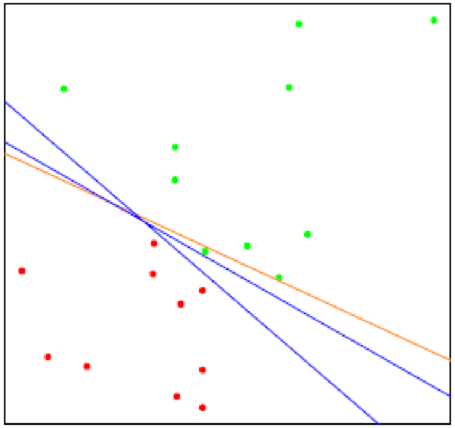


Deep Learning

Introduction to Deep Learning Models

Ernir Erlingsson

PhD Student, University of Iceland & DEEP-EST Application Engineer

LECTURE 2

Fundamentals of Convolutional Neural Networks

June 6th, 2018

JSC, Germany

UNIVERSITY OF ICELAND

SCHOOL OF ENGINEERING AND NATURAL SCIENCES

FACULTY OF INDUSTRIAL ENGINEERING,
MECHANICAL ENGINEERING AND COMPUTER SCIENCE

JÜLICH Forschungszentrum

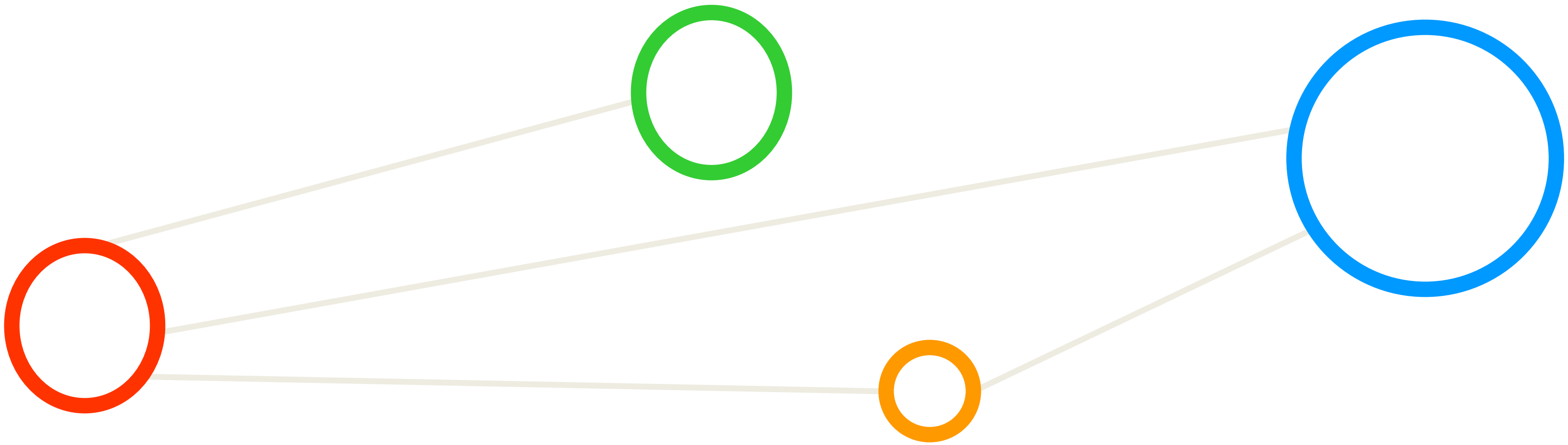
HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

DEEP Projects

Outline of the Course

- 1. Introduction to Deep Learning
- 2. Fundamentals of Convolutional Neural Networks (CNNs)
- 3. Deep Learning in Remote Sensing: Challenges
- 4. Deep Learning in Remote Sensing: Applications
- 5. Model Selection and Regularization
- 6. Fundamentals of Long Short-Term Memory (LSTM)
- 7. LSTM Applications and Challenges
- 8. Deep Reinforcement Learning

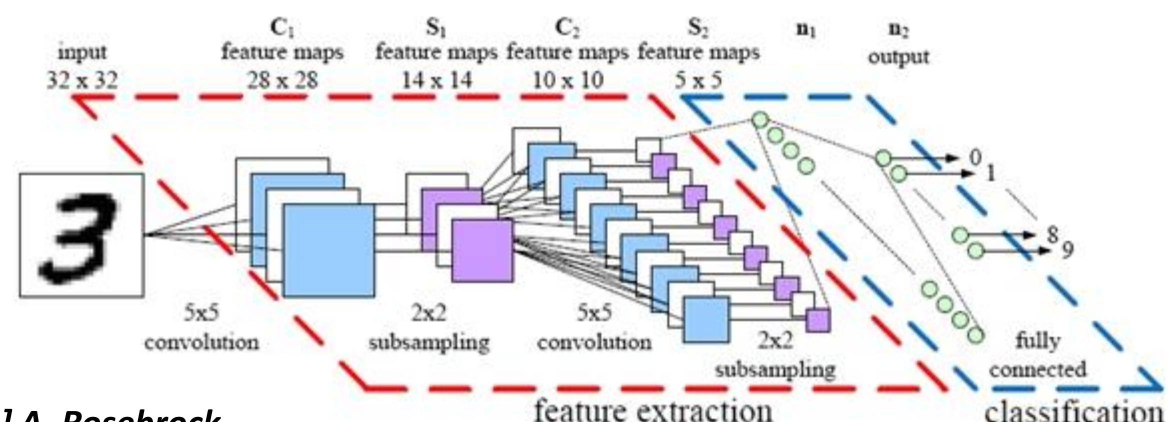
Convolutional Neural Networks (CNNs)



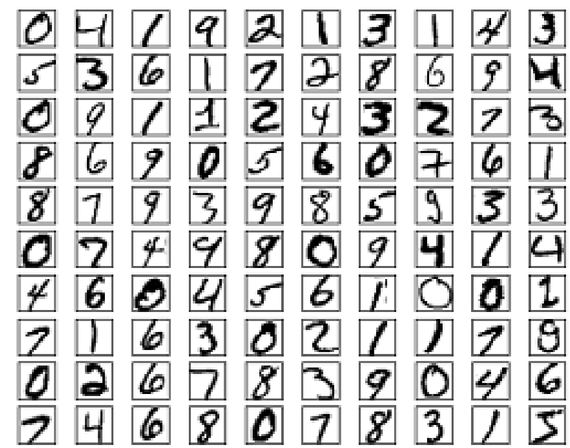
CNNs – Basic Principles

- **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



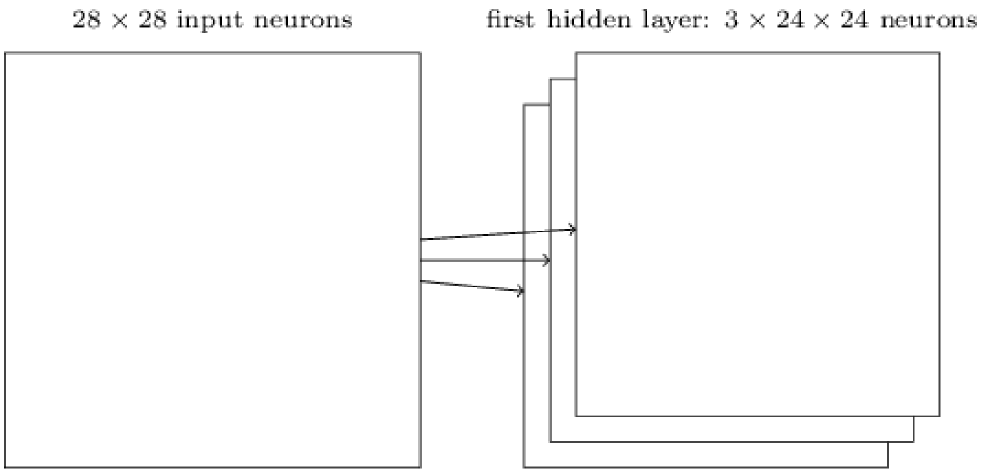
[3] A. Rosebrock



[1] M. Nielsen

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

[1] M. Nielsen

The Convolution Operation

- Assume we are measuring the location of something, e.g. a spaceship, where s(t) is its location at time t.
- To reduce the effect of noise we average several measurements and give more recent measurements more weight than older ones.

$$s(t) = \int x(a)w(t-a)da.$$

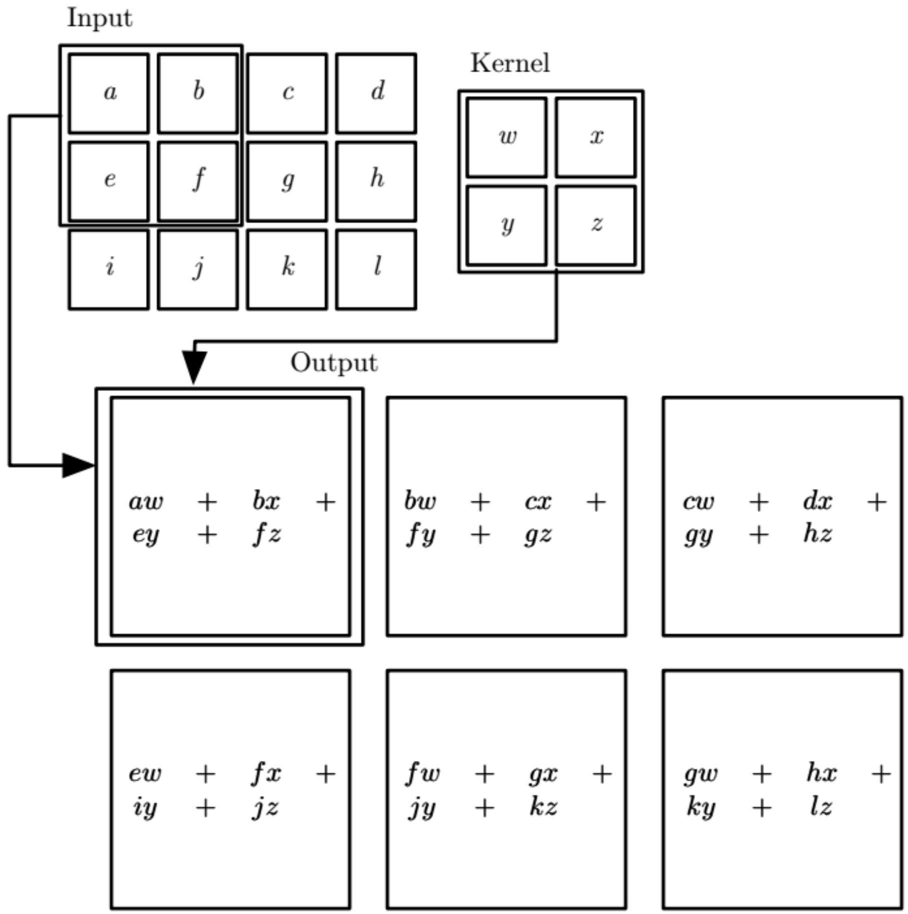
- This operation is called **convolution** and is denoted

$$s(t) = (x * w)(t).$$

- For 2D images (discrete)

