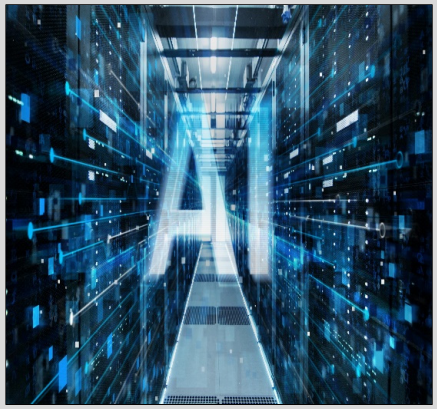


Parallel and Scalable Machine Learning

Introduction to Machine Learning Models

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 5

Introduction to Statistical Learning Theory

February 18, 2020
Juelich Supercomputing Centre, Germany

PRACE PARTNERSHIP FOR ADVANCED COMPUTING IN EUROPE TRAINING PORTAL

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

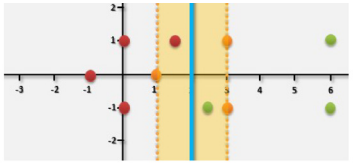
JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

DEEP Projects

HELMHOLTZ AI | ARTIFICIAL INTELLIGENCE COOPERATION UNIT

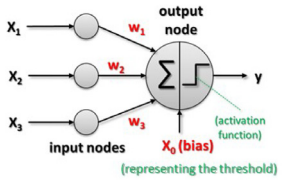
Review of Lecture 4 – Artificial Neural Networks (ANNs)

h_1



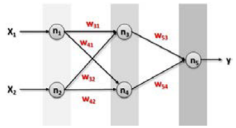
(e.g. support vector machine model)

h_2



(e.g. linear perceptron model)

h_m



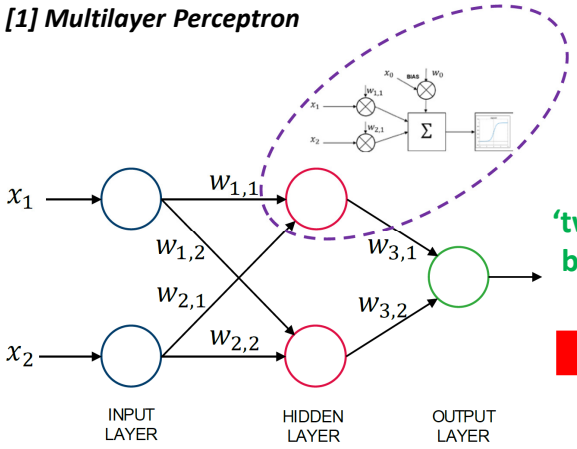
(e.g. artificial neural network model)

introducing hidden layers concept

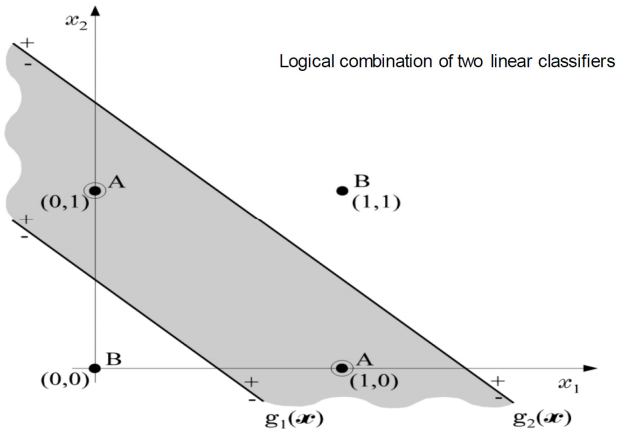
```
import numpy as np
from keras.datasets import mnist
```

