Artificial Intelligence Data Analysis (AIDA)

1st School for Heliophysicists

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

@Morris Riedel @MorrisRiedel @MorrisRiedel

LECTURE 1

Introduction & Differences between AI, ML, NN, and Big Data

January 20, 2020
CINECA, Bologna, Italy

UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE
JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE
DEEP Projects
HELMHOLTZAI | ARTIFICIAL INTELLIGENCE COOPERATION UNIT

Outline of the School

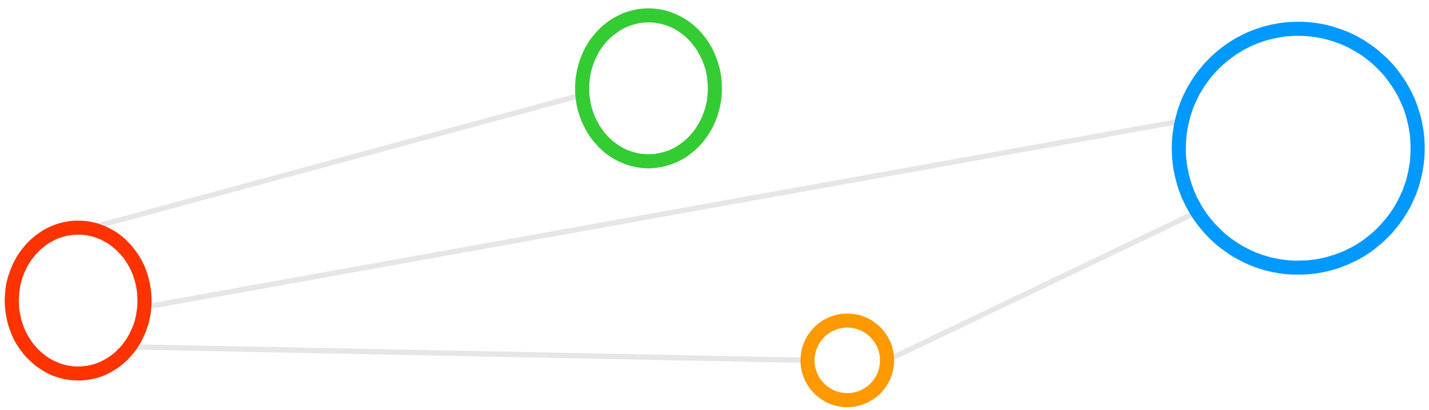
Time	**Day 1**	**Day 2**	**Day 3**
9 - 10	Welcome and intro to the school (Giovanni Lapenta, Jorge Amaya)	Space missions data acquisition (Hugo Breuillard)	Review of ML applied to heliophysics (Peter Wintoft)
10 - 11	Introduction and differences between AI, ML, NN and Big Data (Morris Riedel)	Data manipulation in python with pandas, xarray, and additional python tools (Geert Jan Bex)	Review of ML applied to heliophysics (Peter Wintoft)
	Coffee break	Coffee break	Coffee break
11:30 - 12:30	Unsupervised learning (Morris Riedel)	Feature engineering and data reduction (Geert Jan Bex)	Reinforcement learning (Morris Riedel)
	Lunch	Lunch	Lunch
14 - 15	Unsupervised learning (Morris Riedel)	Data reduction and visualization (Geert Jan Bex)	Physics informed ML (Romain Dupuis)
15 -16	Supervised learning (Morris Riedel)	CNN, DNN (Morris Riedel)	Explainable AI (Jorge Amaya)
	Coffee break	Coffee break	Coffee break
16:30 - 18:00	Supervised learning (Morris Riedel)	CNN, DNN (Morris Riedel)	Performance and tuning of ML (Morris Riedel)

Outline

- Introduction to Machine Learning (ML)
  - Terminology & Differences between AI, ML & DL
  - Machine Learning Models & Learning Approaches
  - Machine Learning Prerequisites & Challenges
  - Simple Application Example to Understand Fundamentals
  - Perceptron Learning Model & Learning from Datasets

- Relationships to High Performance Computing (HPC) & Big Data
  - Innovative Deep Learning (DL) Techniques & Short Examples
  - Understanding Deep Learning Momentum & Startup Example
  - Complex Relationships: ML & DL vs. HPC/Clouds & Big Data
  - DEEP Series of Projects & Modular Supercomputing Architecture (MSA)
  - Hands-On Training System – Data Analytics Module (DAM)

Introduction to Machine Learning (DL)



Terminology & Differences between AI, ML & DL

Artificial Intelligence (AI)

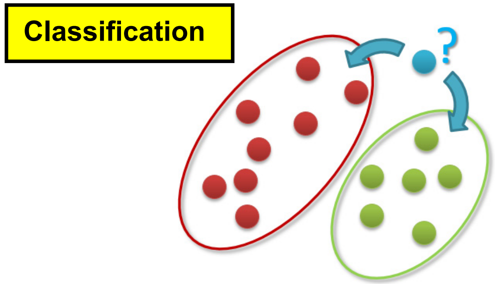
A wide area of techniques and tools that enable computers to mimic human behaviour (+ robotics)

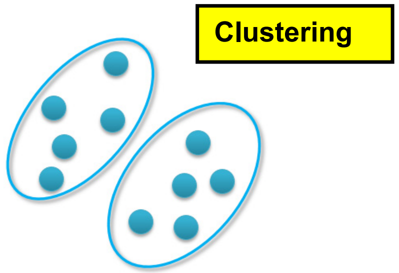
Machine Learning (ML)

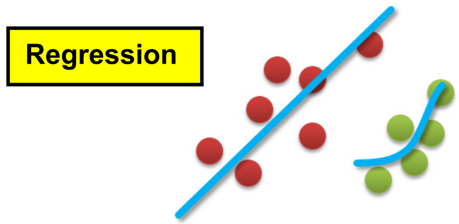
Learning from data without explicitly being programmed with common programming languages

Deep Learning (DL)

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

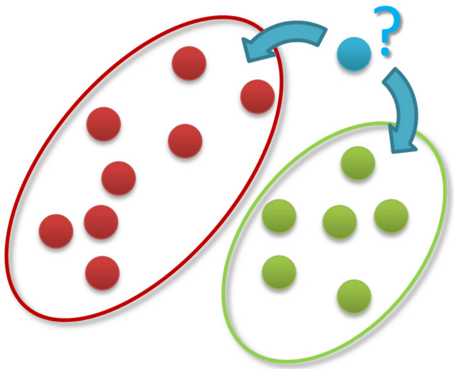






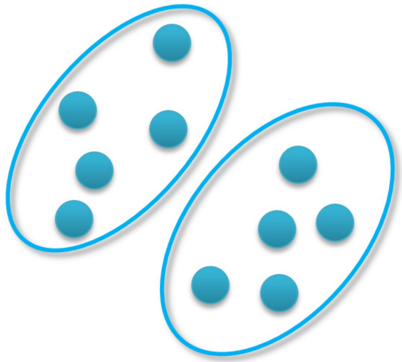
Machine Learning Models – Short Overview

Classification



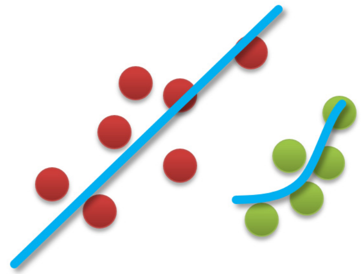
- Groups of data exist
- New data classified to existing groups

Clustering



- No groups of data exist
- Create groups from data close to each other

Regression



- Identify a line with a certain slope describing the data

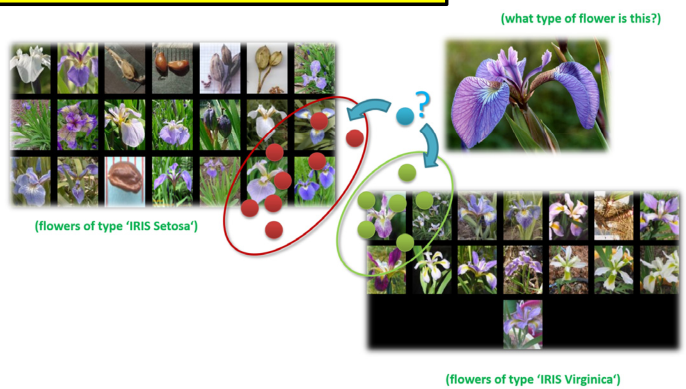
- **Machine learning methods can be roughly categorized in classification, clustering, or regression augmented with various techniques for data exploration, selection, or reduction – despite the momentum of deep learning, traditional machine learning algorithms are still widely relevant today**

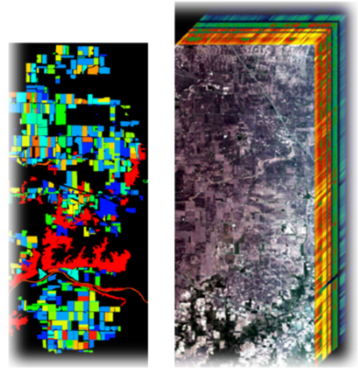
Learning Approaches – What means Learning from data?

- **The basic meaning of learning is 'to use a set of observations to uncover an underlying process'**
- **The three different learning approaches are supervised, unsupervised, and reinforcement learning**

[14] Image sources: Species Iris Group of North America Database, www.signa.org

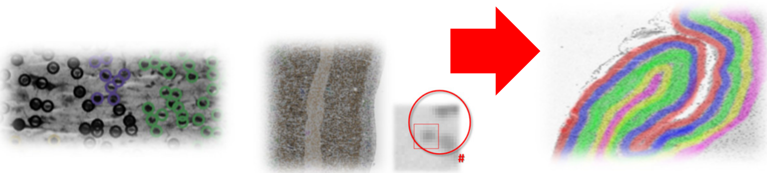
- Supervised Learning
  - Majority of methods follow this approach in this course
  - Example: credit card approval based on previous customer applications

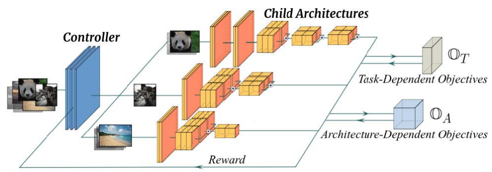




- Unsupervised Learning
  - Often applied before other learning → higher level data representation
  - Example: Coin recognition in vending machine based on weight and size

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





- Reinforcement Learning
  - Typical 'human way' of learning
  - Example: Toddler tries to touch a hot cup of tea (again and again)

➢ **Day 1 offers details about unsupervised & supervised learning with examples & Day 3 offers an introduction to reinforcement learning**

Learning Approaches – Supervised Learning with a Simple Example

- Each observation of the predictor measurement(s) has an associated response measurement:
  - Input $\mathbf{x} = x_1, ..., x_d$
  - Output $y_i, i = 1, .., n$
  - Data $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)$
  - (the output guides the learning process as a 'supervisor')

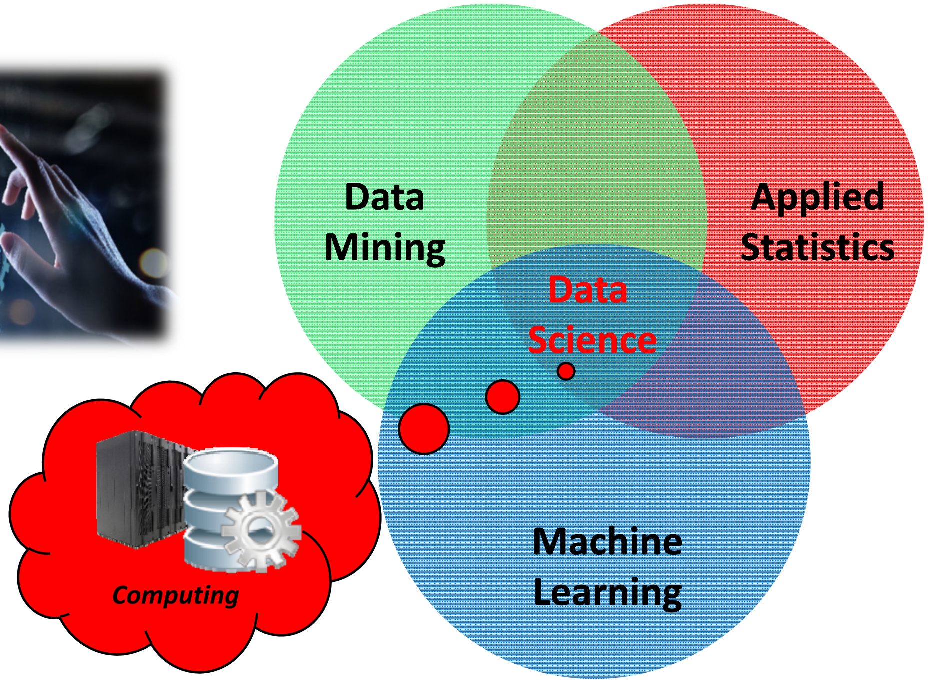
- Goal: Fit a model that relates the response to the predictors
  - Prediction: Aims of accurately predicting the response for future observations
  - Inference: Aims to better understanding the relationship between the response and the predictors

> - **Supervised learning approaches fits a model that related the response to the predictors**
> - **Supervised learning approaches are used in classification algorithms such as SVMs**
> - **Supervised learning works with data = [input, correct output]**

Machine Learning Prerequisites & Challenges

1. Some pattern exists
2. No exact mathematical formula
3. Data exists

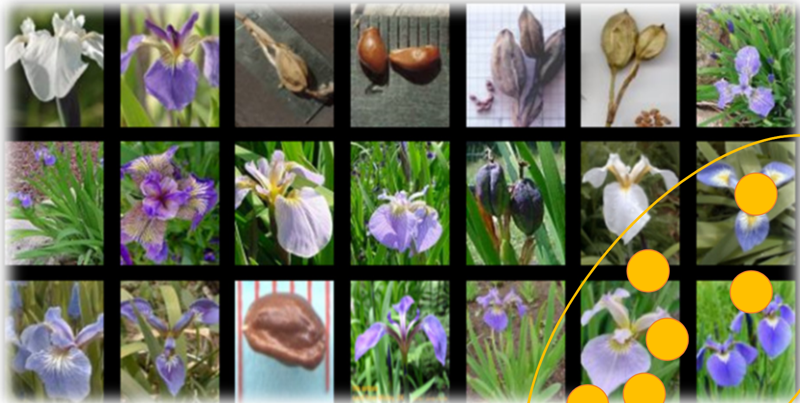
- Idea 'Learning from Data'
  - Shared with a wide variety of other disciplines
  - E.g. signal processing, data mining, etc.
- Challenges
  - Data is often complex
  - Learning from data requires processing time → Clouds



- Machine learning is a very broad subject and goes from very abstract theory to extreme practice ('rules of thumb')
- Training machine learning models needs processing time

Simple Application Example: Classification of a Flower

(what type of flower is this?)



