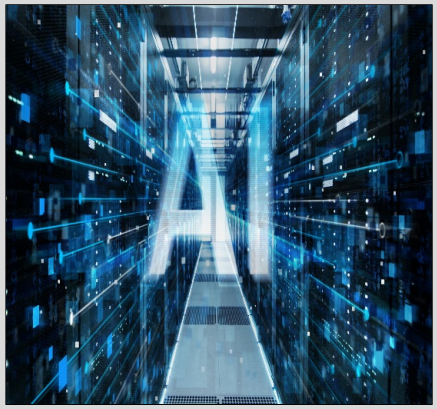


Artificial Intelligence Data Analysis (AIDA)

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 6

Deep Neural Networks & Convolutional Neural Networks – Overview

January 21, 2020
CINECA, Bologna, Italy

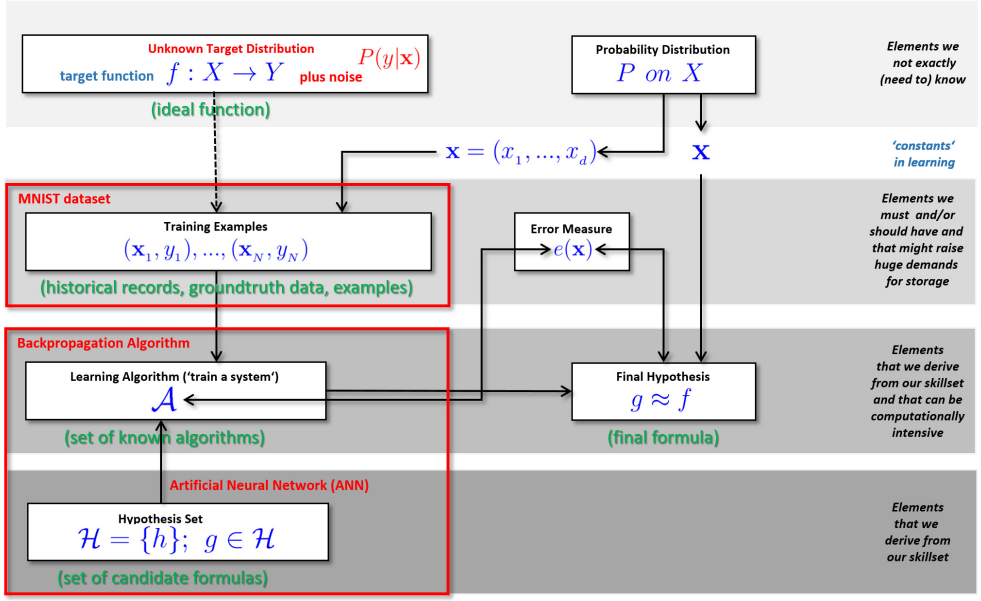
UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

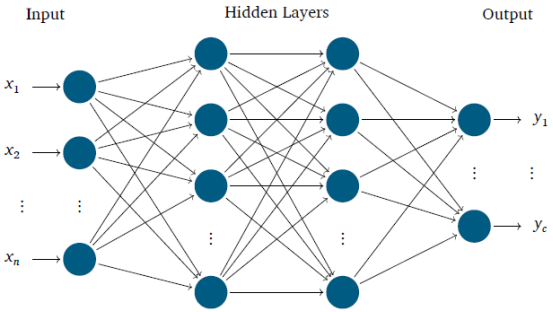
JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

DEEP Projects

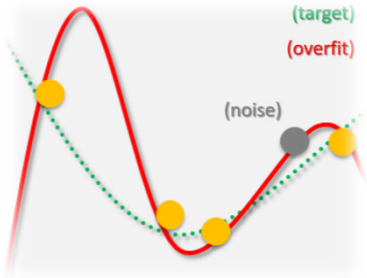
HELMHOLTZ AI | ARTIFICIAL INTELLIGENCE COOPERATION UNIT

Review of Lecture 5 – Artificial Neural Networks & Learning Theory

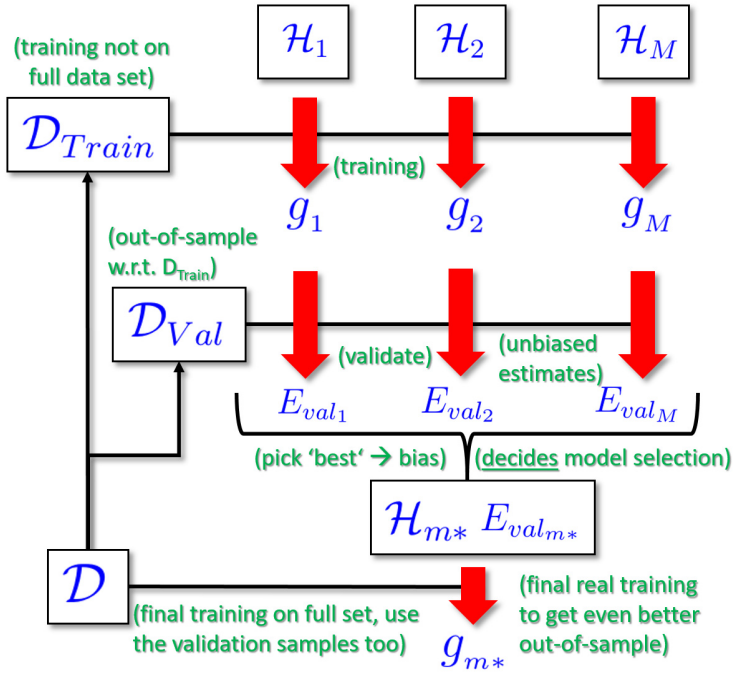





```
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 128)               100480
_________________________________________________________________
activation_1 (Activation)    (None, 128)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               16512
_________________________________________________________________
activation_2 (Activation)    (None, 128)               0
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 10)                1290
_________________________________________________________________
activation_3 (Activation)    (None, 10)                0
=================================================================
Total params: 118,282
Trainable params: 118,282
Non-trainable params: 0
```



(combat overfitting with validation and regularization)




```
# parameter setup
NB_EPOCH = 20
BATCH_SIZE = 128
NB_CLASSES = 10 # number of outputs = number of digits
OPTIMIZER = SGD() # optimization technique
VERBOSE = 1
N_HIDDEN = 128 # number of neurons in one hidden layer
VAL_SPLIT = 0.2 # 1/5 for validation rule of thumb
DROPOUT = 0.3 # regularization
```

(using Dropout as regularizer)

```
# model evaluation
score = model.evaluate(X_test, Y_test, verbose=VERBOSE)
print("Test score:", score[0])
print('Test accuracy:', score[1])
```

```
10000/10000 [==============================] - 0s 27us/step
Test score: 0.07506137332450598
Test accuracy: 0.9775
```

(the best we could do with ANNs)

Outline of the School

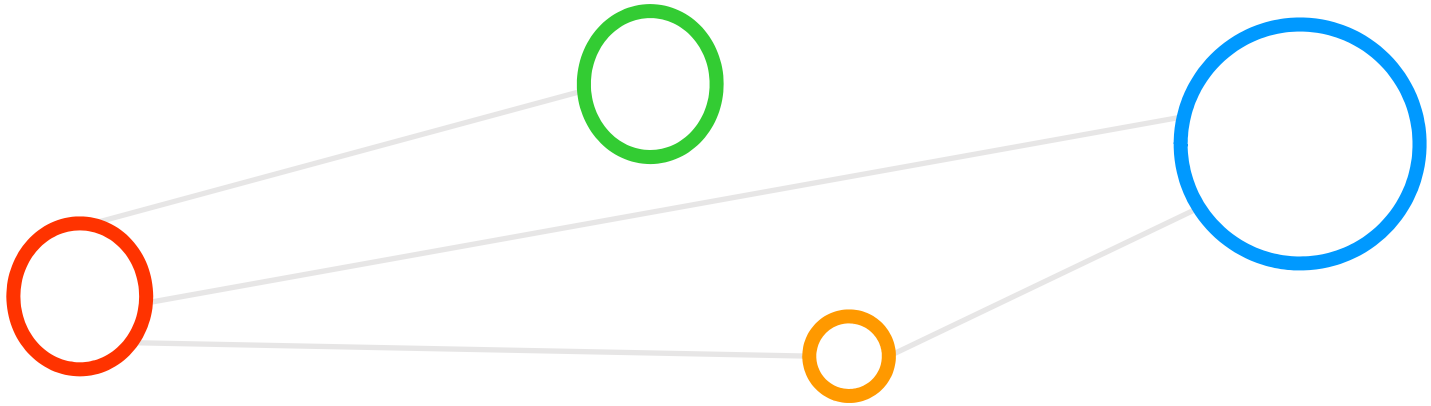
Time	**Day 1**	**Day 2**	**Day 3**
9 - 10	Welcome and intro to the school (Giovanni Lapenta, Jorge Amaya)	Space missions data acquisition (Hugo Breuillard)	Review of ML applied to heliophysics (Peter Wintoft)
10 - 11	Introduction and differences between AI, ML, NN and Big Data (Morris Riedel)	Data manipulation 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

- Deep Neural Networks (DNNs)
  - Terminologies & Deep Learning Technology Foundations
  - Benefits of Many-Core Architectures
  - Distributed Training of Deep Neural Networks
  - Remote Sensing & ImageNet Example using Horovod
  - Parallel Computing & HPC Impact in Deep Learning

- Convolutional Neural Networks (CNNs)
  - Basic Principles & MNIST Application Example
  - Local Receptive Fields & Sliding
  - Revisit ANN Overfitting & Weight Problem
  - Shared Weights & Feature Maps
  - Complexity of Parameters & Neural Architecture Search (NAS)

Deep Neural Networks (DNNs)



Terminology & Differences between AI, ML & DL – Revisited (cf. Lecture 1)

Artificial Intelligence (AI)

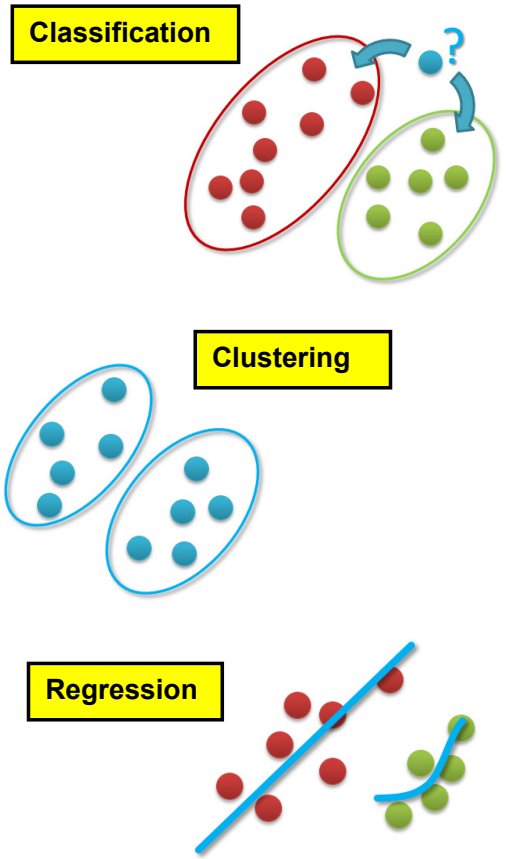
A wide area of techniques and tools that enable computers to mimic human behaviour (+ robotics)

Machine Learning (ML)

Learning from data without explicitly being programmed with common programming languages

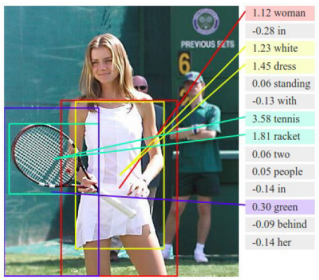
Deep Learning (DL)

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

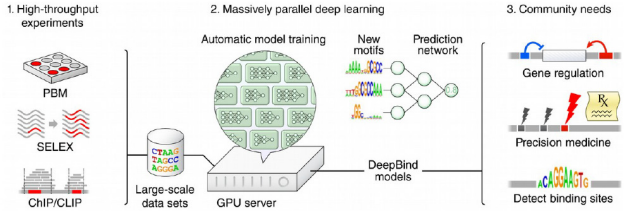


Impact of Deep Learning in Various Application Domains

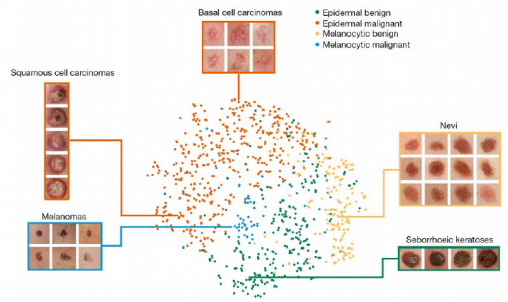
Vision, Natural Speech

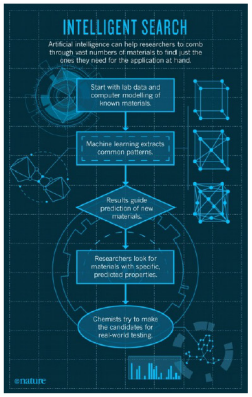


Omics (Genomics, Proteomics, Metabolomics)

