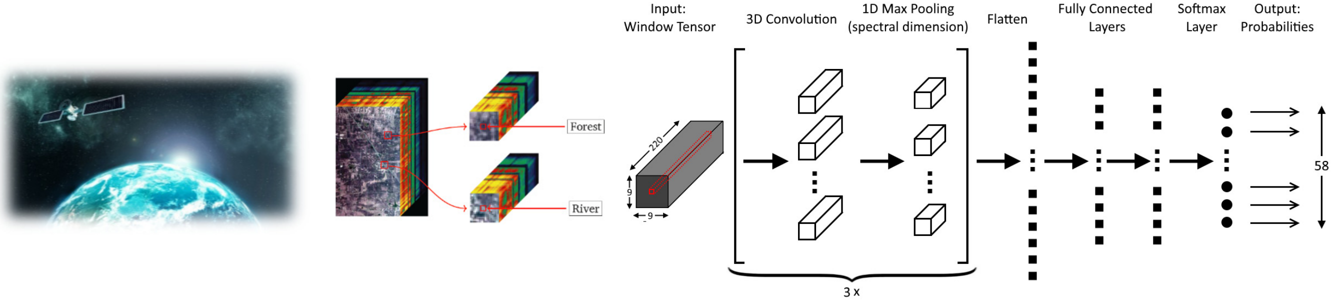
Finding Hyper-Parameters of Neural Network Architectures in a more Systematic Way

KEY CHALLENGE: FIND THE RIGHT PARAMETERS

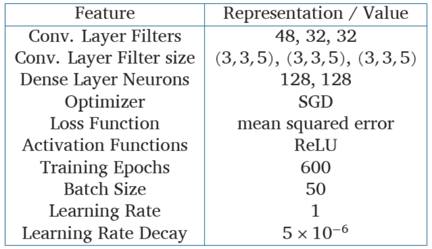
Example of Remote Sensing Applications

- **Using Convolutional Neural Networks (CNNs) with hyperspectral remote sensing image data**

[36] J. Lange and M. Riedel et al., IGARSS Conference, 2018

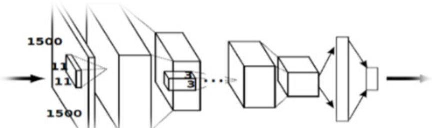


- **Find right set of hyper-parameters and the right neural network architecture is a manual time-consuming and error-prone process**
- **Needs urgently HPC, but a systematic and automated way is required as trying out all options of hyper-parameters and architectures is computationally infeasible**

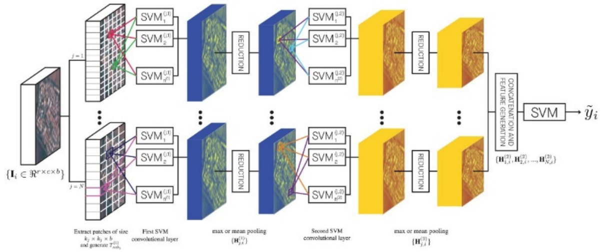


Feature	Representation / Value
Conv. Layer Filters	48, 32, 32
Conv. Layer Filter size	(3,3,5), (3,3,5), (3,3,5)
Dense Layer Neurons	128, 128
Optimizer	SGD
Loss Function	mean squared error
Activation Functions	ReLU
Training Epochs	600
Batch Size	50
Learning Rate	1
Learning Rate Decay	5×10^{-6}

- What is the right optimization method?
- How many convolutional layers we need?
- How many neurons in dense layers?
- What is the right filter size?
- How do we train best?



- **Find Hyperparameters & joint ‘new-old‘ modeling & transfer learning given rare labeled/annotated data in science (e.g. 36,000 vs. 14,197,122 images ImageNet)**

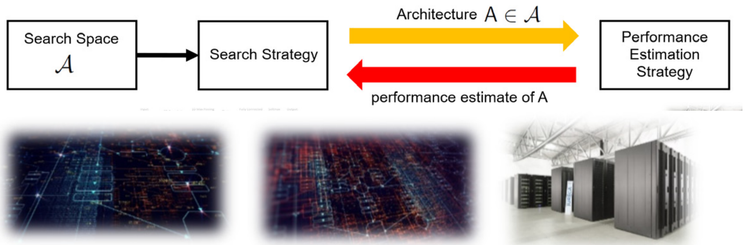

SVM
REDUCTION
CONCATENATION AND FEATURE GENERATION
$\hat{y}_i$
Extract patches of size
First SVM convolutional layer
max or mean pooling
Second SVM convolutional layer
max or mean pooling

[37] G. Cavallaro, M. Riedel et al., IGARSS 2019

NEURAL ARCHITECTURE SEARCH (NAS)

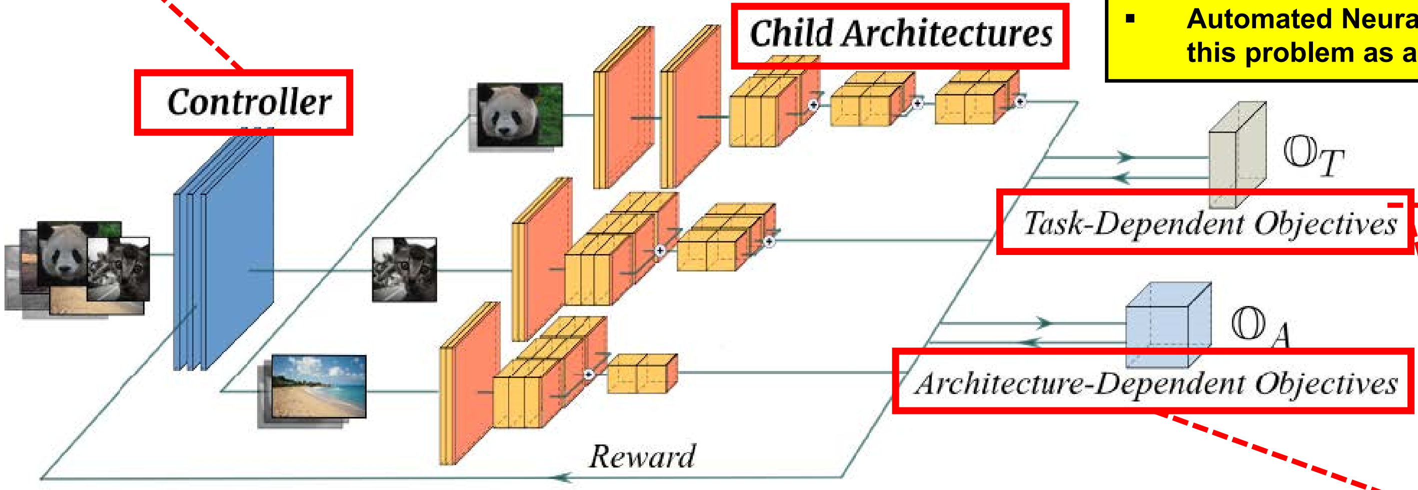
Massive Requirement for HPC Resources

- Often a Recurrent Neural Network (RNN) technique that performs the agent steps


Search Space $\mathcal{A}$
Search Strategy
Architecture $A \in \mathcal{A}$
performance estimate of A
Performance Estimation Strategy

- Employed neural networks architectures are often developed manually by human experts that is time-consuming and error-prone
- Deep learning success has been accompanied by a rising demand for architecture engineering, where increasingly more complex neural architectures are designed manually
- Neural Architecture Search (NAS) methods can be categorized in (a) search space, (b) search strategy, and (c) performance estimation strategy
- Automated Neural Architecture (NAS) search methods aim to solve this problem as a process of automating Architecture engineering

[38] M. Riedel, 'NAS with Reinforcement Learning'



- Derived specific architectures that perform good for specific dataset samples

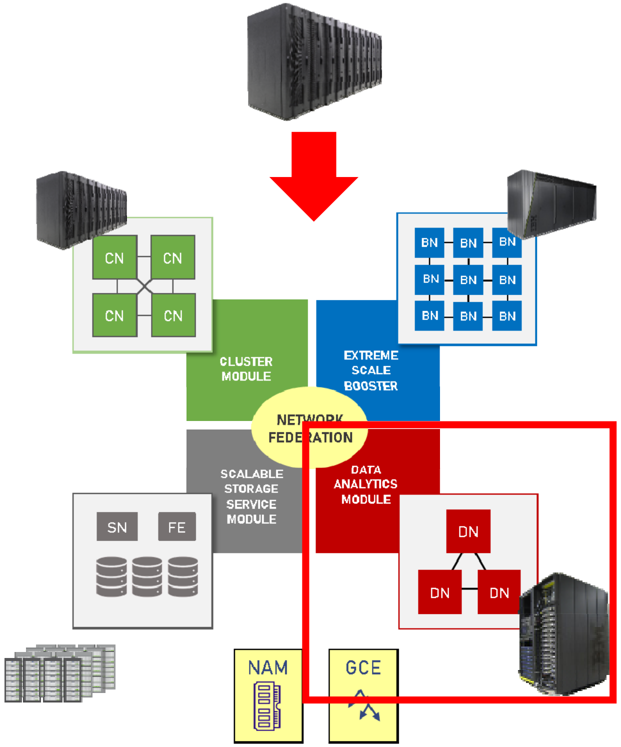
- E.g. what is the accuracy or error rate we obtain as metric to guide the search for specific architectures for specific dataset samples

- E.g. what is the latency of the network for a given dataset sample to guide the search for specific architectures that offer better latency by keeping accuracy(!)

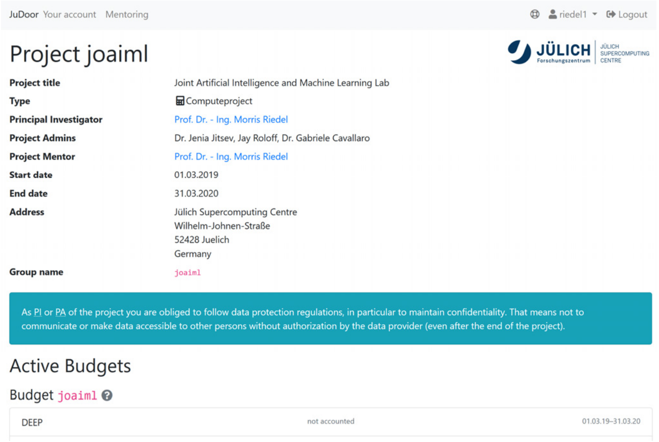
[39] A.C. Cheng et al., 'InstaNAS: Instance-aware Neural Architecture Search', 2018

MODULAR SUPERCOMPUTING ARCHITECTURE

Data Analytics Module (DAM) Prototype Example



- Data Analytics Module (DAM)
  - Specific requirements for data science & analytics frameworks
  - 16 nodes with 2x Intel Xeon Cascade Lake; 24 cores
  - 1x NVIDIA V100 GPU / node
  - 1x Intel STRATIX10 FPGA PCIe3 / node
  - 384 GB DDR4 memory / node
  - 2 TB non-volatile memore / node
- DAM Prototype for teaching
  - 3 x 4 GPUs Tesla Volta V100
  - Slurm scheduling system



Project joaiml

Project title Joint Artificial Intelligence and Machine Learning Lab

Type Computeproject

Principal Investigator Prof. Dr. - Ing. Morris Riedel

Project Admins Dr. Jenia Jitsev, Jay Roloff, Dr. Gabriele Cavallaro

Project Mentor Prof. Dr. - Ing. Morris Riedel

Start date 01.03.2019

End date 31.03.2020

Address Jülich Supercomputing Centre
Wilhelm-Johnen-Straße
52428 Juelich
Germany

Group name joaiml

As PI or PA of the project you are obliged to follow data protection regulations, in particular to maintain confidentiality. That means not to communicate or make data accessible to other persons without authorization by the data provider (even after the end of the project).

Active Budgets

Budget joaiml

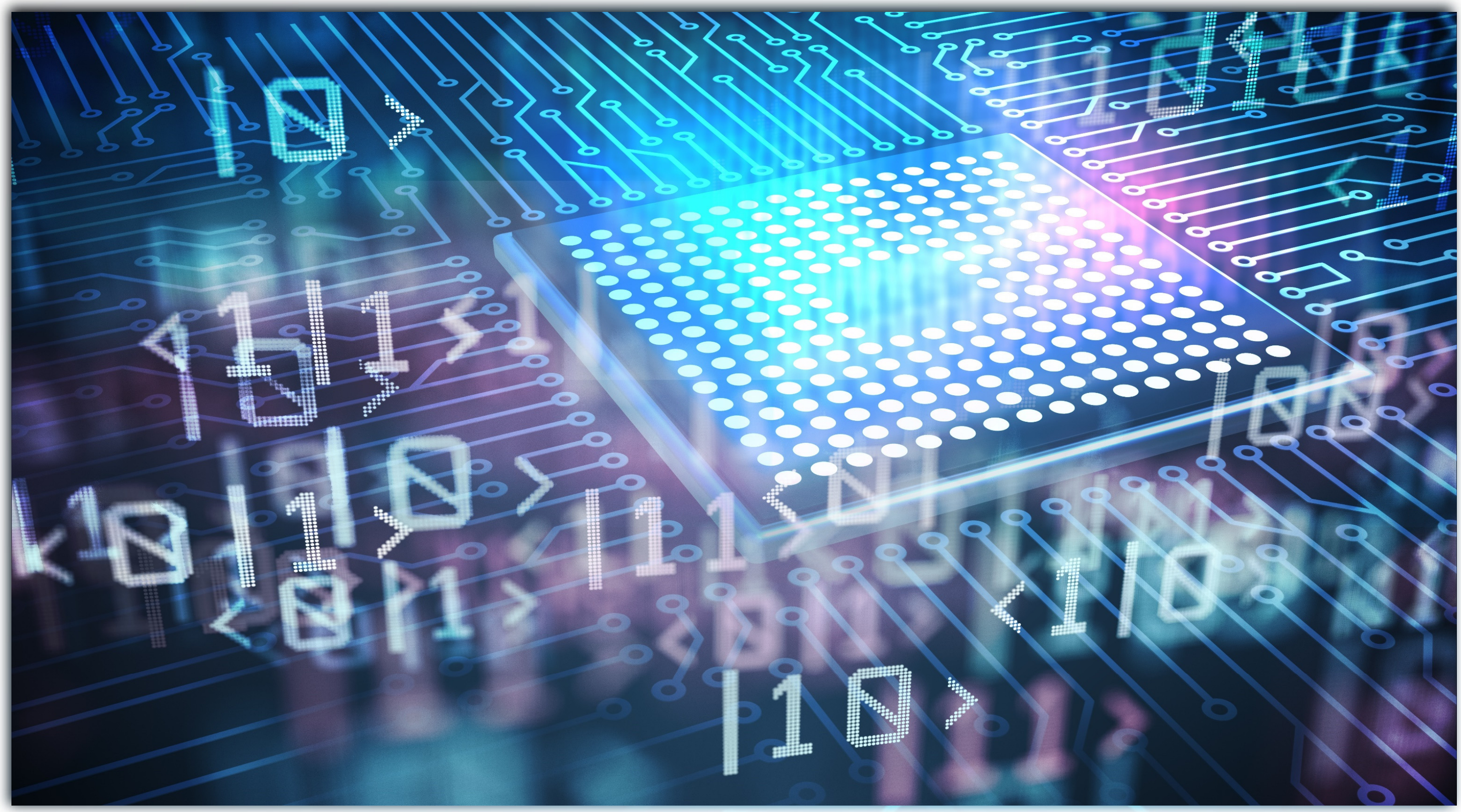
DEEP not accounted 01.03.19–31.03.20

(easy join via JOIAML lab with JuDoor)