(flowers of type ‘IRIS Setosa‘)

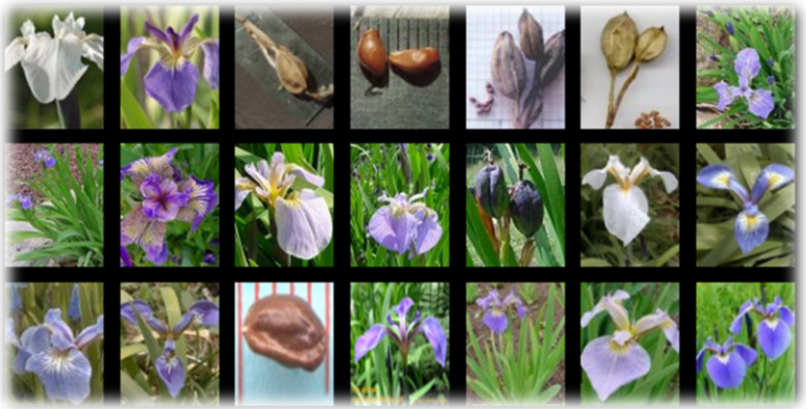
- Groups of data exist
- New data classified to existing groups

[14] Image sources: Species Iris Group of North America Database, www.signa.org

(flowers of type ‘IRIS Virginica‘)

The Learning Problem in the Example

(flowers of type 'IRIS Setosa') (flowers of type 'IRIS Virginica')



[14] Image sources: Species Iris Group of North America Database, www.signa.org

Learning problem: A prediction task

- Determine whether a new Iris flower sample is a "Setosa" or "Virginica"
- Binary (two class) classification problem
- What attributes about the data help?

(what type of flower is this?)

Feasibility of Machine Learning in this Example

1. Some pattern exists:
   - Believe in a ‘pattern with ‘petal length‘ & ‘petal width‘ somehow influence the type
2. No exact mathematical formula
   - To the best of our knowledge there is no precise formula for this problem
3. Data exists
   - Data collection from UCI Dataset „Iris“
   - 150 labelled samples (aka ‘data points‘)
   - Balanced: 50 samples / class

[15] UCI Machine Learning Repository Iris Dataset

[16] Image source: Wikipedia, Sepal

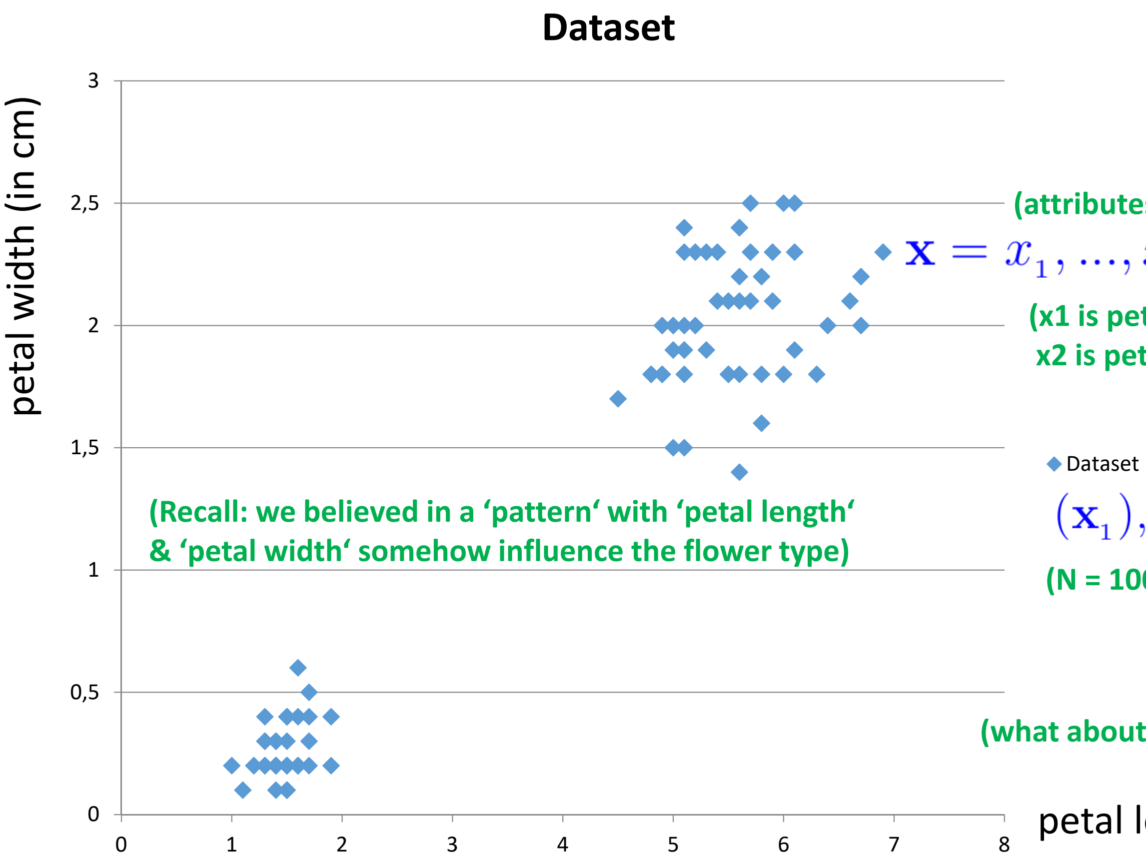
(four data attributes for each sample in the dataset)

- sepal length in cm
- sepal width in cm
- petal length in cm
- petal width in cm

(one class label for each sample in the dataset)

- class: Iris Setosa, or Iris Versicolour, or Iris Virginica

Check Preparation Phase: Plotting the Data (Two Classes)



(attributes with d=2)

$$\mathbf{x} = x_1, \ldots, x_d$$

(x1 is petal length,
x2 is petal width)

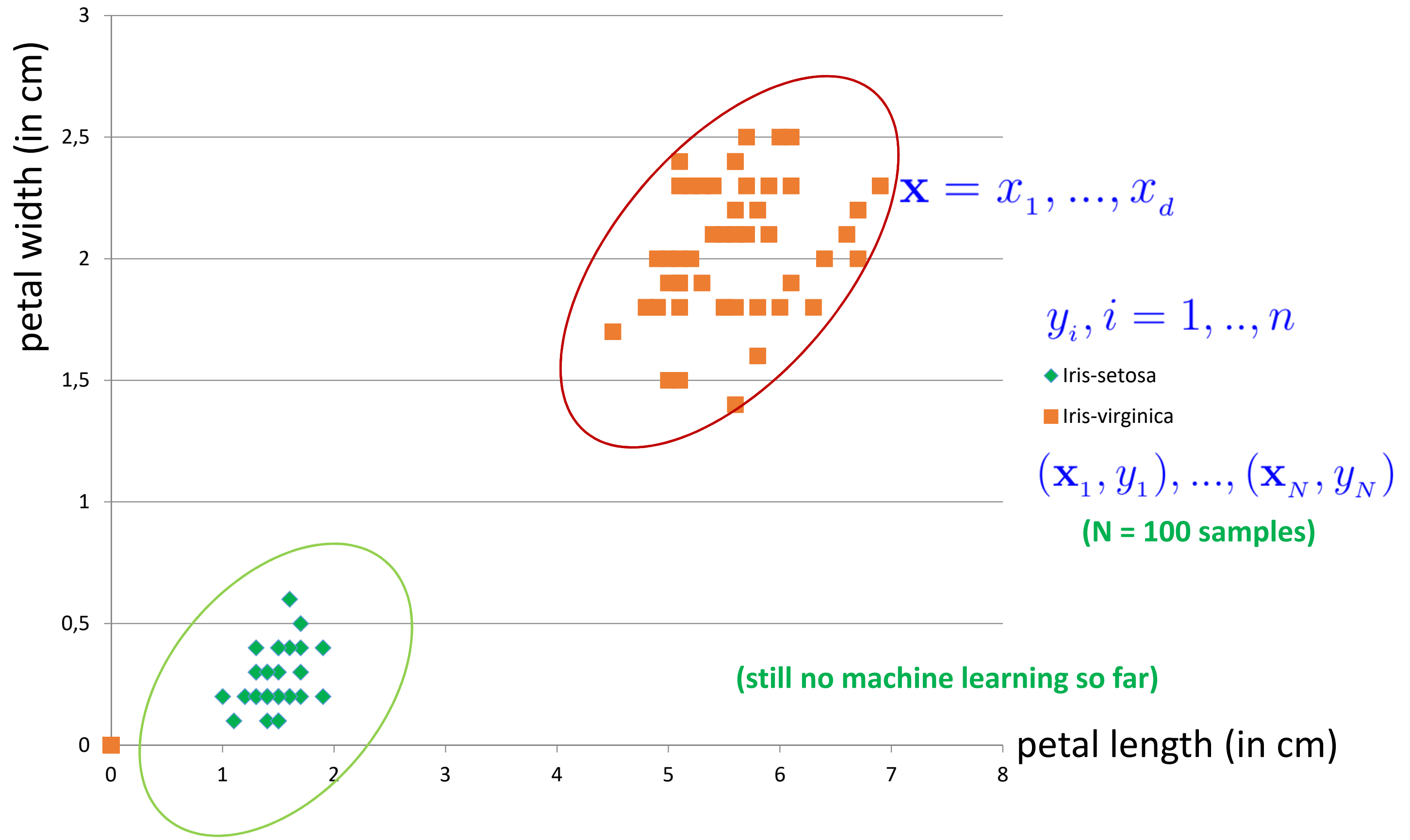
$$(\mathbf{x}_1), \ldots, (\mathbf{x}_N)$$

(N = 100 samples)

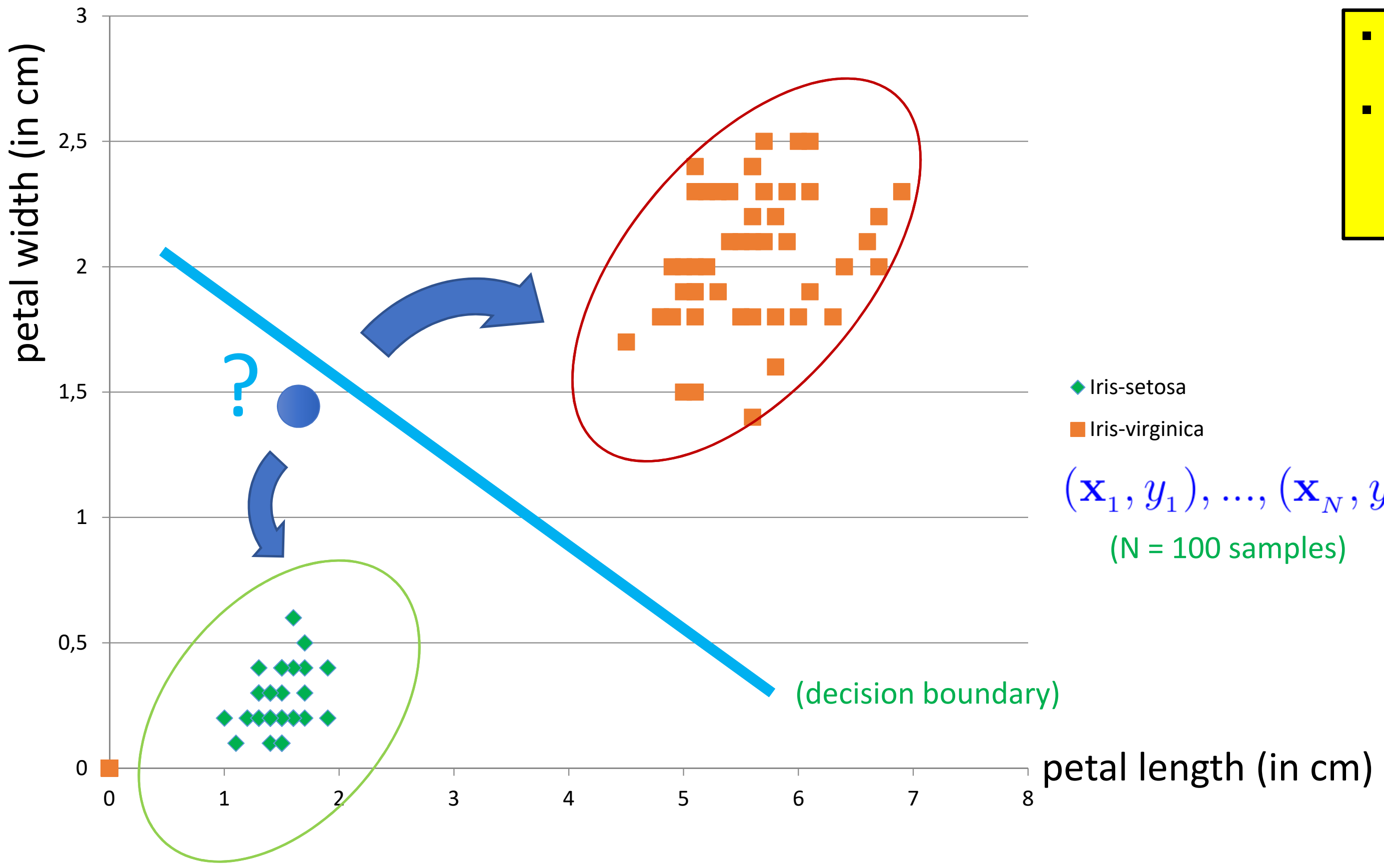
(what about the class labels?)