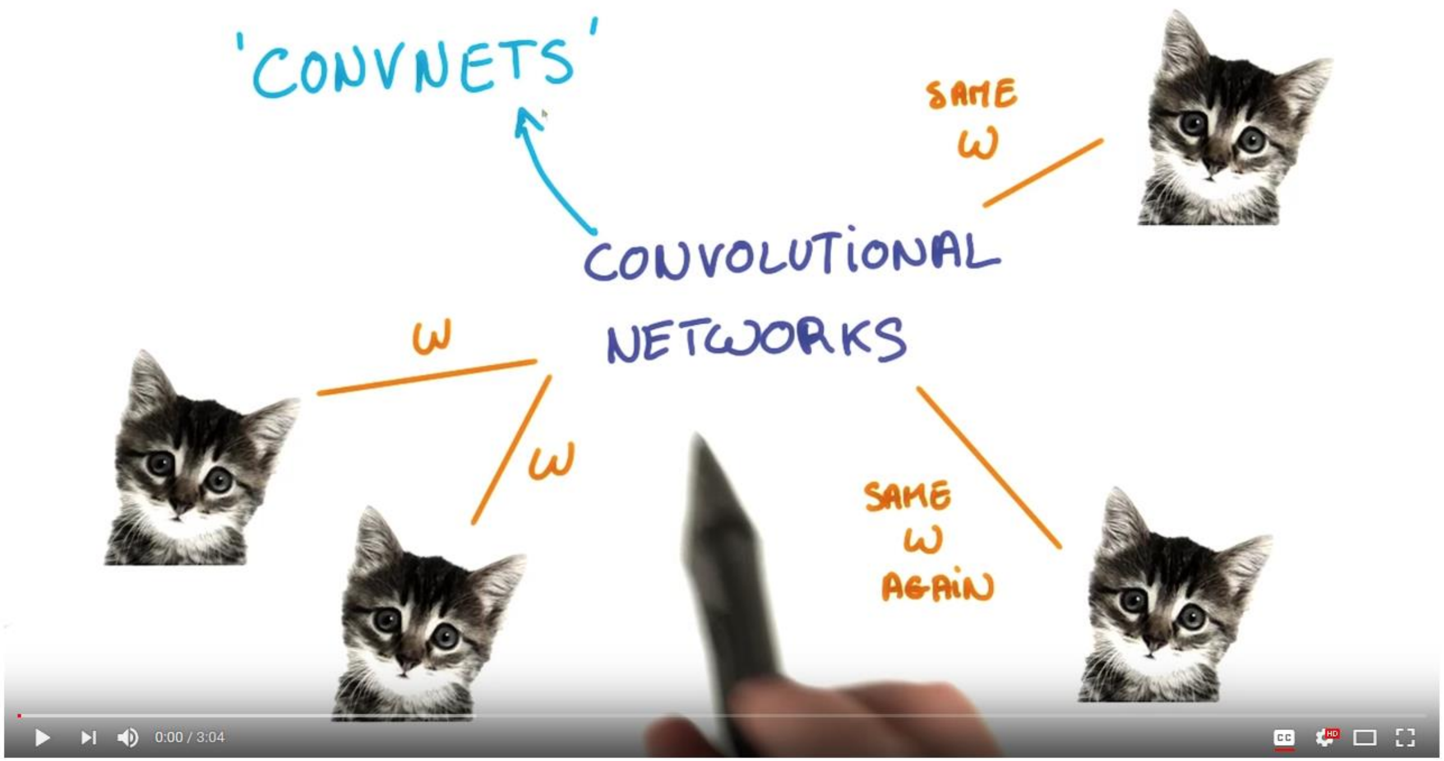
$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n).$$

Valid vs Same Convolution



- 3x4 input matrix processed by a 2x2 kernel with stride=1 that calculates the sum of its content.
- **Valid convolution** does not exceed the input´s boundary
- **Same convolution** adds padding to maintain the input´s dimension for each convolutional layer.

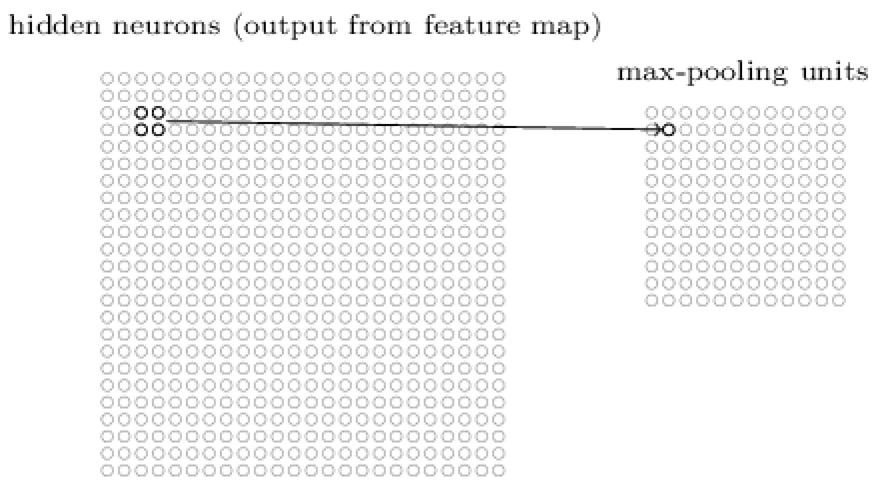
Convolutional Networks

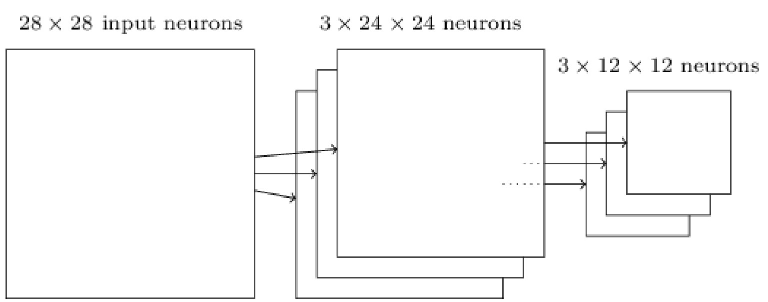


[15] Convolutional Networks

CNNs – Principle of Pooling

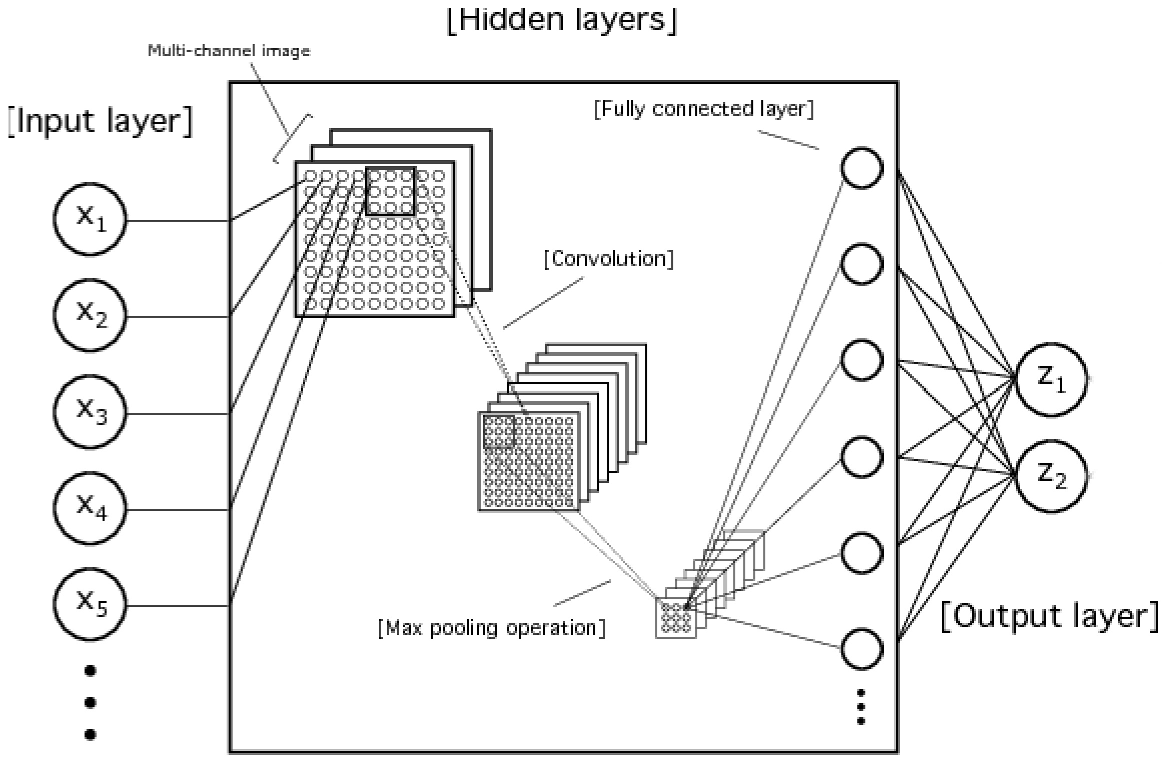
- 'Downsampling' Approach
  - Usually applied directly after convolutional layers
  - Idea is to simplify the information in the output from the convolution
  - Take each feature map output from the convolutional layer and generate a condensed feature map
  - E.g. Pooling with 2 * 2 neurons using 'max-pooling'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