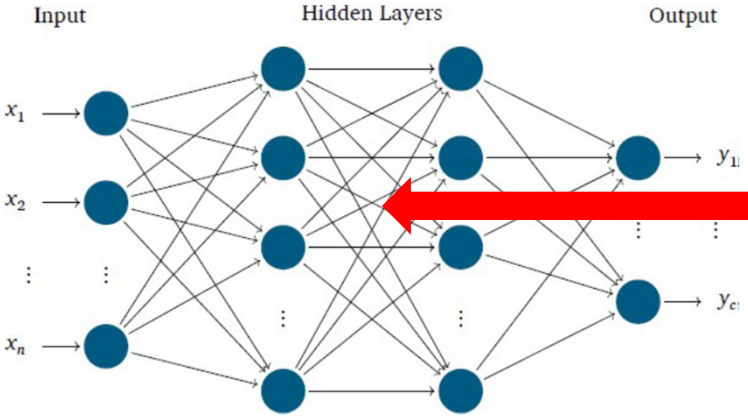
```
# download and shuffled as training and testing set
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

[1] Multilayer Perceptron



enables 'two decision boundaries'

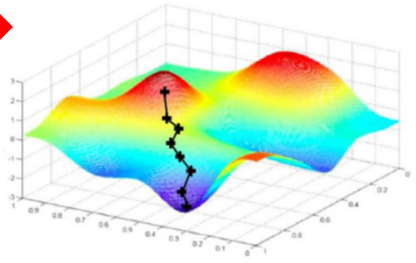




$$W := W - \eta \frac{\partial \mathcal{L}(W)}{\partial W}$$

[2] MIT Course

How to set the **learning rate**?



Outline of the Training Course

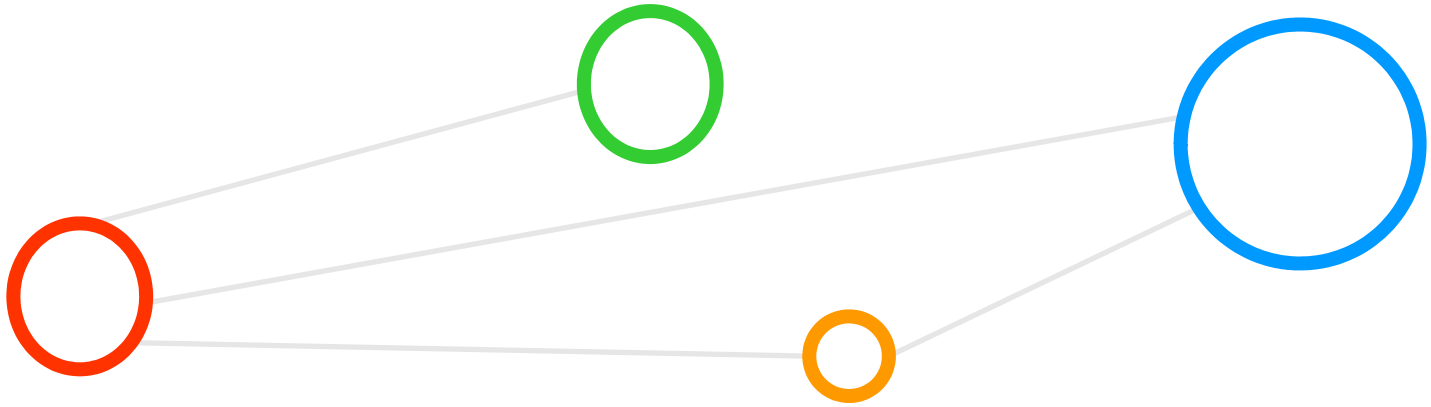
1. Parallel & Scalable Machine Learning driven by HPC
2. Introduction to Machine Learning Fundamentals
3. Supervised Learning with a Simple Learning Model
4. Artificial Neural Networks (ANNs)
5. Introduction to Statistical Learning Theory
6. Validation and Regularization
7. Pattern Recognition Systems
8. Parallel and Distributed Training of ANN
9. Supervised Learning with Deep Learning
10. Unsupervised Learning – Clustering
11. Clustering with HPC
12. Introduction to Deep Reinforcement Learning

Outline

- Supervised Learning & Statistical Learning Theory
  - Statistical Learning Theory is Best Understood for Supervised Learning
  - Refining Mathematic Building Blocks for Supervised Learning
  - Feasibility of Learning from Data & Hoeffding Inequality
  - Probably Approximately Correct (PAC) Learning
  - Learning Model Perceptron Examples in Context

- Vapnik – Chervonenkis (VC) Inequality & Dimension
  - Theory of Generalization – Revisited & Reviewed
  - Infinite Learning Models Challenge & Union Bound Solution
  - Vapnik – Chervonenkis (VC) Inequality & Dimension
  - Relationship of Number of Samples & Model Complexity
  - Overfitting as Key Problem in Machine Learning