petal length (in cm)

Check Preparation Phase: Class Labels



Linearly Seperable Data & Linear Decision Boundary



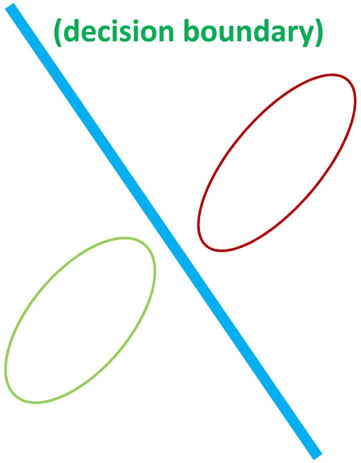
- **The data is linearly seperable (rarely in practice)**
- **A line becomes a decision boundary to determine if a new data point is class red/green**

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$$

(N = 100 samples)

Separating Line & Mathematical Notation

- Data exploration results
  - A line can be crafted between the classes since linearly seperable data
  - All the data points representing Iris-setosa will be below the line
  - All the data points representing Iris-virginica will be above the line
- More formal mathematical notation
  - Input: $\mathbf{x} = x_1, \ldots, x_d$ **(attributes of flowers)**
  - Output:
    class +1 (Iris-virginica)
    or class -1 (Iris-setosa)



Iris-virginica if $\sum_{i=1}^{d} w_i x_i > threshold$

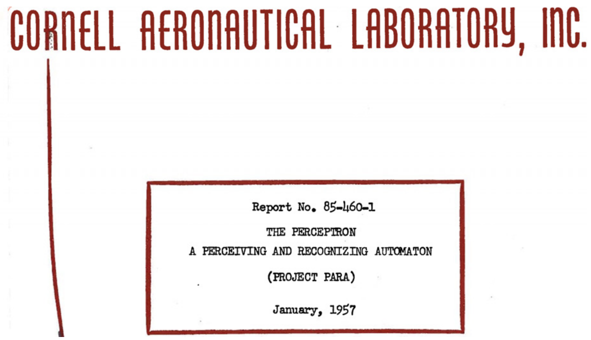
Iris-setosa if $\sum_{i=1}^{d} w_i x_i < threshold$

(w_i and threshold are still unknown to us)

$$sign\left(\left(\sum_{i=1}^{d} w_i x_i\right) - threshold\right)$$ **(compact notation)**

A Simple Linear Learning Model – The Perceptron

- Human analogy in learning
  - Human brain consists of nerve cells called neurons
  - Human brain learns by changing the strength of neuron connections (w_i) upon repeated stimulation by the same impulse (aka a 'training phase')
  - Training a perceptron model means adapting the weights w_i
  - Done until they fit input-output relationships of the given 'training data'


CORNELL AERONAUTICAL LABORATORY, INC.

Report No. 85-460-1

THE PERCEPTRON

A PERCEIVING AND RECOGNIZING AUTOMATON

(PROJECT PARA)

January, 1957

[18] F. Rosenblatt, 1957

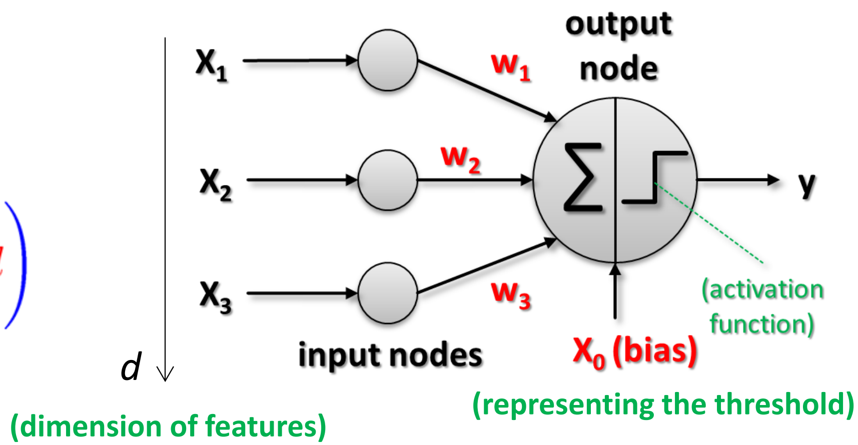
$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$$

(training data)

(modelled as bias term)

$$sign\left(\underbrace{\left(\sum_{i=1}^{d} w_i x_i\right) - threshold}_{\text{(the signal)}}\right)$$

(activation function, +1 or -1)



Perceptron – Example of a Boolean Function

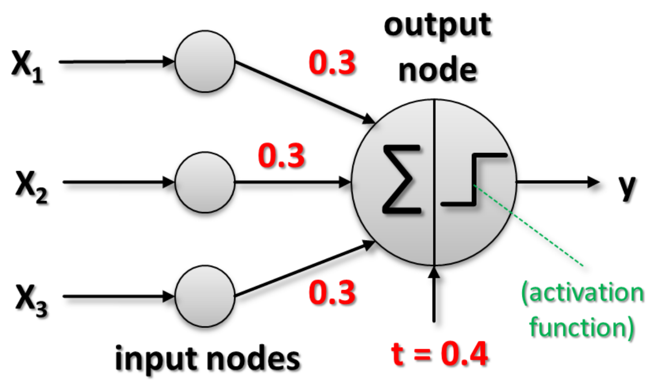
	x1	x2	x3	y
1	1	0	0	-1
2	1	0	1	1
3	1	1	0	1
4	1	1	1	1
5	0	0	1	-1
6	0	1	0	-1
7	0	1	1	1
8	0	0	0	-1

$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$

(training data)

(training phase)

$$\boxed{sign}\left(\left(\sum_{i=1}^{d} w_i x_i\right) - threshold\right)$$



(trained perceptron model)

- Output node interpretation
  - More than just the weighted sum of the inputs – threshold (aka bias)
  - Activation function sign (weighted sum): takes sign of the resulting sum

$$y = 1, \textbf{if } 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 > 0$$

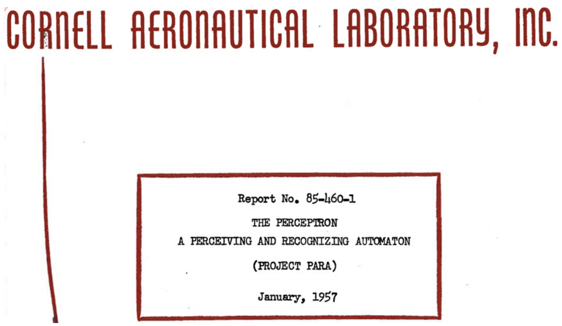
(e.g. consider sample #3, sum is positive (0.2) → +1)

$$y = -1, \textbf{if } 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 < 0$$

(e.g. consider sample #6, sum is negative (-0.1) → -1)

Summary Perceptron & Hypothesis Set h(x)

[18] F. Rosenblatt, 1957


CORNELL AERONAUTICAL LABORATORY, INC.

Report No. 85-460-1

THE PERCEPTRON

A PERCEIVING AND RECOGNIZING AUTOMATON

(PROJECT PARA)

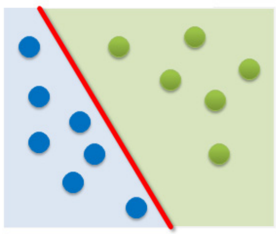
January, 1957

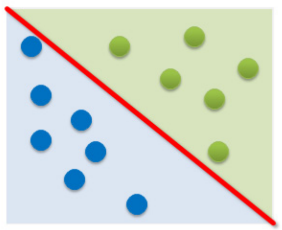
- When: Solving a linear classification problem
  - Goal: learn a simple value (+1/-1) above/below a certain threshold
  - Class label renamed: Iris-setosa = -1 and Iris-virginica = +1
- Input: $\mathbf{x} = x_1, \ldots, x_d$ **(attributes in one dataset)**
- Linear formula **(take attributes and give them different weights – think of 'impact of the attribute')**
  - All learned formulas are different hypothesis for the given problem

$$h(\mathbf{x}) = \boxed{sign}\left(\left(\sum_{i=1}^{d} w_i x_i\right) - threshold\right); h \in \mathcal{H}$$

(parameters that define one hypothesis vs. another)

(each green space and blue space are regions of the same class label determined by sign function)





(red parameters correspond to the redline in graphics)

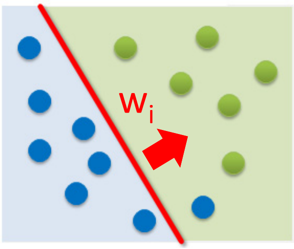
(but question remains: how do we actually learn w_i and threshold?)

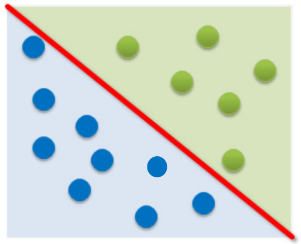
Perceptron Learning Algorithm – Understanding Vector W

- When: If we believe there is a linear pattern to be detected
  - Assumption: Linearly seperable data (lets the algorithm converge)
  - Decision boundary: perpendicular vector w_i fixes orientation of the line

$$\mathbf{w}^T\mathbf{x} = 0$$
$$\mathbf{w} \cdot \mathbf{x} = 0$$

(points on the decision boundary satisfy this equation)





$$h(\mathbf{x}) = sign(\mathbf{w}^T\mathbf{x})$$

(vector notation, using T = transpose)

$$\mathbf{w}_i = (w_{i1}, w_{i2}, \ldots, w_d)$$

$$\mathbf{w}_i^T = \begin{bmatrix} w_{i1} \\ w_{i2} \\ \ldots \\ w_{id} \end{bmatrix}$$

  - Possible via simplifications since we also need to learn the threshold:

$$h(\mathbf{x}) = sign\left(\left(\sum_{i=1}^{d} w_i x_i\right) + w_0\right); w_0 = -threshold$$

$$h(\mathbf{x}) = sign\left(\left(\sum_{i=0}^{d} w_i x_i\right)\right); x_0 = 1$$

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_d)$$

$$h(\mathbf{x}) = sign(\mathbf{w} \cdot \mathbf{x})$$

(equivalent dotproduct notation)

[19] Rosenblatt, 1958

(all notations are equivalent and result is a scalar from which we derive the sign)

Perceptron Learning Algorithm – Learning Step

- Iterative Method using (labelled) training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$

(one point at a time is picked)

1. Pick one misclassified training point where:

   $sign(\mathbf{w}^T \mathbf{x}_n) \neq y_n$

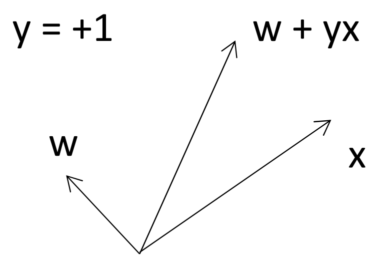
2. Update the weight vector:

   $\mathbf{w} \leftarrow \mathbf{w} + y_n \mathbf{x}_n$

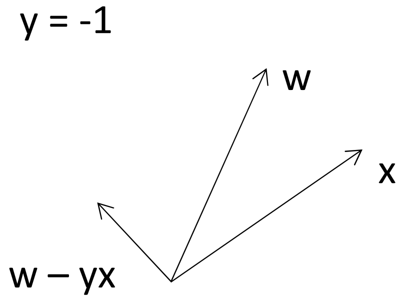
   (y$_n$ is either +1 or -1)

(a) adding a vector or
(b) subtracting a vector

(a)

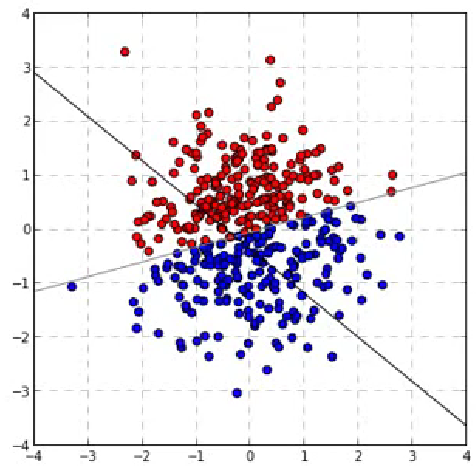


(b)