[1] M. Nielsen

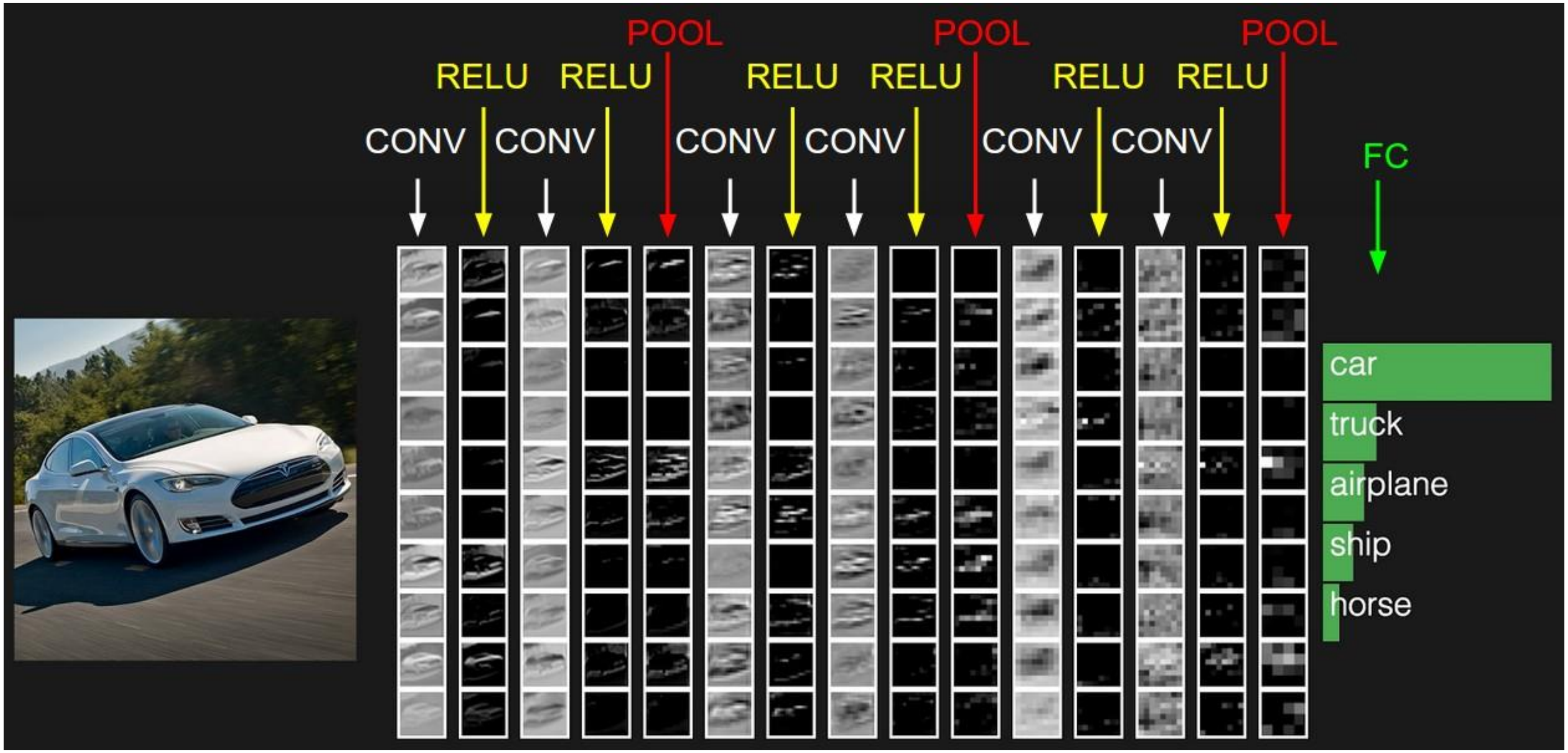
CNNs – Fully Connected Layer



[16] CERN plots

- Sigmoidal or Softmax normalization is a way of reducing the influence of extreme values or outliers in the data without removing them from the dataset

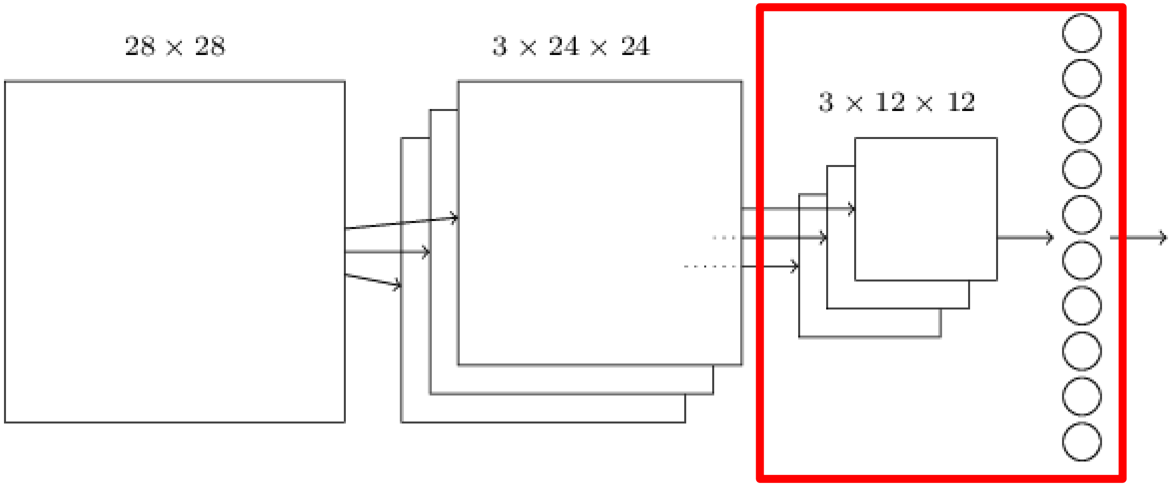
CNNs – Putting it all together



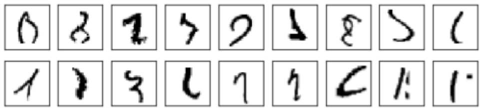
[17] **Convolutional Neural Networks (CNNs / ConvNets)**

CNN – 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



  - Approach works, except for some bad training and test examples



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

[1] M. Nielsen

CNN – Practicals

What do machine learning researcher have to do ?

- Determing the best method for machine learning method for the respective data, i.e. is a CNN applicable or even deep learning ? Computational resources ? Size of the datasets ?
- Pre-processing, i.e. data augmentation
- How many layers, which activation functions, which kernels
- Examine output, inference
- Regularization, e.g. Dropout

Increasing number of Deep Learning Frameworks

- TensorFlow
  - An open-source software library often used
  - Supported device types are CPU and GPU
- Caffe
  - Deep learning framework made with speed and modularity in mind
  - Switch between CPU and GPU by setting a single flag
  - E.g. train on a GPU machine, then deploy to commodity clusters
- Theano
  - Python library for deep learning with integration of NumPY
  - Transparent use of GPGPUs

[4] Tensorflow

[5] Caffe

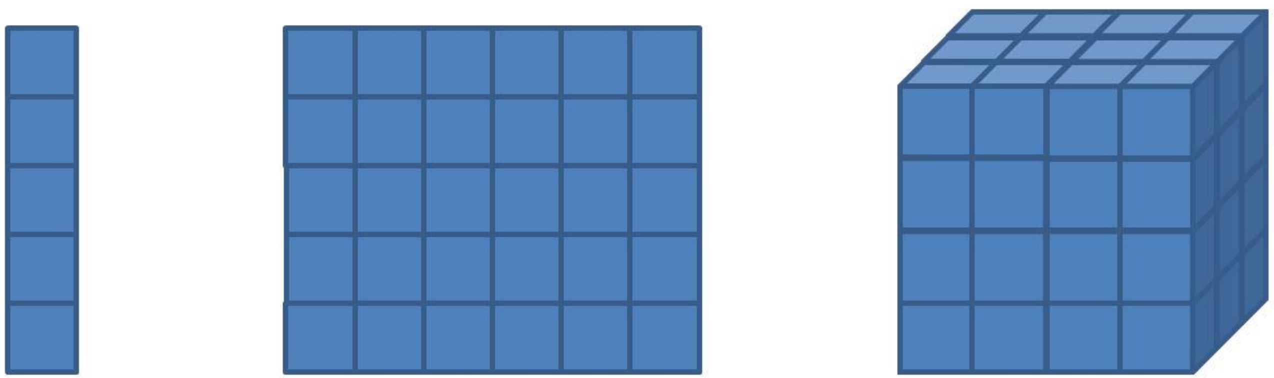
[6] Theano

- **There are a wide variety of deep learning frameworks available that support convolutional neural networks and take advantage of GPGPUs, e.g. TensorFlow, Caffe, Theano**

[7] Deep Learning Framework Comparison

What is a Tensor?

- Meaning
  - Multi-dimensional array used in big data analysis often today
  - Best understood when comparing it with vectors or matrices



(one dimensional tensor) (vector of dimension [5])

(two dimensional tensor) (matrix of dimensions [5,6])

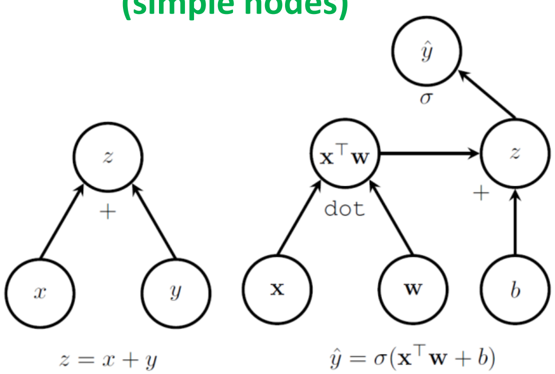
(three dimensional tensor) (tensor of dimension [4,4,3])

[10] Big Data Tips, What is a Tensor?