[15] DEEP Projects Web Page

SHORT INTRODUCTION TO QUANTUM COMPUTING FOR AI

Focus on Quantum Annealing Approach to Quantum Computing

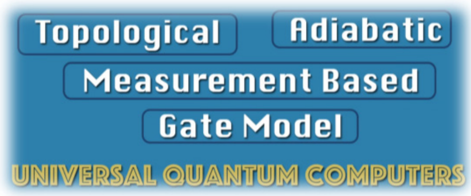


QUANTUM COMPUTING IS STILL VERY COMPLEX

Many different Approaches exist for Quantum Computing

- Quantum Annealing (focus in this talk)
  - D-Wave System 2000Q (annealer system) will be part of the emerging Juelich Unified Infrastructure for Quantum Computing (JUNIQ)
  - Uses intrinsic effects of Quantum Physics (QP) to help in optimization problems or probabilistic sampling (i.e., is not a mainstream computer!)
  - Setup a problem, then natural evolution of quantum states, and finally configuration at the end of evolution is one/some answer (but no control)
- Gate-Model Quantum Computing
  - Much more ambitious to control and manipulate the evolution of quantum states over time, but more difficult as quantum systems hard to work with
  - But enables to solve bigger problems, ~ 10 Qubits only
  - Hard to let Qubits working together coherently

(quantum annealer vs. universal quantum computer approaches)


Topological
Adiabatic
Measurement Based
Gate Model
UNIVERSAL QUANTUM COMPUTERS

[50] Launch of JUNIQ

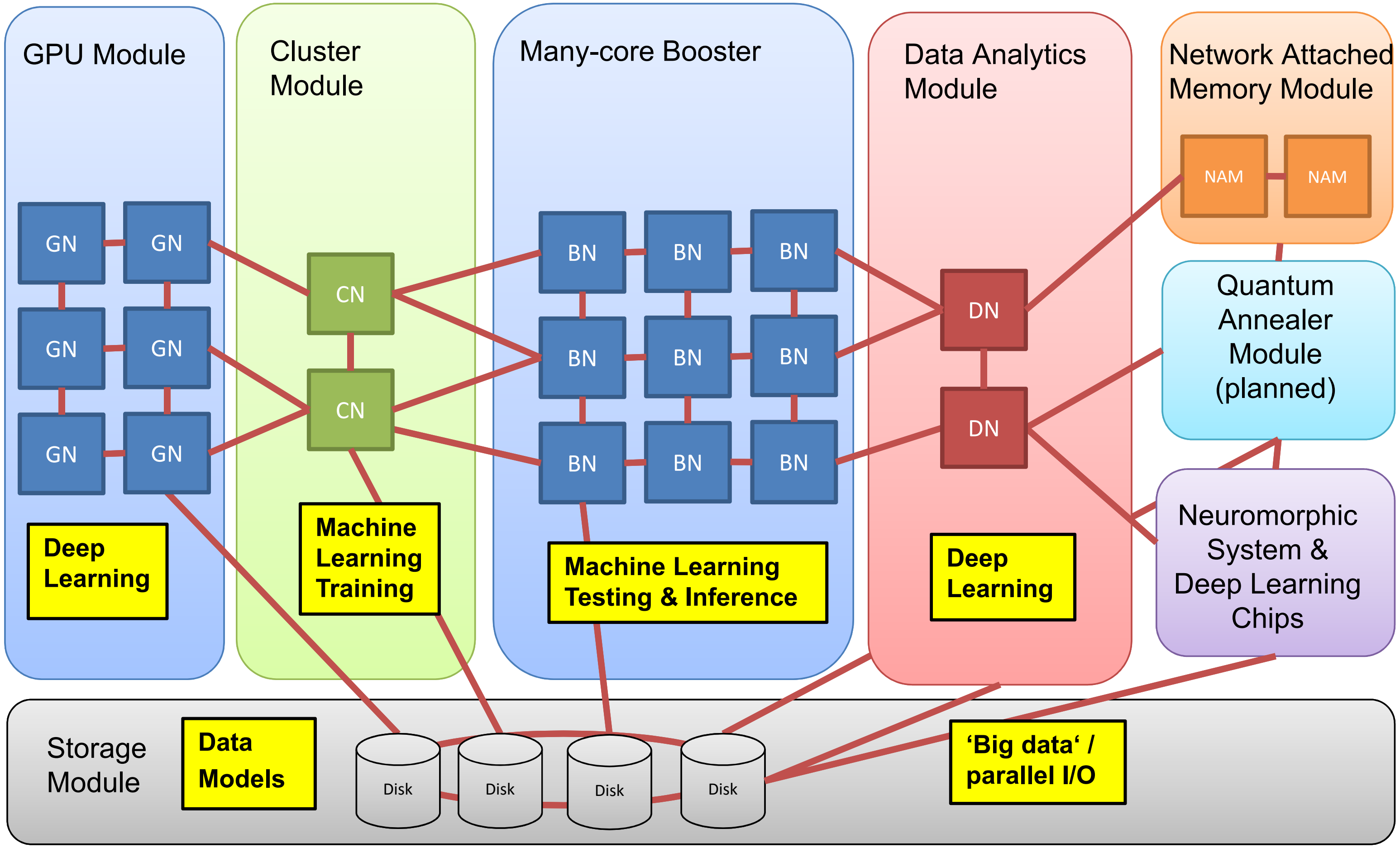
[51] D-Wave Systems

MODULAR SUPERCOMPUTING ARCHITECTURE

Innovation through Cutting-Edge Technologies

DEEP Projects

[15] DEEP Projects Web Page



GPU Module

GN GN GN GN GN GN

Deep Learning

Cluster Module

CN CN

Machine Learning Training

Many-core Booster

BN BN BN BN BN BN BN BN BN

Machine Learning Testing & Inference

Data Analytics Module

DN DN

Deep Learning

Network Attached Memory Module

NAM NAM

Quantum Annealer Module (planned)

Neuromorphic System & Deep Learning Chips

Storage Module

Data Models

Disk Disk Disk Disk

'Big data' / parallel I/O

Innovative Ideas, e.g. trained models in memory, put/get store for data, non-volatile memory, etc.

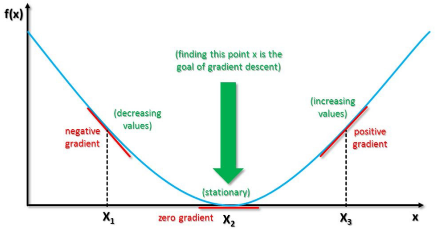
Innovative computing paradigms for specific tasks, e.g. solving optimization tasks in machine learning algorithms

Innovative chips, e.g. use of deep learning optimized chip designs

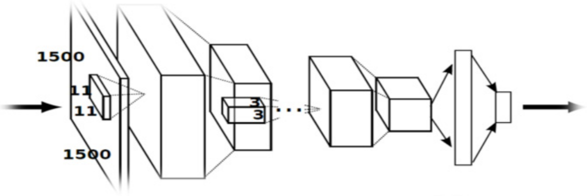
QUANTUM ANNEALING FOR OPTIMIZATION PROBLEMS

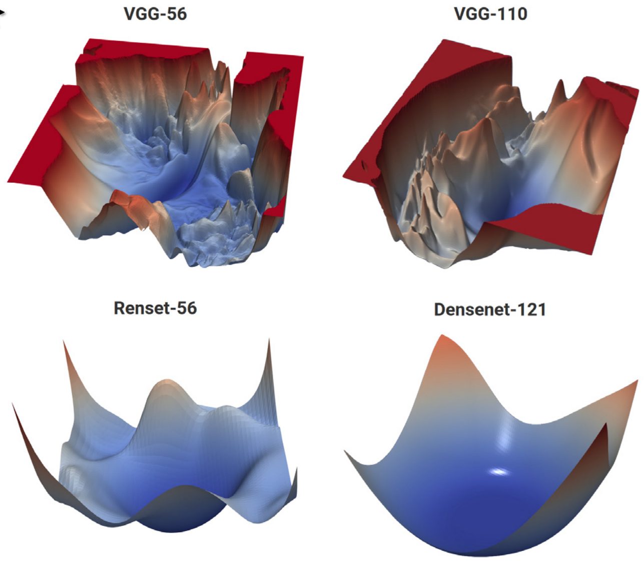
Optimization Problems can be found in many machine & deep learning algorithms

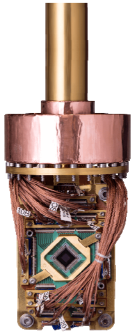
- Key Problem(not only existing in AI)
  - Trying to search for the best configuration out of extremely many configurations
  - E.g. optimization during training of deep neural networks (i.e., error/loss minimization for learning correct weights)
  - What is the best combination of all the different configuration options?
  - Also called ‘energy minimization problem‘ (i.e., low is good)
  - Fundamental part of physics is trying to find its minimum energy state



[53] Gradient Descent Example







- **Quantum Annealing is using Quantum Physics to find the minimum energy state of a given problem**
- **Quantum Annealing is harnessing the natural evolution of quantum states (no direct control of evolution**

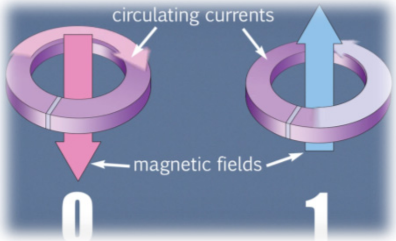
[52] Loss Visualization

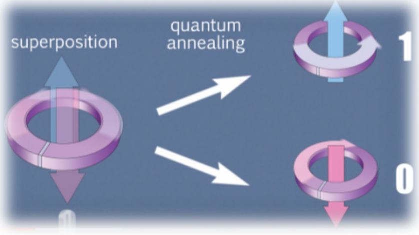
[51] D-Wave Systems

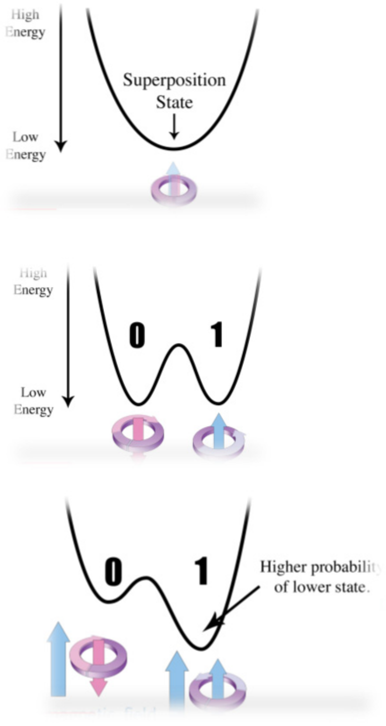
ELEMENTS OF QUANTUM ANNEALING: SUPERPOSITION

Perform Calculations via Qubits by Exploiting 'Superposition' Applying Magnetic Fields

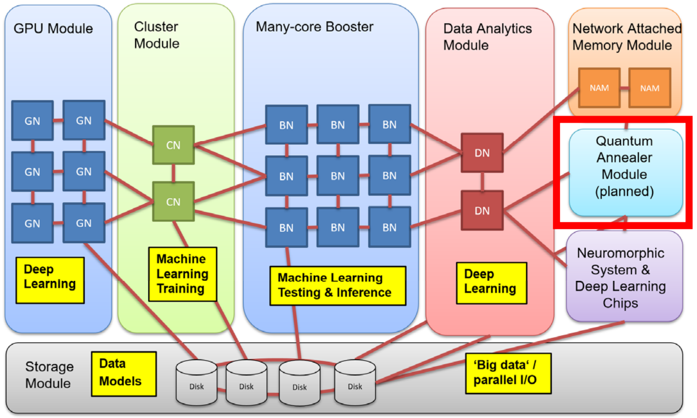
- Quantum Bits
  - The key element of quantum computers are based on circuits that are called 'quantum bits' or 'qubits' for short
  - Compared to traditional computers: qubit not represent 0/1, but 0 and 1 simultaneously ('superposition')
  - N qubits can represent 2^N bits of information (e.g., 2 = 4 states; 3 = 8 states)







(applied magnetic field influences the probability)


GPU Module
Cluster Module
Many-core Booster
Data Analytics Module
Network Attached Memory Module
NAM
NAM
GN
CN
BN
DN
Quantum Annealer Module (planned)
Deep Learning
Machine Learning Training
Machine Learning Testing & Inference
Deep Learning
Neuromorphic System & Deep Learning Chips
Storage Module
Data Models
Disk
'Big data' / parallel I/O

D-Wave Systems @dwavesys · 3h

Register for the webinar where speakers from @Accenture will discuss an important problem and a promising quantum approach to solving it: the live kidney donor exchange system.

bit.ly/32elloQ #QuantumComputing

Welcome! You are invited to join a webinar: Delivering Valu...
Since 2016, Accenture has been working on delivering value through quantum computing. In this talk, speakers from ...
dwavesys.zoom.us