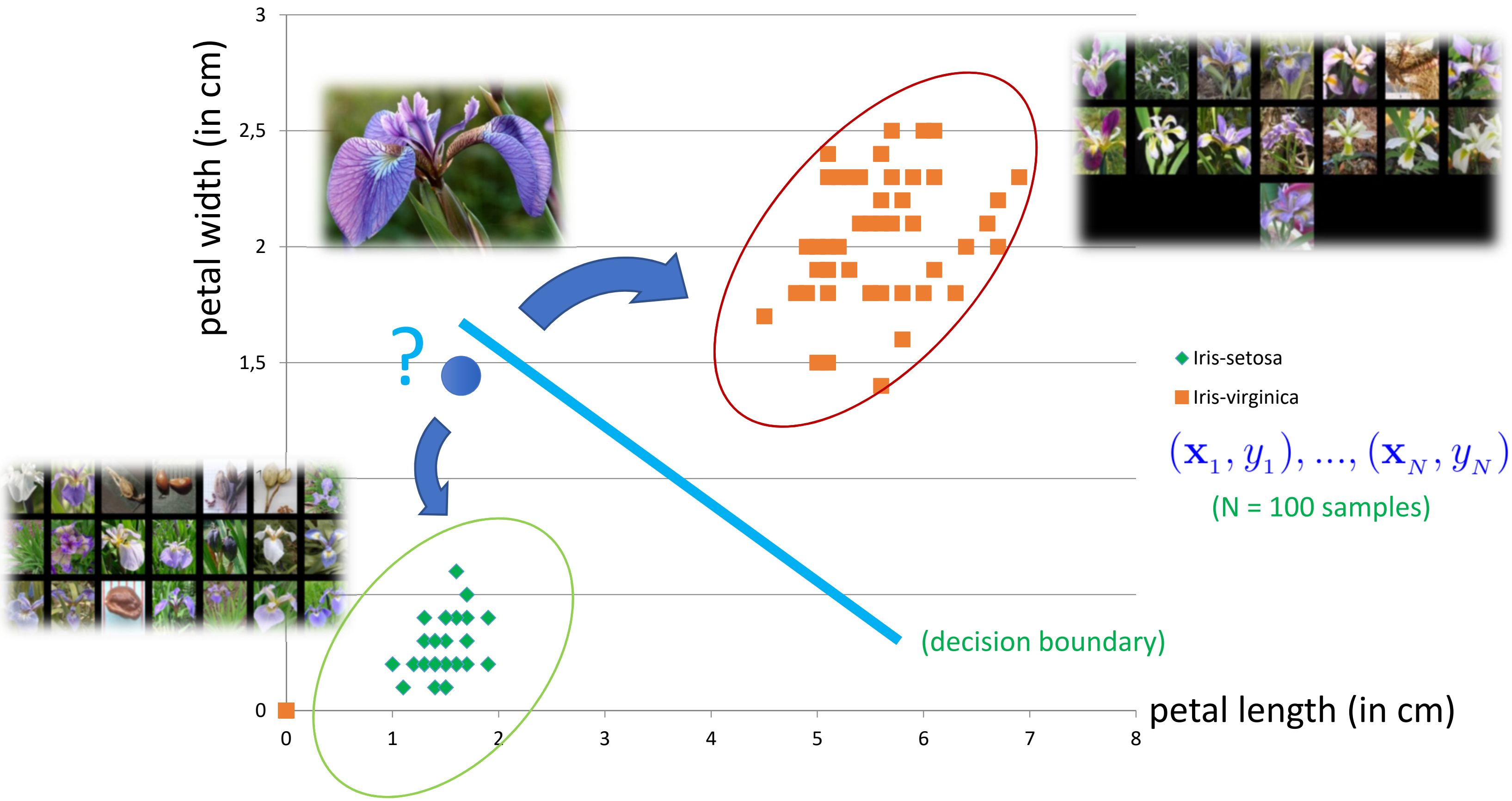
- Terminates when there are no misclassified points

(converges only with linearly seperable data)



[20] Perceptron Visualization

Predicting Task: Obtain Class of a new Flower 'Data Point'



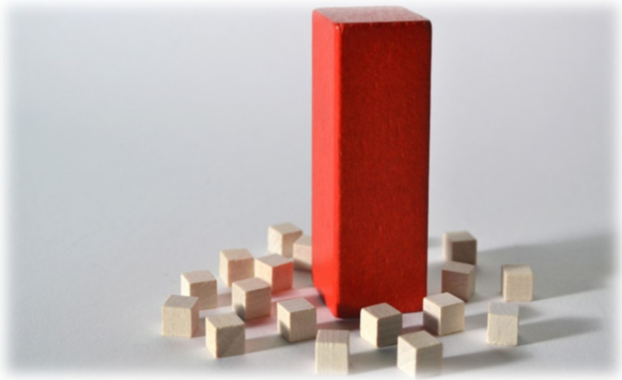
$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$$

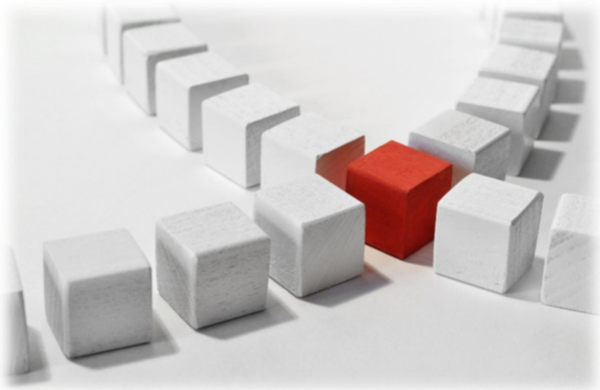
(N = 100 samples)

[14] Image sources: Species Iris Group of North America Database, www.signa.org

Summary Terminologies & Different Dataset Elements

- Target Function
  - Ideal function that ‘explains‘ the data we want to learn
- Labelled Dataset (samples)
  - ‘in-sample‘ data given to us:
- Learning vs. Memorizing
  - The goal is to create a system that works well ‘out of sample‘
  - In other words we want to classify ‘future data‘ (ouf of sample) correct
- Dataset Part One: Training set
  - Used for training a machine learning algorithms
  - Result after using a training set: a trained system
- Dataset Part Two: Test set
  - Used for testing whether the trained system might work well
  - Result after using a test set: accuracy of the trained model

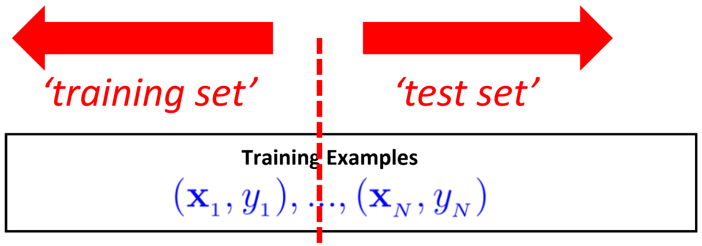




Model Evaluation – Training and Testing Phases

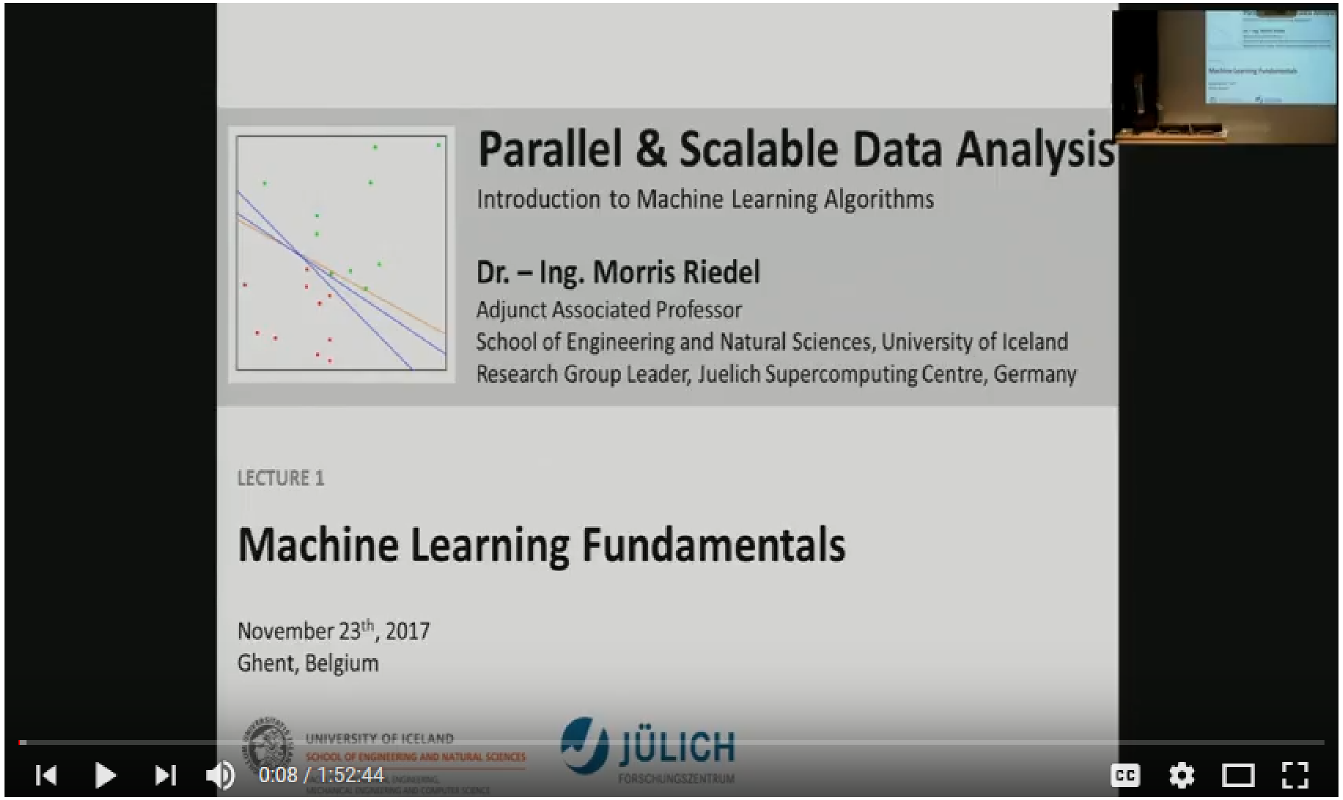
- Different Phases in Learning
  - Training phase is a hypothesis search
  - Testing phase checks if we are on right track (once the hypothesis clear)
- Work on 'training examples'
  - Create two disjoint datasets
  - One used for training only (aka training set)
  - Another used for testing only (aka test set)
  - Exact seperation is rule of thumb per use case (e.g. 10 % training, 90% test)
  - Practice: If you get a dataset take immediately test data away ('throw it into the corner and forget about it during modelling')
  - Reasoning: Once we learned from training data it has an 'optimistic bias'

(e.g. student exam training on examples to get E_{in} 'down', then test via exam)



(historical records, groundtruth data, examples)

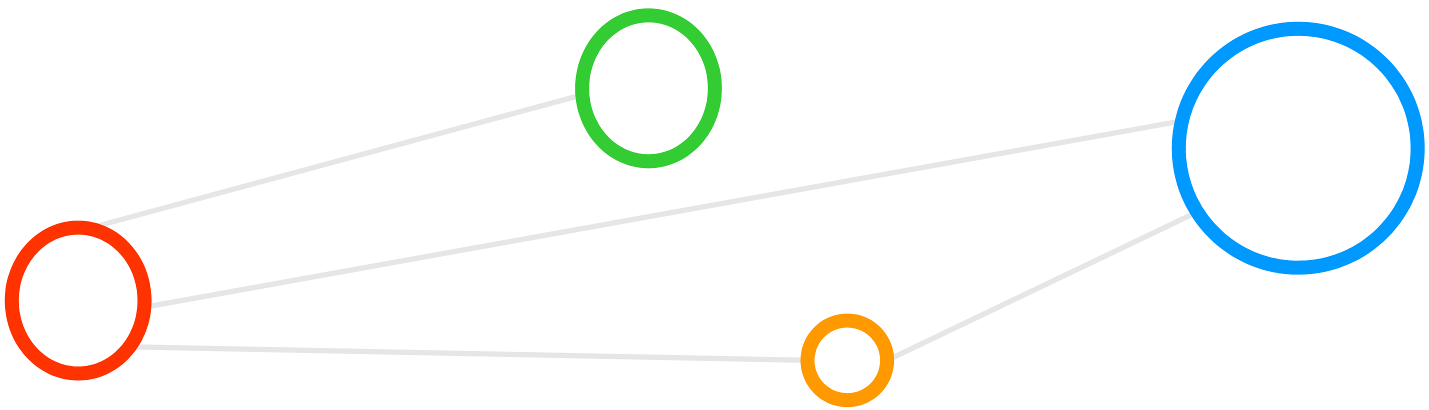
[YouTube Lectures] More Machine Learning Fundamentals



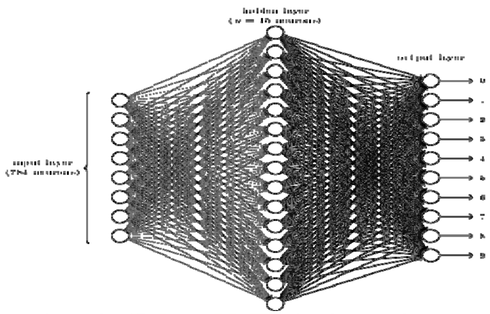
[21] Morris Riedel, 'Introduction to Machine Learning Algorithms',
Invited YouTube Lecture, six lectures, University of Ghent, 2017

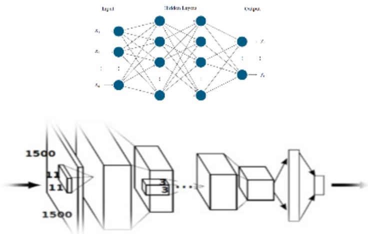
➢ **Note that this lecture series in this school is not a full course on Machine Learning like a 3 days course in Juelich or University Lectures**

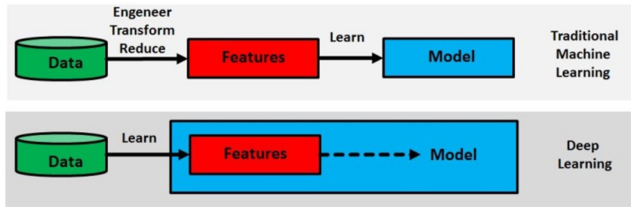
Relationships to High Performance Computing & Big Data

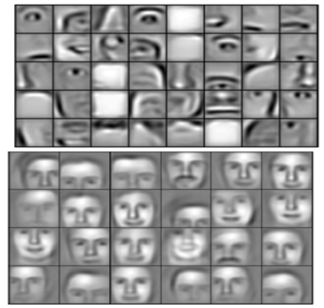


Innovative Deep Learning Techniques









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

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

JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

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

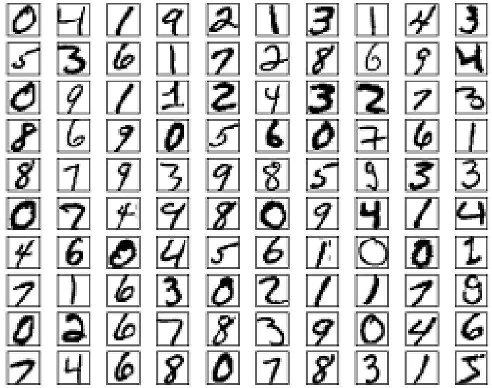
Cross-Sectional Team Deep Learning

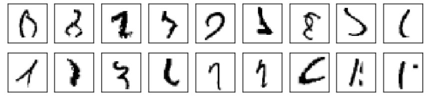
- **Provide deep learning tools that work with HPC machines (e.g. Python/Keras/Tensorflow)**
- **Advance deep learning applications and research on HPC prototypes (e.g. DEEP-EST, SMITH, etc.)**
- **Engage with industry (industrial relations team) & support SMEs (e.g. Soccerwatch, ON4OFF)**
- **Offer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