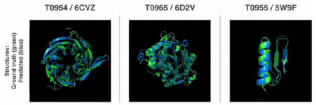


Medical Imaging and Diagnostics

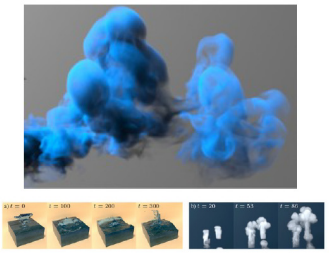




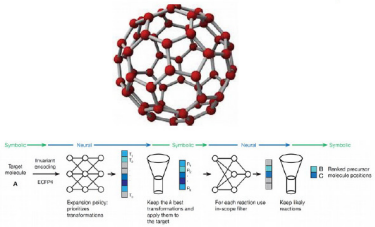
Protein Folding



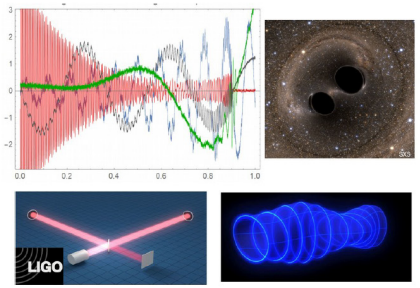
Fluid, Gas Dynamics



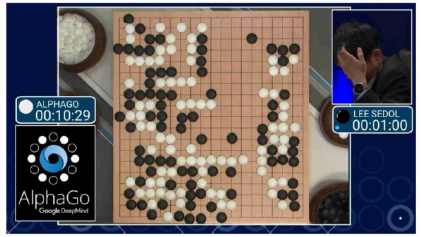
Material Science, Chemistry



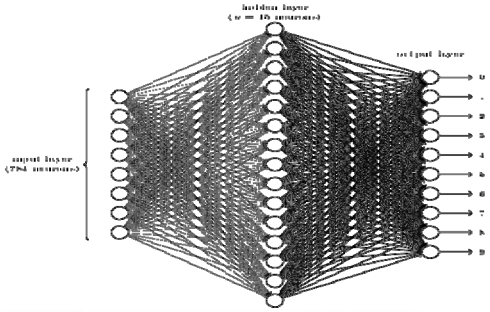
Gravitational Waves Detection

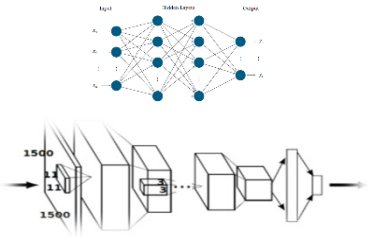


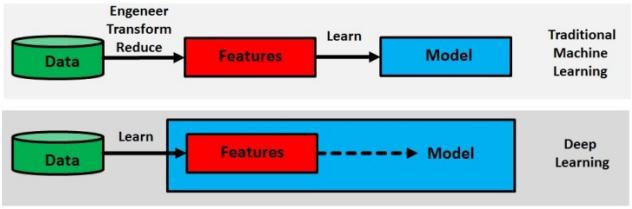
Games, Control Optimization



Innovative Deep Learning Techniques – Revisited (cf. Lecture 1)

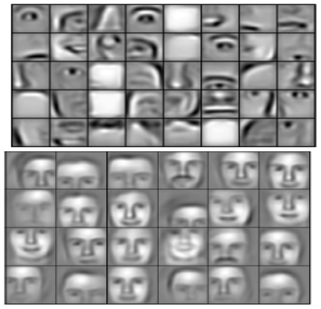






[1] M. Riedel, 'Deep Learning - Using a Convolutional Neural Network', Invited YouTube Lecture, six lectures, University of Ghent, 2017

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



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

JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

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)**

Award for Deep Learning Godfathers in Computing

Turing Award 2018: Recognition of **Deep Neural Networks** as **critical component of computing**

Association for Computing Machinery

FATHERS OF THE DEEP LEARNING REVOLUTION

RECEIVE ACM A.M. TURING AWARD

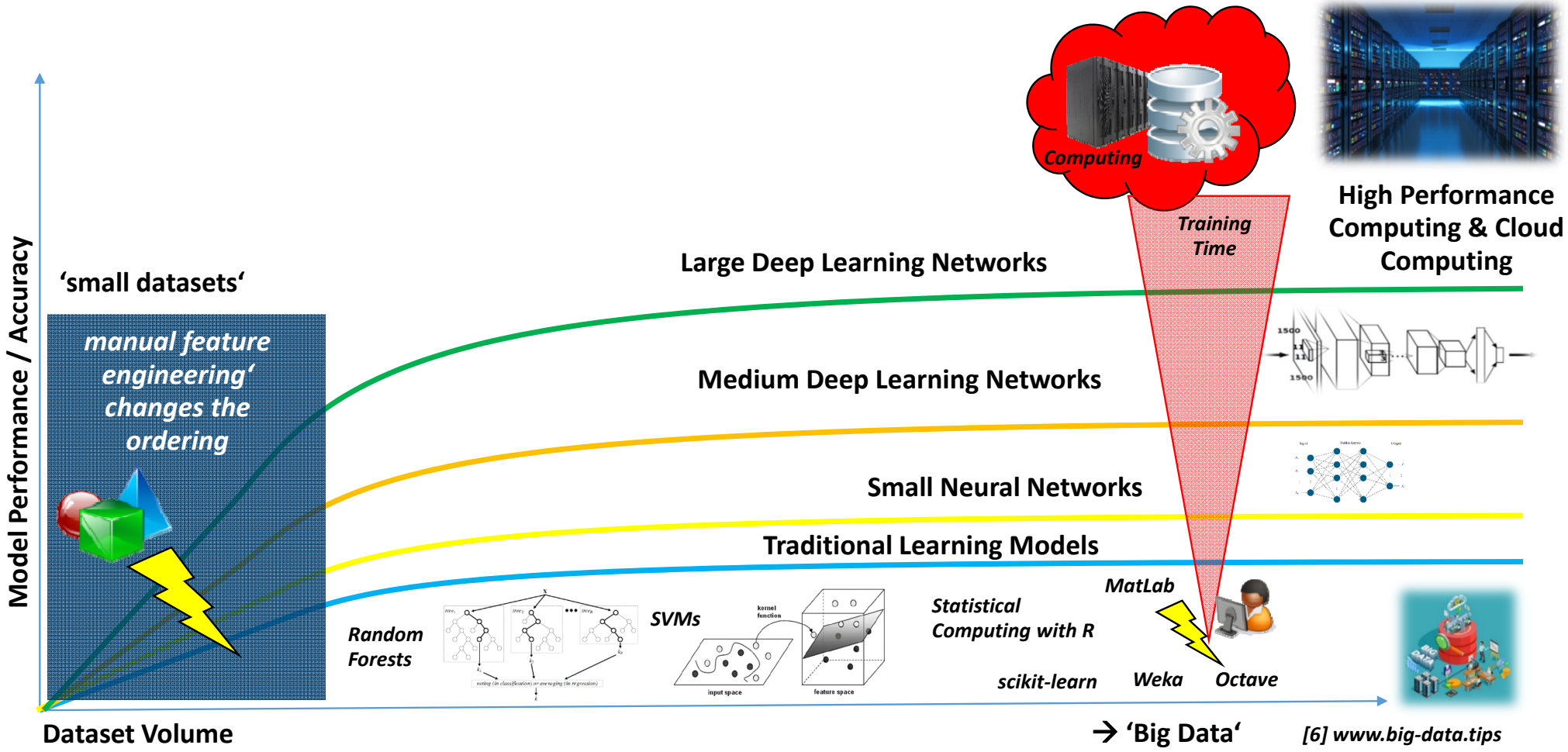
Bengio, Hinton, and LeCun Ushered in Major Breakthroughs in Artificial Intelligence

AWARDS & RECOGNITION

ACM Announces 2018 Turing Award Recipients

ACM has named Yoshua Bengio of the University of Montreal, Geoffrey Hinton of Google, and Yann LeCun of New York University recipients of the 2018 ACM A.M. Turing Award for conceptual and engineering breakthroughs that have made deep neural networks a critical component of computing. Working independently and together, Hinton, LeCun and Bengio developed conceptual foundations for the field, identified surprising phenomena through experiments, and contributed engineering advances that demonstrated the practical advantages of deep neural networks.

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

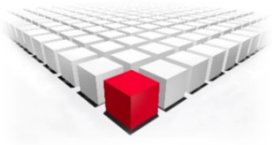
1995 Deep Convolutional Neural Networks
- Significant improvements in image analysis

...

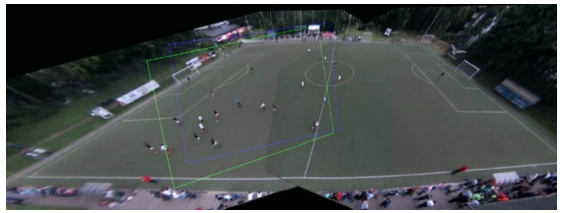
Impact in AI & HPC in industry & science

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

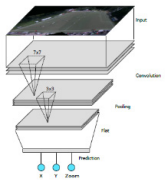


[7] NVIDIA

[8] Keras

[9] TensorFlow

[10] soccerwatch.tv

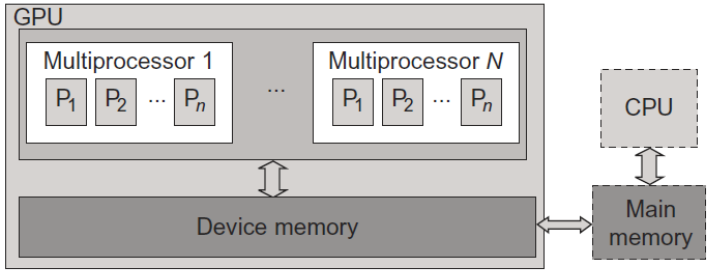


Combination: Start-up Example of my research group

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

Many-core GPGPUs Driving Deep Learning Momentum

- 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



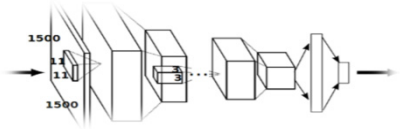
[23] Distributed & Cloud Computing Book

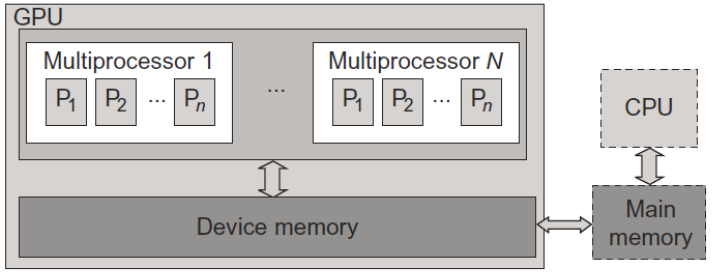
- Many-core GPUs are used in large clusters and within massively parallel supercomputers today
  - Named General-Purpose Computing on GPUs (GPGPU)
  - Different programming models emerge

- **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**

GPU Acceleration Benefits for Deep Learning

- GPU accelerator architecture example (e.g. NVIDIA card)
  - GPUs can have 128 cores on one single GPU chip
  - Each core can work with eight threads of instructions
  - GPU is able to concurrently execute 128 * 8 = 1024 threads
  - Interaction and thus major (bandwidth) bottleneck between CPU and GPU is via memory interactions
  - E.g. applications that use matrix – vector/matrix multiplication (e.g. deep learning algorithms)



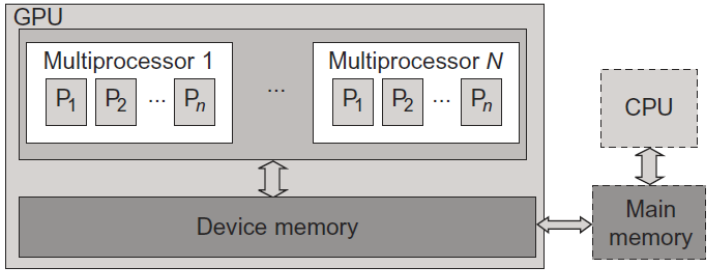


[23] Distributed & Cloud Computing Book

- **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**

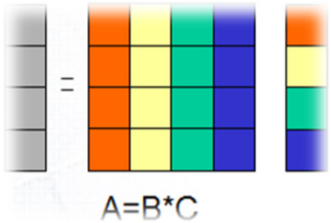
Parallel Computing Example on Many-Core GPUs – A Case for using Libraries

- General Purpose Graphical Processing Unit (GPGPU)
  - Designed to compute large numbers of floating point operations in parallel, but with moderate performance



[23] Distributed & Cloud Computing Book

- **Step 'zero': Data is loaded via the main memory of the CPU (i.e., host CPU memory) to the device memory of the GPU accessed by the many cores**
- **Step one: each GPU core has a column of matrix B (named as Bpart)**
- **Step one: each GPU core has an element of column vector C (named Cpart)**
- **Step two: Each GPU core performs an independent vector-scalar multiplication (i.e., independently based on their Bpart and Cpart contents)**
- **Step three: Each GPU core has a part of the result vector A (named Apart) and is written in device memory; results go to the main memory of CPU**



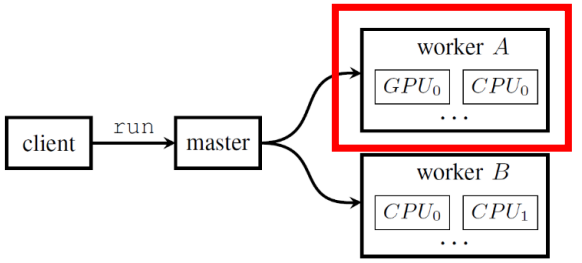
(nice parallelization possible via independent computing)

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} b_{0,0}c_0 + & b_{0,1}c_1 + & b_{0,2}c_2 + & b_{0,3}c_3 \\ b_{1,0}c_0 + & b_{1,1}c_1 + & b_{1,2}c_2 + & b_{1,3}c_3 \\ b_{2,0}c_0 + & b_{2,1}c_1 + & b_{2,2}c_2 + & b_{2,3}c_3 \\ b_{3,0}c_0 + & b_{3,1}c_1 + & b_{3,2}c_2 + & b_{3,3}c_3 \end{bmatrix}$$