Supervised Learning & Statistical Learning Theory



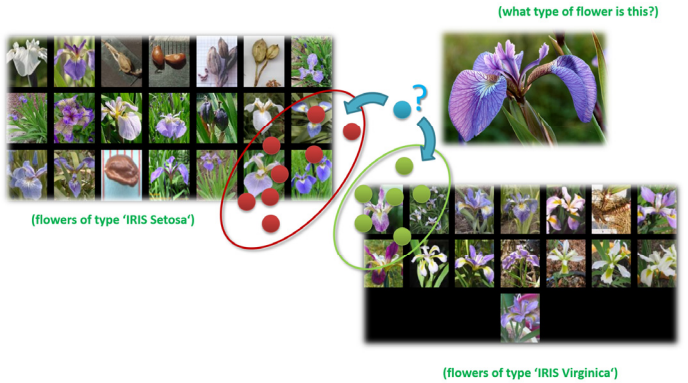
Learning Approaches – What means Learning from data – Revisited

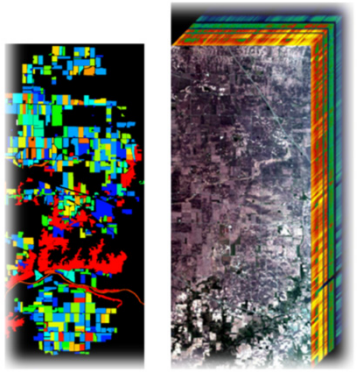
- **The basic meaning of learning is 'to use a set of observations to uncover an underlying process'**
- **The three different learning approaches are supervised, unsupervised, and reinforcement learning**

Supervised Learning

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

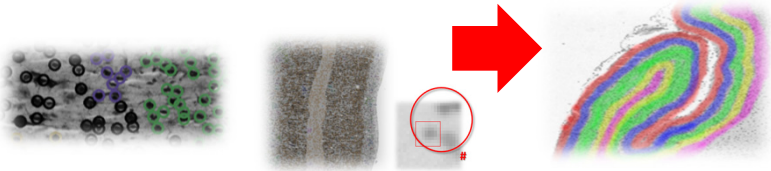
[3] Image sources: Species Iris Group of North America Database, www.signa.org





Unsupervised Learning

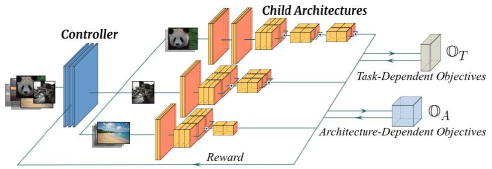
- Often applied before other learning → higher level data representation
- Example: Coin recognition in vending machine based on weight and size



Reinforcement Learning

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

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

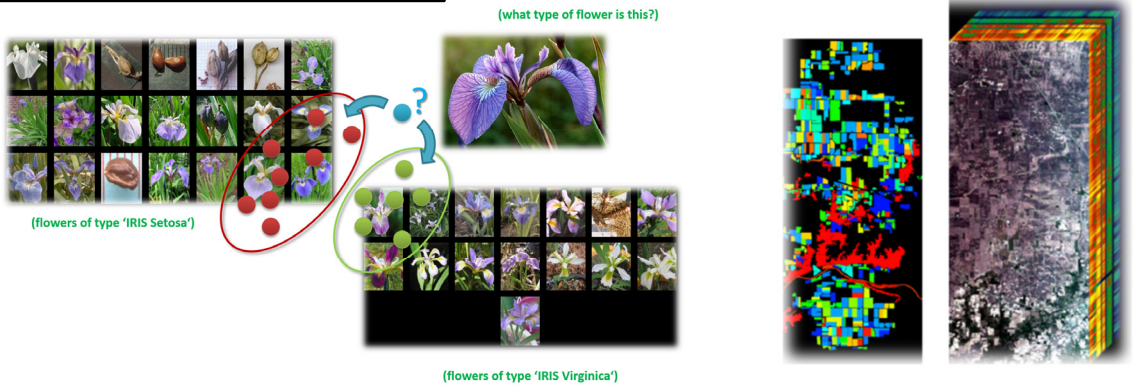


Supervised Learning – Statistical Learning Theory Perspective

- **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**

[3] 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

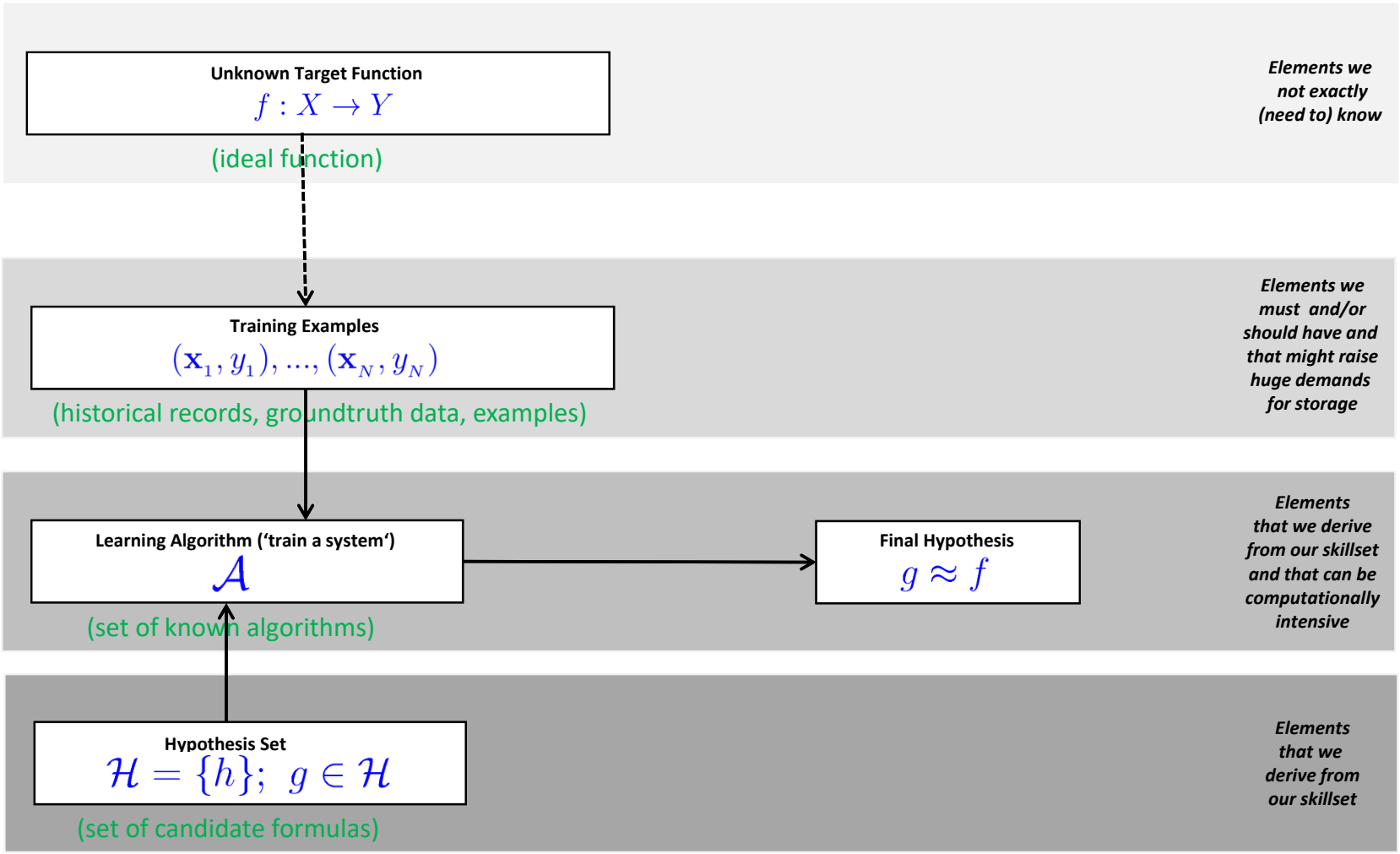


- Statistical Learning Theory Perspective
  - Supervised Learning is best understood
  - Represents a 'framework' in which learning from examples can be studied in a principled way
  - Gives indicators for selected questions: How much data we need for the model we want to use?
  - This lecture just introduces the topic (much more deeper & complex)

Statistical Learning Theory: A Primer

THEODOROS EVGENIOU, MASSIMILIANO PONTIL AND TOMASO POGGIO