➢ Day 2 offers a more detailed introduction to Deep Learning Techniques with examples and Convolutional Neural Networks (CNNs)

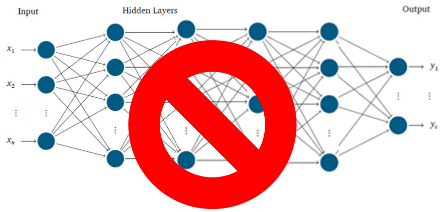
Deep Learning Technique Example – Convolutional Neural Networks (CNNs)

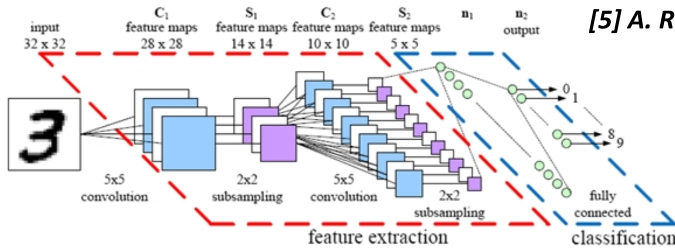
[4] Neural Network 3D Simulation





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

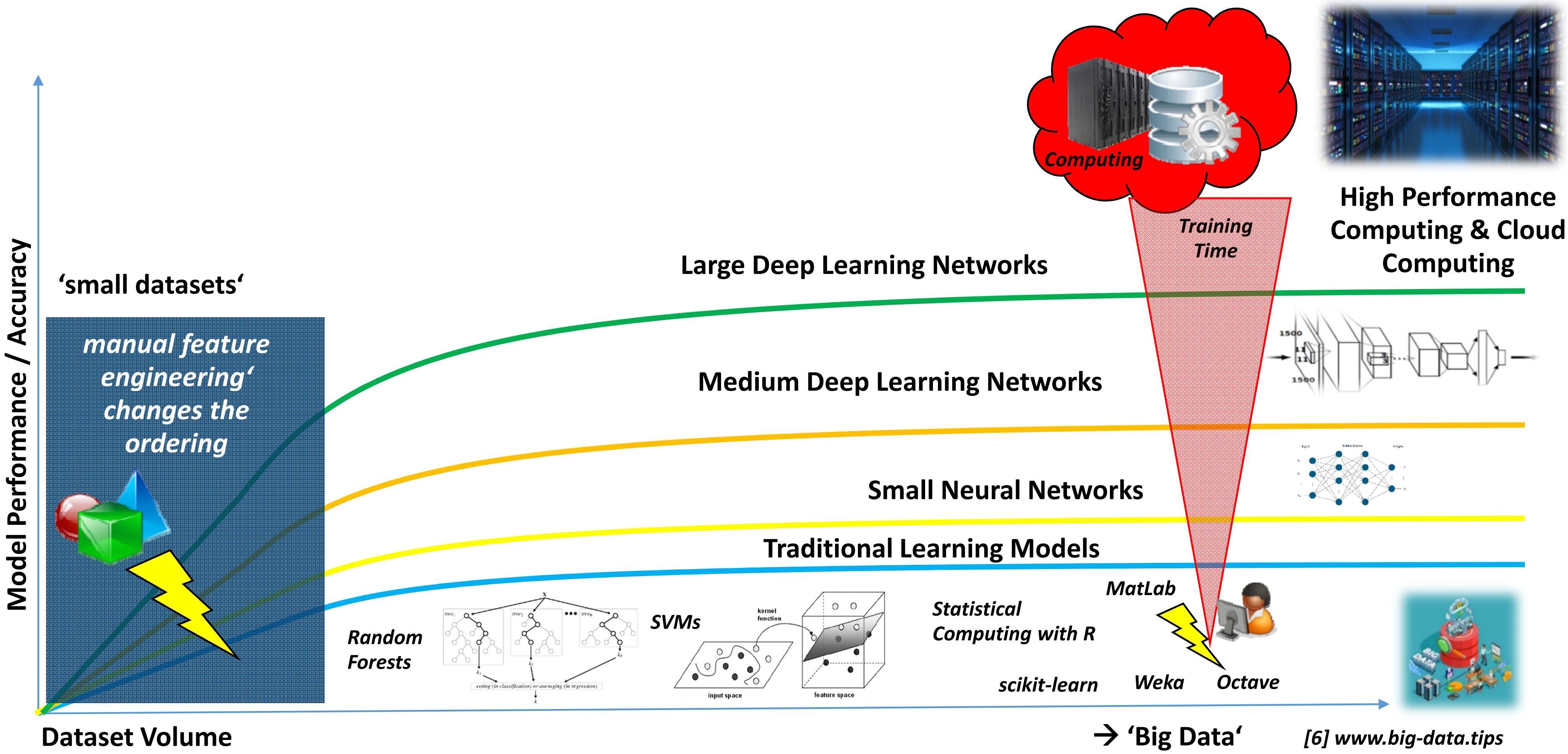




[5] A. Rosebrock

> ➢ **Day 2 offers a more detailed introduction to Deep Learning Techniques with examples and Convolutional Neural Networks (CNNs)**

Complex Relationships: ML & DL vs. HPC/Clouds & Big Data



Understanding Deep Learning Momentum & Startup Example

1952 Stochastic Gradient Descent
- Solving optimization problems

1958 Perceptron Learning Model
- Learning weights

⋮

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

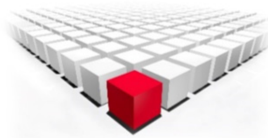
1995 Deep Convolutional Neural Networks
- Significant improvements in image analysis

⋮

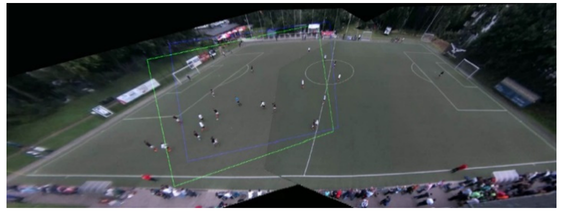
Impact in AI & HPC in industry & science

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

[7] NVIDIA

Keras

[8] Keras

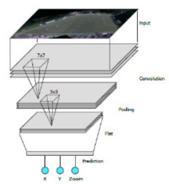
TensorFlow

[9] TensorFlow

UNIVERSITÄT DUISBURG ESSEN

EXIST Existenzgründungen aus der Wissenschaft

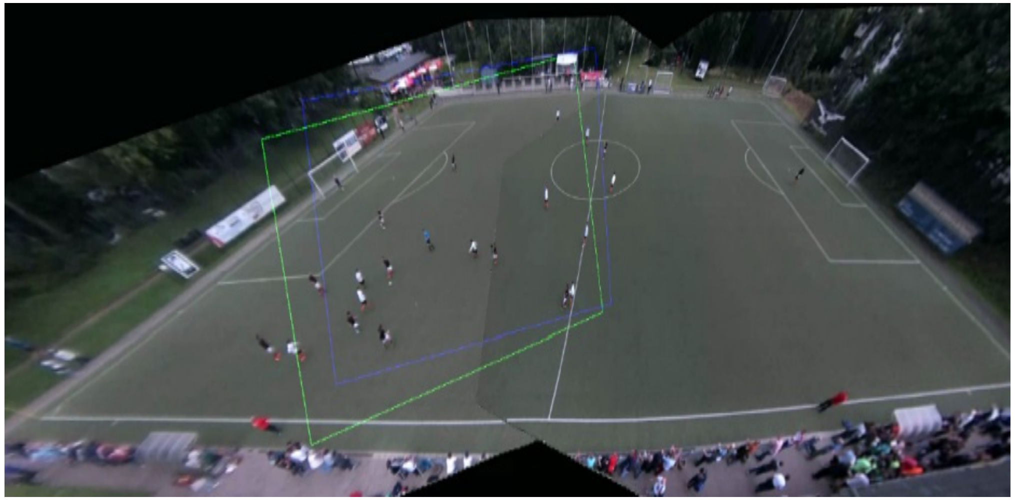
[10] soccerwatch.tv

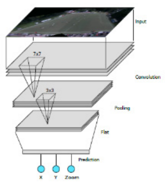


Combination: Start-up Example of my research group

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

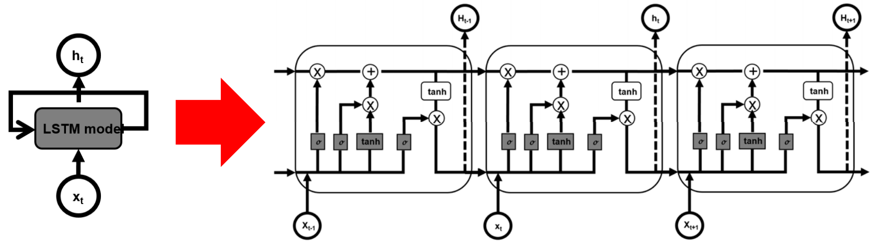
Impacts of Deep Learning Techniques for Different Types of Data





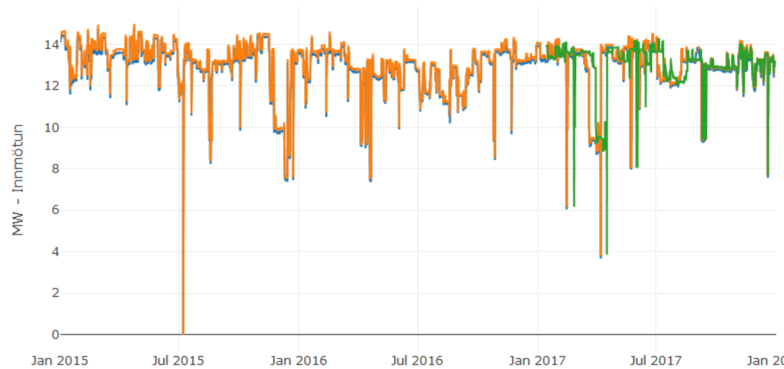
- Using Deep Learning to enable automatic camera tracking of soccer

[11] 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



High Performance Computing & Data Sciences getting more intertwined

1.000.000 FLOP/s

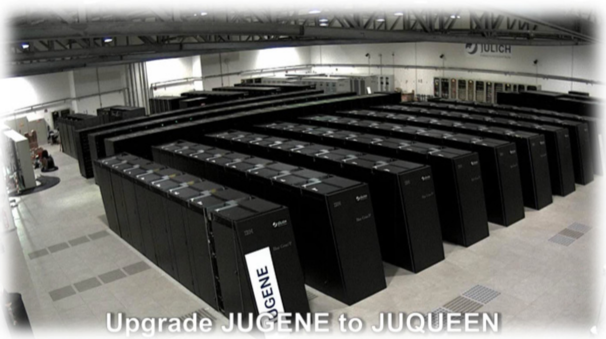
~1984

© Photograph by Rama, Wikimedia Commons

- **Floating Point Operations per one second (FLOPS or FLOP/s)**
- **1 GigaFlop/s = 10^9 FLOPS**
- **1 TeraFlop/s = 10^{12} FLOPS**
- **1 PetaFlop/s = 10^{15} FLOPS**
- **1 ExaFlop/s = 10^{18} FLOPS**

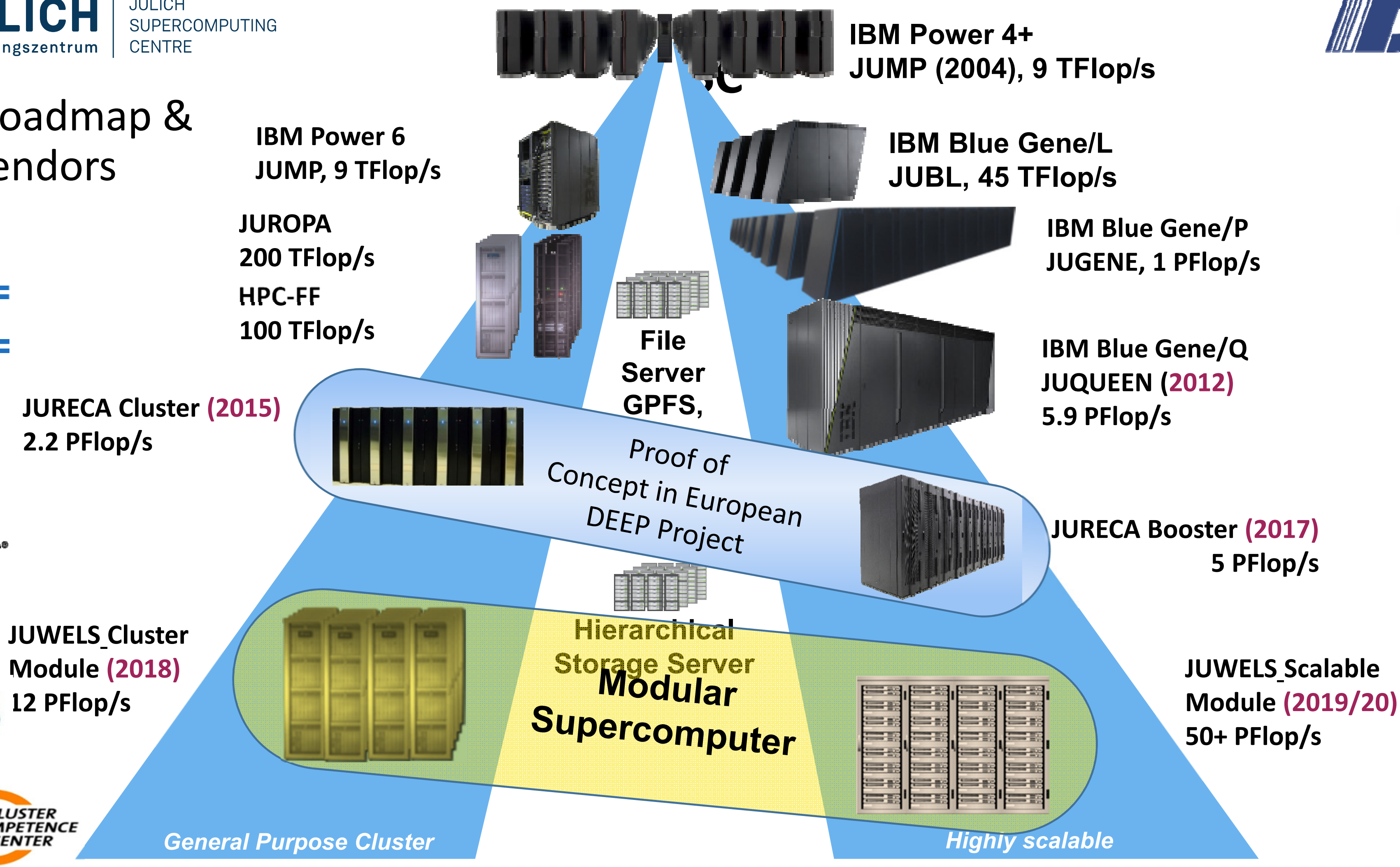
1.000.000.000.000.000 FLOP/s

~295.000 cores~2009 (JUGENE)