[45] Big Data Tips, Quantum Computing

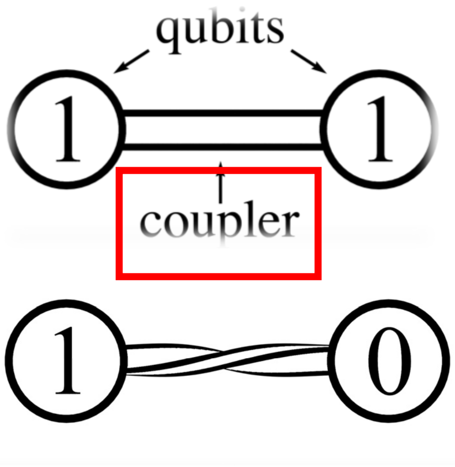
[51] D-Wave Systems

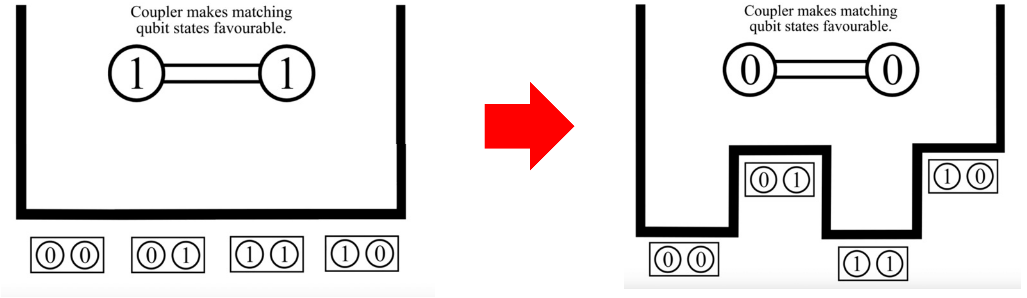
[46] D-Wave Systems on Twitter

ELEMENTS OF QUANTUM ANNEALING: ENTANGLEMENT

Innovative Potential of Quantum Devices for Solving Difficult Optimization Problems with Entanglement

- Entanglement
  - Two quantum systems (e.g., like an electron or a nucleus) interaction: both become connected (‘entanglement’) using a coupler
  - They retain a very specific ‘correlation’ in their energy states
  - ‘Correlations’ represent combinations of 0 & 1
  - Thus ‘entanglement’ enables qubits to work together to represent multiple combinations of values simultanously (e.g., compared to today with traditional computers: just one combination at a time)
- Particular calculation finished in ~ ms time:
  - Qubits can be observed as 0 or 1 values to determine solutions almost like in classical computers today



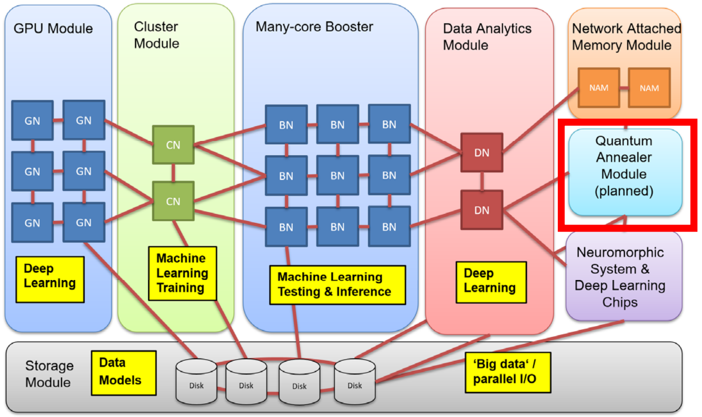

Coupler makes matching qubit states favourable.
1
1
00
01
11
10
Coupler makes matching qubit states favourable.
0
0
01
10
00
11

[51] D-Wave Systems

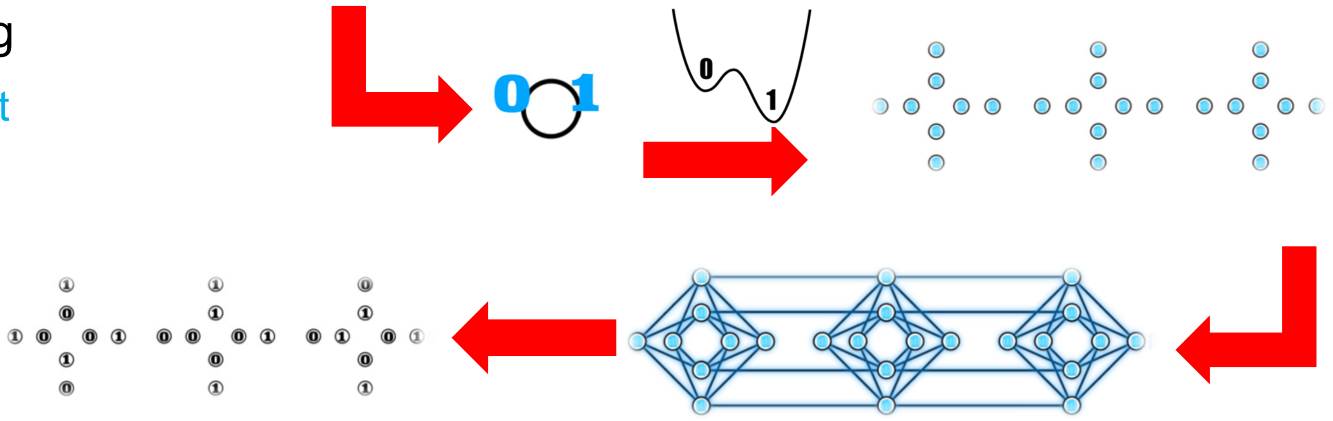
PROGRAMMING QUANTUM ANNEALING IN PRACTICE

Using Ocean SDK & Small Datasets (in the moment)

- Ocean SDK Python API
  - Enables interaction from standard computer with Quantum Annealer
  - E.g. formalization as a specific optimization problem
  - (not like usual programming of just application logic)
  - Most time consuming element of programming, requires rather thinking and math knowledge
- Data View for Machine/Deep Learning
  - Works only for small data in the moment (e.g. just 30 samples libsvm format)
  - No access to parallel filesystem or storage module directly from Annealer
  - E.g. using Python data structures

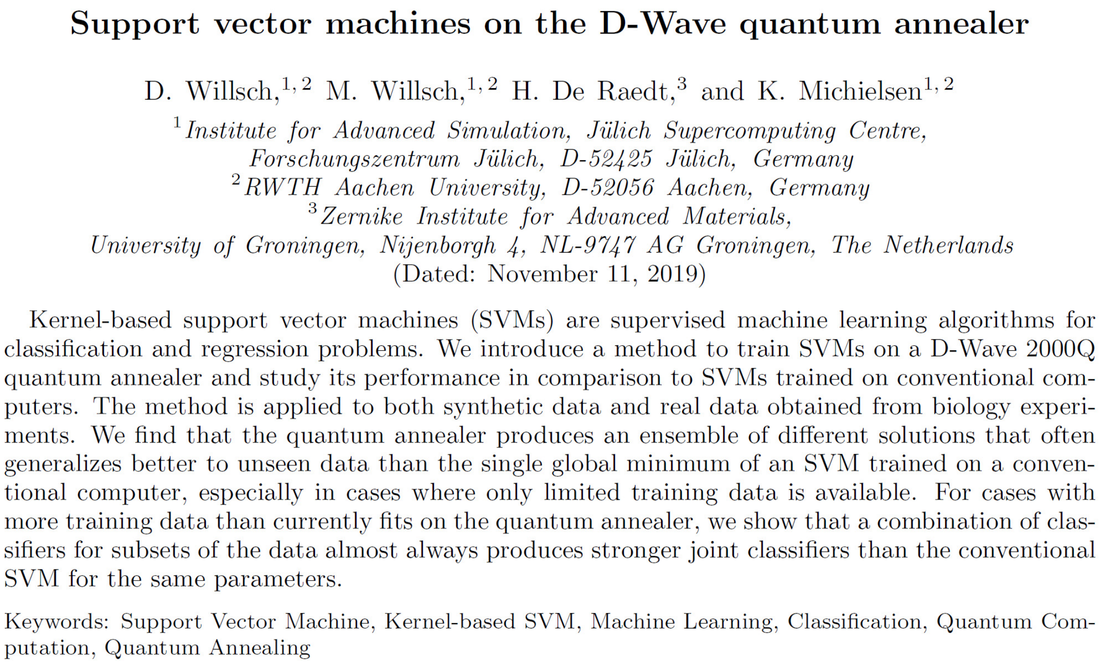

GPU Module
GN
Deep Learning
Cluster Module
CN
Machine Learning Training
Many-core Booster
BN
Machine Learning Testing & Inference
Data Analytics Module
DN
Deep Learning
Network Attached Memory Module
NAM
Quantum Annealer Module (planned)
Neuromorphic System & Deep Learning Chips
Storage Module
Data Models
Disk
'Big data' / parallel I/O

[51] D-Wave Systems



SUPPORT VECTOR MACHINE ON QUANTUM ANNEALER

Solving a Quadratic Optimization Problem that is inherent in this Machine Learning Technique



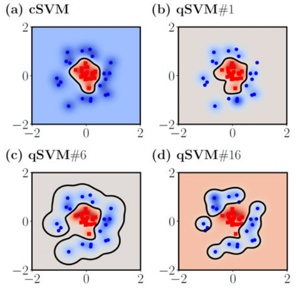
Support vector machines on the D-Wave quantum annealer

D. Willsch,[1, 2] M. Willsch,[1, 2] H. De Raedt,[3] and K. Michielsen[1, 2]
[1] *Institute for Advanced Simulation, Jülich Supercomputing Centre,*
Forschungszentrum Jülich, D-52425 Jülich, Germany
[2] *RWTH Aachen University, D-52056 Aachen, Germany*
[3] *Zernike Institute for Advanced Materials,*
University of Groningen, Nijenborgh 4, NL-9747 AG Groningen, The Netherlands
(Dated: November 11, 2019)

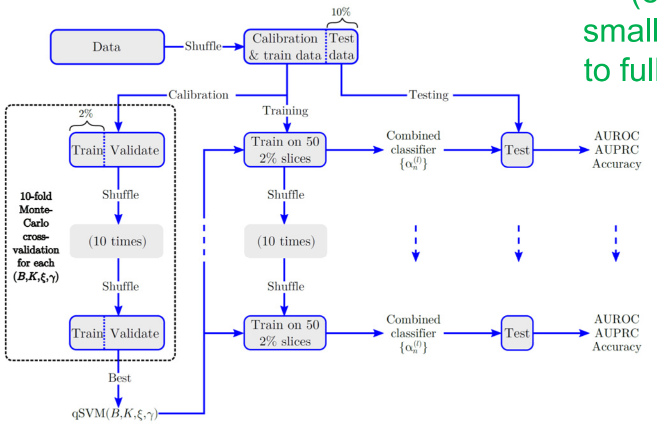
Kernel-based support vector machines (SVMs) are supervised machine learning algorithms for classification and regression problems. We introduce a method to train SVMs on a D-Wave 2000Q quantum annealer and study its performance in comparison to SVMs trained on conventional computers. The method is applied to both synthetic data and real data obtained from biology experiments. We find that the quantum annealer produces an ensemble of different solutions that often generalizes better to unseen data than the single global minimum of an SVM trained on a conventional computer, especially in cases where only limited training data is available. For cases with more training data than currently fits on the quantum annealer, we show that a combination of classifiers for subsets of the data almost always produces stronger joint classifiers than the conventional SVM for the same parameters.

Keywords: Support Vector Machine, Kernel-based SVM, Machine Learning, Classification, Quantum Computation, Quantum Annealing

[54] Quantum SVM, D. Willsch et al.



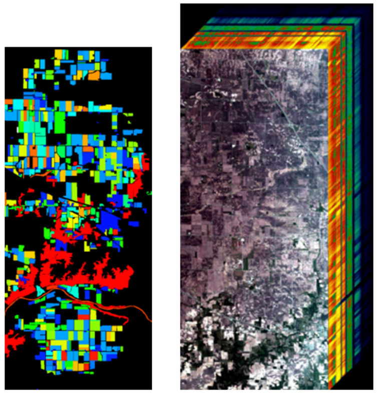
(ensembles due to small datasets compared to full datasets on CPUs)



➢ **Appendix offers details on understanding Support Vector Machines (SVMs) & Kernel Methods with a geometric SVM interpretation**

PARALLEL & SCALABLE ALGORITHM DEVELOPMENT

Using Support Vector Machines with Quantum Annealing with Remote Sensing (work in progress…)



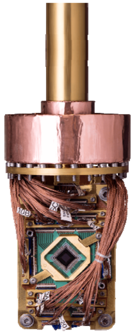
Parallel & Scalable Classification with SVMs based on Message Passing Interface (MPI) using HPC resources

Scenario ‘pre-processed data‘, 10xCV **serial**: accuracy (min)

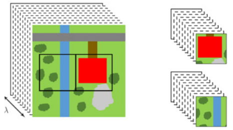
γ/C	1	10	100	1000	10 000
2	48.90 (18.81)	65.01 (19.57)	73.21 (20.11)	75.55 (22.53)	74.42 (21.21)
4	57.53 (16.82)	70.74 (13.94)	75.94 (13.53)	76.04 (14.04)	74.06 (15.55)
8	64.18 (18.30)	74.45 (15.04)	**77.00** (14.41)	75.78 (14.65)	74.58 (14.92)
16	68.37 (23.21)	76.20 (21.88)	76.51 (20.69)	75.32 (19.60)	74.72 (19.66)
32	70.17 (34.45)	75.48 (34.76)	74.88 (34.05)	74.08 (34.03)	73.84 (38.78)

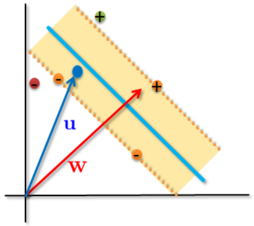
$$E = \sum_{i \leq j} a_i Q_{ij} a_j,$$

$$\alpha_n = \sum_{k=0}^{K-1} B^k a_{Kn+k}$$



$$\begin{aligned} \text{minimize} \quad & E = \frac{1}{2}\sum_{nm} \alpha_n \alpha_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \\ & \quad - \sum_n \alpha_n, \\ \text{subject to} \quad & 0 \leq \alpha_n \leq C, \\ \text{and} \quad & \sum_n \alpha_n t_n = 0, \end{aligned}$$





Scenario ‘pre-processed data‘, 10xCV **parallel**: accuracy (min)

γ/C	1	10	100	1000	10 000
2	75.26 (1.02)	65.12 (1.03)	73.18 (1.33)	75.76 (2.35)	74.53 (4.40)
4	57.60 (1.03)	70.88 (1.02)	75.87 (1.03)	76.01 (1.33)	74.06 (2.35)
8	64.17 (1.02)	74.52 (1.03)	**77.02 (1.02)**	75.79 (1.04)	74.42 (1.34)
16	68.57 (1.33)	76.07 (1.33)	76.40 (1.34)	75.26 (1.05)	74.53 (1.34)
32	70.21 (1.33)	75.38 (1.34)	74.69 (1.34)	73.91 (1.47)	73.73 (1.33)