Center for Biological and Computational Learning, Artificial Intelligence Laboratory, MIT, Cambridge, MA, USA

[5] Evgeniou T., Pontil M., Poggio T., 'Statistical Learning Theory: A Primer'


Unknown Target Function
$f : X \rightarrow Y$
(ideal function)
Elements we not exactly (need to) know
Training Examples
$(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)$
(historical records, groundtruth data, examples)
Elements we must and/or should have and that might raise huge demands for storage
Learning Algorithm ('train a system')
$\mathcal{A}$
(set of known algorithms)
Final Hypothesis
$g \approx f$
Elements that we derive from our skillset and that can be computationally intensive
Hypothesis Set
$\mathcal{H} = \{h\}; \; g \in \mathcal{H}$
(set of candidate formulas)
Elements that we derive from our skillset

Different Models – Hypothesis Set & Unlimited ‘Degrees of Freedom‘ – Revisited

Hypothesis Set

$$\mathcal{H} = \{h\};\ g \in \mathcal{H}$$

$$\mathcal{H} = \{h_1, \ldots, h_m\};$$

(all candidate functions
derived from models
and their parameters)

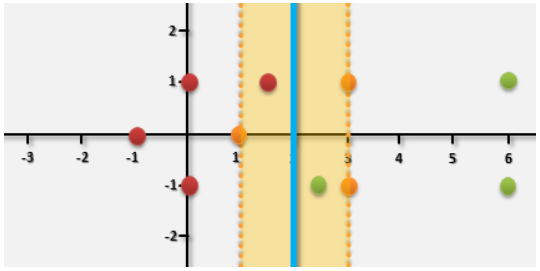
- **Choosing from various model approaches h_1, …, h_m is a different hypothesis**
- **Additionally a change in model parameters of h_1, …, h_m means a different hypothesis too**

‘select one function‘
that best approximates

Final Hypothesis

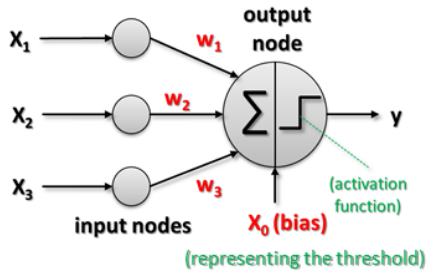
$$g \approx f$$

h_1



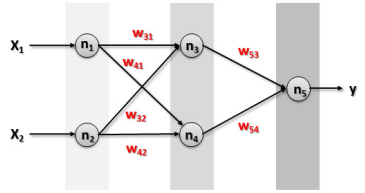
(e.g. support vector machine model)

h_2



(e.g. linear perceptron model)

h_m



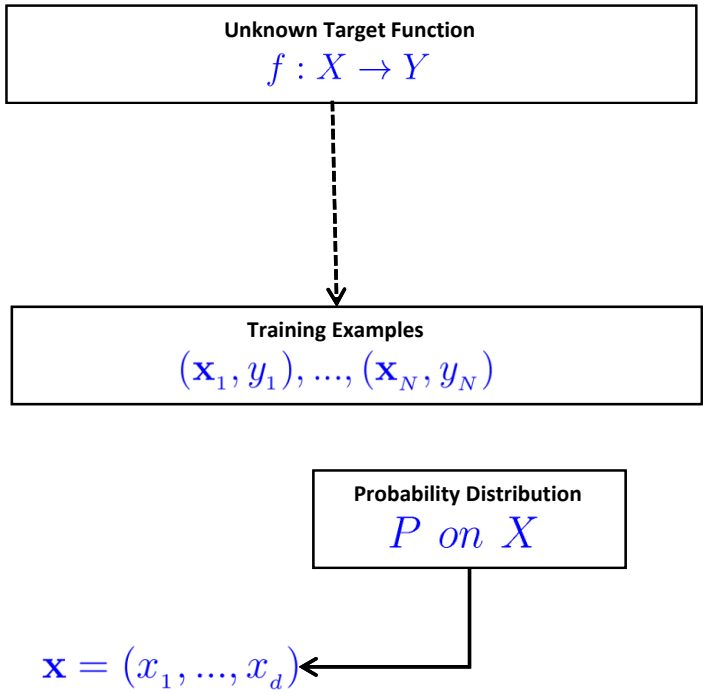
(e.g. artificial neural network model)

Solutions – Train on Testing Dataset & Test on Training Dataset & Increase Epochs