P0 P1 P2 P3

Selected Deep Learning Frameworks & Tools used with GPUs

- TensorFlow
  - One of the most popular deep learning frameworks available today
  - Execution on multi-core CPUs or many-core GPUs



TensorFlow

[24] Tensorflow Web page

- **Tensorflow is an open source library for deep learning models using a flow graph approach**
- **Tensorflow nodes model mathematical operations and graph edges between the nodes are so-called tensors (also known as multi-dimensional arrays)**
- **The Tensorflow tool supports the use of CPUs and GPUs (much more faster than CPUs)**
- **Tensorflow work with the high-level deep learning tool Keras in order to create models fast**
- **New versions of Tensorflow have Keras shipped with it as well & many further tools**

- Keras
  - Often used in combination with low-level frameworks like Tensorflow

Keras

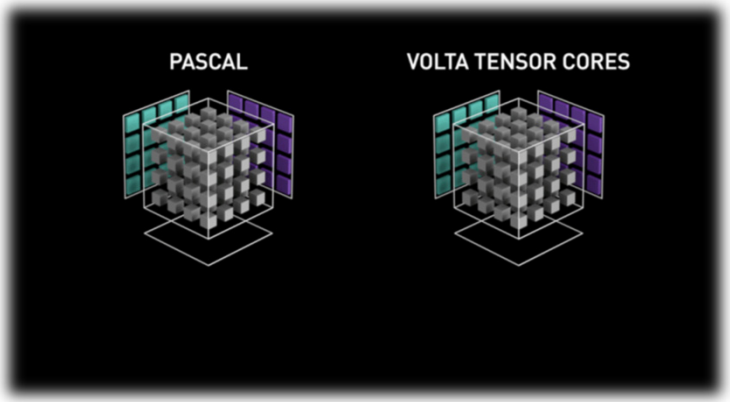
[25] Keras Web page

- **Keras is a high-level deep learning library implemented in Python that works on top of existing other rather low-level deep learning frameworks like Tensorflow, CNTK, or Theano**
- **Created deep learning models with Keras run seamlessly on CPU and GPU via low-level deep learning frameworks**
- **The key idea behind the Keras tool is to enable faster experimentation with deep networks**

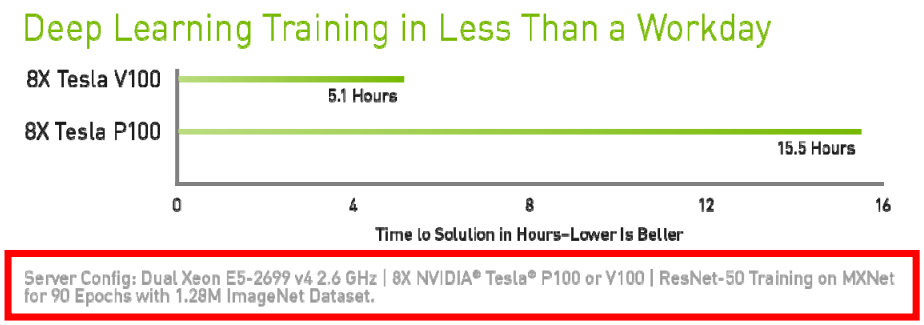
NVIDIA Tesla Volta (v100) GPU Example with Tensor Cores

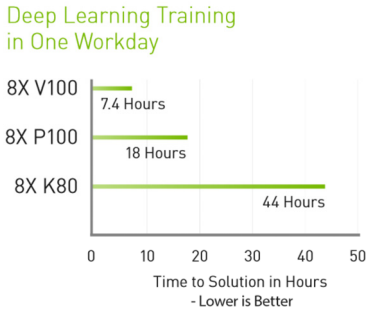
- Selected Facts
  - 150 + 150 GB/s total bandwidth (NVLink v2)

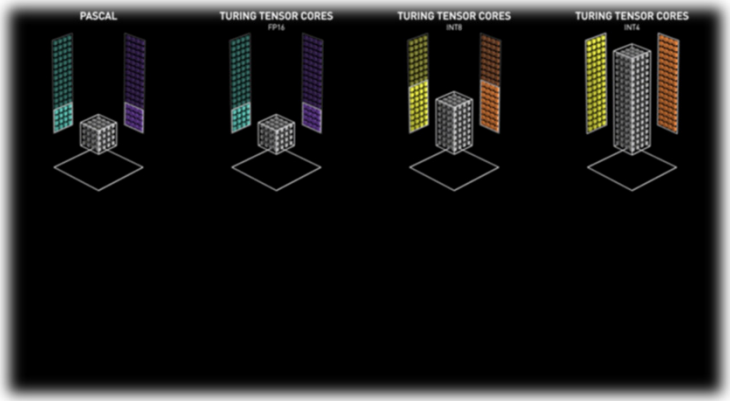
- **NVIDIA Tesla Volta (v100) is equipped with over 21 billion transistors with 5120 CUDA cores & 640 tensor cores**
- **A tensor core is optimized for deep learning workloads by accelerating large matrix operations & perform mixed-precision matrix multiply & accumulate calculations in a single operation**
- **Predecessor of NVIDIA v100 was NVIDIA Tesla Pascal (p100) and widely used in HPC systems**
- **Predecessor of NVIDIA p100 was NVIDIA Tesla Kepler (e.g., K80 or K40) & used in HPC systems**



Mixed precision (e.g. FP16 vs. FP32 better for deep learning due to a regularization effect for example)



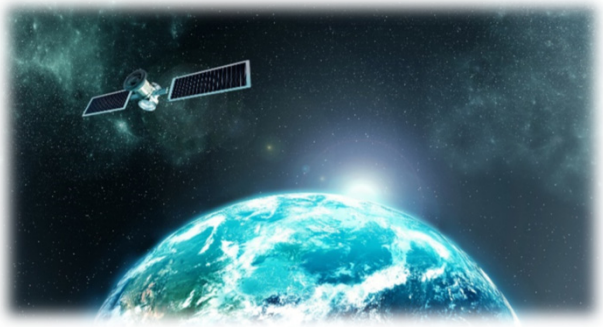


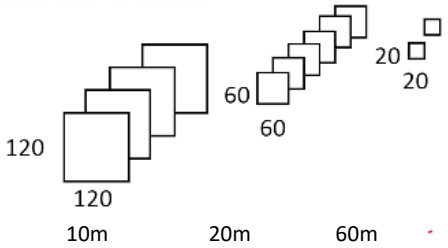


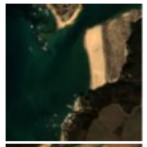
[29] Summit Architecture Overview [30] NVIDIA Tesla Volta v100

Multispectral Remote Sensing Dataset Example

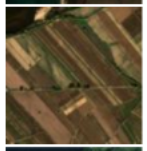
Datasets	Image type	Image per class	Scene classes	Annotation type	Total images	Spatial resolution (m)	Image sizes	Year	Ref. [9] G. Sumbul et al.
BigEarthNet	Satellite MS	328 to 217119	43	Multi label	590,326	10 20 60	120x120 60x60 20x20	2018	

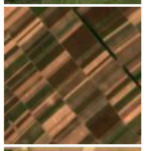





permanently irrigated land, sclerophyllous vegetation, beaches, dunes, sands, estuaries, sea and ocean


non-irrigated arable land, fruit trees and berry plantations, agro-forestry areas, transitional woodland/shrub


permanently irrigated land, vineyards, beaches, dunes, sands, water courses


non-irrigated arable land


coniferous forest, mixed forest, water bodies


discontinuous urban fabric, non-irrigated arable land, land principally occupied by agriculture, broad-leaved forest

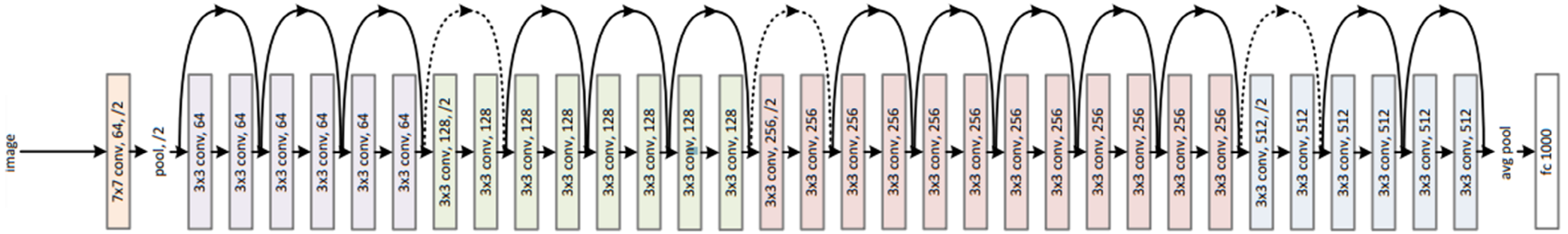
[26] G. Sumbul et al.

[27] Big Earth Net Dataset

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

More Computation: Deep Learning via RESNET-50 Architecture

- Classification of land cover in scenes
  - Very suitable for parallelization via distributed training on multi GPUs

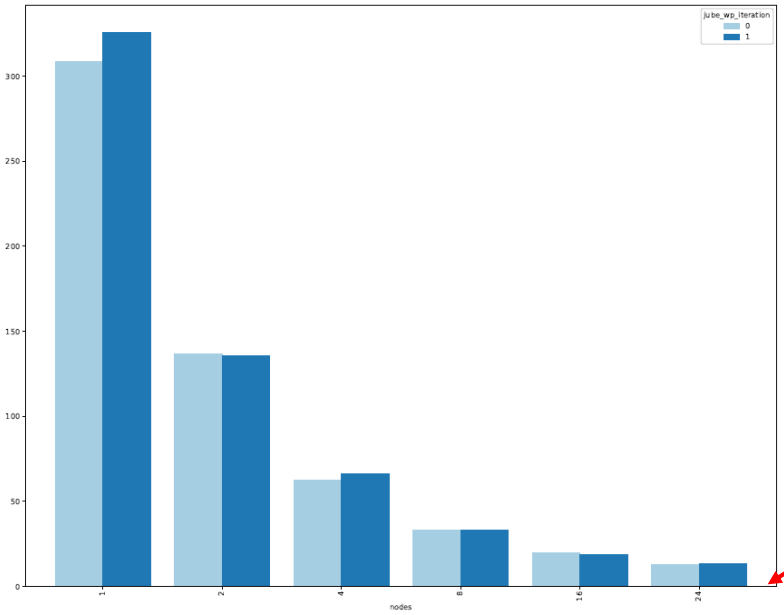


[14] RESNET

- **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**

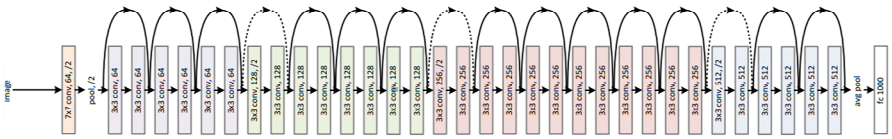
Distributed Training via Multi GPUs with Horovod – Remote Sensing Example

Time per epoch [sec]

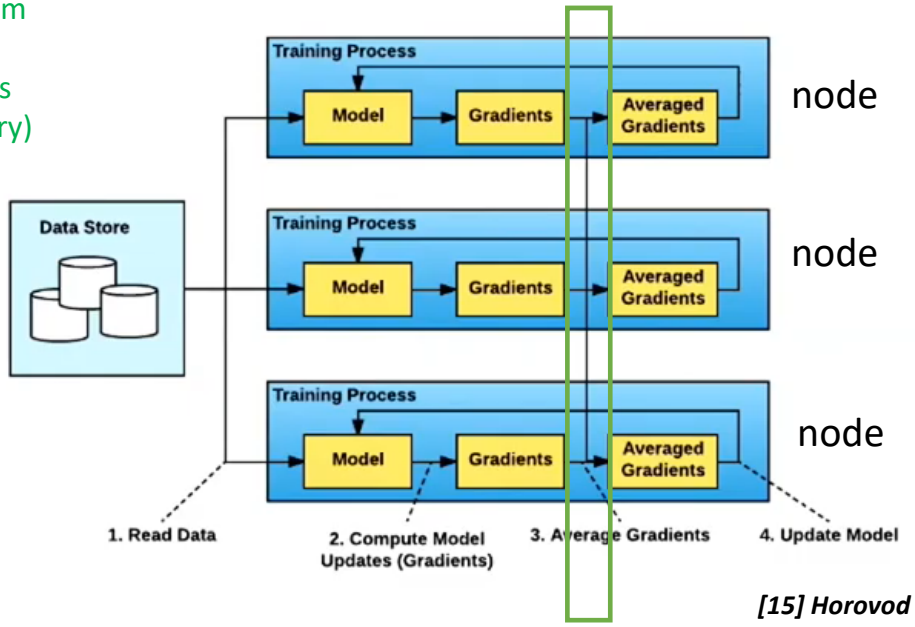


A partition of the JUWELS system has 56 compute nodes, each with 4 NVIDIA V100 GPUs (equipped with 16 GB of memory)

24 nodes x 4 GPUs = 96 GPUs



[14] RESNET



[15] 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 Training via Multi GPUs with Horovod – ImageNet Example

- Dataset: ImageNet
  - Total number of images: 14.197.122
  - Images with bounding box annotations: 1.034.908

(huge collection of images with high level categories)