First Result: best parameter set from 14.41 min to 1.02 min
Second Result: all parameter sets from ~9 hours to ~35 min

D:WAVE
The Quantum Computing Company™

Quantum Annealer requires the formulation of the computational problem as a quadratic unconstrained binary optimization(QUBO)

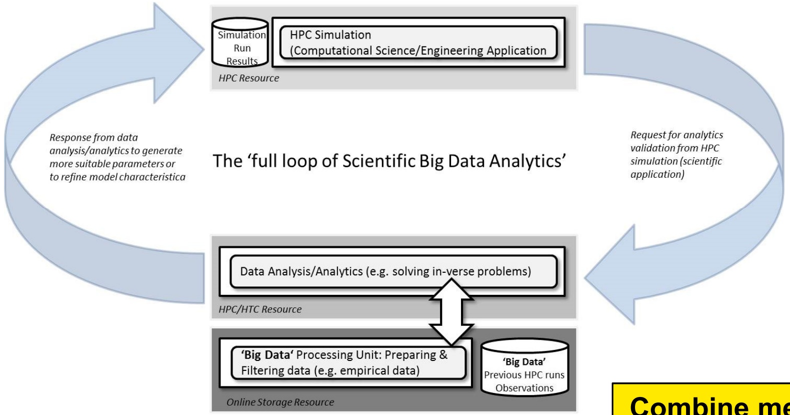
[20] G. Cavallaro and M. Riedel et al., Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 2015

IMPACTS OF ARTIFICIAL INTELLIGENCE IN APPLICATIONS

Emerging Medical Application Examples

INTERTWINED HPC SIMULATIONS & MACHINE LEARNING

Enabling ‘full loop‘ in research – forward numerical simulations – backwards machine & deep learning


Simulation Run Results
HPC Simulation
(Computational Science/Engineering Application
HPC Resource
Response from data analysis/analytics to generate more suitable parameters or to refine model characteristica
The ‘full loop of Scientific Big Data Analytics’
Request for analytics validation from HPC simulation (scientific application)
Data Analysis/Analytics (e.g. solving in-verse problems)
HPC/HTC Resource
‘Big Data‘ Processing Unit: Preparing & Filtering data (e.g. empirical data)
‘Big Data’
Previous HPC runs
Observations
Online Storage Resource

[31] Th. Lippert, D. Mallmann, M. Riedel, ‘Scientific Big Data Analytics by HPC’, NIC Series 48, 2016

MEDIZIN INFORMATIK INITIATIVE

UNIKLINIK RWTHAACHEN

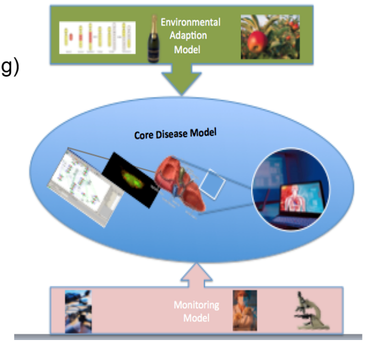
SMITH
Smart Medical Information Technology for Healthcare

SPONSORED BY THE

Federal Ministry of Education and Research

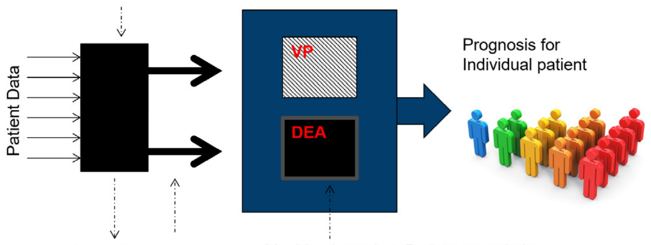
Combine mechanistic/numeric modeling with machine learning modeling in one ‘full loop‘ (~ ‘hybrid modeling‘)

Patient + state paramters:
individualisation
(Machine Learning)



generic model
(mechanistic)

adaption of generic model
to available data (Machine Learning)



Unsupervised Patient Stratification
- Dynamic clustering
- Critical state detection

Predictive modelling Machine for Algorithmic Surveillance of ICU Patients

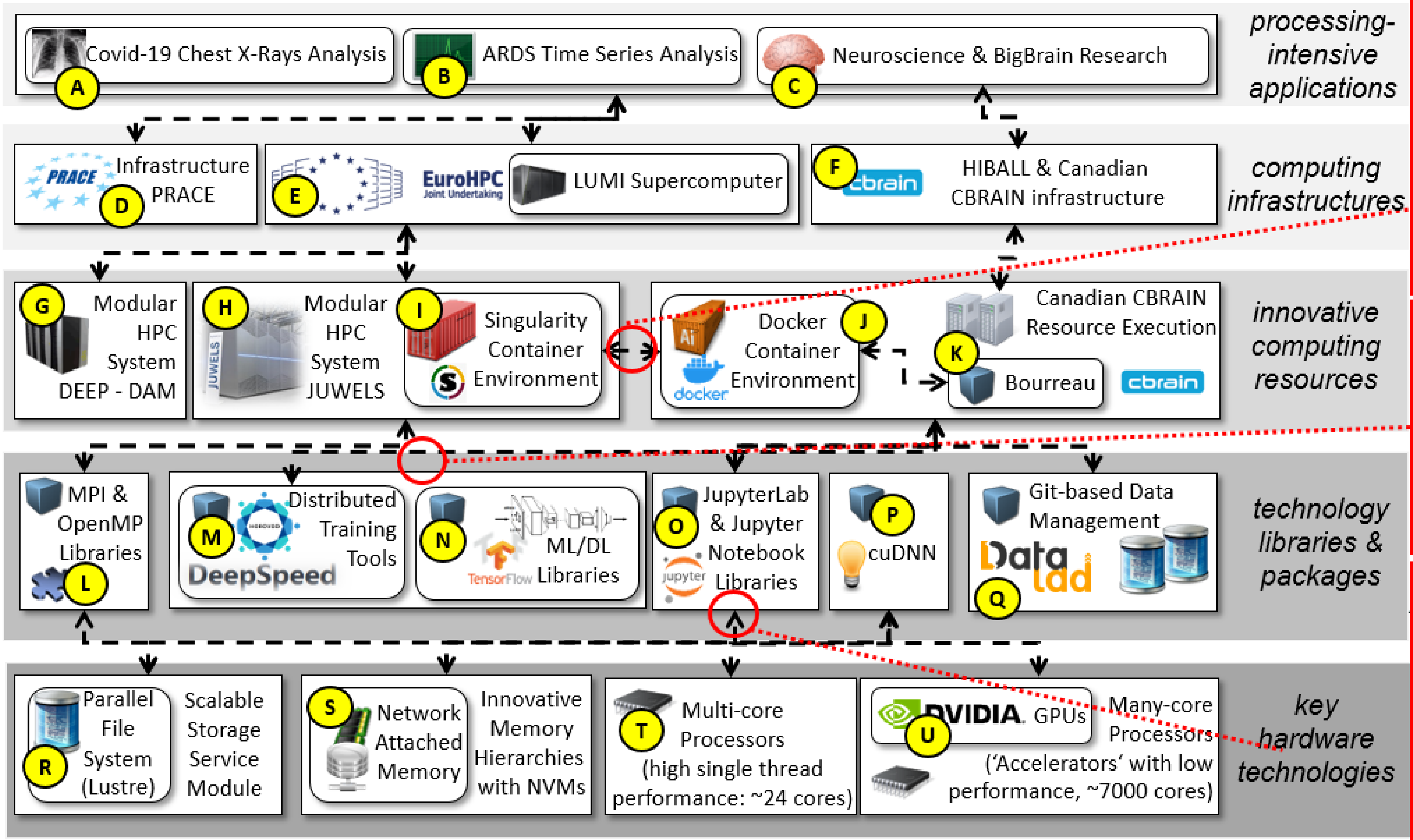
Prognosis for
Individual patient

Patient subgroups
& classifiers

Machine Learning, Patient association,
Subgroup specific prediction

[32] Alfred Winter, A. Schuppert, M. Riedel et al., Journal of Methods of Information in Medicine, 2018

OVERVIEW OF HEALTH APPLICATIONS IN HPC & AI



processing-intensive applications

- A Covid-19 Chest X-Rays Analysis
- B ARDS Time Series Analysis
- C Neuroscience & BigBrain Research

computing infrastructures

- D Infrastructure PRACE
- E EuroHPC Joint Undertaking LUMI Supercomputer
- F HIBALL & Canadian CBRAIN infrastructure

innovative computing resources

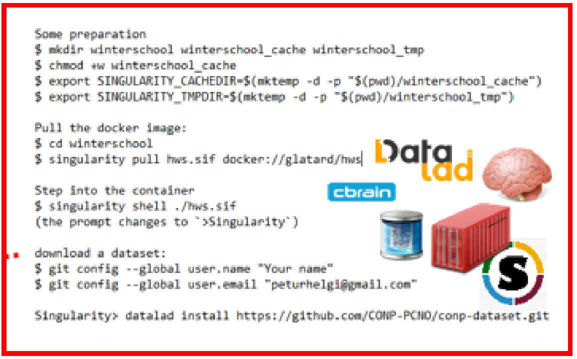
- G Modular HPC System DEEP - DAM
- H Modular HPC System JUWELS
- I Singularity Container Environment
- J Docker Container Environment
- K Canadian CBRAIN Resource Execution Bourreau cbrain

technology libraries & packages

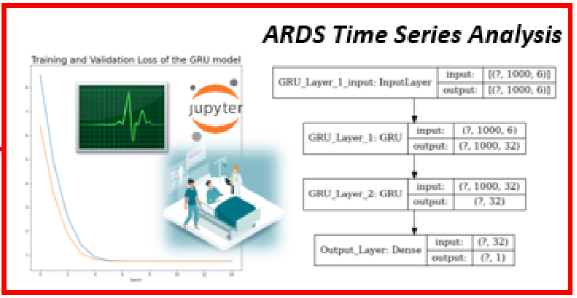
- L MPI & OpenMP Libraries
- M Distributed Training Tools DeepSpeed
- N ML/DL Libraries TensorFlow
- O JupyterLab & Jupyter Notebook Libraries
- P cuDNN
- Q Git-based Data Management DataLad

key hardware technologies

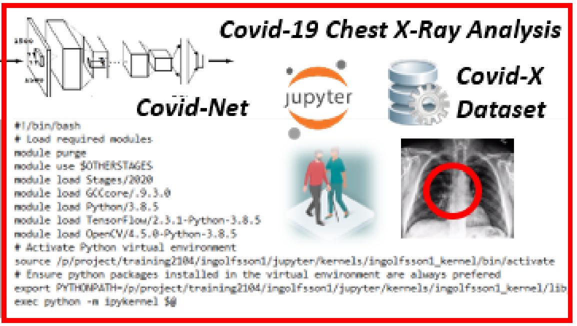
- R Parallel File System (Lustre) Scalable Storage Service Module
- S Network Attached Memory Innovative Memory Hierarchies with NVMs
- T Multi-core Processors (high single thread performance: ~24 cores)
- U NVIDIA GPUs Many-core Processors ('Accelerators' with low performance, ~7000 cores)



```
Some preparation
$ mkdir winterschool winterschool_cache winterschool_tmp
$ chmod +w winterschool_cache
$ export SINGULARITY_CACHEDIR=$(mktemp -d -p "$(pwd)/winterschool_cache")
$ export SINGULARITY_TMPDIR=$(mktemp -d -p "$(pwd)/winterschool_tmp")

Pull the docker image:
$ cd winterschool
$ singularity pull hws.sif docker://glatard/hws

Step into the container
$ singularity shell ./hws.sif
(the prompt changes to `>Singularity`)

download a dataset:
$ git config --global user.name "Your name"
$ git config --global user.email "peturhelgi@gmail.com"

Singularity> datalad install https://github.com/CONP-PCNO/conp-dataset.git
```



ARDS Time Series Analysis



Covid-19 Chest X-Ray Analysis

```
#!/bin/bash
# Load required modules
module purge
module use $OTHERSTAGES
module load Stages/2020
module load GCCcore/.9.3.0
module load Python/3.8.5
module load TensorFlow/2.3.1-Python-3.8.5
module load OpenCV/4.5.0-Python-3.8.5
# Activate Python virtual environment
source /p/project/training2104/ingolfsson1/jupyter/kernels/ingolfsson1_kernel/bin/activate
# Ensure python packages installed in the virtual environment are always prefered
export PYTHONPATH=/p/project/training2104/ingolfsson1/jupyter/kernels/ingolfsson1_kernel/lib
exec python -m ipykernel $@
```

IMPACTS OF ARTIFICIAL INTELLIGENCE IN APPLICATIONS

Selected Commercial and Industry Application Examples

IMPACTS OF ARTIFICIAL INTELLIGENCE IN APPLICATIONS

Retail Examples

ON 4 OFF

UNIVERSITÄT DUISBURG ESSEN

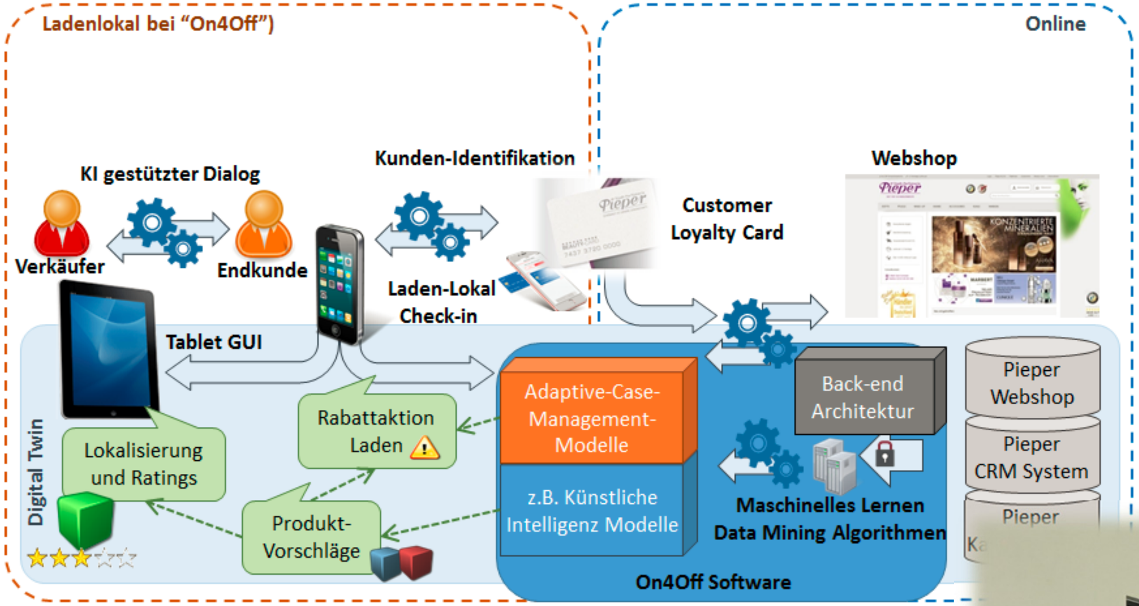
adesso

IN-telegence

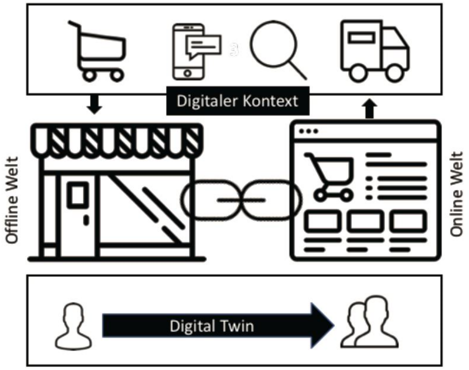
JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