- It seems the number of N samples matter in learning – why?

Feasibility of Learning – Probability Distribution

- Predict output from future input
  (fitting existing data is not enough)
  - In-sample ‘1000 points‘ fit well
  - Possible: Out-of-sample >= ‘1001 point‘ doesn‘t fit very well
  - Learning ‘any target function‘ is not feasible (can be anything)
- Assumptions about ‘future input‘
  - Statement is possible to define about the data outside the in-sample data $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)$
  - All samples (also future ones) are derived from same ‘unknown probability‘ distribution $P \ on \ X$



(which exact probability is not important, but should not be completely random)

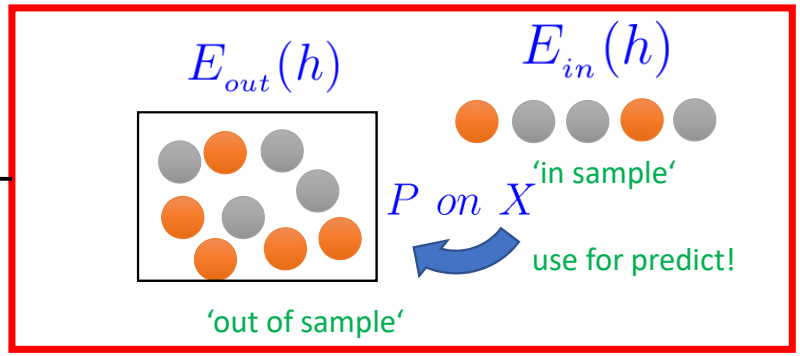
- **Statistical Learning Theory assumes an unknown probability distribution over the input space X**

Feasibility of Learning – In Sample vs. Out of Sample

- Given ‘unknown‘ probability $P \ on \ X$
  - Given large sample N for $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)$
  - There is a probability of ‘picking one point or another‘
  - ‘Error on in sample‘ is known quantity (using labelled data): $E_{in}(h)$
  - ‘Error on out of sample‘ is unknown quantity: $E_{out}(h)$
  - In-sample frequency is likely close to out-of-sample frequency

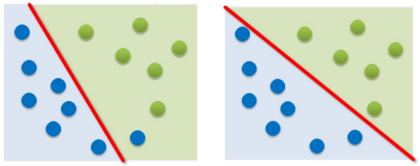
- **Statistical Learning Theory part that enables that learning is feasible in a probabilistic sense (P on X)**

depend on which hypothesis h out of M different ones



$\mathcal{H} = \{h_1, ..., h_m\};$

E_{in} tracks E_{out}



$$E_{in}(h) \approx E_{out}(h)$$