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

[17] ImageNet Web page

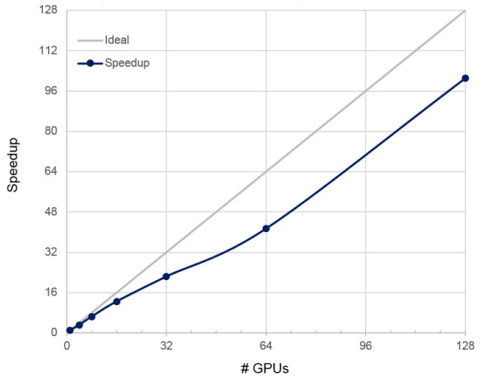
- **Open source tool Horovod enables distributed deep learning with TensorFlow / Keras**
- **Machine & Deep Learning: speed-up is just secondary goal after 1st goal accuracy**
- **Speed-up & parallelization good for faster hyperparameter tuning, training, inference**
- **Third goal is to avoid much feature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

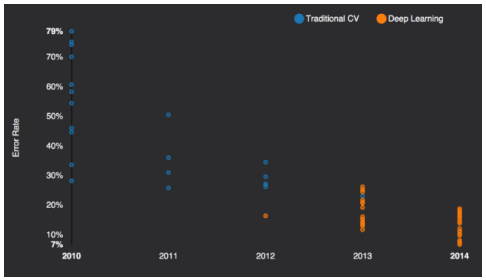
(ImageNet as a benchmark in deep learning community)

(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

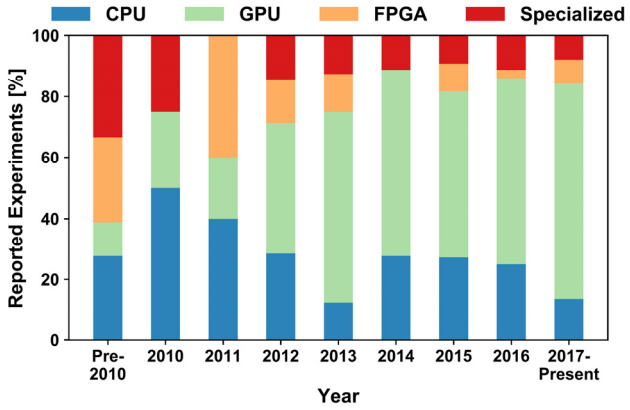
[15] Horovod

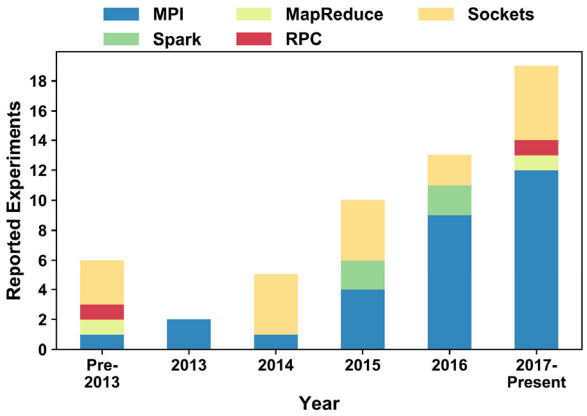


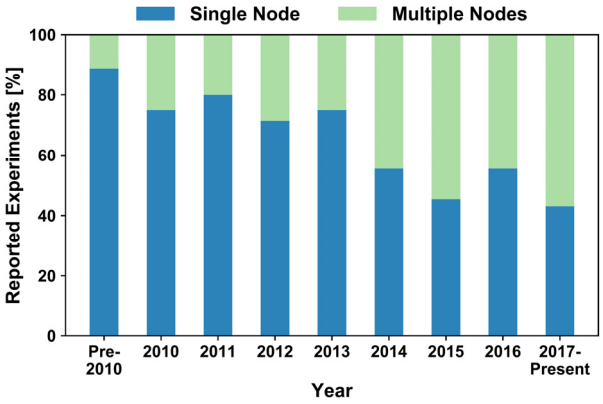


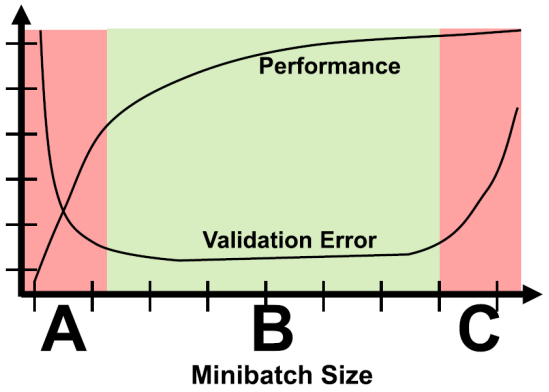
[16] J. Dean et al., 'Large-Scale Deep Learning'

Parallel Computing & HPC using GPUs for Deep Learning – Selected Impacts

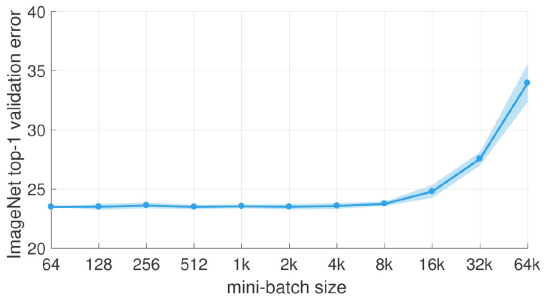






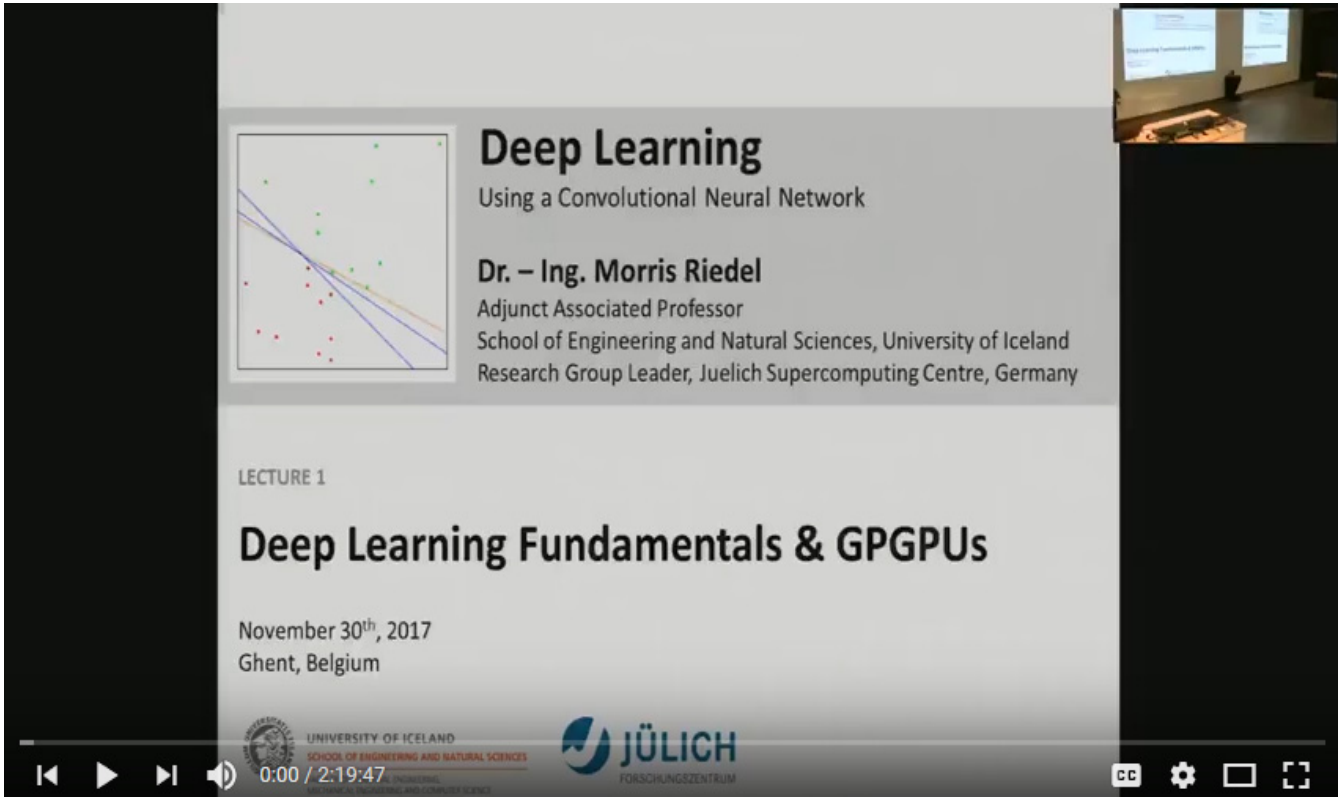


- **Facts: GPUs are mostly used today for deep learning compared to CPUs, FPGA, and specialized hardware**
- **Facts: ~55% of all users that use deep learning use it with multiple nodes instead of just a single node**
- **Facts: The communication layer MPI is mostly used as communication layer for distributed 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**



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

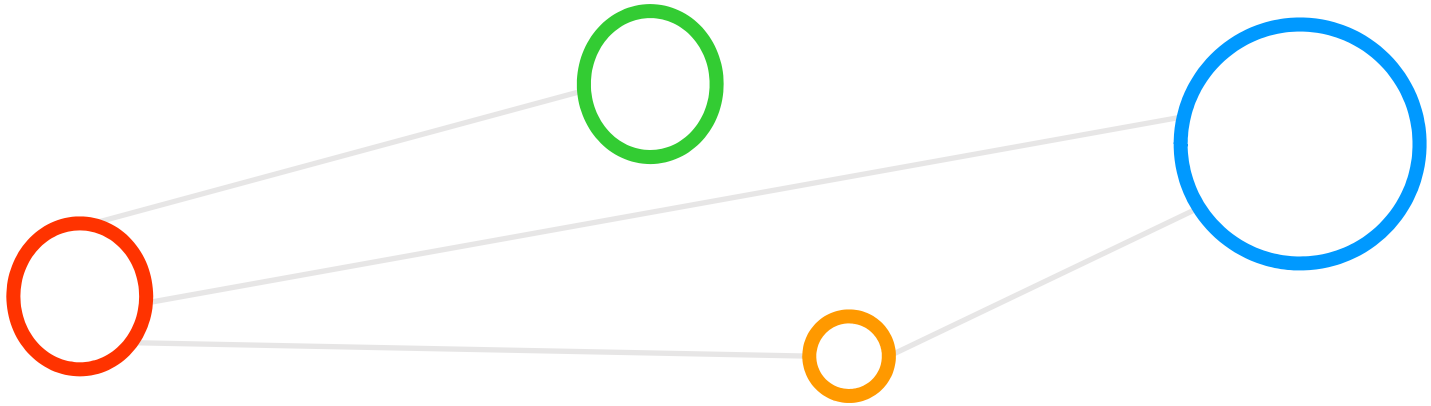
[YouTube Lectures] More Deep Learning Fundamentals

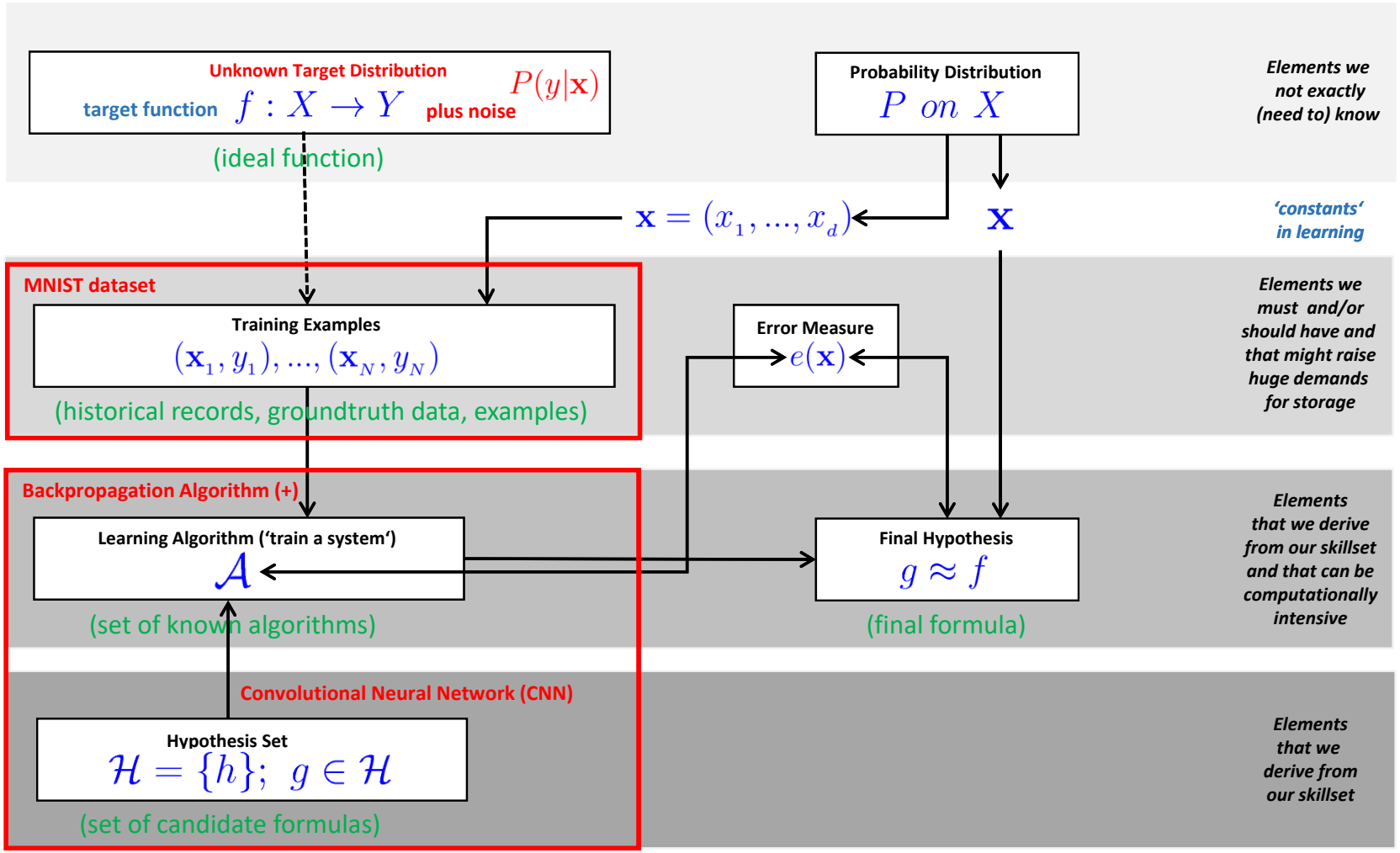


[13] Morris Riedel, 'Deep Learning - Using a Convolutional Neural Network', Invited YouTube Lecture, six lectures, University of Ghent, 2017