Tensorflow Computational Graph

- Keras as a High-Level Framework (on top of Tensorflow)
  - Abstracts from the computational graph and focus on layers
- Machine learning algorithms as computational graph
  - Sometimes also called ‘dataflow graph‘ to emphasize data elements
  - Edges represent data (i.e. often tensors) flowing between nodes
  - Vertices / nodes are operations of various types (i.e. combination or transformation of data flowing through the graph)

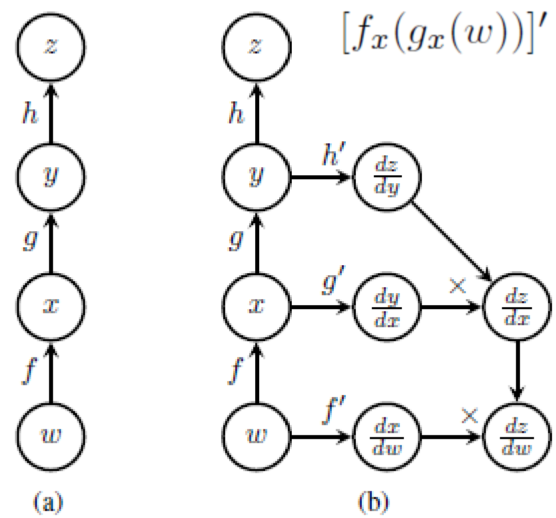
(simple nodes)



[8] A Tour of Tensorflow

(adds gradient node for each operation that takes the gradient of the previous link – outer functions – and multiplies with its own gradient)

$$[f_x(g_x(w))]' = f'_x(g_x(w)) \cdot g'_x(w)$$



(backpropagation algorithm traverses Tensorflow graph in reverse to compute this chain rule)

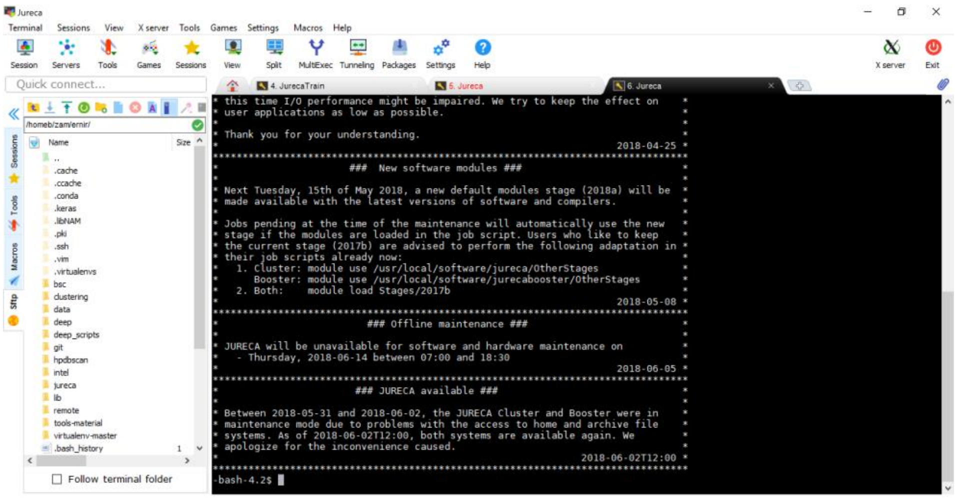
[4] Tensorflow

Exercises – MNIST Dataset – CNN Model Example

SSH Login

- Open https://goo.gl/tTzach
  password is **JSC_dl_2018**

- Log into JURECA using your trainXXX username, ssh-key and password



- For Windows users we recommend MobaXterm

File Copy and Modification

- Copy the job script file
  /homea/hpclab/train001/scripts/submit_train_cnn_mnist.sh
  to your local workspace

- Copy the Python script
  /homea/hpclab/train001/tools/mnist/dl_mnist.py
  to your local workspace

- Modify the Job Script **submit_train_cnn_mnist.sh**, changing the path to the Python script, e.g.
  vi submit_train_cnn_mnist.sh

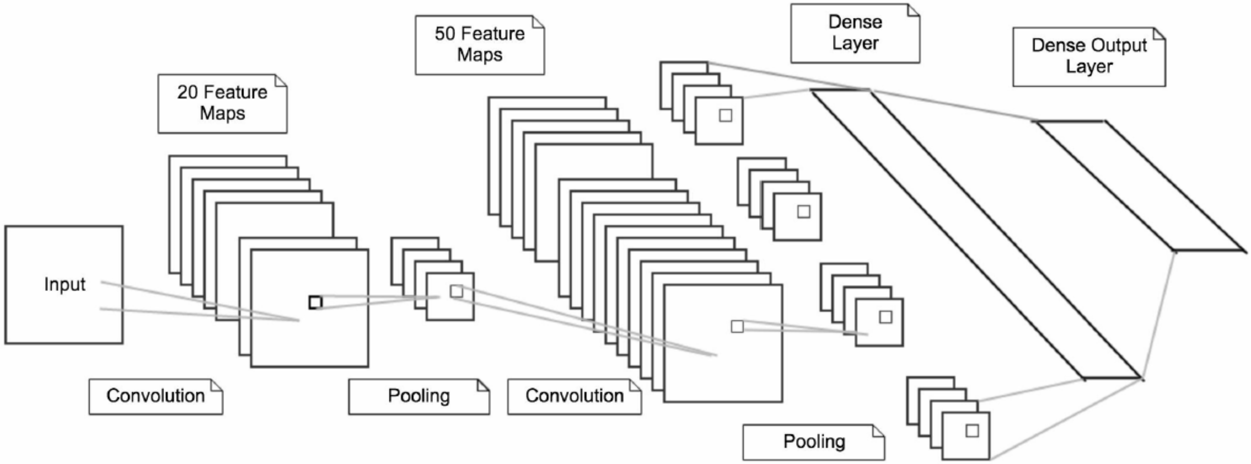
- Run the script by executing
  sbatch submit_train_cnn_mnist.sh

The Job Script

```
#!/bin/bash -x
#SBATCH--nodes=1
#SBATCH--ntasks=1
#SBATCH--output=mnist_out.%j
#SBATCH--error=mnist_err.%j
#SBATCH--time=01:00:00
#SBATCH--mail-user=g.cavallaro@fz-juelich.de
#SBATCH--mail-type=ALL
#SBATCH--job-name=train_mnist
#SBATCH--partition=gpus
#SBATCH--gres=gpu:1
#SBATCH--reservation=deep_learning

### location executable
MNIST=/homea/hpclab/train001/tools/mnist/dl_mnist.py

module restore dl_tutorial

### submit
python $MNIST
```

MNIST Dataset – CNN Python Script

```python
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers.core import Dense, Activation, Flatten
from keras.utils import np_utils
from keras import backend as K
from keras.layers.convolutional import Convolution2D, MaxPooling2D
from keras.optimizers import SGD, RMSprop, Adam

# model
class CNN:
  @staticmethod
  def build(input_shape, classes):
    model = Sequential()
    model.add(Convolution2D(20, kernel_size=5, padding="same", input_shape=input_shape))
    model.add(Activation("relu"))
    model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))
    model.add(Convolution2D(50, kernel_size=5, border_mode="same"))
    model.add(Activation("relu"))
    model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))
    model.add(Flatten())
    model.add(Dense(500))
    model.add(Activation("relu"))
    model.add(Dense(classes))
    model.add(Activation("softmax"))
    return model
```



[9] A. Gulli et al.