Stadt-Parfümerie Pieper SEIT 1931 IN FAMILIENBESITZ

Hochschule Niederrhein University of Applied Sciences



[33] ON4OFF Web page



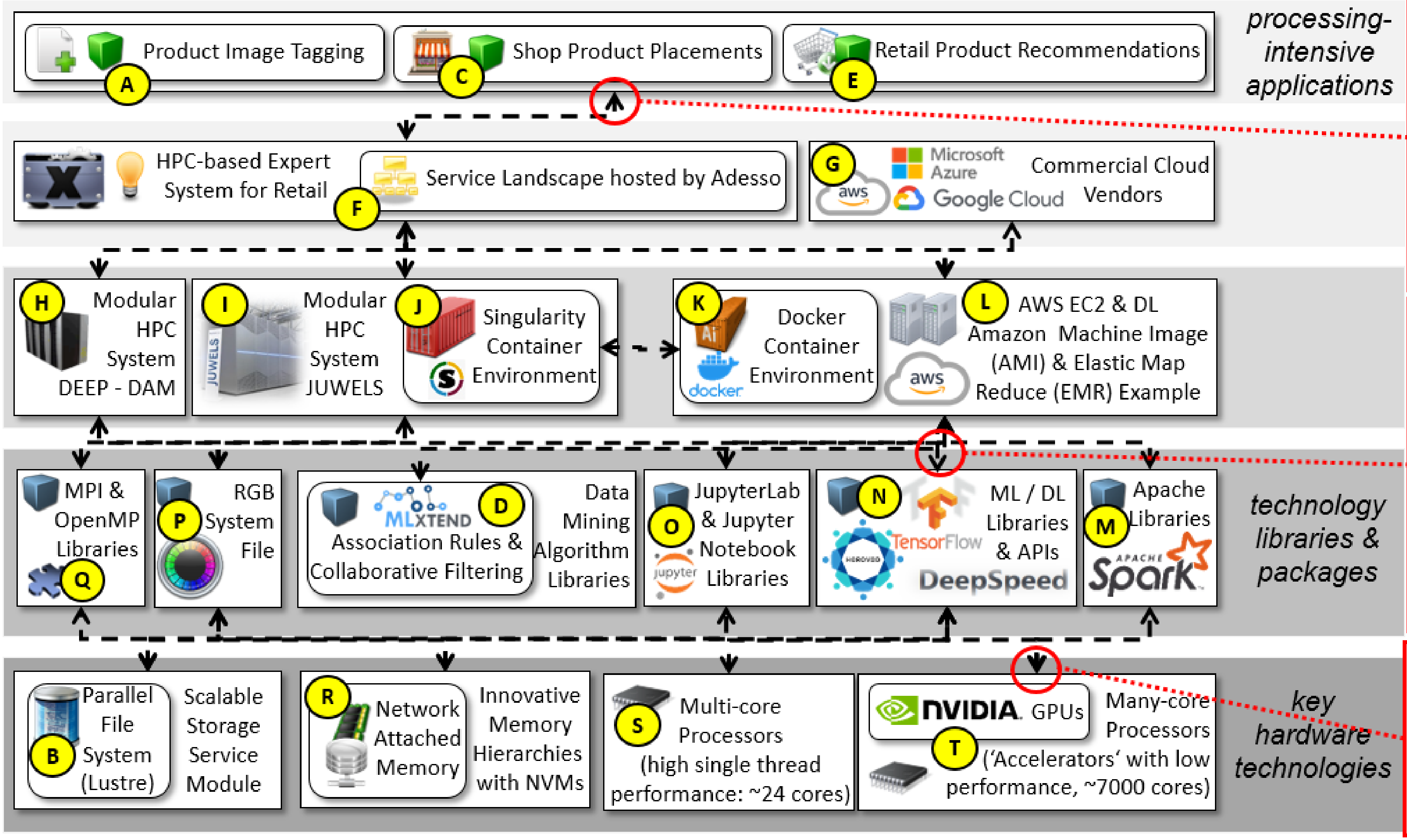
Using Machine & Deep Learning to enable better online-offline shopping in Germany

Leitmarkt Agentur.NRW

EUROPÄISCHE UNION
Investition in unsere Zukunft
Europäischer Fonds
für regionale Entwicklung

2014 2020 EFRE.NRW
Investitionen in Wachstum
und Beschäftigung

OVERVIEW OF RETAIL APPLICATIONS IN HPC & AI



processing-intensive applications

A Product Image Tagging

C Shop Product Placements

E Retail Product Recommendations

HPC-based Expert System for Retail

F Service Landscape hosted by Adesso

G Microsoft Azure, aws, Google Cloud: Commercial Cloud Vendors

H Modular HPC System DEEP - DAM

I Modular HPC System JUWELS

J Singularity Container Environment

K Docker Container Environment

L AWS EC2 & DL Amazon Machine Image (AMI) & Elastic Map Reduce (EMR) Example

technology libraries & packages

Q MPI & OpenMP Libraries

P RGB System File

D MLxtend Association Rules & Collaborative Filtering; Data Mining Algorithm Libraries

O JupyterLab & Jupyter Notebook Libraries

N ML / DL Libraries & APIs: TensorFlow, DeepSpeed

M Apache Libraries: Apache Spark

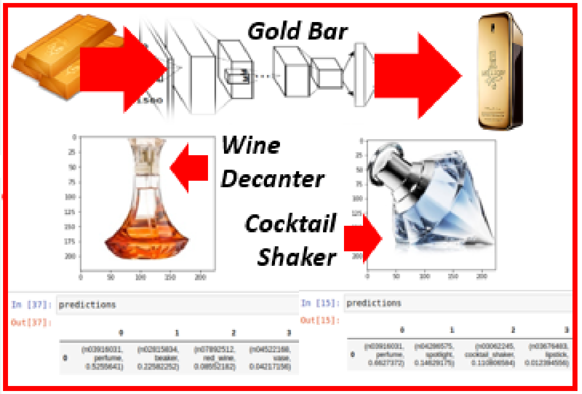
key hardware technologies

B Parallel File System (Lustre); Scalable Storage Service Module

R Network Attached Memory; Innovative Memory Hierarchies with NVMs

S Multi-core Processors (high single thread performance: ~24 cores)

T NVIDIA GPUs; Many-core Processors ('Accelerators' with low performance, ~7000 cores)



#GPUs	images/s	speedup	Performance per GPU [images/s]
1	55	1.0	55
4	178	3.2	44.5
8	357	6.5	44.63
16	689	12.5	43.06
32	1230	22.4	38.44
64	2276	41.4	35.56
128	5562	101.1	43.45

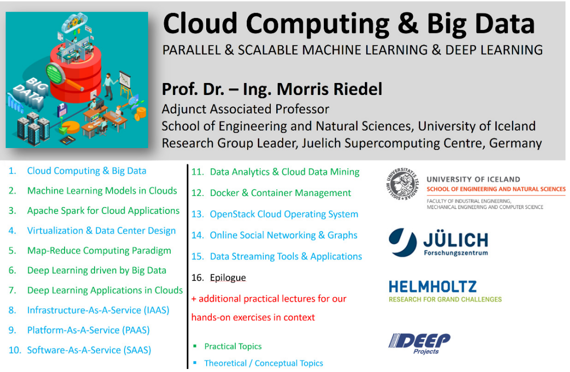
#128 GPUs in parallel

IMPACTS OF ARTIFICIAL INTELLIGENCE IN TEACHING

More and More Courses & Trainings for Machine & Deep Learning

TEACHING & TRAINING PARALLEL & SCALABLE ML/DL

Selected University Lectures at University of Iceland & Training Courses @JSC & Online via YouTube



Cloud Computing & Big Data

PARALLEL & SCALABLE MACHINE LEARNING & DEEP LEARNING

Prof. Dr. – Ing. Morris Riedel
Adjunct Associated Professor
School of Engineering and Natural Sciences, University of Iceland
Research Group Leader, Juelich Supercomputing Centre, Germany

1. Cloud Computing & Big Data
2. Machine Learning Models in Clouds
3. Apache Spark for Cloud Applications
4. Virtualization & Data Center Design
5. Map-Reduce Computing Paradigm
6. Deep Learning driven by Big Data
7. Deep Learning Applications in Clouds
8. Infrastructure-As-A-Service (IAAS)
9. Platform-As-A-Service (PAAS)
10. Software-As-A-Service (SAAS)
11. Data Analytics & Cloud Data Mining
12. Docker & Container Management
13. OpenStack Cloud Operating System
14. Online Social Networking & Graphs
15. Data Streaming Tools & Applications
16. Epilogue

+ additional practical lectures for our hands-on exercises in context

- Practical Topics
- Theoretical / Conceptual Topics

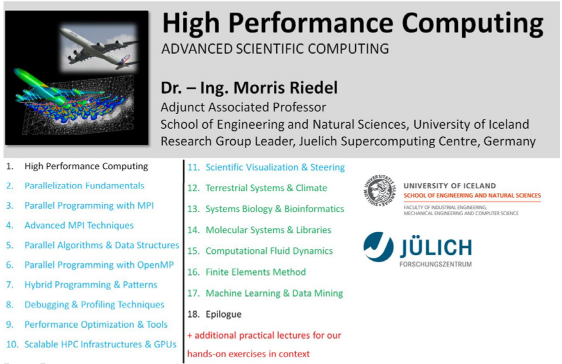
UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

JÜLICH Forschungszentrum

HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

DEEP Projects

[40] M. Riedel, 'Cloud Computing & Big Data – Parallel & Scalable Machine Learning & Deep Learning', 2018



High Performance Computing

ADVANCED SCIENTIFIC COMPUTING

Dr. – Ing. Morris Riedel
Adjunct Associated Professor
School of Engineering and Natural Sciences, University of Iceland
Research Group Leader, Juelich Supercomputing Centre, Germany

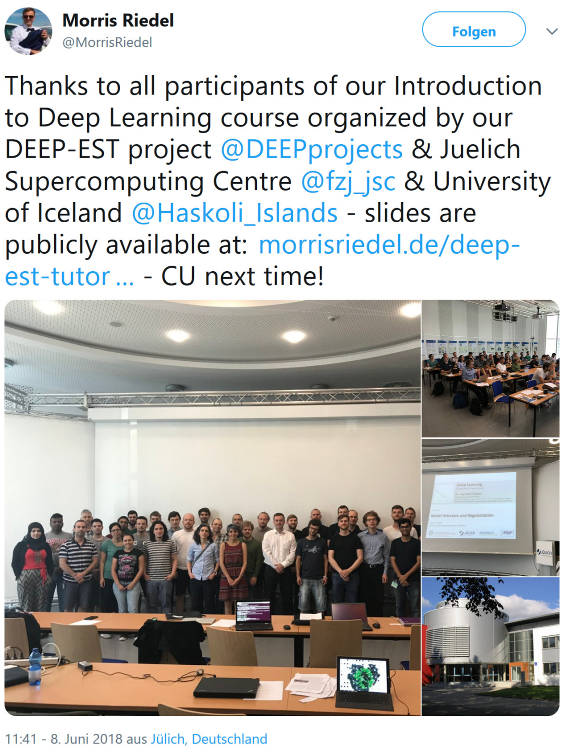
1. High Performance Computing
2. Parallelization Fundamentals
3. Parallel Programming with MPI
4. Advanced MPI Techniques
5. Parallel Algorithms & Data Structures
6. Parallel Programming with OpenMP
7. Hybrid Programming & Patterns
8. Debugging & Profiling Techniques
9. Performance Optimization & Tools
10. Scalable HPC Infrastructures & GPUs
11. Scientific Visualization & Steering
12. Terrestrial Systems & Climate
13. Systems Biology & Bioinformatics
14. Molecular Systems & Libraries
15. Computational Fluid Dynamics
16. Finite Elements Method
17. Machine Learning & Data Mining
18. Epilogue

+ additional practical lectures for our hands-on exercises in context

UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

JÜLICH FORSCHUNGSZENTRUM

[41] M. Riedel, 'High Performance Computing – Advanced Scientific Computing', 2017

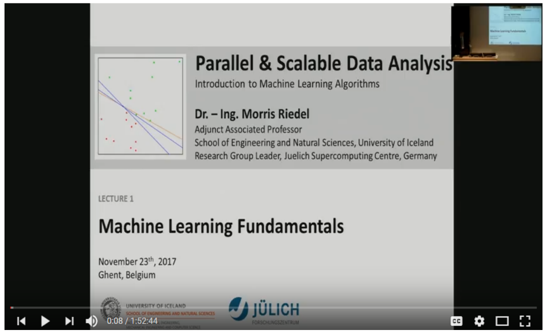


Morris Riedel
@MorrisRiedel

Thanks to all participants of our Introduction to Deep Learning course organized by our DEEP-EST project @DEEPprojects & Juelich Supercomputing Centre @fzj_jsc & University of Iceland @Haskoli_Islands - slides are publicly available at: morrisriedel.de/deep-est-tutor … - CU next time!