>5.900.000.000.000.000 FLOP/s
~ 500.000 cores
~2013 → end of service in 2018

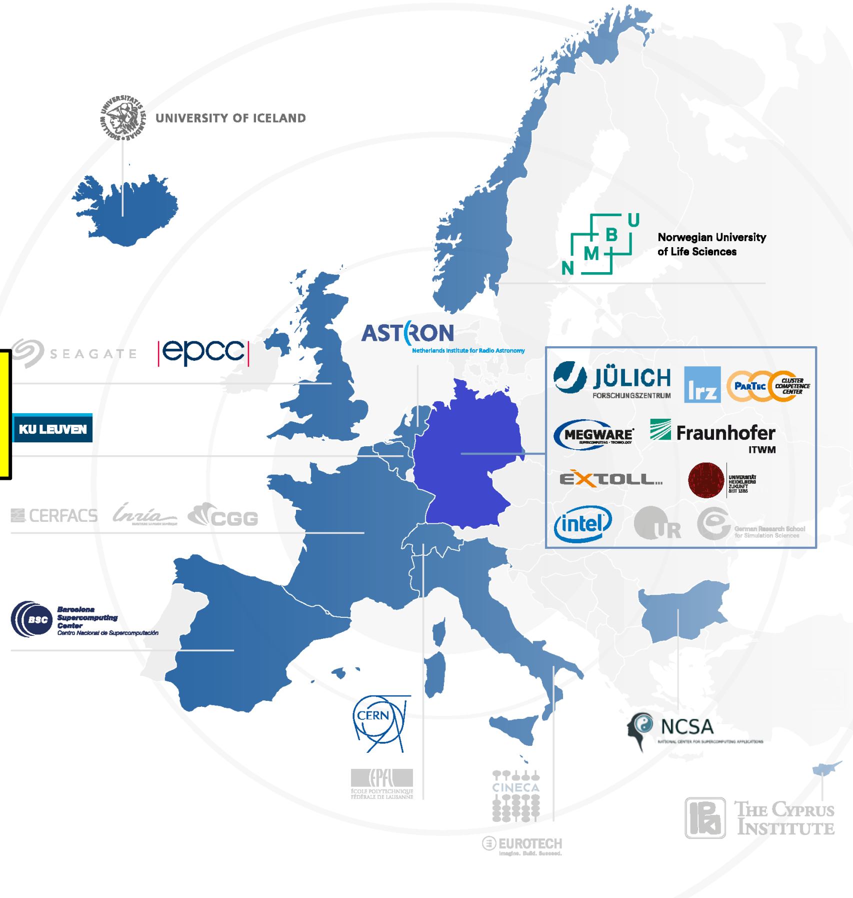
- HPC Roadmap & Key Vendors



DEEP Series of Projects – Modular Supercomputing Architecture Research

[12] DEEP Projects Web Page

Strong collaboration with our industry partners Intel, Extoll & Megware

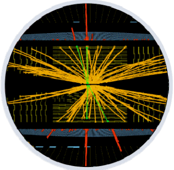


- 3 EU Exascale projects
  DEEP, DEEP-ER, DEEP-EST
- 27 partners
  Coordinated by JSC
- EU-funding: 30 M€
  JSC-part > 5,3 M€
- Nov 2011 – Dec 2020

- **Strong collaboration with industry partners Intel, Extoll & Megware**

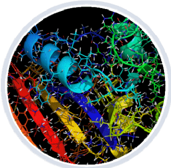
- **Juelich Supercomputing Centre implements the DEEP projects designs in its HPC infrastructure**

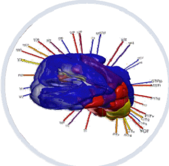
Application Co-Design for Machine & Deep Learning in HPC


High Energy Physics

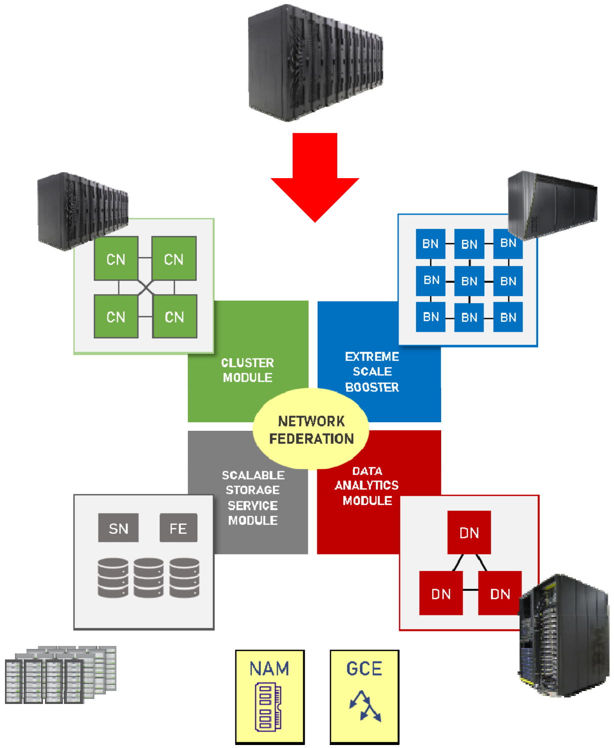

Earth Science

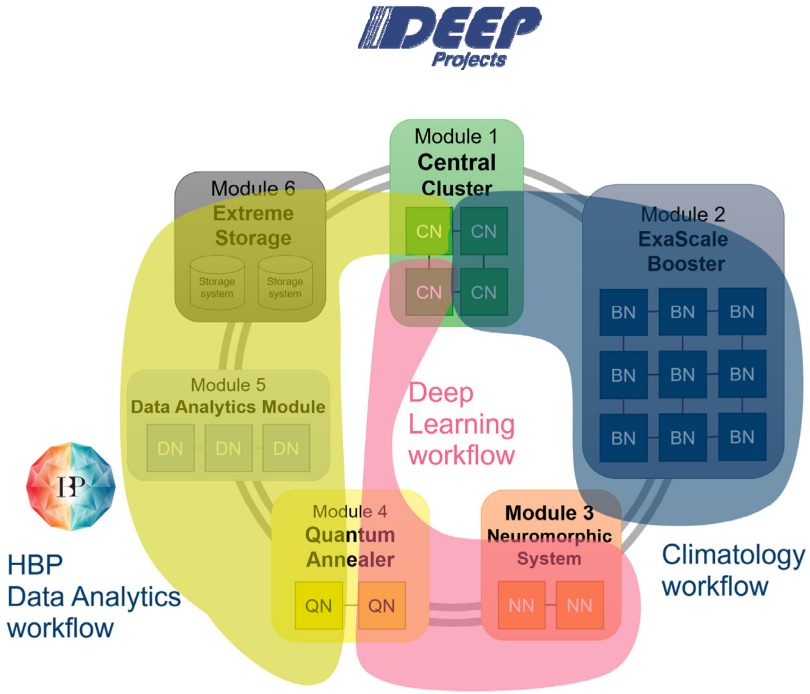

Space Weather


Molecular Dynamics


Neuroscience


Radio Astronomy

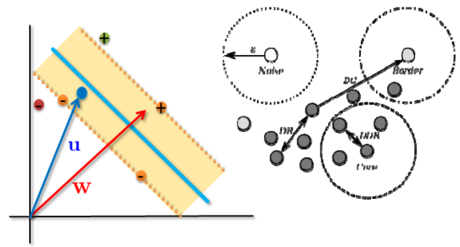




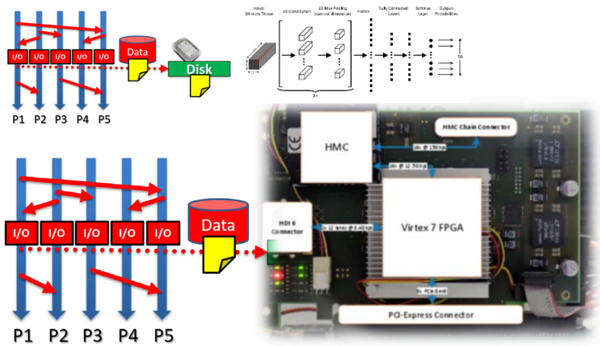
- **The modular supercomputing architecture (MSA) enables a flexible HPC system design co-designed by the need of different application workloads**

[12] DEEP Projects Web Page

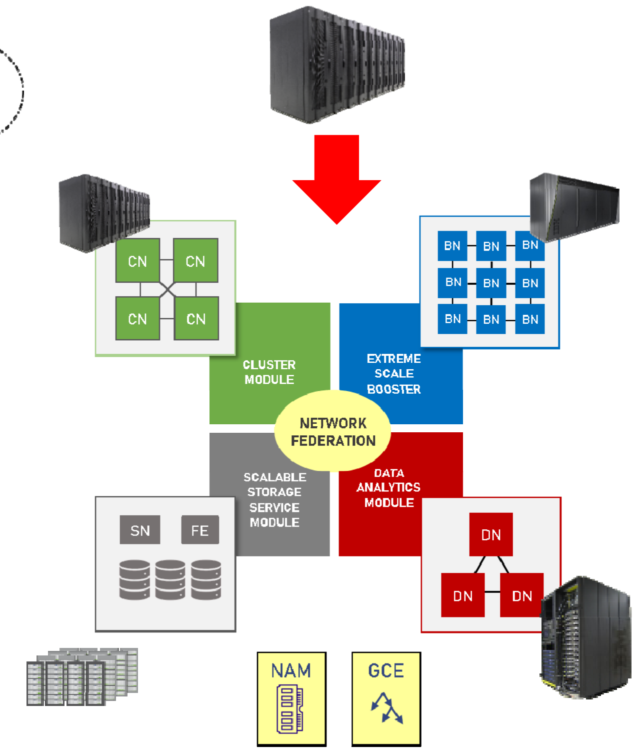
Innovative HPC Hardware via Machine/Deep Learning Co-Design



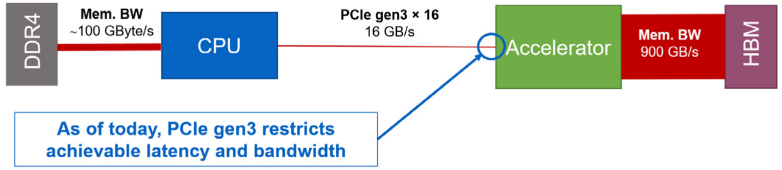
- Explore Network Attached Memory (NAM)



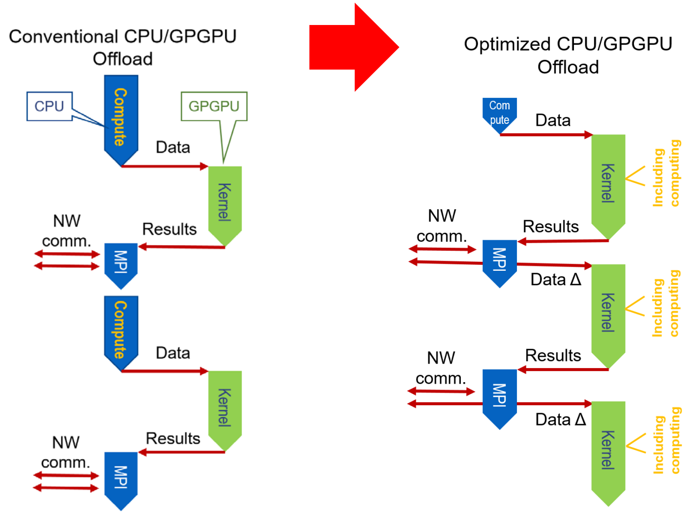
[13] E. Erlingsson, M. Riedel et al.,
IEEE MIPRO Conference, 2018



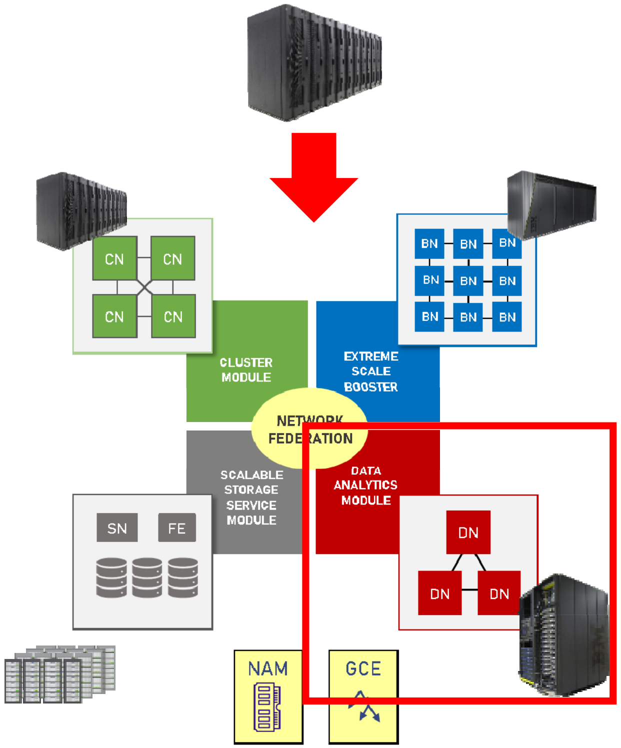
- The modular supercomputing architecture (MSA) enables a flexible HPC system design co-designed by the need of different application workloads