MNIST Dataset – CNN Python Script

```python
# parameters
NB_CLASSES = 10
NB_EPOCH = 20
BATCH_SIZE = 128
VERBOSE = 1
OPTIMIZER = 'Adam'
VALIDATION_SPLIT = 0.2
IMG_ROWS, IMG_COLS = 28, 28
INPUT_SHAPE = (1, IMG_ROWS, IMG_COLS)

# dataset 28 x 28 pixels
(X_train, y_train), (X_test, y_test) = mnist.load_data()
K.set_image_dim_ordering("th")
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')

# normalization
X_train /= 255
X_test /= 255

# input convnet
X_train = X_train[:, np.newaxis, :, :]
X_test = X_test[:, np.newaxis, :, :]

# data output
print(X_train.shape[0], 'train samples')
print(X_test.shape[0], 'test samples')

# convert vectors to binary matrices of classes
Y_train = np_utils.to_categorical(y_train, NB_CLASSES)
Y_test = np_utils.to_categorical(y_test, NB_CLASSES)

# Simple CNN model
model = CNN.build(input_shape=INPUT_SHAPE, classes=NB_CLASSES)

# Compilation
model.compile(loss='categorical_crossentropy', optimizer=OPTIMIZER, metrics=['accuracy'])

# Fit the model
history = model.fit(X_train, Y_train, batch_size=BATCH_SIZE, epochs=NB_EPOCH, verbose=VERBOSE, validation_split=VALIDATION_SPLIT)

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

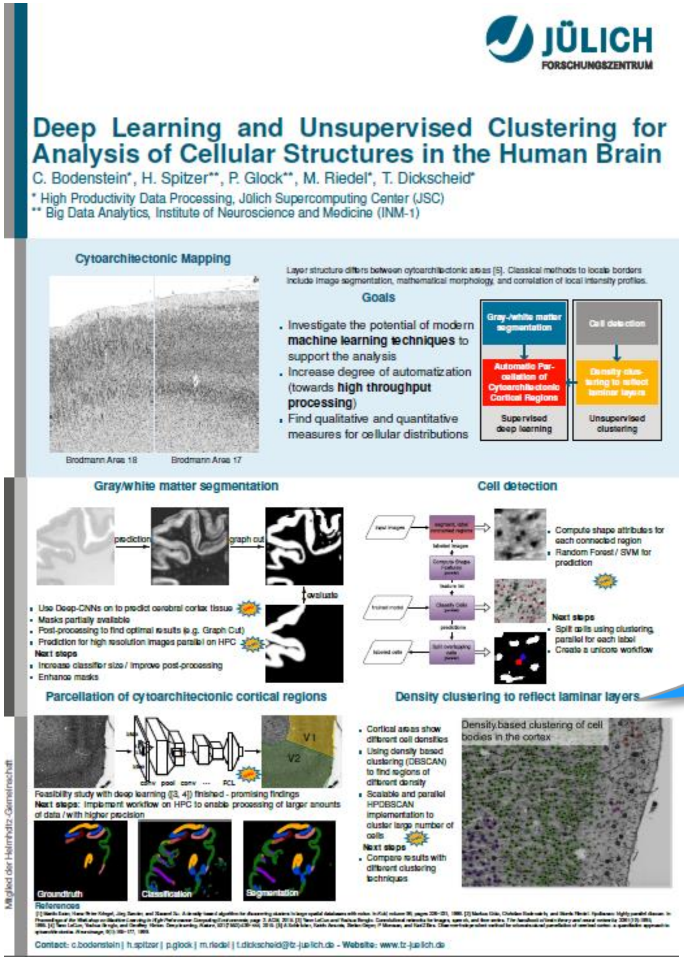
- **OPTIMIZER: Adam - advanced optimization technique that includes the concept of a momentum (a certain velocity component) in addition to the acceleration component of Stochastic Gradient Descent (SGD)**
- **Adam computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients**
- **Adam enables faster convergence at the cost of more computation and is currently recommended as the default algorithm to use (or SGD + Nesterov Momentum)**

[12] D. Kingma et al., 'Adam: A Method for Stochastic Optimization'

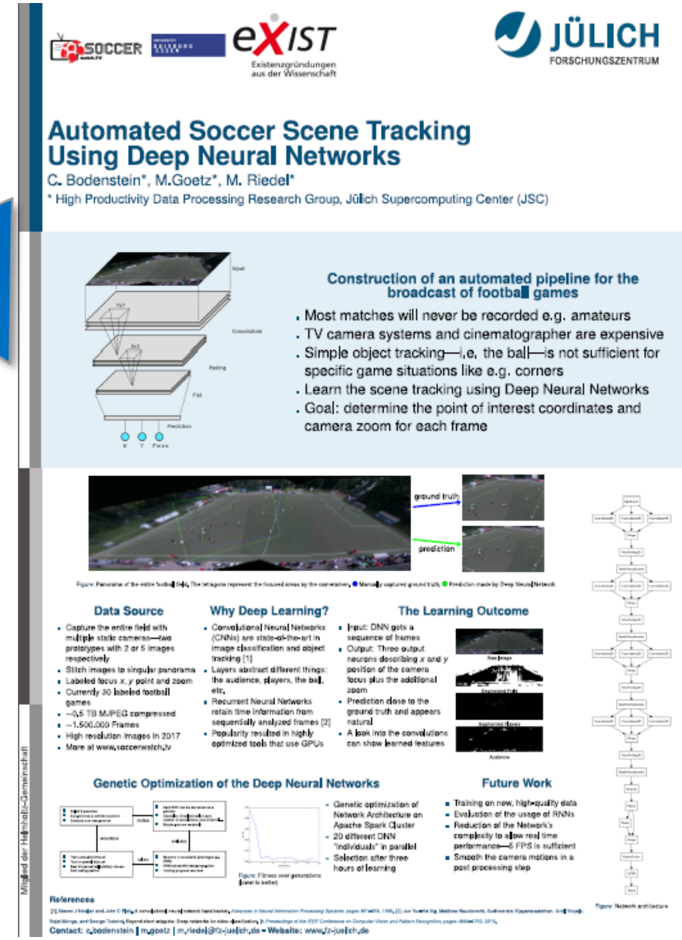
MNIST Dataset – CNN Model – Output

Epoch 19/20
48000/48000
[==============================] - 31s
641us/step - loss: 0.0021 - acc: 0.9992 -
val_loss: 0.0436 - val_acc: 0.9928
Epoch 20/20
48000/48000
[==============================] - 31s
636us/step - loss: 0.0061 - acc: 0.9981 -
val_loss: 0.0397 - val_acc: 0.9917
10000/10000
[==============================] - 3s
262us/step
('Test score: ', 0.03320675646913296)