11:41 - 8. Juni 2018 aus Jülich, Deutschland

[42] M. Riedel et al., 'DEEP-EST Tutorial: Introduction to Deep Learning', 2018



Parallel & Scalable Data Analysis

Introduction to Machine Learning Algorithms

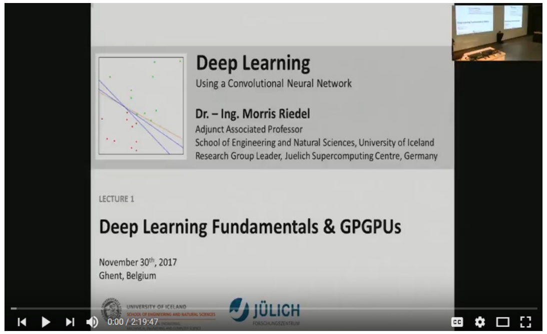
Dr. – Ing. Morris Riedel
Adjunct Associated Professor
School of Engineering and Natural Sciences, University of Iceland
Research Group Leader, Juelich Supercomputing Centre, Germany

LECTURE 1

Machine Learning Fundamentals

November 23th, 2017
Ghent, Belgium

[43] M. Riedel, 'Introduction to Machine Learning Algorithms', Invited YouTube Lecture, six lectures, University of Ghent, 2017



Deep Learning

Using a Convolutional Neural Network

Dr. – Ing. Morris Riedel
Adjunct Associated Professor
School of Engineering and Natural Sciences, University of Iceland
Research Group Leader, Juelich Supercomputing Centre, Germany

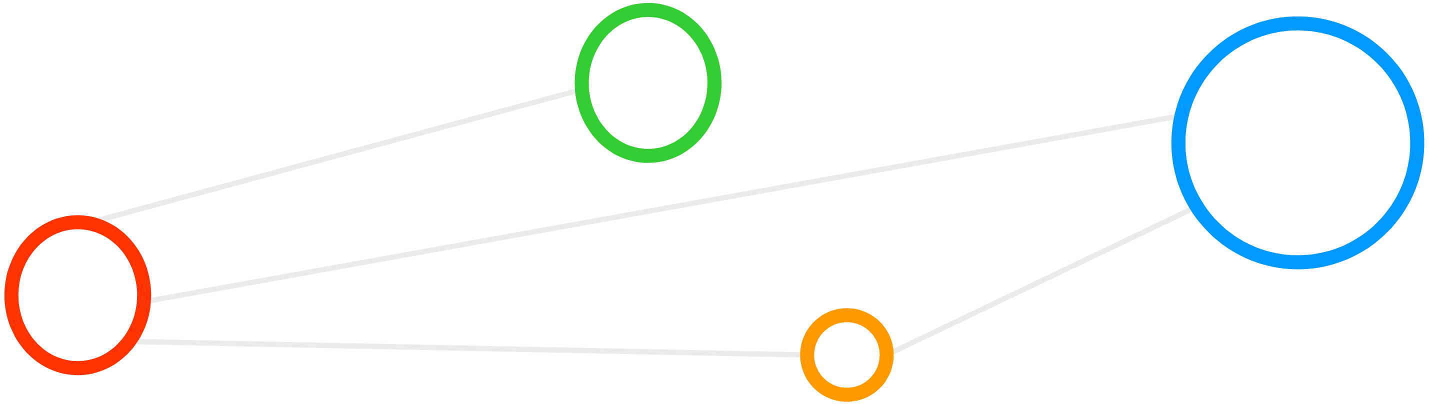
LECTURE 1

Deep Learning Fundamentals & GPGPUs

November 30th, 2017
Ghent, Belgium

[44] M. Riedel, 'Deep Learning - Using a Convolutional Neural Network', Invited YouTube Lecture, six lectures, University of Ghent, 2017

REFERENCES



REFERENCES (1)

- [1] Helmholtz Association Web Page, Online:
  https://www.helmholtz.de/en/
- [2] University of Iceland, School of Enginering and Natural Sciences, Online:
  https://english.hi.is/school_of_engineering_and_natural_sciences
- [3] Morris Riedel, ‘Deep Learning - Using a Convolutional Neural Network‘, Invited YouTube Lecture, six lectures & exercises, University of Ghent, 2017, Online:
  https://www.youtube.com/watch?v=gOL1_YIosYk&list=PLrmNhuZo9sgZUdaZ-f6OHK2yFW1kTS2qF
- [4] M. Riedel et al., ‘Introduction to Deep Learning Models‘, JSC Tutorial, three days, JSC, 2019, Online:
  http://www.morrisriedel.de/introduction-to-deep-learning-models
- [5] H. Lee et al., ‘Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations’, Online:
  http://doi.acm.org/10.1145/1553374.1553453
- [6] YouTube Video, ‘Neural Network 3D Simulation‘, Online:
  https://www.youtube.com/watch?v=3JQ3hYko51Y
- [7] A. Rosebrock, ‘Get off the deep learning bandwagon and get some perspective‘, Online:
  http://www.pyimagesearch.com/2014/06/09/get-deep-learning-bandwagon-get-perspective/
- [8] Big Data Tips – Big Data Mining & Machine Learning, Online:
  http://www.big-data.tips/

REFERENCES (2)

- [9] European Commission, COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE EUROPEAN COUNCIL, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS, EC, 2018, 2nd May, Online: https://ec.europa.eu/commission/sites/beta-political/files/communication-modern-budget-may2018_en.pdf?utm_source=POLITICO.EU&utm_campaign=e3a8a86cc6-EMAIL_CAMPAIGN_2018_05_02&utm_medium=email&utm_term=0_10959edeb5-e3a8a86cc6-189710085
- [10] NVIDIA Web Page, Online: https://www.nvidia.com/en-us/
- [11] Keras Python High-Level Deep Learning Library, Online: https://keras.io/
- [12] TensorFlow Python Low-Level Deep learning Library, Online: https://www.tensorflow.org/
- [13] Deep Learning Start-Up Beispiel Deutschland, Online: https://soccerwatch.tv/
- [14] C. Bodenstein, M. Goetz, M. Riedel, ‘Automated Soccer Scene Tracking using Deep Neural Networks’, Poster IAS Symposium, Online: https://www.researchgate.net/publication/328997974_Automated_Soccer_Scene_Tracking_Using_Deep_Neural_Networks
- [15] DEEP Series Projects Web Page, Online: http://www.deep-projects.eu/

REFERENCES (3)

- [16] ETP4HPC, Online:
  https://www.etp4hpc.eu/
- [17] E. Erlingsson, G. Cavallaro, A. Galonska, M. Riedel, H. Neukirchen, ‘Modular Supercomputing Design Supporting Machine Learning Applications‘, in conference proceedings of the 41st IEEE MIPRO 2018, May 21-25, 2018, Opatija, Croatia, Online:
  https://www.researchgate.net/publication/326708137_Modular_supercomputing_design_supporting_machine_learning_applications
- [18] M. Goetz, C. Bodenstein, M. Riedel, ‘HPDBSCAN – Highly Parallel DBSCAN’, in proceedings of the ACM/IEEE International Conference for High Performance Computing, Networking, Storage, and Analysis (SC2015), Machine Learning in HPC Environments (MLHPC) Workshop, 2015, Online:
  http://dx.doi.org/10.1145/2834892.2834894
- [19] M. Goetz, G. Cavallaro, T. Geraud, M. Book, M. Riedel, ‘Parallel Computation of Component Trees on Distributed Memory Machines‘, Journal of Transactions on Parallel and Distributed Systems, 2018, Online:
  https://www.researchgate.net/publication/325212343_Parallel_Computation_of_Component_Trees_on_Distributed_Memory_Machines
- [20] G. Cavallaro, M. Riedel, J.A. Benediktsson et al., ‘On Understanding Big Data Impacts in Remotely Sensed Image Classification using Support Vector Machine Methods’, IEEE Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 2015, DOI: 10.1109/JSTARS.2015.2458855, Online:
  https://www.researchgate.net/publication/282524415_On_Understanding_Big_Data_Impacts_in_Remotely_Sensed_Image_Classification_Using_Support_Vector_Machine_Methods
- [21] Introduction to High Performance Computing for Scientists and Engineers, Georg Hager & Gerhard Wellein, Chapman & Hall/CRC Computational Science, ISBN 143981192X, English, ~330 pages, 2010, Online:
  http://www.amazon.de/Introduction-Performance-Computing-Scientists-Computational/dp/143981192X

REFERENCES (4)

- [22] K. Hwang, G. C. Fox, J. J. Dongarra, 'Distributed and Cloud Computing', Book, Online: http://store.elsevier.com/product.jsp?locale=en_EU&isbn=9780128002049
- [23] Haut, J.M., Gallardo, J.A., Paoletti, M.E., Cavallaro, G., Plaza, J., Plaza, A., Riedel, M.: Cloud Deep Networks for Hyperspectral Image Analysis, IEEE Transactions on Geoscience and Remote Sensing, PP(99):1-17, 2019, Online: https://doi.org/10.1109/TGRS.2019.2929731, Online: https://www.researchgate.net/publication/335181248_Cloud_Deep_Networks_for_Hyperspectral_Image_Analysis
- [24] Apache Spark, Online: https://spark.apache.org/
- [25] The OpenMP API specification for parallel programming, Online: http://openmp.org/wp/openmp-specifications/
- [26] The MPI Standard, Online: http://www.mpi-forum.org/docs/
- [27] A. Herten et al., Introduction to GPU Programming JSC Course
- [28] R. Sedona, G. Cavallaro, J. Jitsev, A. Strube, M. Riedel, J.A. Benediktsson, 'Remote Sensing Big Data Classification with High Performance Distributed Deep Learning', MDPI Journal of Remote Sensing, to appear, to appear
- [29] Kaiming He et al., 'Deep Residual Learning for Image Recognition', Online: https://arxiv.org/pdf/1512.03385.pdf

REFERENCES (5)

- [30] Horovod: Uber's Open Source Distributed Deep Learning Framework for TensorFlow, Online: https://www.slideshare.net/databricks/horovod-ubers-open-source-distributed-deep-learning-framework-for-tensorflow
- [31] Th. Lippert, D. Mallmann, M. Riedel, 'Scientific Big Data Analytics by HPC', Publication Series of the John von Neumann Institute for Computing (NIC) NIC Series 48, 417, ISBN 978-3-95806-109-5, pp. 1 - 10, 2016
- [32] A. Winter, A. Schuppert, M. Riedel et al., 'Smart Medical Information Technology for Healthcare (SMITH) – Data Integration based on Interoperability Standards', Journal of Methods of Information in Medicine, 57 (S 01), e92-e105, 2018, Online: https://www.researchgate.net/publication/326465733_Smart_Medical_Information_Technology_for_Healthcare_SMITH
- [33] ON4OFF Project Web Page (German), Online: http://www.on-4-off.de
- [34] J. Dean et al., 'Large scale deep learning', Keynote GPU Technical Conference, 2015
- [35] ImageNet Web page, Online: http://image-net.org
- [36] J. Lange, G. Cavallaro, M. Goetz, E. Erlingsson, M. Riedel, 'The Influence of Sampling Methods on Pixel-Wise Hyperspectral Image Classification with 3D Convolutional Neural Networks', Proceedings of the IGARSS 2018 Conference, Online: https://www.researchgate.net/publication/328991957_The_Influence_of_Sampling_Methods_on_Pixel-Wise_Hyperspectral_Image_Classification_with_3D_Convolutional_Neural_Networks

REFERENCES (6)

- [37] Cavallaro, G., Bazi, Y., Melgani, F., Riedel, M.: Multi-Scale Convolutional SVM Networks for Multi-Class Classification Problems of Remote Sensing Images, in conference proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2019), July 28 – August 2nd, 2019, Yokohama, Japan, Online: https://www.researchgate.net/publication/337439088_Multi-Scale_Convolutional_SVM_Networks_for_Multi-Class_Classification_Problems_of_Remote_Sensing_Images
- [38] M. Riedel, 'Neural Architecture Search with Reinforcement Learning', Online: http://www.morrisriedel.de/neural-architecture-search-with-reinforcement-learning
- [39] Cheng, A.C, Lin, C.H., Juan, D.C., InstaNAS: Instance-aware Neural Architecture Search, Online: https://arxiv.org/abs/1811.10201
- [40] M. Riedel, 'Cloud Computing & Big Data – Parallel & Scalable Machine Learning & Deep Learning', 2018, Online: http://www.morrisriedel.de/cloud-computing-and-big-data-course-fall-2018
- [41] M. Riedel, 'High Performance Computing – Advanced Scientific Computing', 2017, Online: http://www.morrisriedel.de/hpc-course-fall-2017
- [42] M. Riedel et al., 'DEEP-EST Tutorial: Introduction to Deep Learning', Online: http://www.morrisriedel.de/deep-est-tutorial-deep-learning
- [43] Morris Riedel, 'Introduction to Machine Learning Algorithms', Invited YouTube Lecture, six lectures & exercises, University of Ghent, 2017, Online: https://www.youtube.com/watch?v=KgiuUZ3WeP8&list=PLrmNhuZo9sgbcWtMGN0i6G9HEvh08JG0J

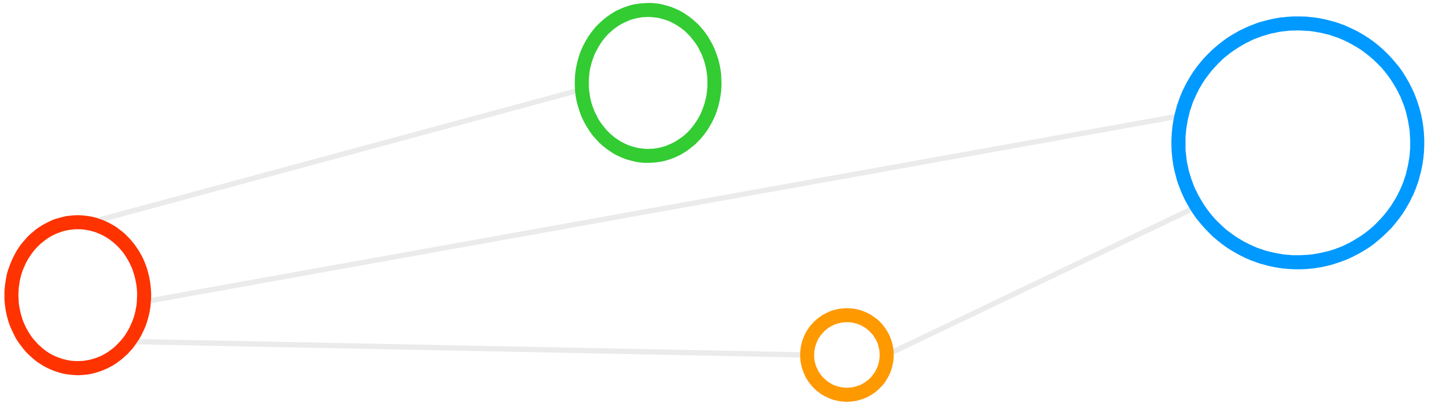
REFERENCES (7)