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

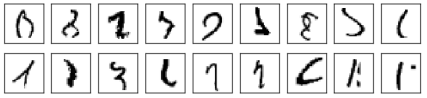
Convolutional Neural Networks (CNNs)



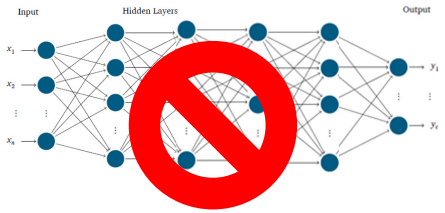

Unknown Target Distribution
target function $f : X \rightarrow Y$ plus noise $P(y|\mathbf{x})$
(ideal function)
Probability Distribution
P on X
Elements we not exactly (need to) know
$\mathbf{x} = (x_1, ..., x_d)$
$\mathbf{X}$
'constants' in learning
MNIST dataset
Training Examples
$(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)$
(historical records, groundtruth data, examples)
Error Measure
$e(\mathbf{x})$
Elements we must and/or should have and that might raise huge demands for storage
Backpropagation Algorithm (+)
Learning Algorithm ('train a system')
$\mathcal{A}$
(set of known algorithms)
Final Hypothesis
$g \approx f$
(final formula)
Elements that we derive from our skillset and that can be computationally intensive
Convolutional Neural Network (CNN)
Hypothesis Set
$\mathcal{H} = \{h\}; \ g \in \mathcal{H}$
(set of candidate formulas)
Elements that we derive from our skillset

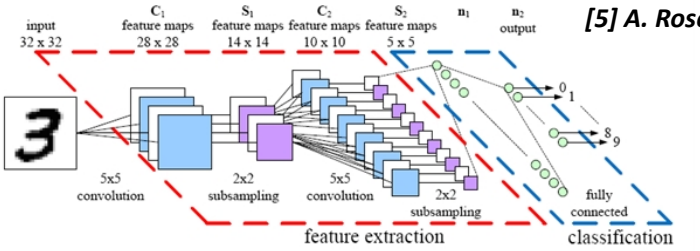
Deep Learning Technique Example – Convolutional Neural Networks (CNNs)

[4] Neural Network 3D Simulation



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



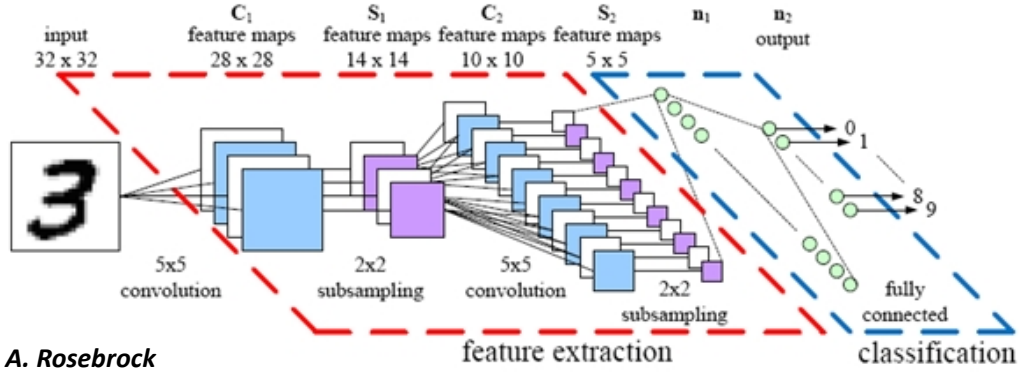


[5] A. Rosebrock

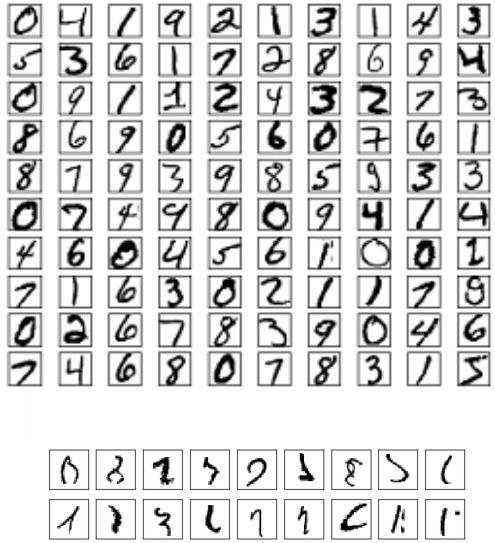
CNNs – Basic Principles

- **Convolutional Neural Networks (CNNs/ConvNets) implement a connectivity pattner between neurons inspired by the animal visual cortex and use several types of layers (convolution, pooling)**
- **CNN key principles are local receptive fields, shared weights, and pooling (or down/sub-sampling)**
- **CNNs are optimized to take advantage of the spatial structure of the data**

- Simple application example
  - MNIST database written characters
  - Use CNN architecture with different layers
  - Goal: automatic classification of characters



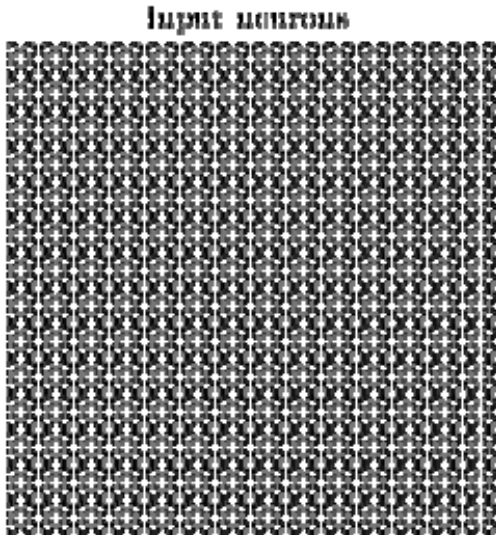
[5] A. Rosebrock



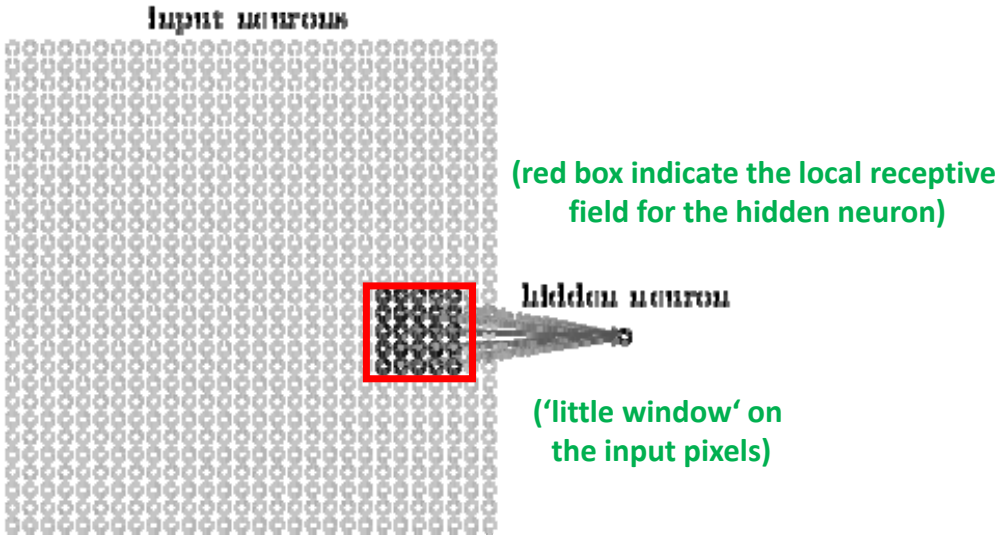
[31] M. Nielsen

CNNs – Principle Local Receptive Fields

- MNIST dataset example
  - 28 * 28 pixels modeled as square of neurons in a convolutional net
  - Values correspond to the 28 * 28 pixel intensities as inputs



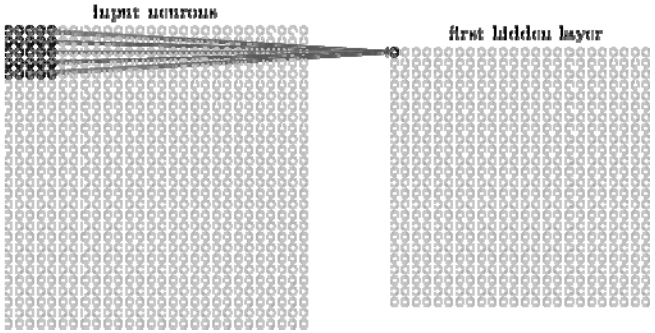
(28 * 28 pixel image)



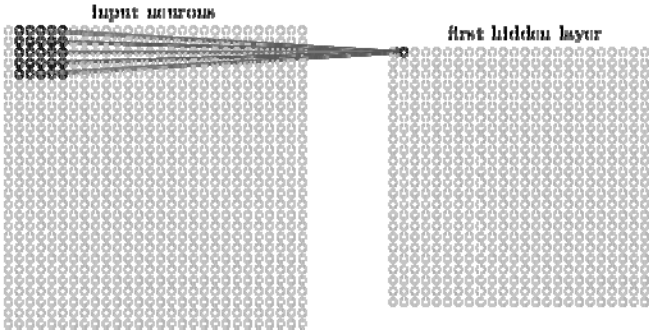
(5 * 5 local connectivity)

CNNs – Principle Local Receptive Fields & Sliding

- MNIST database example
  - Apply stride length = 1
  - Different configurations possible and depends on application goals
  - Creates ‘feature map‘ of 24 * 24 neurons (hidden layer)



(28 * 28 pixel image) (24 * 24 feature map)



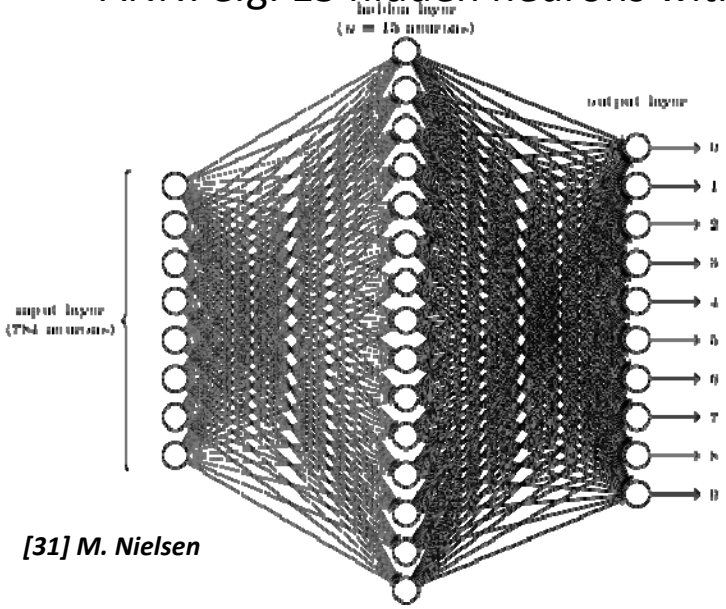
(28 * 28 pixel image) (24 * 24 feature map)

Exercise: Why might it be good in Deep Learning to share Weights?