('Test accuracy: ', 0.9927)

Advanced Application Examples & Opportunities

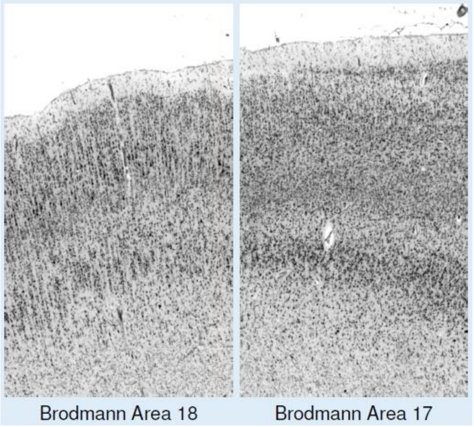


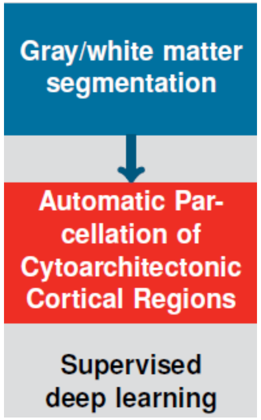
Enabling new Start-ups

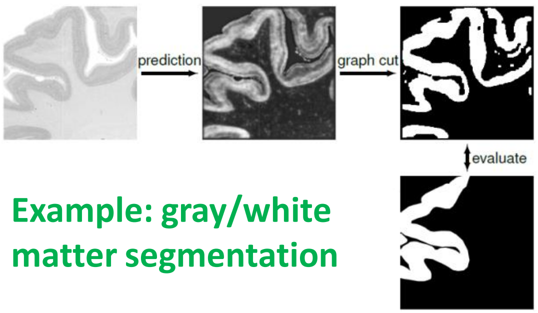


CNN – Neuroscience Application

- Goal: Cytoarchitectonic Mapping
  - Layer structure differs between cytoarchitectonic areas
  - Classical methods to locate borders consists of much manual work: e.g. image segmentation, mathematical morphology, etc.
  - Deep Learning: Automate the process of learning ‘border features‘ by providing large quantities of labelled image data
  - However: the structure setup of the deep learning network still requires manual setup (e.g. how many hidden layers, etc.)

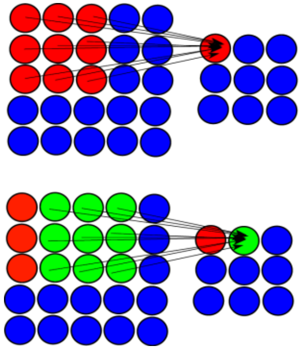


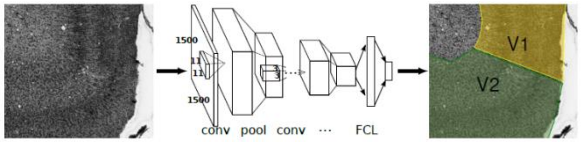




Example: gray/white matter segmentation

Use Convolution Neural Networks: arbitrary dimension, move ‘filter‘ kernel over input space, take local space into account, much cheaper, less parameters than fully connected (e.g. ANNs)

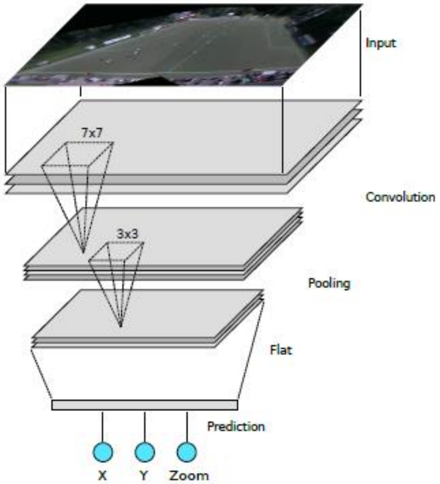


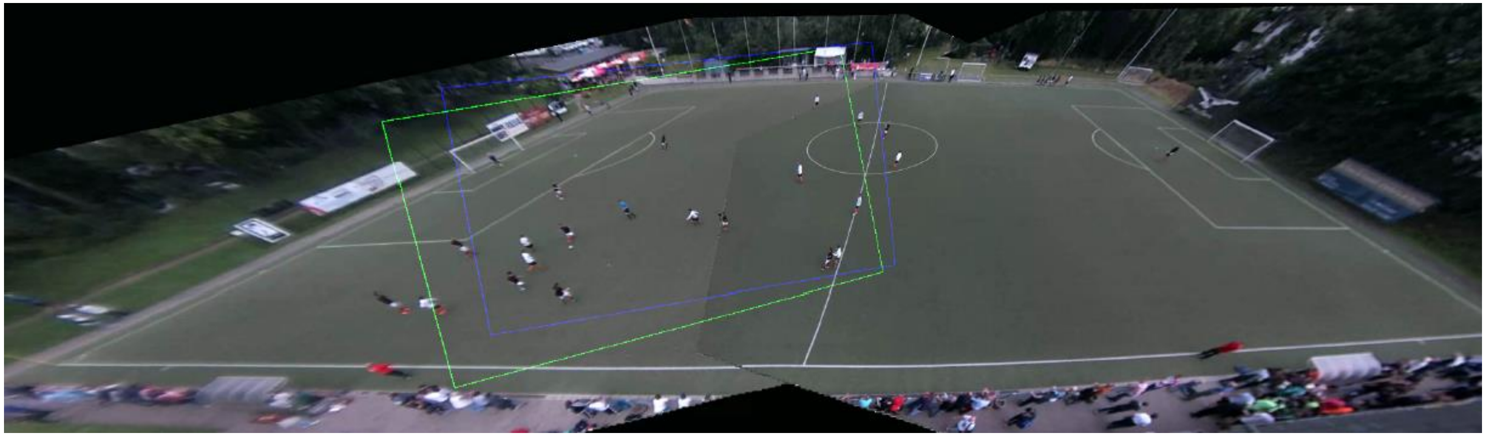


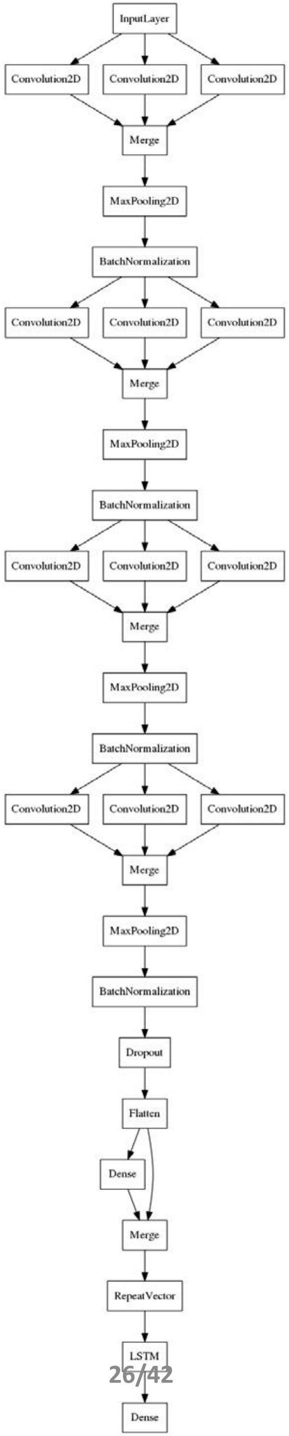
Example: Parcellation of cytoarchitectonic cortical regions

CNN – Soccerwatch.tv Application

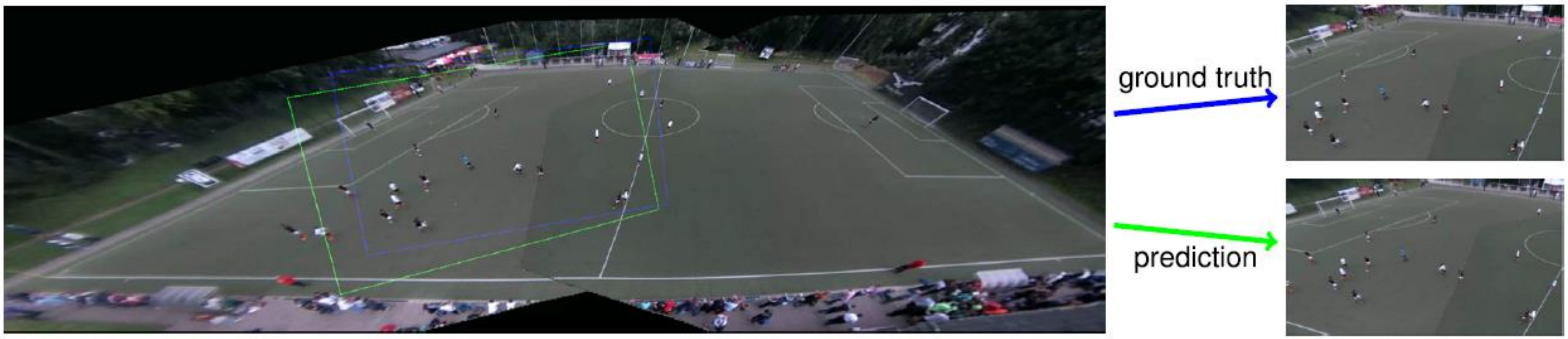
- Goal: Automatic zoom w/o camera man
  - Besides upper leagues: 80k matches/week
  - Recording too expensive (amateurs)
  - Camera man needed
  - Soccerwatch.tv provides panorama
  - Approach: Find X,Y center and zoom on panorama using Deep Learning



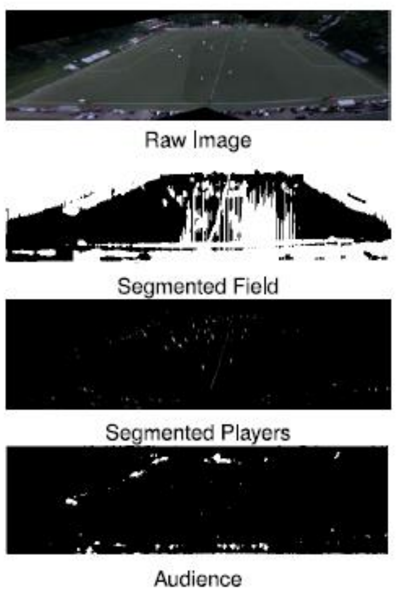




CNN – Soccerwatch.tv Application – Results



(Look into convolutions shows learned features)



UNIVERSITÄT DUISBURG ESSEN

eXIST Existenzgründungen aus der Wissenschaft

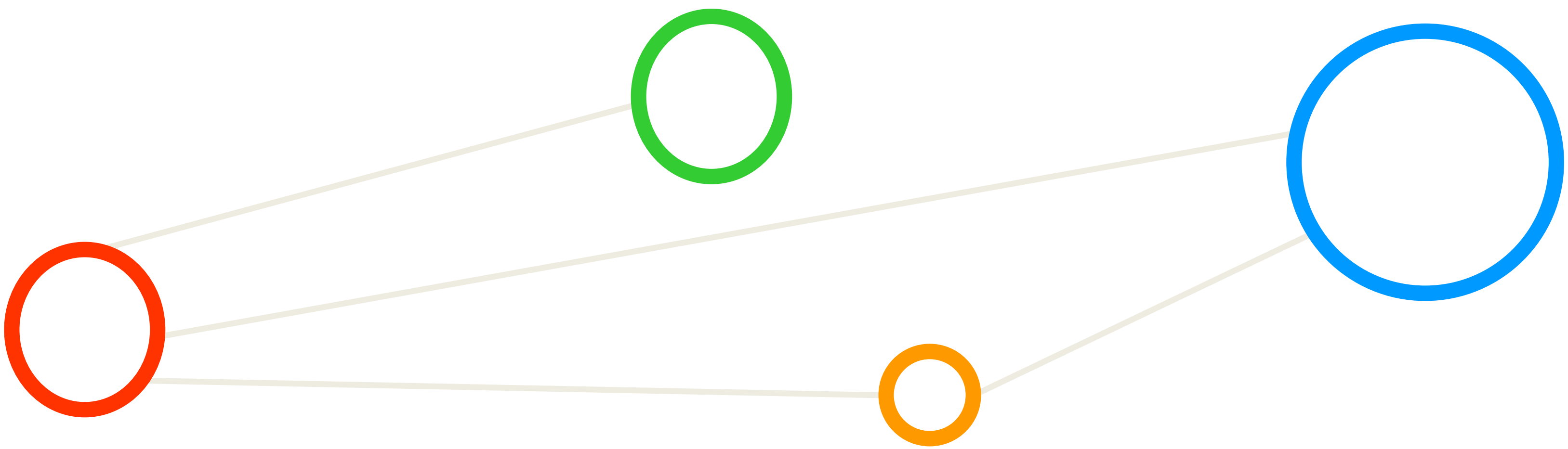
[11] Soccerwatch.tv

[Video] CNN Application in Autonomous Driving

[14] YouTube Video, Speed Sign Recognition

Exercises – MNIST Dataset – CNN Model Check

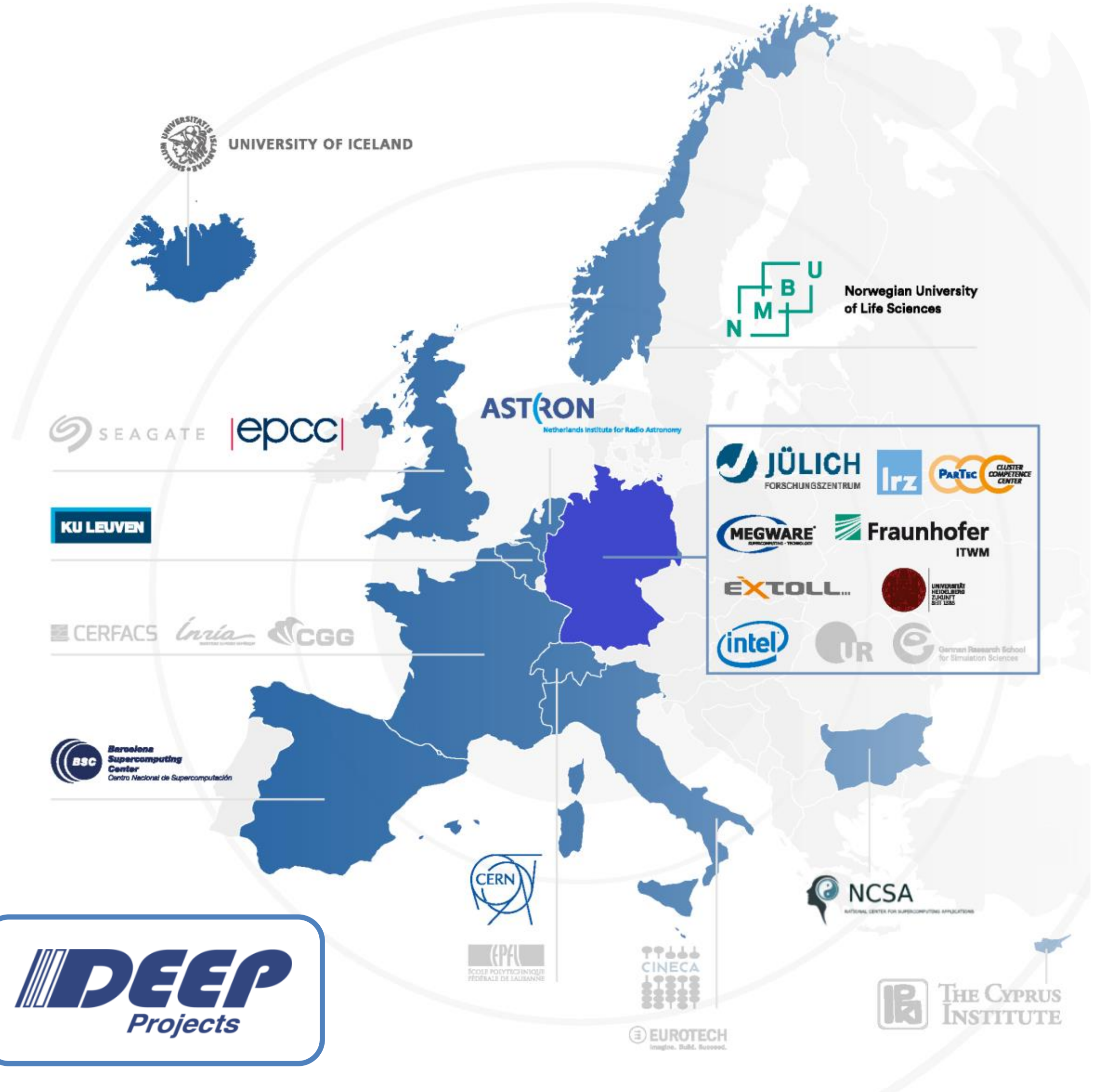
Deep Learning Applications in DEEP-EST



DEEP Projects & Partners

- DEEP
  - Dynamic Exascale Entry Platform