- [44] Morris Riedel, ‘Deep Learning - Using a Convolutional Neural Network‘, Invited YouTube Lecture, six lectures & exercises, University of Ghent, 2017, Online: https://www.youtube.com/watch?v=gOL1_YIosYk&list=PLrmNhuZo9sgZUdaZ-f6OHK2yFW1kTS2qF
- [45] Big Data Tips, ‘Quantum Computing‘, Online: http://www.big-data.tips/quantum-computing
- [46] D-Wave Systems on Twitter, Online: https://twitter.com/dwavesys
- [47] Dimitrios Marmanis et al., ‘Deep Learning Earth Obervation Classification Using ImageNet Pretrained Networks‘, IEEE Geoscience and Remote Sensing Letters, Volume 13 (1), 2016, Online: http://ieeexplore.ieee.org/document/7342907/
- [48] P. Sermanet et al., ‘OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks’, Online: http://arxiv.org/abs/1312.6229
- [49] T. Ben-Nun & T. Hoefler, ‘Demystifying Parallel and Distributed Deep Learning: An In-depth Concurrency Analysis‘, Online: http://doi.acm.org/10.1145/3320060
- [50] Launch of JUNIQ Press Release FZJ, Online: https://www.fz-juelich.de/SharedDocs/Pressemitteilungen/UK/EN/2019/2019-10-25-juniq.html

REFERENCES (8)

- [51] D-Wave Systems YouTube Channel, Online: https://www.youtube.com/channel/UC6_etbfDnWMxAuYj9qD1qmA
- [52] Visualizing the Loss of Neural Networks, online: https://www.cs.umd.edu/~tomg/projects/landscapes/
- [53] Big Data Tips Gradient Descent, Online: http://www.big-data.tips/gradient-descent
- [54] D. Willsch, M. Willsch, H. De Raedt, K. Michielsen, ‘Support Vector Machines on the D-Wave Quantum Annealer‘, Online: https://www.sciencedirect.com/science/article/pii/S001046551930342X

ACKNOWLEDGEMENTS

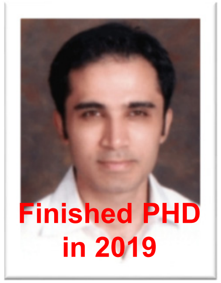


ACKNOWLEDGEMENTS

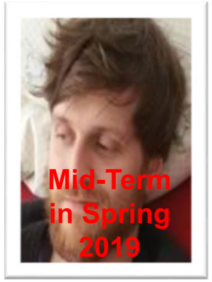
Previous & current members of the High Productivity Data Processing Research Group

PD Dr.
G. Cavallaro

Senior PhD
Student
A.S. Memon



PD Dr.
M.S. Memon



PhD Student
E. Erlingsson

PhD Student
S. Bakarat

PhD Student
R. Sedona

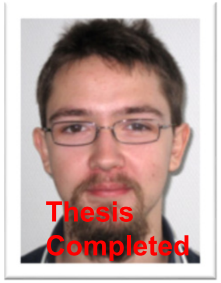
PhD Student
P. H. Einarsson

Morris Riedel @MorrisRiedel · Feb 10

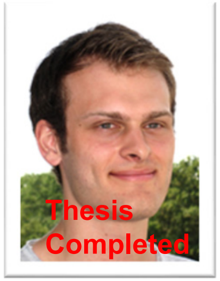
Enjoying our yearly research group dinner 'Iceland Section' to celebrate our productive collaboration of @uni_iceland @uisens @Haskoli_Islands & @fzj_jsc @fz_juelich & E.Erlingsson @ernire passed mid-term in modular supercomputing driven by @DEEPprojects - morrisriedel.de/research

You and Ernir Erlingsson

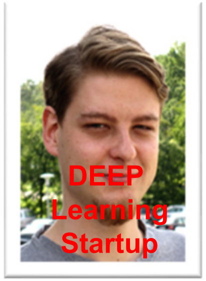
Dr. M. Goetz
(now KIT)



MSc M.
Richerzhagen
(now other division)



MSc
P. Glock
(now INM-1)



MSc
C. Bodenstein
(now
Soccerwatch.tv)

MSc G.S.
Guðmundsson
(Landsverkjun)

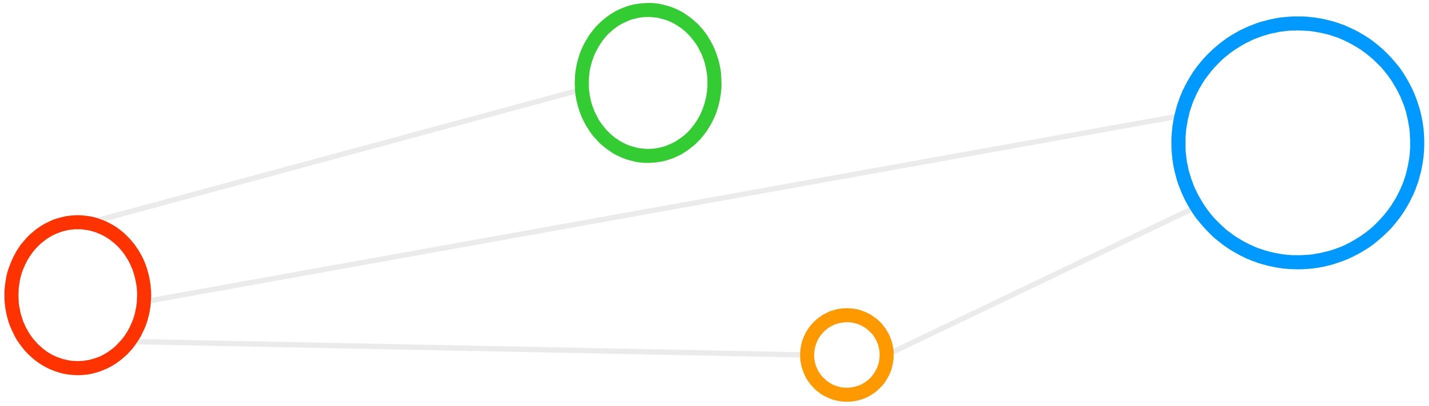
PhD Student
Reza

This research group has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 763558 (DEEP-EST EU Project) and grant agreement No 951740 (EuroCC EU Project)

THANKS

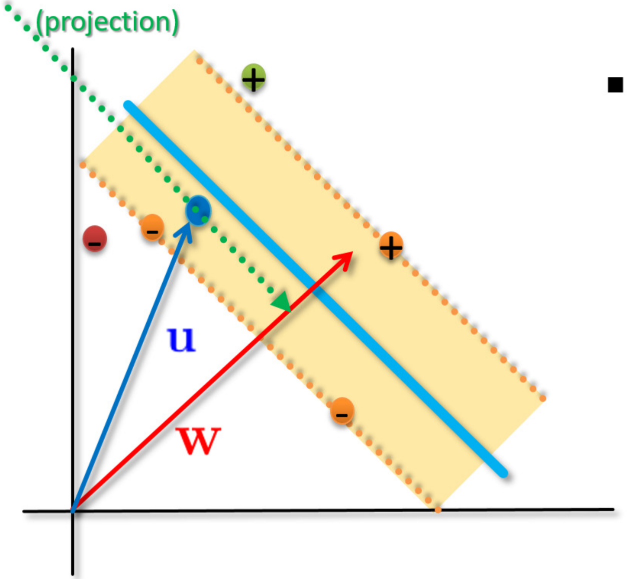
Talk shortly available under <u>www.morrisriedel.de</u>

APPENDIX: SUPPORT VECTOR MACHINES



GEOMETRIC SVM INTERPRETATION AND SETUP (1)

- Think ‘simplified coordinate system‘ and use ‘Linear Algebra‘
  - Many other samples are removed (red and green not SVs) - +
  - Vector $\mathbf{w}$ of ‘any length‘ perpendicular to the decision boundary
  - Vector $\mathbf{u}$ points to an unknown quantity (e.g. new sample to classify)
  - Is $\mathbf{u}$ on the left or right side of the decision boundary?



- Dot product $\mathbf{w} \cdot \mathbf{u} \geq C; C = -b$
  - With $\mathbf{u}$ takes the projection on the $\mathbf{w}$
  - Depending on where projection is it is left or right from the decision boundary
  - Simple transformation brings decison rule:

① $\mathbf{w} \cdot \mathbf{u} + b \geq 0$ → means +

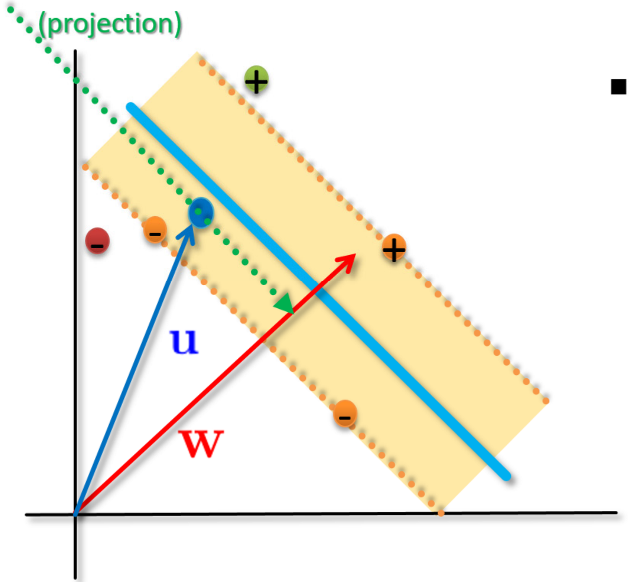
  - (given that b and $\mathbf{w}$ are unknown to us)

(constraints are not enough to fix particular b or w, need more constraints to calculate b or w)

GEOMETRIC SVM INTERPRETATION AND SETUP (2)

- Creating our constraints to get b or $\mathbf{w}$ computed
  - First constraint set for positive samples + $\mathbf{w} \cdot \mathbf{x}_+ + b \geq 1$
  - Second constraint set for negative samples - $\mathbf{w} \cdot \mathbf{x}_- + b \leq 1$
  - For mathematical convenience introduce variables (i.e. labelled samples)

$y_i = +$ for + and $y_i = -$ for -



- Multiply equations by y_i
  - Positive samples: $y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1$
  - Negative samples: $y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1$
  - Both same due to $y_i = +$ and $y_i = -$

(brings us mathematical convenience often quoted)

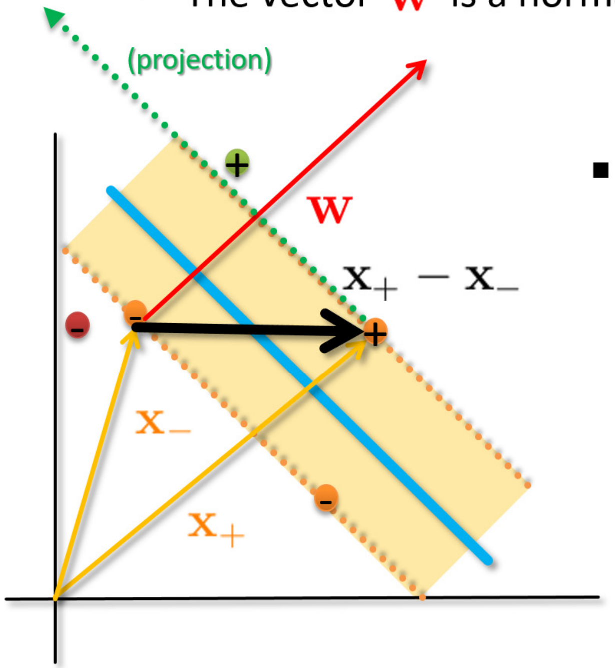
$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0$$

(additional constraints just for support vectors itself helps)

② $y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 = 0$

GEOMETRIC SVM INTERPRETATION AND SETUP (3)

- Determine the ‘width of the margin‘
  - Difference between positive and negative SVs: $\mathbf{x}_+ - \mathbf{x}_-$
  - Projection of $\mathbf{x}_+ - \mathbf{x}_-$ onto the vector $\mathbf{w}$
  - The vector $\mathbf{w}$ is a normal vector, magnitude is $\|\mathbf{w}\|$



(Dot product of two vectors is a scalar, here the width of the margin)

- Unit vector is helpful for ‘margin width‘
  - Projection (dot product) for margin width:

$$\underset{1-b}{\underset{\Downarrow}{\mathbf{x}_+}} - \underset{1+b}{\underset{\Downarrow}{\mathbf{x}_-}} \cdot \boxed{\frac{\mathbf{w}}{\|\mathbf{w}\|}} \Longrightarrow \frac{2}{\|\mathbf{w}\|}$$

(unit vector)

③

- When enforce constraint:

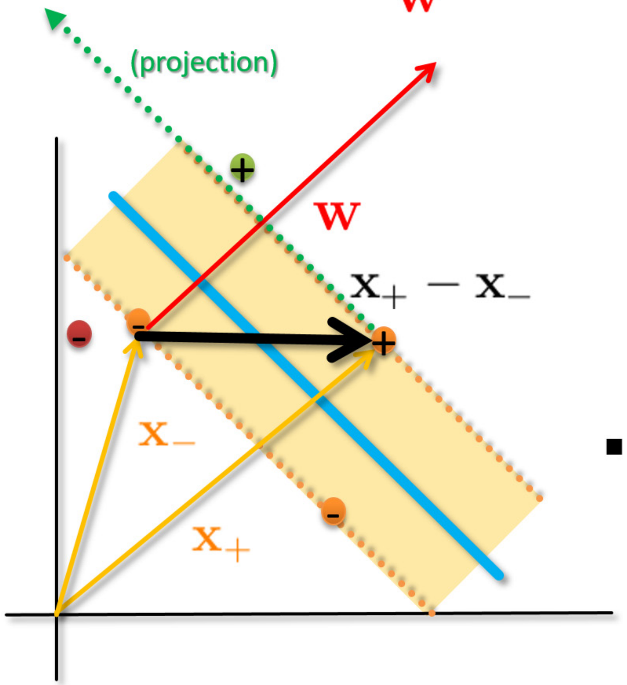
② $y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 = 0$

$y_i = +$ ⊕

$y_i = -$ ⊖

CONSTRAINED OPTIMIZATION STEPS SVM (1)

- Use ‘constraint optimization‘ of mathematical toolkit
  - Idea is to ‘maximize the width‘ of the margin: $max \frac{2}{\|\mathbf{w}\|}$ (drop the constant 2 is possible here)



➡ $max \frac{1}{\|\mathbf{w}\|}$ (equivalent)