CNNs – Compare Example with an ANN with risk of Overfitting (cf. Lecture 5)

- MNIST database example
  - CNN: e.g. 20 feature maps with 5 * 5 (+bias) = 520 weights to learn
  - Apply ANN that is fully connected between neurons
  - ANN: fully connected first layer with 28 * 28 = 784 input neurons
  - ANN: e.g. 15 hidden neurons with 784 * 15 = 11760 weights to learn

(eventually lead to overfitting and much computing time)

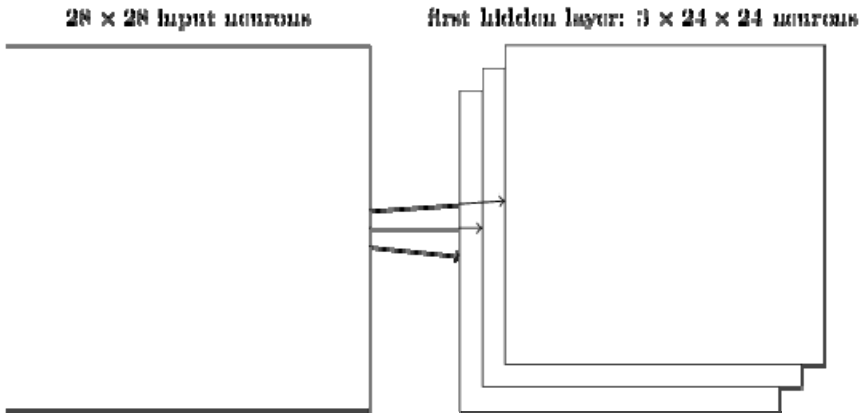


[31] M. Nielsen

- **Overfitting refers to fit the data too well – more than is warranted – thus may misguide the learning – rather memorizing than realy learning**
- **A good model must have low training error (E_{in}) and low generalization error (E_{out})**
- **Model overfitting is if a model fits the data too well (E_{in}) with a poorer generalization error (E_{out}) than another model with a higher training error (E_{in})**
- **The two general approaches to prevent overfitting are (1) regularization and (2) validation**

CNNs – Principle Shared Weights & Feature Maps

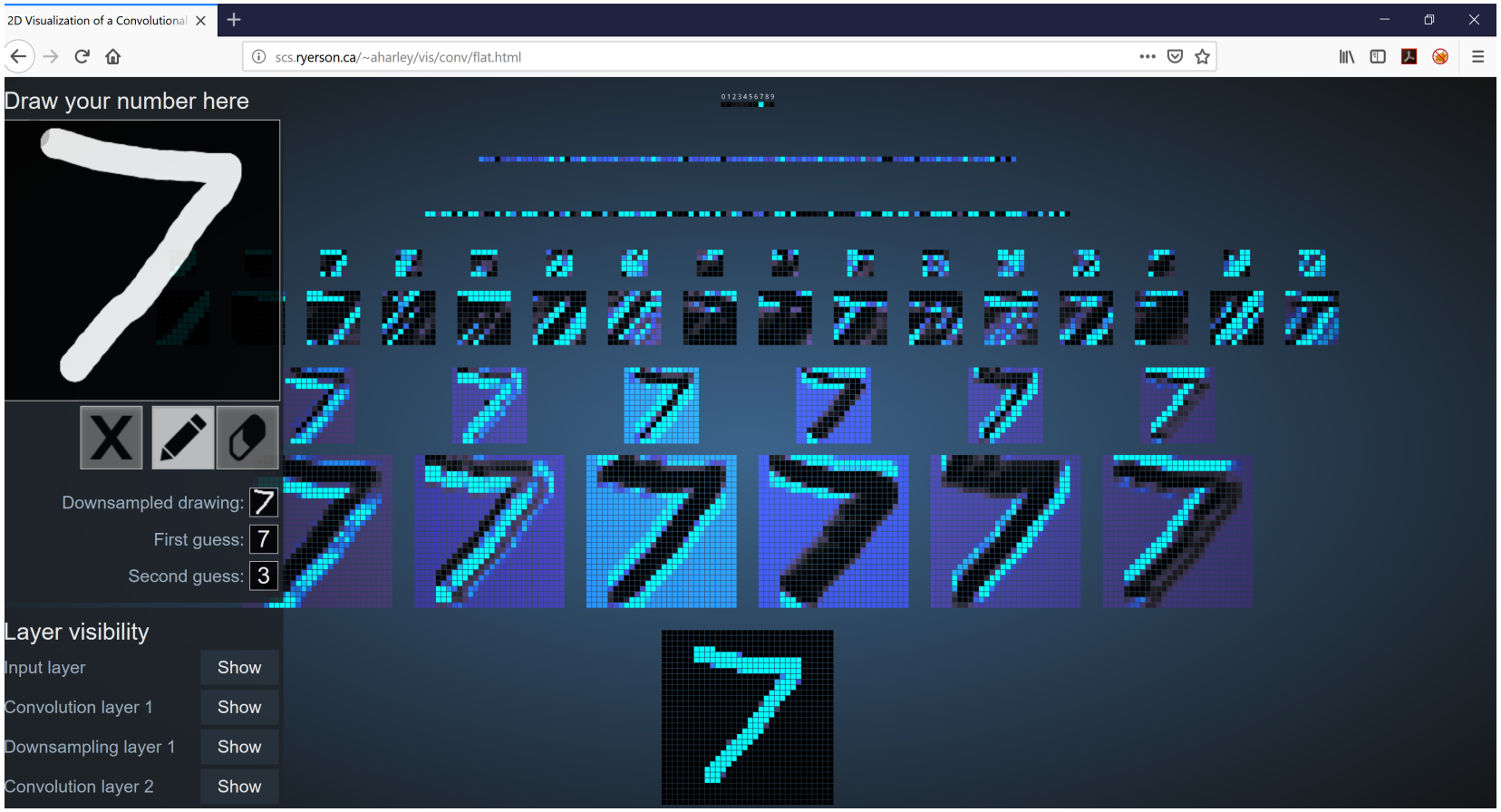
- Approach
  - CNNs use same shared weights for each of the 24 * 24 hidden neurons
  - Goals: significant reduction of number of parameters (prevent overfitting)
  - Example: 5 * 5 receptive field → 25 shared weights + shared bias
- Feature Map
  - Detects one local feature
  - E.g. 3: each feature map is defined by a set of 5 * 5 shared weights and a single shared bias leading to 24 * 24
  - Goal: The network can now detect 3 different kind of features **(many more in practice)**



(shared weights are also known to define a kernel or filter)

  - Benefit: learned feature being detectable across the entire image

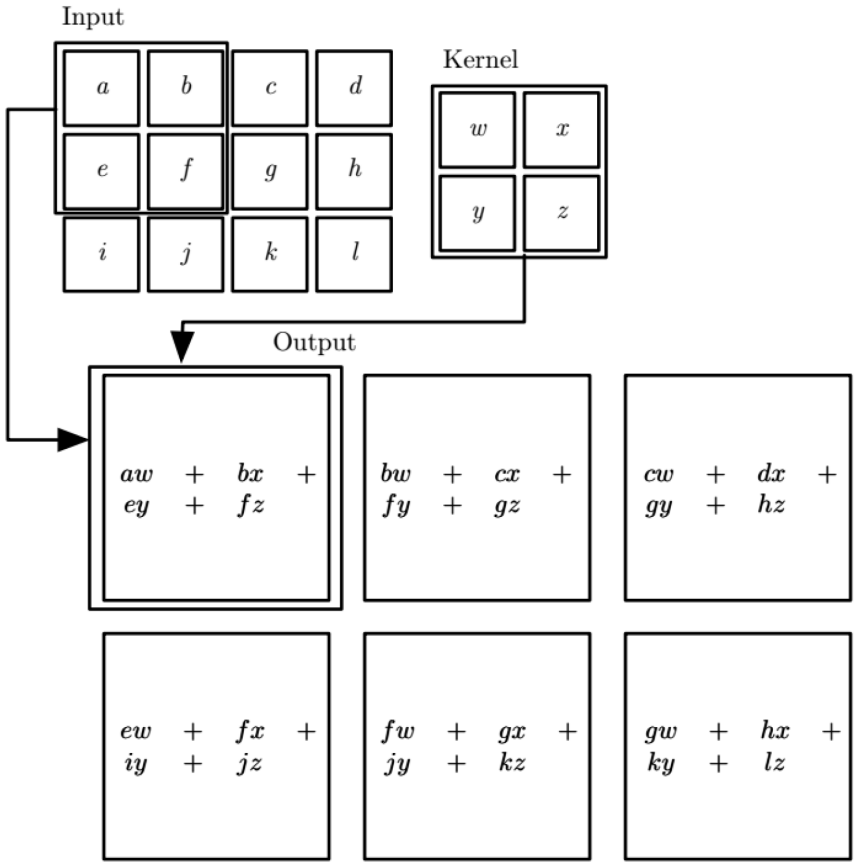
Understanding Feature Maps & Convolutions – Online Web Tool



[12] Harley, A.W., An Interactive Node-Link Visualization of Convolutional Neural Networks

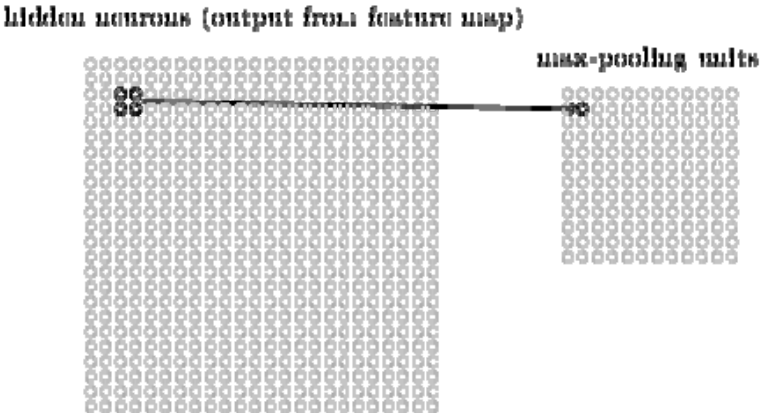
CNNs – Principle of Convolution & Example

- Example
  - 3x4 input matrix processed by a 2x2 kernel with stride=1 that calculates the sum of ist content
- Terminology
  - Valid convolution does not exceed the input‘s boundary
  - Same convolution adds a so called ‘padding‘ to maintain the input‘s dimension for each convolutional layer

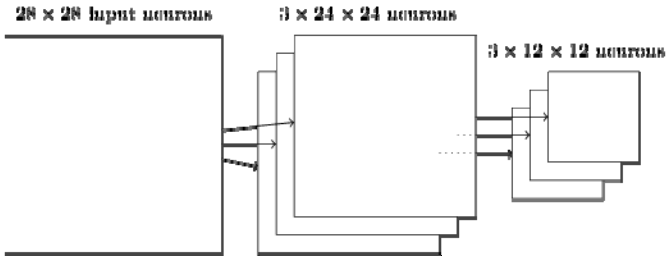


CNNs – Principle of Pooling

- ‘Downsampling‘ Approach
  - Usually applied directly after convolutional layers
  - Idea is to simplify the information in the output from the convolution
  - Take each feature map output from the convolutional layer and generate a condensed feature map
  - E.g. Pooling with 2 * 2 neurons using ‘max-pooling‘
  - Max-Pooling outputs the maximum activation in the 2 * 2 region

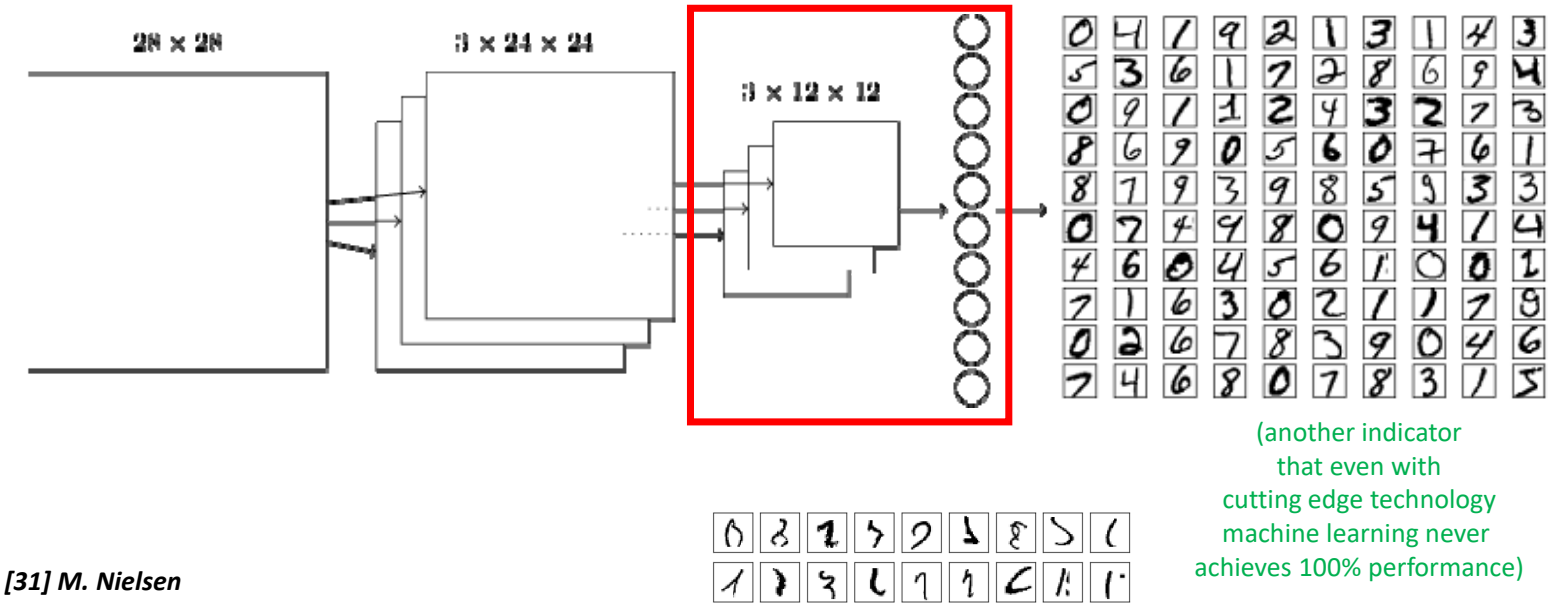


[31] M. Nielsen



CNN – Understanding Application Example MNIST

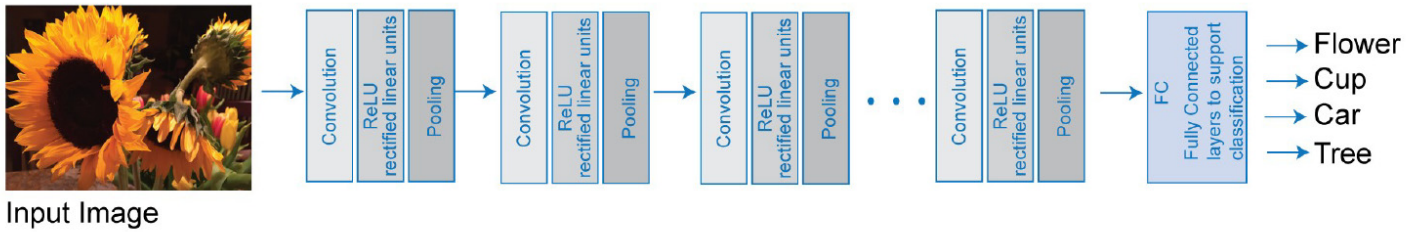
- MNIST database example
  - Full CNN with the addition of output neurons per class of digits
  - Apply ‘fully connected layer‘: layer connects every neuron from the max-pooling outcome layer to every neuron of the 10 out neurons
  - Train with backpropagation algorithm (gradient descent), only small modifications for new layers