- 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 – Jun 2020

[6] DEEP-EST EU Project



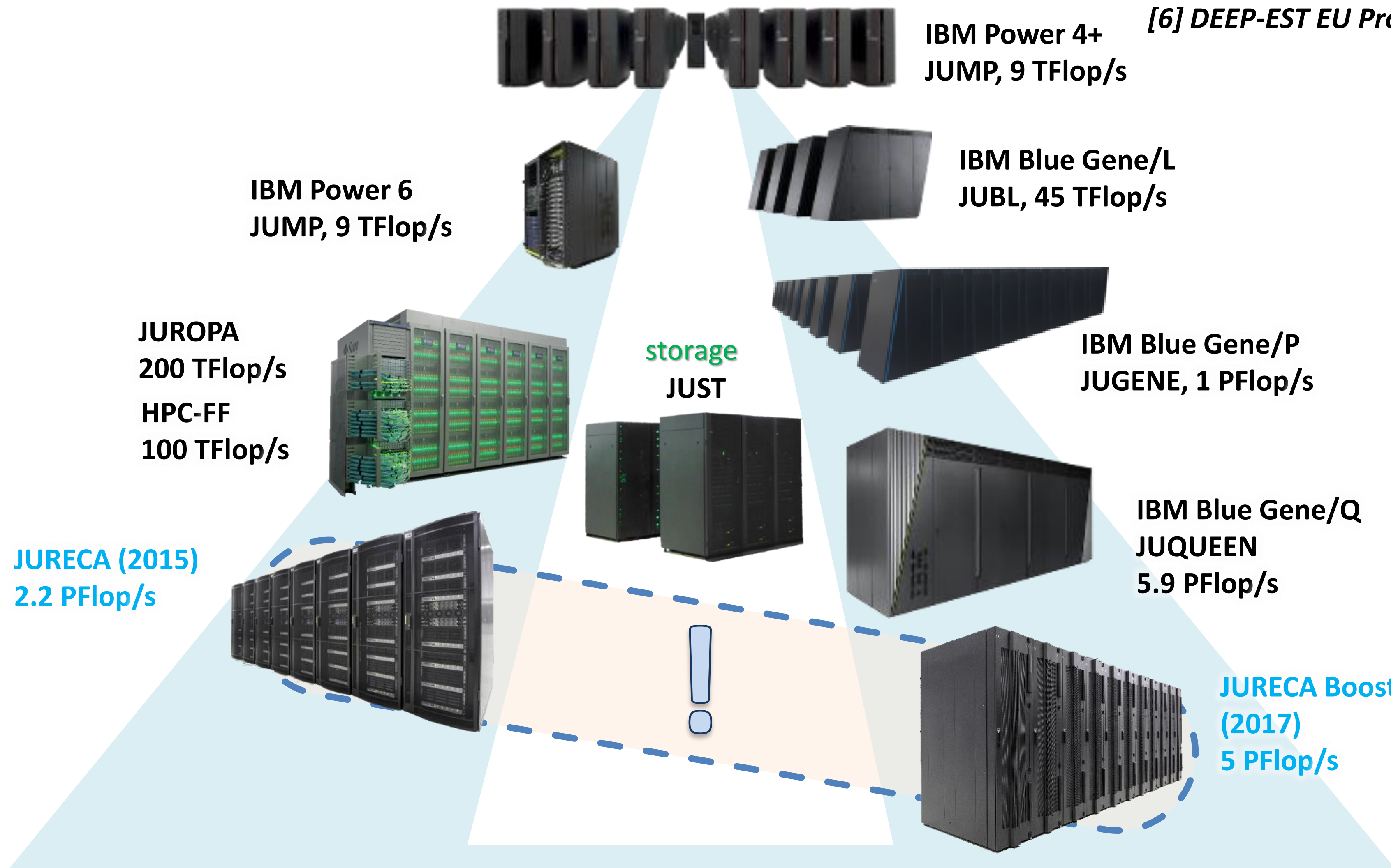
Deep Projects – Application Co-Design → Heterogenity

DEEP Projects

[6] DEEP-EST EU Project

Modular Supercomputing @ JSC

[6] DEEP-EST EU Project



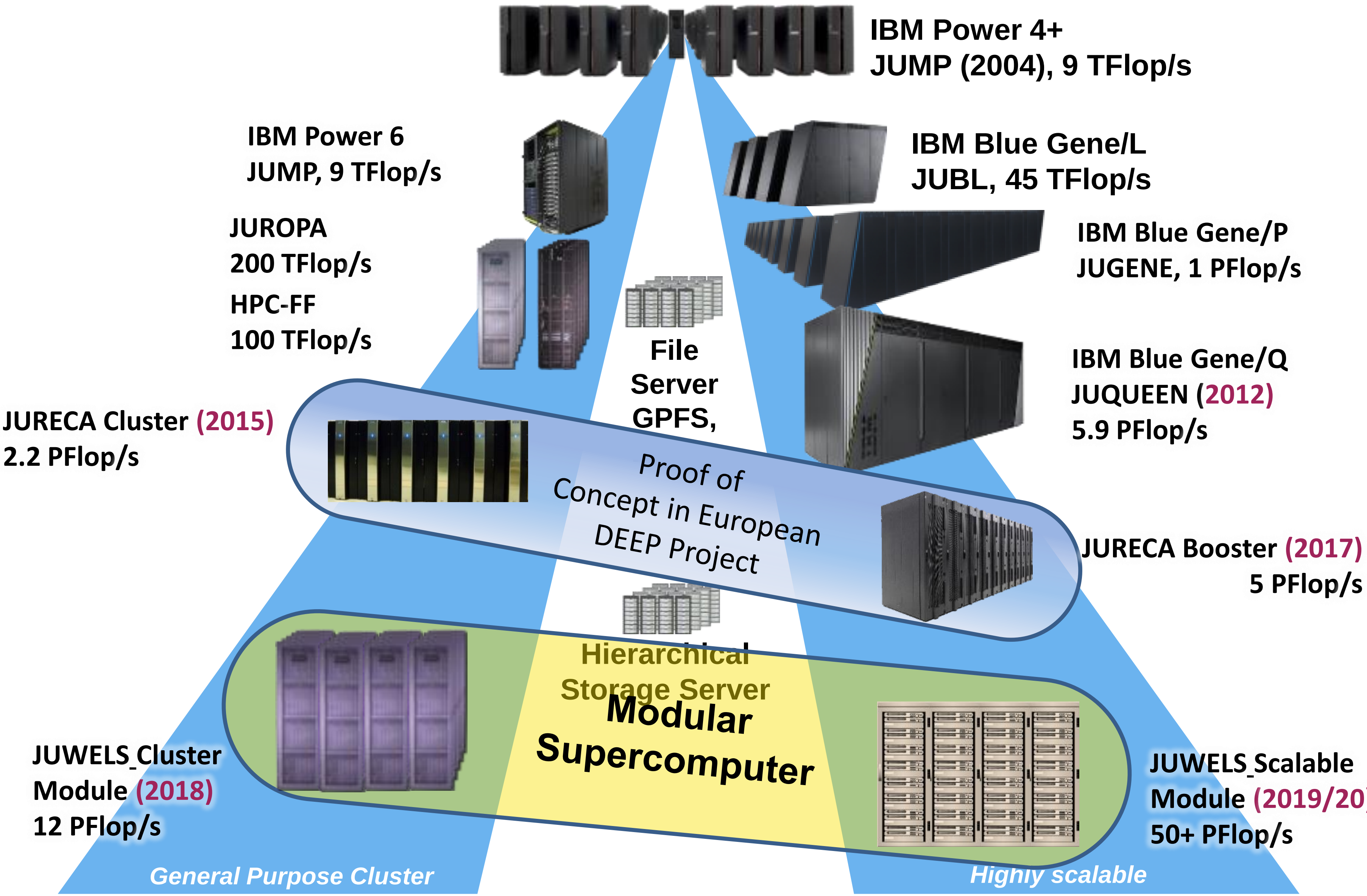
JURECA HPC System

[6] DEEP-EST EU Project

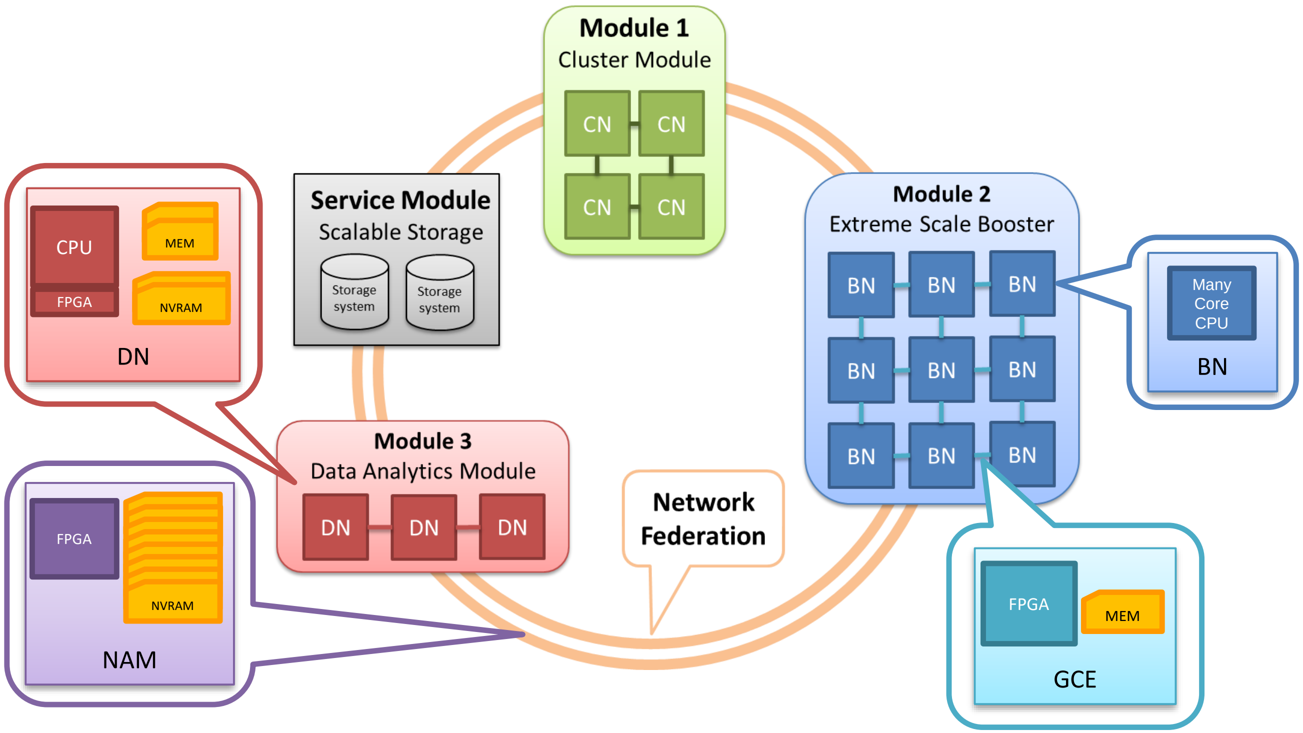
JURECA

- T-Platforms V210 blade server solution
  - Dual-socket Intel Xeon Haswell CPUs
- Mellanox InfiniBand EDR network
- Peak: 1.8 PF (CPUs) + 0.4 PF (GPUs)
- 281 TiB main memory
- 100 GBps storage bandwidth

- Dell PowerEdge C6320P solution
  - Intel Xeon Phi “Knights Landing” 7250-F
- Intel OPA network
- Peak: 5 PF
- 157 TiB main memory + 26 TiB MCDRAM
- 200 GBps storage BW


IBM Power 4+
JUMP (2004), 9 TFlop/s
IBM Power 6
JUMP, 9 TFlop/s
IBM Blue Gene/L
JUBL, 45 TFlop/s
JUROPA
200 TFlop/s
HPC-FF
100 TFlop/s
IBM Blue Gene/P
JUGENE, 1 PFlop/s
File Server GPFS,
IBM Blue Gene/Q
JUQUEEN (2012)
5.9 PFlop/s
JURECA Cluster (2015)
2.2 PFlop/s
Proof of Concept in European DEEP Project
JURECA Booster (2017)
5 PFlop/s
Hierarchical Storage Server
Modular Supercomputer
JUWELS Cluster Module (2018)
12 PFlop/s
JUWELS Scalable Module (2019/20)
50+ PFlop/s
General Purpose Cluster
Highly scalable

DEEP-EST Modular Supercomputing Architecture

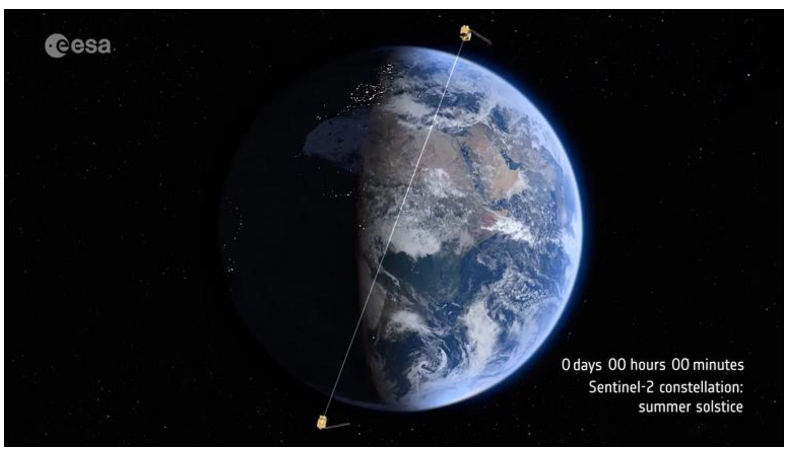


DNNs – Convolutional Neural Networks

- Goal is to leverage the DEEP-EST Modular Supercomputing Architecture (MSA) to enhance machine learning workflows
- For Deep Learning the key is to use **Transfer Learning**, i.e. train new models based on other pretrained models. Multiple models will be trained in parallel and put in a queue. Inference will be carried out simultaneously in another DEEP-EST MSA module by processing multiple trained models in parallel from the queue. Finally, another component sorts/discards models based on the results.

Big Data – The Sentinel 2 Mission

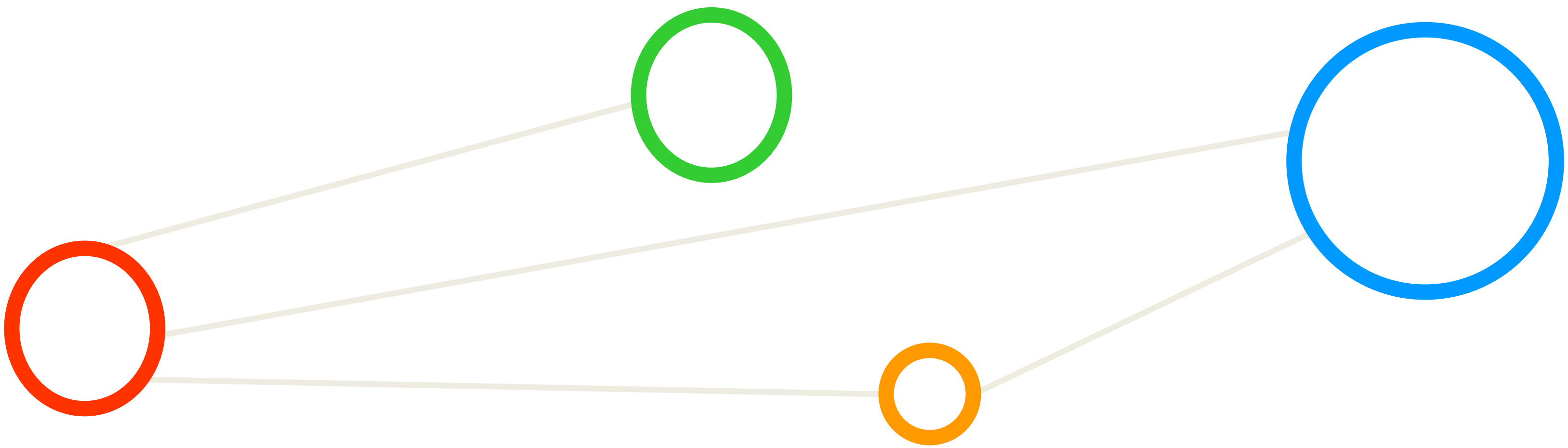
- Twin polar-orbiting satellites, phased at 180° to each other with a temporal resolution of 5 days at the equator in cloud free conditions.