➡ $min \|\mathbf{w}\|$ (equivalent for max)

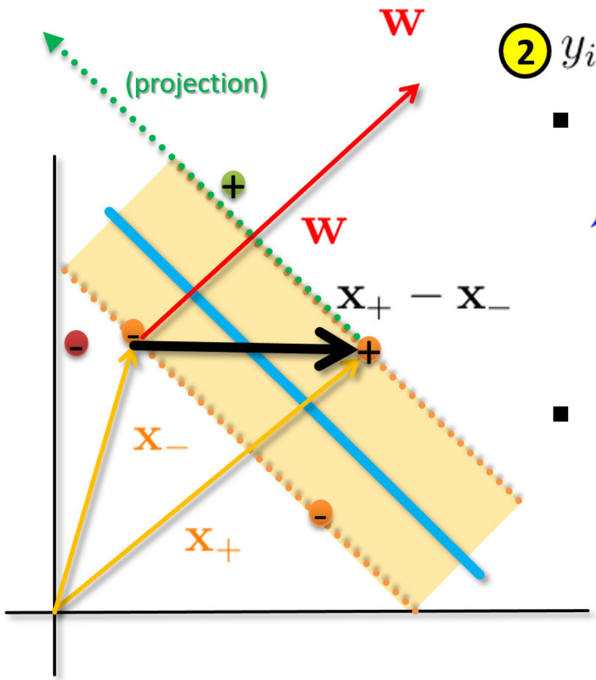
➡ $min \frac{1}{2}\|\mathbf{w}\|^2$ (mathematical convenience) ③

- Next: Find the extreme values
  - Subject to constraints

② $y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 = 0$

CONSTRAINED OPTIMIZATION STEPS SVM (2)

- Use ‘Lagrange Multipliers‘ of mathematical toolkit
  - Established tool in ‘constrained optimization‘ to find function extremum
  - ‘Get rid‘ of constraints by using Lagrange Multipliers ④



② $y_i(\mathbf{x}_i \cdot \mathbf{w} + b - 1) = 0$

- Introduce a multiplier for each constraint

$$\mathcal{L}(\alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum \alpha_i [y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1]$$

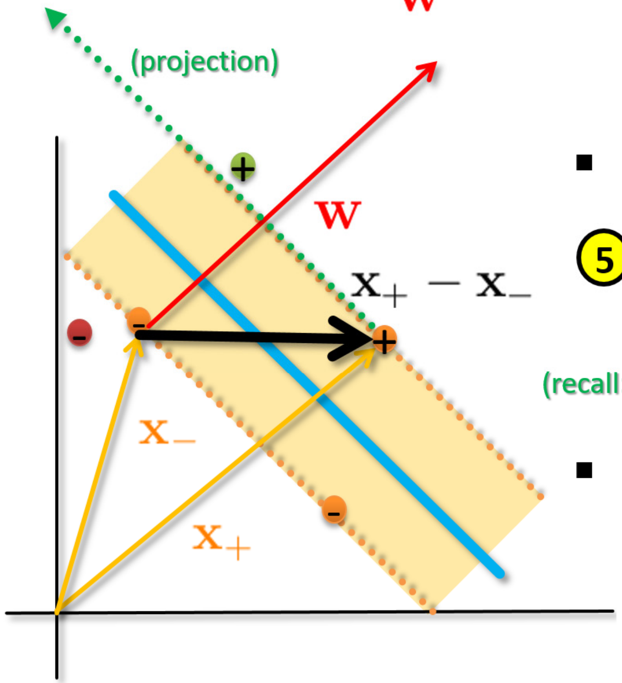
(interesting: non zero for support vectors, rest zero)

- Find derivatives for extremum & set 0
  - But two unknowns that might vary
  - First differentiate w.r.t. $\mathbf{w}$
  - Second differentiate w.r.t. b

(derivative gives the gradient, setting 0 means extremum like min)

CONSTRAINED OPTIMIZATION STEPS SVM (3)

- Lagrange gives: $\mathcal{L}(\alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum \alpha_i[y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1]$



- First differentiate w.r.t $\mathbf{w}$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \boxed{\sum \alpha_i y_i \mathbf{x}_i} = 0$$

(derivative gives the gradient, setting 0 means extremum like min)

- Simple transformation brings:

5 $\mathbf{w} = \boxed{\sum \alpha_i y_i \mathbf{x}_i}$ (i.e. vector is linear sum of samples)

(recall: non zero for support vectors, rest zero → even less samples)

- Second differentiate w.r.t. b

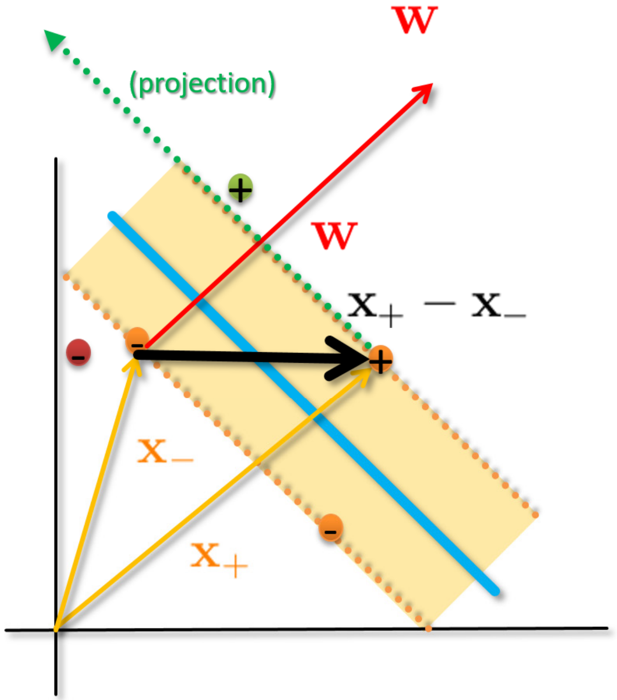
$$\frac{\partial \mathcal{L}}{\partial b} = -\sum \alpha_i y_i = 0 \Rightarrow \sum \alpha_i y_i = 0$$ 5

CONSTRAINED OPTIMIZATION STEPS SVM (4)

- Lagrange gives: $\mathcal{L}(\alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum \alpha_i[y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1]$

(plug into)

- Find minimum
  - Quadratic optimization problem
  - Take advantage of ⑤ $\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i$



$$\begin{aligned}\mathcal{L} = \frac{1}{2}\left(\sum \alpha_i y_i \mathbf{x}_i\right) \cdot \left(\sum \alpha_j y_j \mathbf{x}_j\right) \\ - \sum \alpha_i y_i \mathbf{x}_i \cdot \left(\sum \alpha_j y_j \mathbf{x}_j\right) \\ - \sum \alpha_i y_i b + \sum \alpha_i\end{aligned}$$

(b constant in front sum)

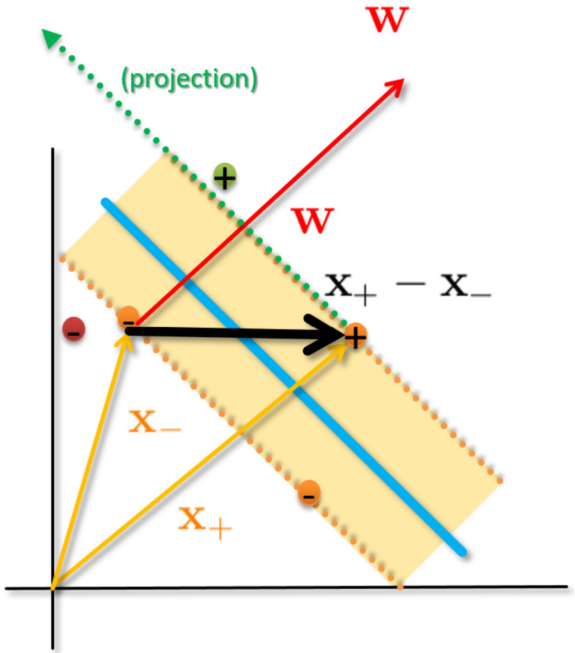
⑤ $\sum \alpha_i y_i = 0$

CONSTRAINED OPTIMIZATION STEPS SVM (5)

- Rewrite formula: $\mathcal{L} = \frac{1}{2}\boxed{\begin{array}{l}(\sum \alpha_i y_i \mathbf{x}_i) \cdot (\sum \alpha_j y_j \mathbf{x}_j) \\ - \sum \alpha_i y_i \mathbf{x}_i \cdot (\sum \alpha_j y_j \mathbf{x}_j)\end{array}}$ (the same)

$$- \cancel{\sum \alpha_i y_i b} + \sum \alpha_i$$

(was 0)



(results in)

(optimization depends only on dot product of samples)

$$\mathcal{L} = \sum \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \boxed{\mathbf{x}_i \cdot \mathbf{x}_j}$$ 6

- Equation to be solved by some quadratic programming package

USE OF SVM CLASSIFIER TO PERFORM CLASSIFICATION

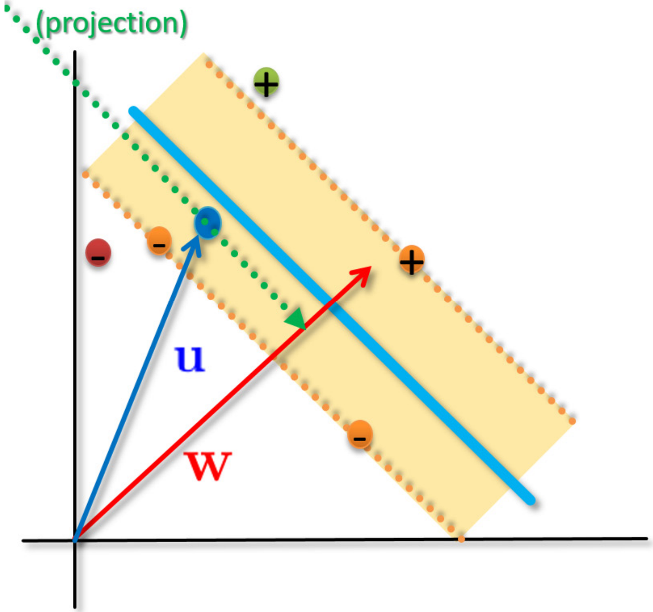
- Use findings for decision rule

⑤ $\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i$

⇓

① $\mathbf{w} \cdot \mathbf{u} + b \geq 0$ + ⇒ $\sum \alpha_i y_i \boxed{\mathbf{x}_i \cdot \mathbf{u}_i} + b \geq 0$ +

(decision rule also depends on dotproduct)



CONSTRAINED OPTIMIZATION STEPS & DOT PRODUCT

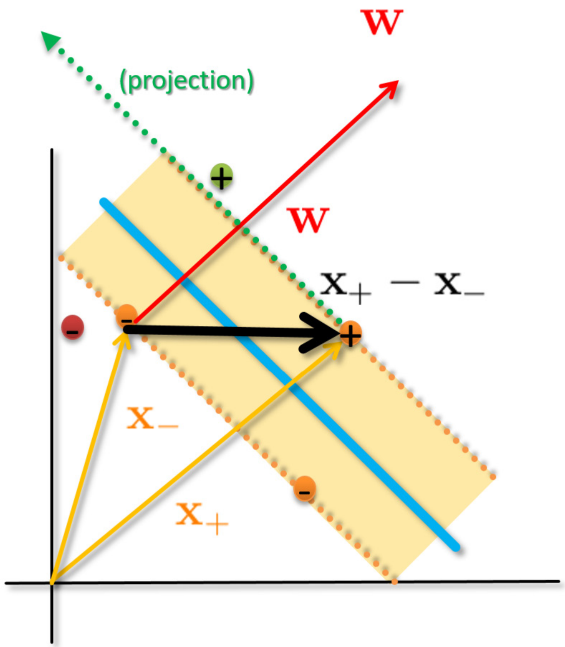
- Rewrite formula: $\mathcal{L} = \frac{1}{2}\boxed{(\sum \alpha_i y_i \mathbf{x}_i) \cdot (\sum \alpha_j y_j \mathbf{x}_j)}$

(the same)

$$\boxed{- \sum \alpha_i y_i \mathbf{x}_i \cdot (\sum \alpha_j y_j \mathbf{x}_j)}$$

$$-\cancel{\sum \alpha_i y_i b} + \sum \alpha_i$$

(was 0)



$\Downarrow$ (results in)

(optimization depends only on dot product of samples)

$$\mathcal{L} = \sum \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \boxed{\mathbf{x}_i \cdot \mathbf{x}_j}$$ 6

- Equation to be solved by some quadratic programming package

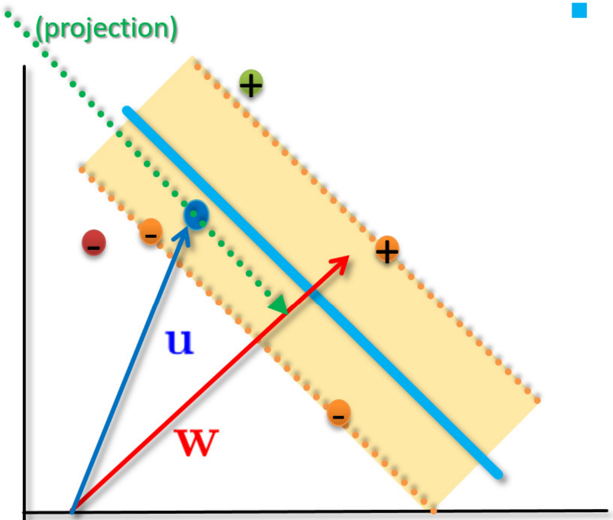
KERNEL METHODS & DOT PRODUCT DEPENDENCY

- Use findings for decision rule

5 $\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i$

(decision rule also depends on dotproduct)

1 $\mathbf{w} \cdot \mathbf{u} + b \geq 0$ + $\Rightarrow$ $\sum \alpha_i y_i \boxed{\mathbf{x}_i \cdot \mathbf{u}_i} + b \geq 0$ +



- Dotproduct enables nice more elements
  - E.g. consider non linearly seperable data
  - Perform non-linear transformation Φ of the samples into another space (work on features)

$$\mathcal{L} = \sum \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \boxed{\mathbf{x}_i \cdot \mathbf{x}_j}$$ 6

(optimization depends only on dot product of samples)

$\Rightarrow$ $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$ (in optimization)

$\Rightarrow$ $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{u}_i)$ (for decision rule above too)

(kernel trick is substitution) $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$ 7 $\boxed{K(\mathbf{x}_i, \mathbf{x}_j)} = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$ (trusted Kernel avoids to know Phi)