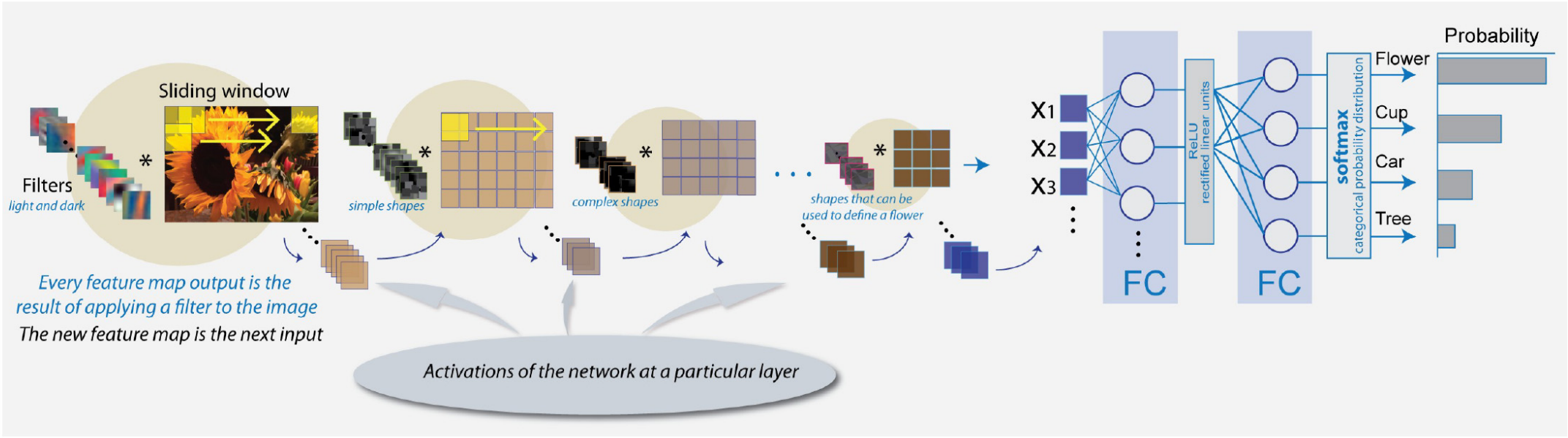
(another indicator that even with cutting edge technology machine learning never achieves 100% performance)

[31] M. Nielsen

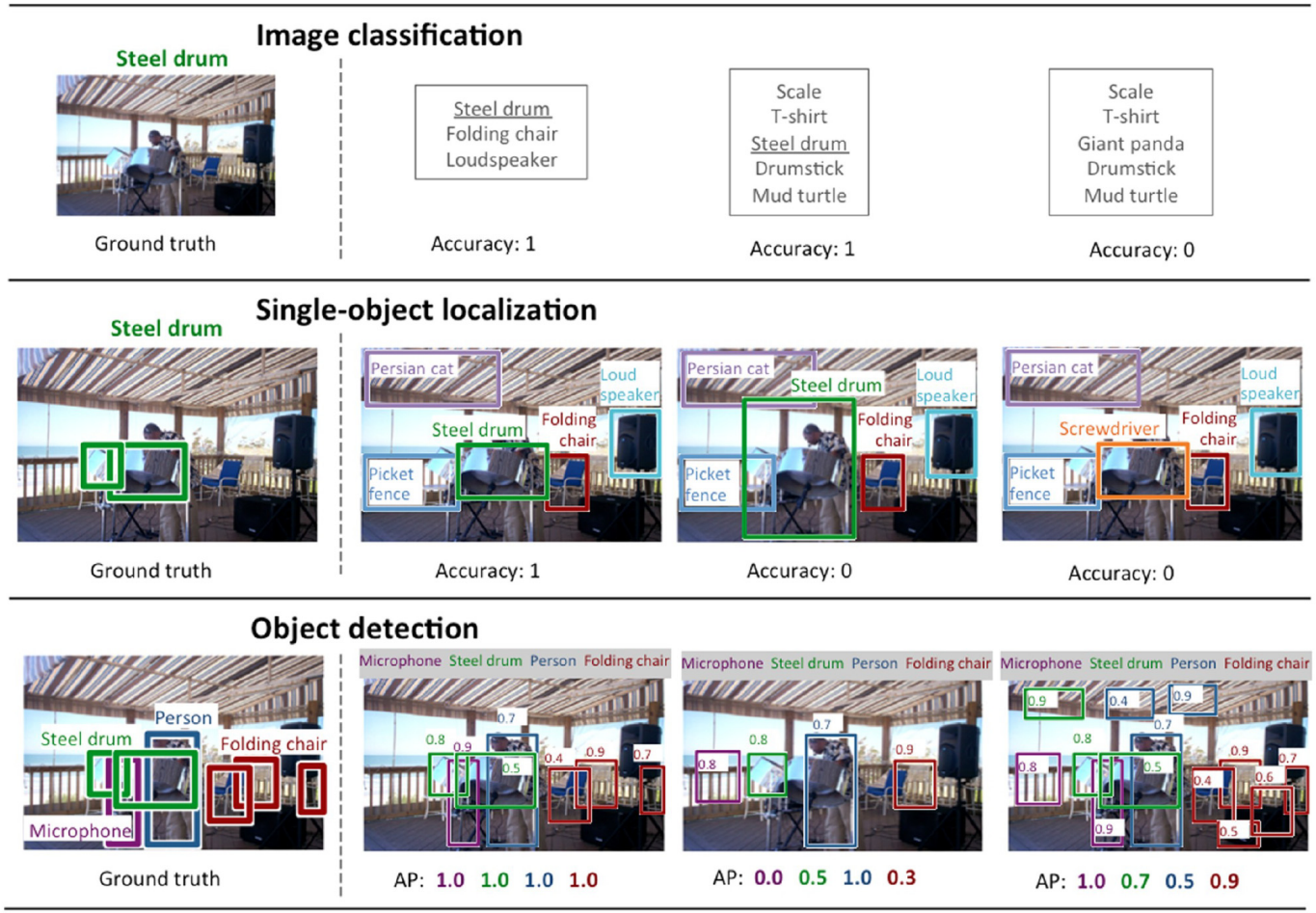
CNN – ImageNet Application Example



[17] ImageNet Web page



CNN – ImageNet Advanced Application Examples & Benchmark



[33] ImageNet Large Scale Visual Recognition Challenge

Distributed Training via Multi GPUs with Horovod – ImageNet – Revisited

- Dataset: ImageNet
  - Total number of images: 14.197.122
  - Images with bounding box annotations: 1.034.908

(huge collection of images with high level categories)

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

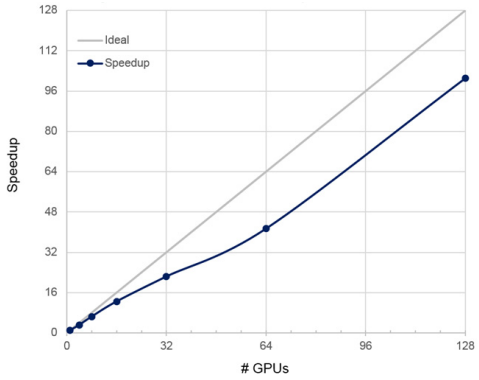
- **Open source tool Horovod enables distributed deep learning with TensorFlow / Keras**
- **Machine & Deep Learning: speed-up is just secondary goal after 1st goal accuracy**
- **Speed-up & parallelization good for faster hyperparameter tuning, training, inference**
- **Third goal is to avoid much feature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

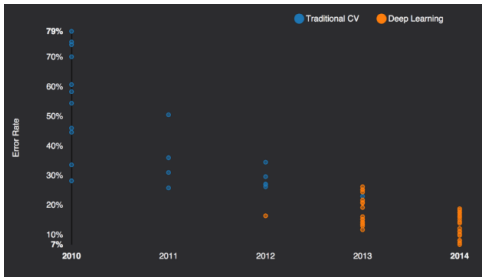
(ImageNet as a benchmark in deep learning community)

(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

[15] Horovod

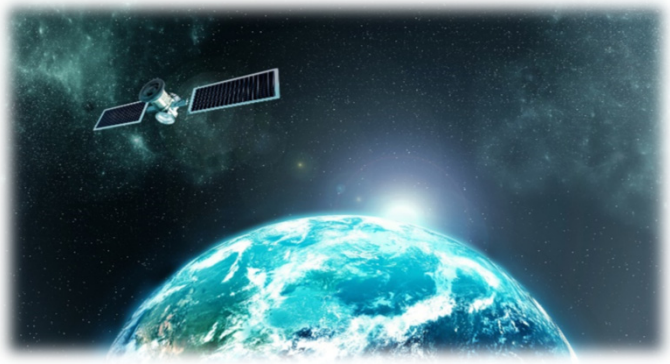




[16] J. Dean et al., 'Large-Scale Deep Learning'

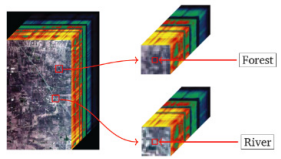
[17] ImageNet Web page

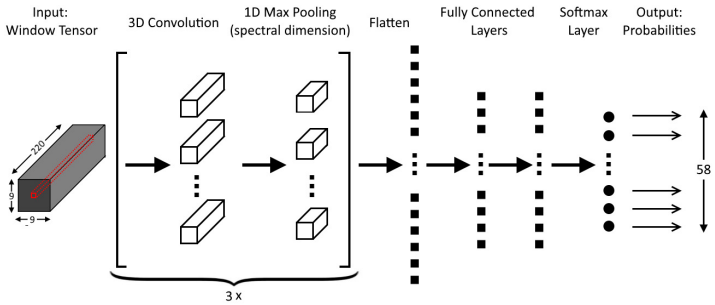
Deep Learning Application Examples – Key Challenge: Find the Right Parameters



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

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

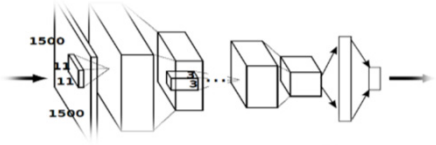




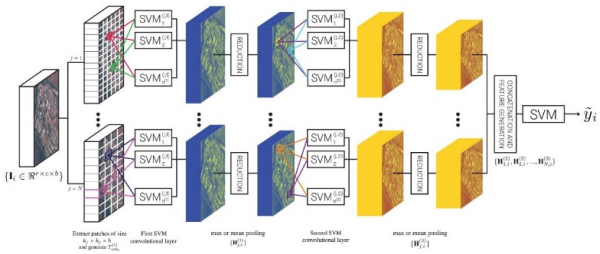
- **Find right set of hyper-parameters and the right neural network architecture is a manual time-consuming and error-prone process**
- **Needs urgently HPC, but a systematic and automated way is required as trying out all options of hyper-parameters and architectures is computationally infeasible**

- 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?

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}



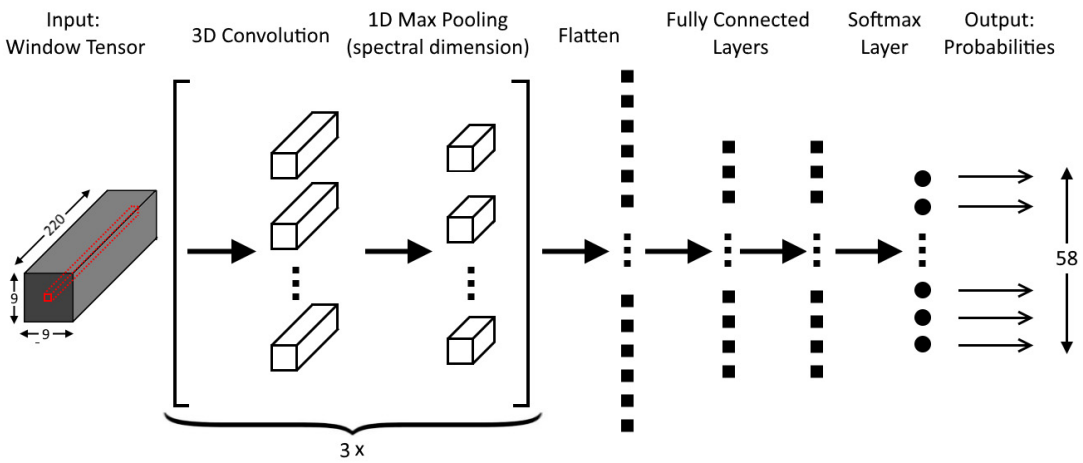
- **Find Hyperparameters & joint 'new-old' modeling & transfer learning given rare labeled/annotated data in science (e.g. 36,000 vs. 14,197,122 images ImageNet)**