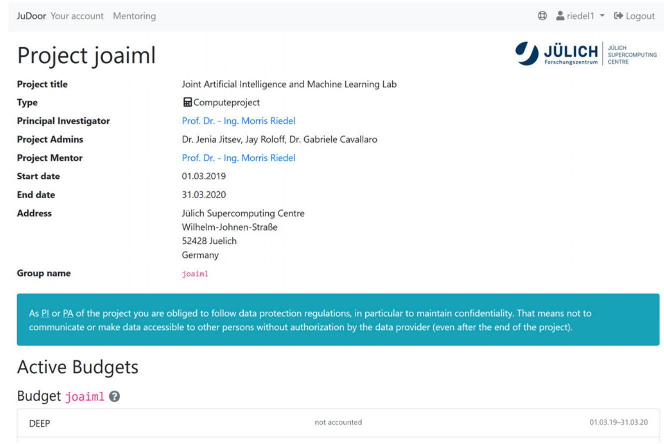
- Explore more scalability with NVIDIA compared to NVIDIA NVLink/NVSwitch 'Islands'



Hands-On Training System – Data Analytics Module (DAM)



- 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
  - 3 x 4 GPUs Tesla Volta V100
  - Slurm scheduling system



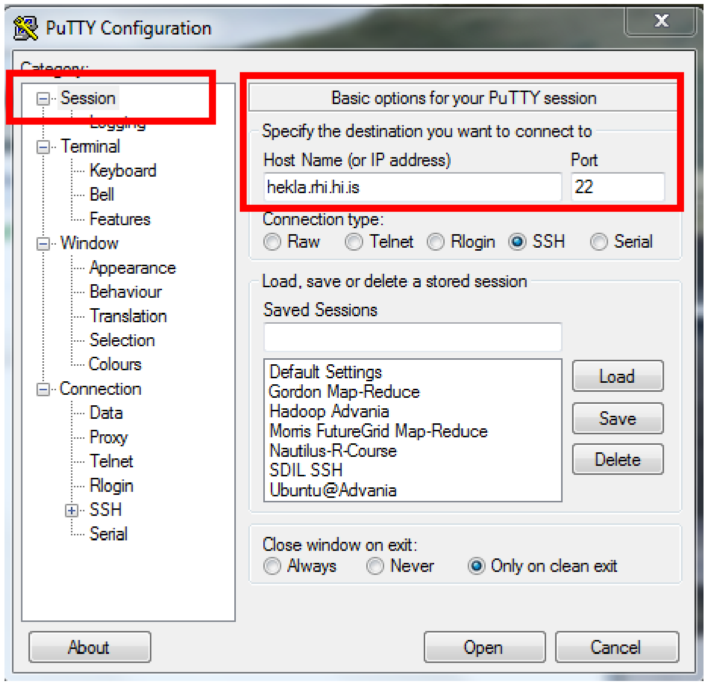
(easy join via JOAIML ab with JuDoor)

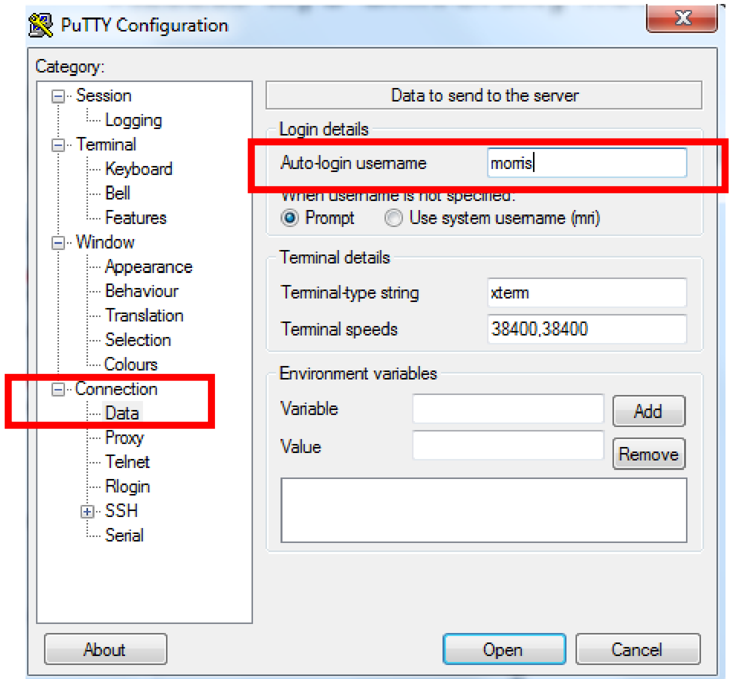
[12] DEEP Projects Web Page

➢ **The Data Analytics Module (DAM) will be used for a couple of different machine & deep learning exercises in the context of lectures**

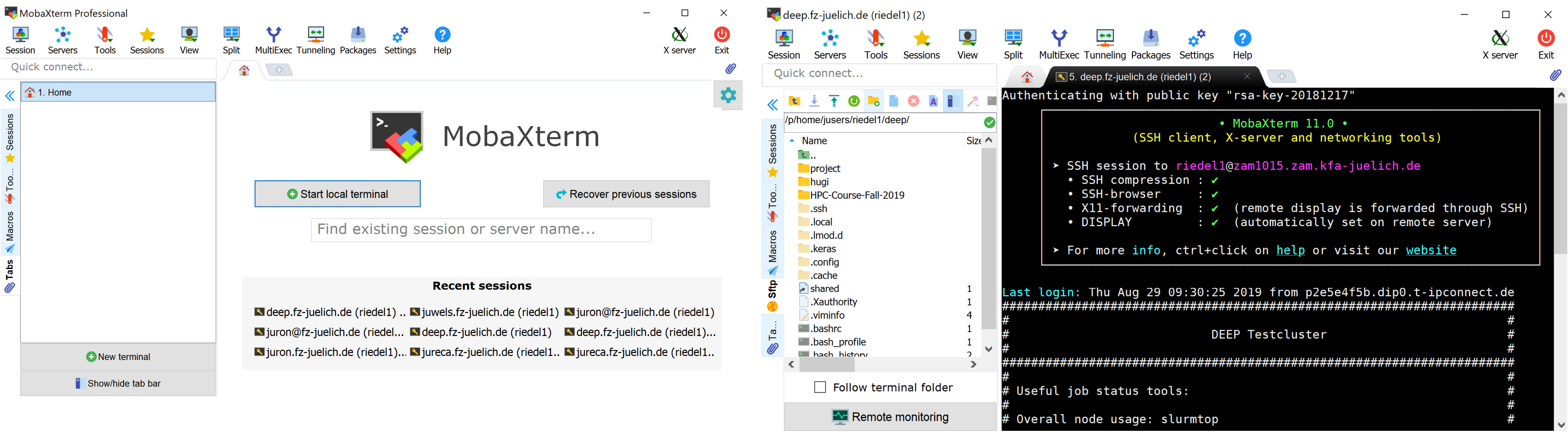
SSH Clients – Putty for Windows

- Example: Putty SSH Client for Windows
  - Not recommended, better install MobaXterm





MobaXterm SSH Client



[22] MobaXterm SSH Client

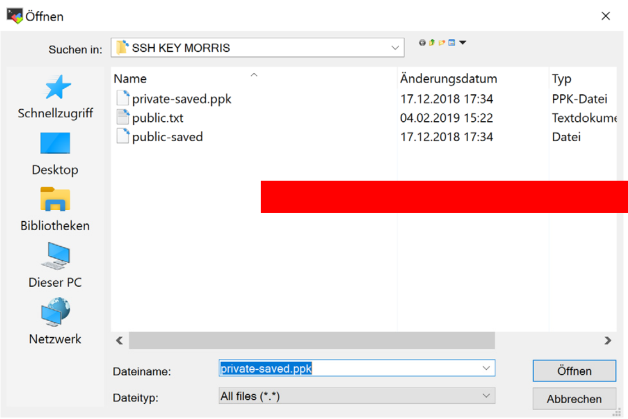
SSH Keys – Example Private & Public Key Pair

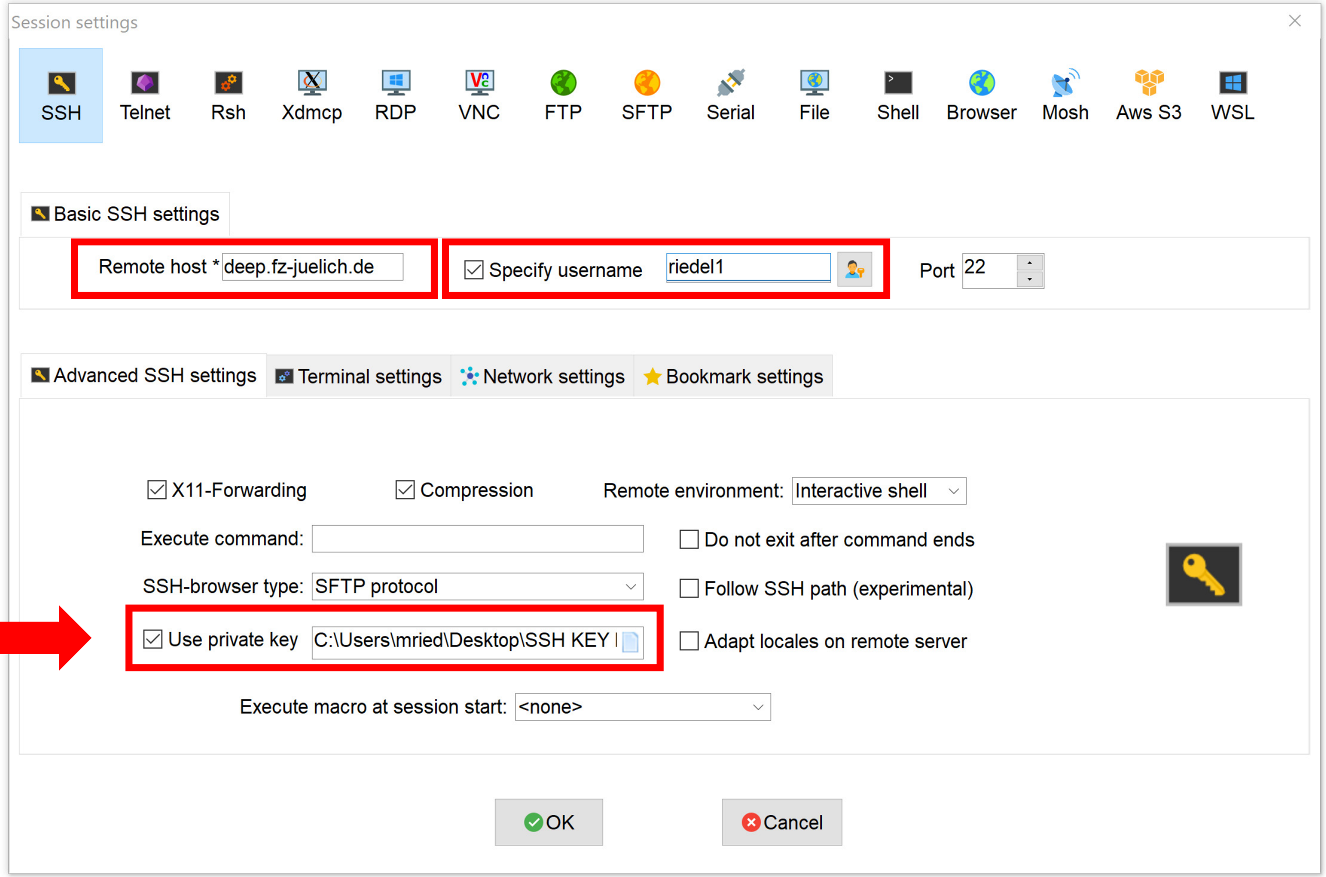
- Everybody is allowed to see the public key
  - Given to HPC administrators
  - Sometimes uploaded in Web forms
  - Only the owner has the corresponding private key that never leaves the laptop of the owner (!)

- SSH Key Example (public key)
  - ssh-rsa
    AAAAB3NzaC1yc2EAAAABJQAAAQEAruA2IJQmVEwVjsQ6N9PUJP0KukCGQV2yAMs3hop0stsvfb4Iac7s2Pqk
    wOgoFPZGwRCSGcA2/rISJX3MxEmx7EQLD5sw63r8LqvETXy4hmefIlBwpcIxMBYSLujWdCH9K60Q6TApMz4h
    V+fsZRiGbTx7hs9Y2a3TiiSE032IvzxMYTvW8NYlhXOP9PzTR1jebVj3rgcOIYLPMGzl4YIbCZJVIeJlwfkZscOH9zT4
    Kl5SpQuk5Q+LyMl95X3xsk3xPMCuocqsYmIY6Gp+BCAYJsdCXFNDJ3SCcphziTqrE+F2EroI4AoegVlH/vhPaAZg
    Q222nV2rDsN+uDhaBf+76Q== rsa-key-20181217

SSH Keys – Use Private/Public Key Pair to Access DEEP HPC System

- Remember to use your Private SSH Key to connect to the DEEP system
  - Corresponding Public SSH key is already uploaded on the HPC System (remote host) per username(!)