- Provides images for agriculture, forests, land-use change and land-cover change, mapping biophysical variables, and monitoring of coastal and inland waters.
- Data is unlabelled but can be merged with labelled data from other sources, e.g. the Corine programme, an inventory on land cover in 44 classes.



~23 TB data stored per day

http://www.copernicus.eu/

Lecture Bibliography



Lecture Bibliography (1)

- [1] M. Nielsen, ‘Neural Networks and Deep Learning‘,
  Online: http://neuralnetworksanddeeplearning.com/
- [2] Ugent Tier-2 Clusters,
  Online: https://www.vscentrum.be/infrastructure/hardware/hardware-ugent
- [3] 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/
- [4] Tensorflow,
  Online: https://www.tensorflow.org/
- [5] Cafe Deep Learning Framework,
  Online: http://caffe.berkeleyvision.org/
- [6] Theono Deep Learning Framework,
  Online: http://deeplearning.net/software/theano/
- [7] Deep Learning Framework Comparisons,
  Online: http://neuralnetworksanddeeplearning.com/chap6.html
- [8] A Tour of Tensorflow,
  Online: https://arxiv.org/pdf/1610.01178.pdf
- [9] A. Gulli and S. Pal, ‘Deep Learning with Keras‘ Book, ISBN-13 9781787128422, 318 pages,
  Online: https://www.packtpub.com/big-data-and-business-intelligence/deep-learning-keras
- [10] Big Data Tips, ‘What is a Tensor?‘,
  Online: http://www.big-data.tips/what-is-a-tensor
- [11] Soccerwatch.tv,
  Online: http://www.soccerwatch.tv

Lecture Bibliography (2)

- [12] D. Kingma and Jimmy Ba, ‘Adam: A Method for Stochastic Optimization‘,
  Online: https://arxiv.org/abs/1412.6980
- [13] YouTube Video, ‘Simple explanation of how backpropagation works in deep learning libraries‘,
  Online: https://www.youtube.com/watch?v=zhKWBye_RgE
- [14] YouTube Video, ‘Speed Sign Recognition by Convolutional Neural Networks‘,
  Online: https://www.youtube.com/watch?v=kkha3sPoU70
- [15] YouTube Video, ‘Convolutional Networks’,
  Online: https://www.youtube.com/watch?v=jajksuQW4mc
- [16] Cern Plots
  Online: http://cds.cern.ch/record/2217394/plots
- [17] Convolutional Neural Networks (CNNs / ConvNets)
  Online: http://cs231n.github.io/convolutional-networks/


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