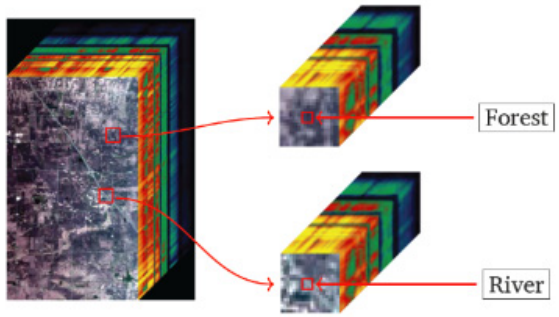


[20] G. Cavallaro, M. Riedel et al., IGARSS 2019

CNN Challenges & Traps – Careful with Training & Testing Datasets

- Window Tensor
  - Possible problem
  - Re-using pixel for training and testing
  - Need to keep them separate





Training Examples

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

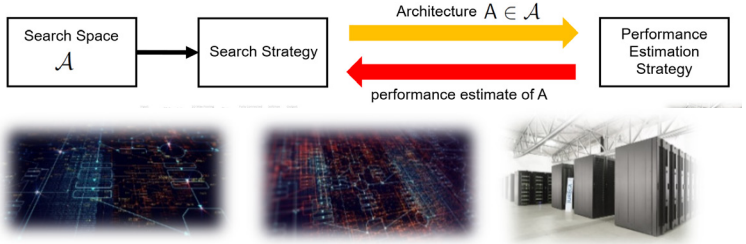
Test Examples

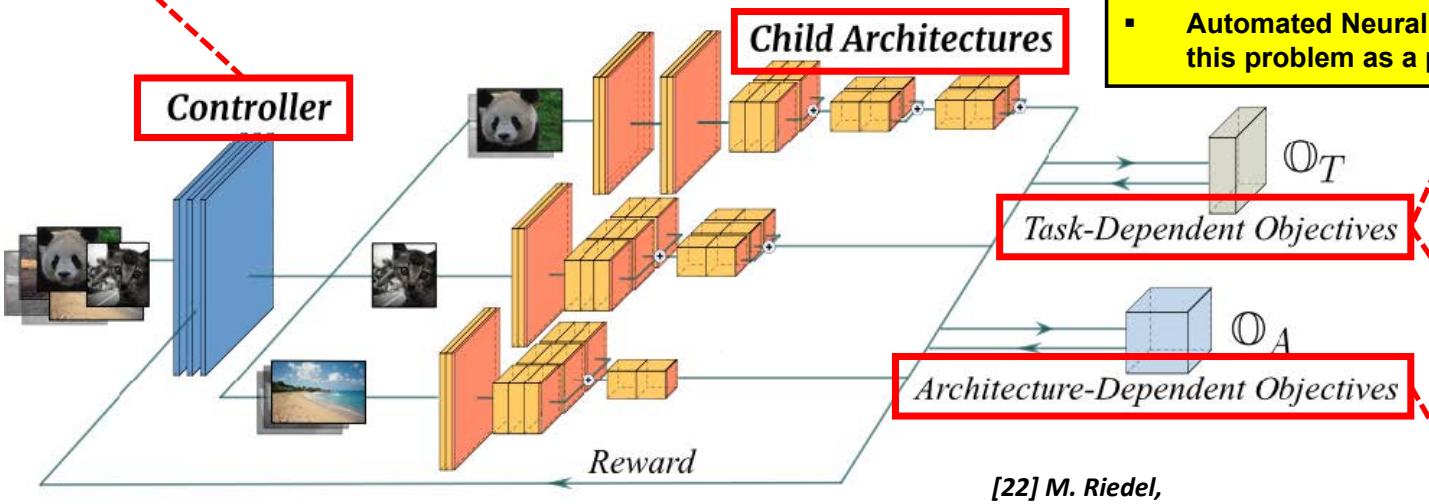
$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$$

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

Massive Requirement for HPC Resources: Neural Architecture Search (NAS)

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





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

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

- 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

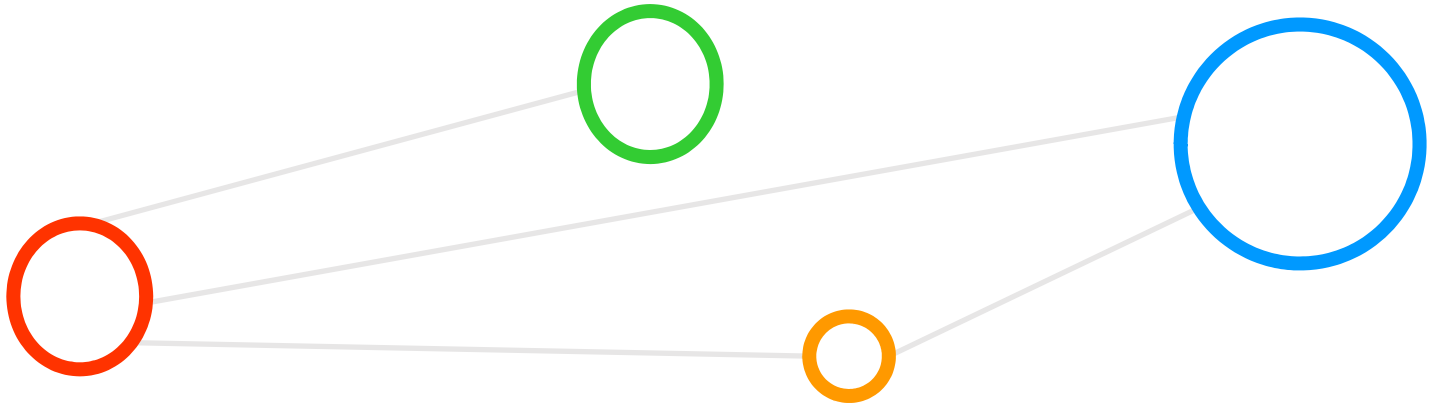
- Derived specific architectures that perform good for specific dataset samples
- E.g. what is the accuracy or error rate we obtain as metric to guide the search for specific architectures for specific dataset samples
- E.g. what is the latency of the network for a given dataset sample to guide the search for specific architectures that offer better latency by keeping accuracy(!)

➢ Day 3 offers details about reinforcement learning and also partly uses Neural Architecture Search (NAS) as one application example

[Video] CNN Application in Autonomous Driving

[32] YouTube Video, CNN in Speed Sign Recognition

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] Harley, A.W., An Interactive Node-Link Visualization of Convolutional Neural Networks, Online: http://scs.ryerson.ca/~aharley/vis/conv/flat.html
- [13] Morris Riedel, 'Deep Learning - Using a Convolutional Neural Network', Invited YouTube Lecture, six lectures, University of Ghent, 2017, Online: https://www.youtube.com/watch?v=gOL1_YIosYk&list=PLrmNhuZo9sgZUdaZ-f6OHK2yFW1kTS2qF
- [14] Kaiming He et al., 'Deep Residual Learning for Image Recognition', Online: https://arxiv.org/pdf/1512.03385.pdf
- [15] 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
- [16] J. Dean et al., 'Large scale deep learning', Keynote GPU Technical Conference, 2015
- [17] ImageNet Web page, Online: http://image-net.org
- [18] T. Ben-Nun & T. Hoefler, 'Demystifying Parallel and Distributed Deep Learning: An In-depth Concurrency Analysis', Online: http://doi.acm.org/10.1145/3320060
- [19] 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
- [20] 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, Japan, Online: https://www.researchgate.net/publication/337439088_Multi-Scale_Convolutional_SVM_Networks_for_Multi-Class_Classification_Problems_of_Remote_Sensing_Images
- [21] Cheng, A.C, Lin, C.H., Juan, D.C., InstaNAS: Instance-aware Neural Architecture Search, Online: https://arxiv.org/abs/1811.10201

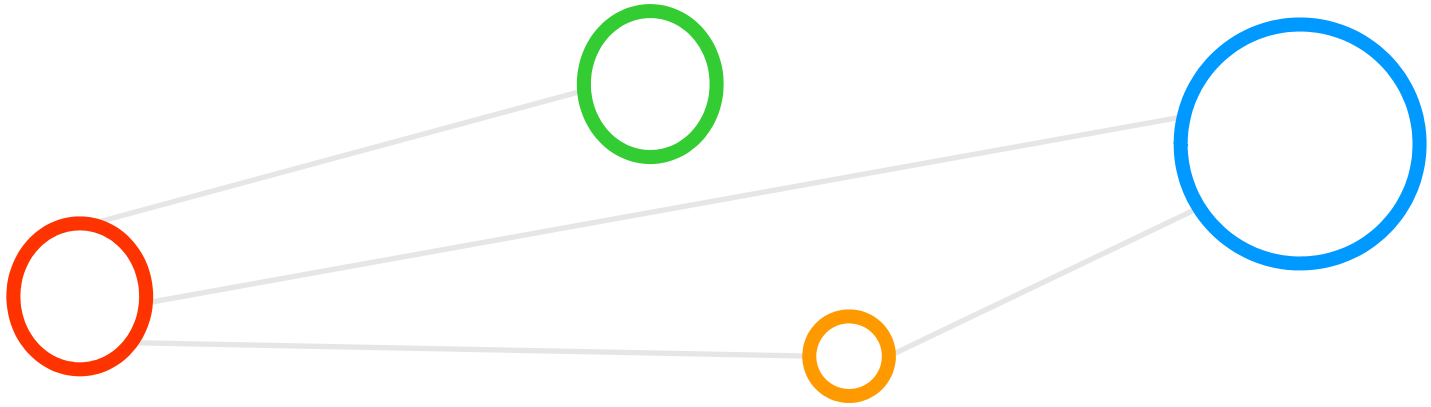
Lecture Bibliography (3)

- [22] M. Riedel, 'Neural Architecture Search with Reinforcement Learning', Online: http://www.morrisriedel.de/neural-architecture-search-with-reinforcement-learning
- [23] 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
- [24] Tensorflow, Online: https://www.tensorflow.org/
- [25] Keras Python Deep Learning Library, Online: https://keras.io/
- [26] G. Sumbul, M. Charfuelan, B. Demir, V. Markl, BigEarthNet: A Large-Scale Benchmark Archive for Remote Sensing Image Understanding, IEEE International Conference on Geoscience and Remote Sensing Symposium, Yokohama, Japan, 2019.
- [27] Tensorflow Dataset 'Big Earth Net', Online: https://www.tensorflow.org/datasets/datasets#bigearthnet
- [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, Online: https://www.researchgate.net/publication/338077024_Remote_Sensing_Big_Data_Classification_with_High_Performance_Distributed_Deep_Learning
- [29] Summit Supercomputer Architecture Overview, Online: https://www.olcf.ornl.gov/wp-content/uploads/2019/05/Summit_System_Overview_20190520.pdf
- [30] NVIDIA Tesla Volta v100, Online: https://www.nvidia.com/en-us/data-center/tesla-v100/
- [31] M. Nielsen, 'Neural Networks and Deep Learning', Online: http://neuralnetworksanddeeplearning.com/

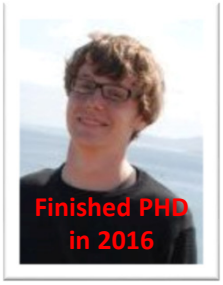
Lecture Bibliography (4)

- [32] YouTube Video, ‘Speed Sign Recognition by Convolutional Neural Networks‘, Online: https://www.youtube.com/watch?v=kkha3sPoU70
- [33] Russakovsky et al. ‘ImageNet Large Scale Visual Recognition Challenge‘, 2015, Online: https://arxiv.org/pdf/1409.0575.pdf

Acknowledgements



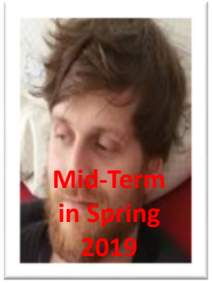
Acknowledgements – High Productivity Data Processing Research Group



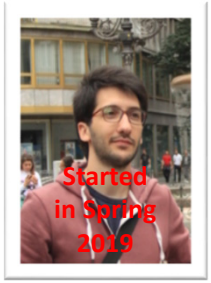
PD Dr.
G. Cavallaro

Senior PhD
Student A.S. Memon

Senior PhD
Student M.S. Memon



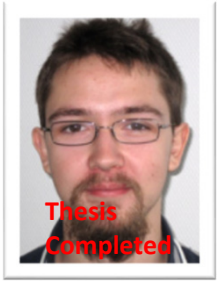
PhD Student
E. Erlingsson



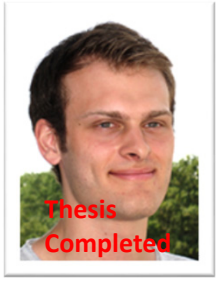
PhD Student
S. Bakarat

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)

services
measurement
within
fusion
new
funding
Policy-based
concepts
forms
cross-disciplinary
resources
device
Centre
Analysis
climate
Computer
expertise
year
like
Computational
Test
E.g
Computing
disciplines
results
Approaches
construction
Fusion
Enable
Cross-Disciplinary
High
big
nodes
PB
community
Environment
Modelling
different
EU
methods
centers
Key
scientific
important
files
Science
etc
DKRZ
Use
storage
basis
Infrastructure
international
model
simulations
ENES
technologies
SMAQ
engineering
References
formats
brain
increasing
via
EUDAT
physical
IPCC
Big
Project
Data
processing
Common
TB
Methods
MapReduce
create
computing
Using
European
databases
GB
Supercomputing
Work
images
codes
analysis
stored
Resources
analyze
Large
Summary
data-intensive
Service
Scientific
levels
often
DLCL
Level
external
use
Matters
Concrete
performance
JSC
computational
Research
research
e.g
science
HPC
Juelich
datasets
acquisition
large-scale
compute
system
Landscape
management
System
Climate
modelling
AR5
Hardware
distributed
cores
access
hundreds
Nutshell
Services
Technologies
Earth
Structure
current
network
DLCLs
Understanding
structures
directory
project
General
Germany
Simulation
Processing
Provide
NASA
Energy
systems
day
Health