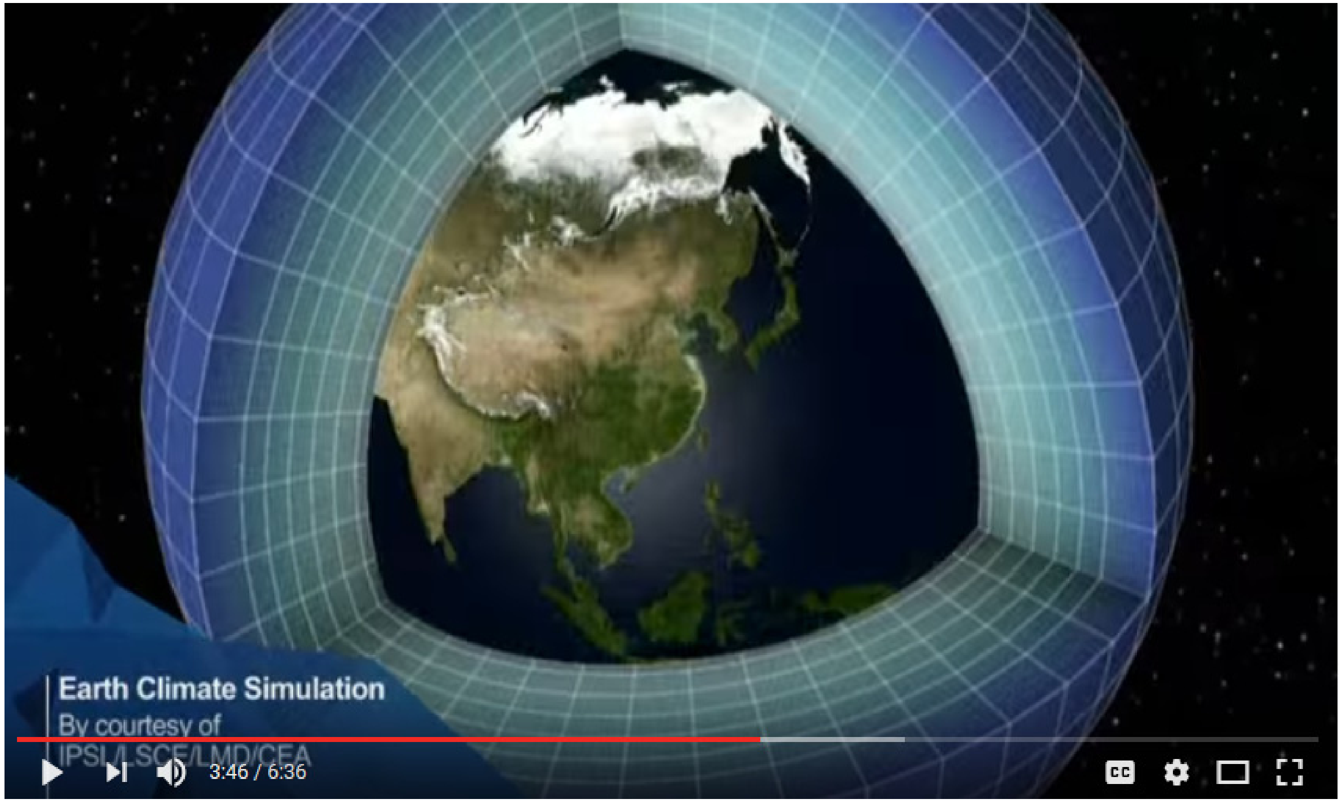


SSH Access to HPC Systems – DEEP HPC System Example

```
Authenticating with public key "rsa-key-20181217"

              • MobaXterm 11.0 •
      (SSH client, X-server and networking tools)

 ➤ SSH session to riedel1@zam1015.zam.kfa-juelich.de
   • SSH compression : ✔
   • SSH-browser     : ✔
   • X11-forwarding  : ✔  (remote display is forwarded through SSH)
   • DISPLAY         : ✔  (automatically set on remote server)

 ➤ For more info, ctrl+click on help or visit our website

Last login: Fri Aug 16 16:36:51 2019 from zam106.zam.kfa-juelich.de
########################################################################
#                                                                      #
#                           DEEP Testcluster                           #
#                                                                      #
########################################################################
#                                                                      #
# Useful job status tools:                                             #
#                                                                      #
# Overall node usage: slurmtop                                         #
# Show jobs in the system: squeue                                      #
# List reservations: scontrol show res                                 #
# Check job issues: scontrol show job <jobid>                          #
# See /etc/slurm/README for details and known problems                 #
#                                                                      #
########################################################################
#                                                                      #
# /usr/local now served by local beegfs file system.                   #
# performance issues under investigation.                              #
# pn Thu Jun 13 14:26:48 CEST 2019                                     #
#                                                                      #
########################################################################
#                                                                      #
# The transition to the new software stack will happen on Tuesday 9th #
# To enable it now use "enable_new_software_stack"                     #
# To enable the current stack use "enable_old_stack"                   #
# To enable the legacy stack use "enable_legacy_stack"                 #
#                                                                      #
########################################################################
#                                                                      #
# /usr/local on deepv, dp-cn nodes:                                    #
# now mounted without acls and extended attributes.                    #
# cm/pn Fri Jul 26 10:30:50 CEST 2019                                  #
#                                                                      #
########################################################################
[riedel1@deepv ~]$
```

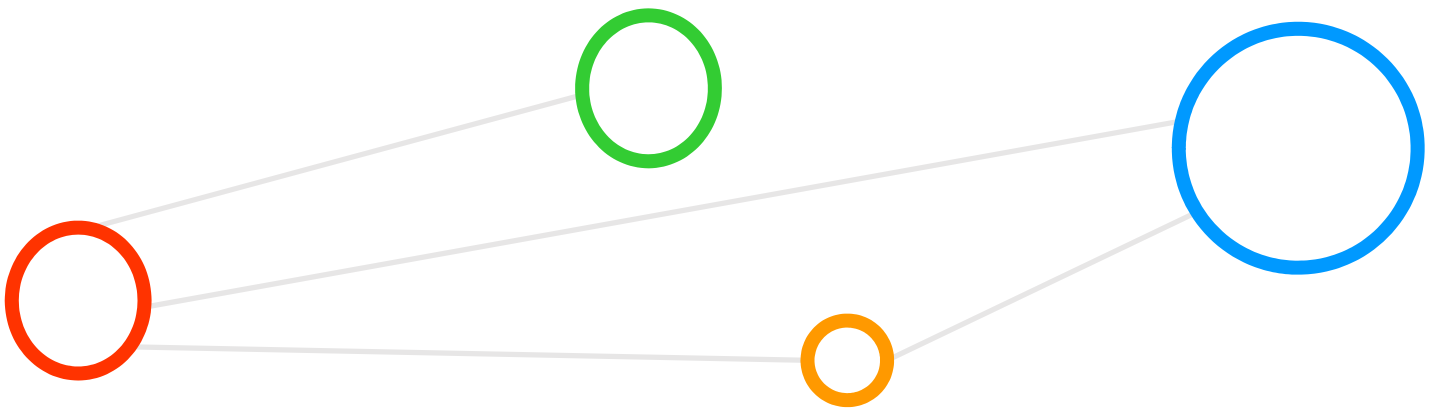
- HPC System Address
  - deep.fz-juelich.de
  - alias for zam1015.zam.kfa-juelich.de
- HPC System Username
  - Example: riedel1
  - Every student will get a different username
- HPC System Welcome Screen
  - If SSH login was succesful
  - Shows useful information about the system
  - E.g. status of the file system or known errors / bugs
  - E.g. help with important commands

[Video] PRACE – Introduction to Supercomputing



[23] PRACE – Introduction to Supercomputing

Lecture Bibliography



Lecture Bibliography (1)

- [1] 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
- [2] 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
- [3] H. Lee et al., ‘Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations’, Online: http://doi.acm.org/10.1145/1553374.1553453
- [4] YouTube Video, ‘Neural Network 3D Simulation‘, Online: https://www.youtube.com/watch?v=3JQ3hYko51Y
- [5] 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/
- [6] Big Data Tips – Big Data Mining & Machine Learning, Online: http://www.big-data.tips/
- [7] NVIDIA Web Page, Online: https://www.nvidia.com/en-us/
- [8] Keras Python High-Level Deep Learning Library, Online: https://keras.io/
- [9] TensorFlow Python Low-Level Deep learning Library, Online: https://www.tensorflow.org/
- [10] Deep Learning Start-Up Beispiel Deutschland, Online: https://soccerwatch.tv/
- [11] 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

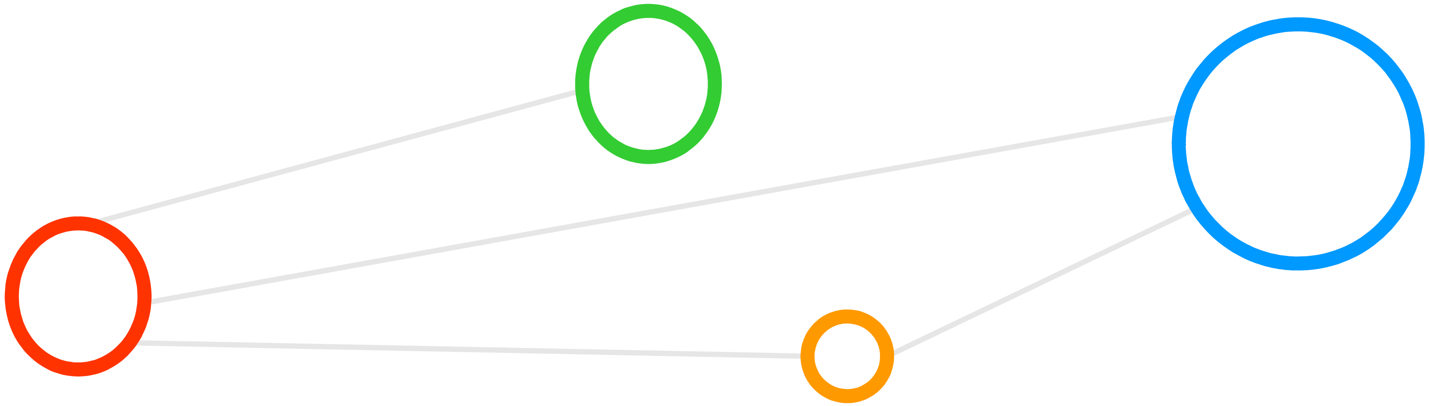
Lecture Bibliography (2)

- [12] DEEP Series Projects Web Page, Online:
  http://www.deep-projects.eu/
- [13] 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
- [14] Species Iris Group of North America Database, Online:
  http://www.signa.org
- [15] Cheng, A.C, Lin, C.H., Juan, D.C., InstaNAS: Instance-aware Neural Architecture Search, Online:
  https://arxiv.org/abs/1811.10201
- [16] UCI Machine Learning Repository Iris Dataset, Online:
  https://archive.ics.uci.edu/ml/datasets/Iris
- [17] Wikipedia 'Sepal', Online:
  https://en.wikipedia.org/wiki/Sepal
- [18] F. Rosenblatt, 'The Perceptron--a perceiving and recognizing automaton',
  Report 85-460-1, Cornell Aeronautical Laboratory, 1957, Online:
  https://blogs.umass.edu/brain-wars/files/2016/03/rosenblatt-1957.pdf
- [19] Rosenblatt,'The Perceptron: A probabilistic model for information storage and orgainzation in the brain',
  Psychological Review 65(6), pp. 386-408, 1958, Online:
  https://psycnet.apa.org/doi/10.1037/h0042519
- [20] YouTube Video, 'Perceptron, Linear', Online:
  https://www.youtube.com/watch?v=FLPvNdwC6Qo

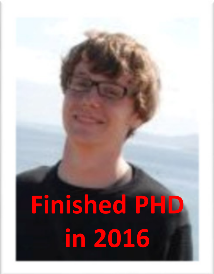
Lecture Bibliography (3)

- [21] Morris Riedel, ‘Introduction to Machine Learning Algorithms‘, Invited YouTube Lecture, six lectures, University of Ghent, 2017, Online: https://www.youtube.com/watch?v=KgiuUZ3WeP8&list=PLrmNhuZo9sgbcWtMGN0i6G9HEvh08JG0J
- [22] MobaXterm SSH Client, Online: https://mobaxterm.mobatek.net/
- [23] PRACE – Introduction to Supercomputing, Online: https://www.youtube.com/watch?v=D94FJx9vxFA

Acknowledgements

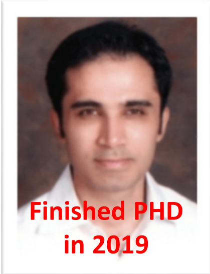


Acknowledgements – High Productivity Data Processing Research Group

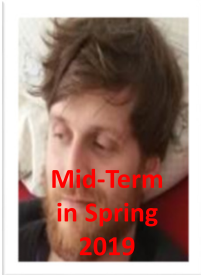


PD Dr.
G. Cavallaro

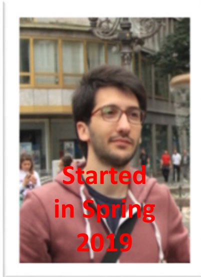
Senior PhD
Student A.S. Memon



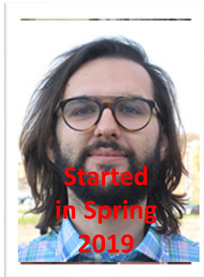
Senior PhD
Student M.S. Memon



PhD Student
E. Erlingsson

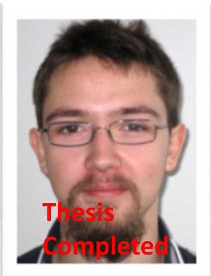


PhD Student
S. Bakarat



PhD Student
R. Sedona

Dr. M. Goetz
(now KIT)



MSc M.
Richerzhagen
(now other division)

MSc
P. Glock
(now INM-1)

MSc
C. Bodenstein
(now
Soccerwatch.tv)

MSc Student
G.S. Guðmundsson
(Landsverkjun)

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)