use E_{in}(h) as a proxy thus the other way around in learning

$$E_{out}(h) \approx E_{in}(h)$$

Feasibility of Learning – Union Bound & Factor M

- Assuming no overlaps in hypothesis set
  - Apply very 'poor' mathematical rule 'union bound'
  - (Note the usage of g instead of h, we need to visit all)

Final Hypothesis

$$g \approx f$$

Think if E_{in} deviates from E_{out} with more than tolerance Є it is a 'bad event' in order to apply union bound

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ > \epsilon\ ]\ <= \Pr\ [\ |\ E_{in}(h_1) - E_{out}(h_1)\ |\ > \epsilon$$

$$\text{or}\ \ |\ E_{in}(h_2) - E_{out}(h_2)\ |\ > \epsilon\ \ \ldots$$

$$\text{or}\ \ |\ E_{in}(h_M) - E_{out}(h_M)\ |\ > \epsilon\ \ ]$$

'visiting M different hypothesis'

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ > \epsilon\ ]\ <=\ \sum_{m=1}^{M} \Pr\ [\ |\ E_{in}(h_m) - E_{out}(h_m)\ |\ > \epsilon\ ]$$

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ > \epsilon\ ]\ <=\ \sum_{m=1}^{M} 2e^{-2\epsilon^2 N}$$

fixed quantity for each hypothesis obtained from Hoeffdings Inequality

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ > \epsilon\ ]\ <=\ 2Me^{-2\epsilon^2 N}$$

problematic: if M is too big we loose the link between the in-sample and out-of-sample

- **The union bound means that (for any countable set of m 'events') the probability that at least one of the events happens is not greater that the sum of the probabilities of the m individual 'events'**

Feasibility of Learning – Modified Hoeffding's Inequality – PAC Learning

- Errors in-sample $E_{in}(g)$ track errors out-of-sample $E_{out}(g)$
  - Statement is made being 'Probably Approximately Correct (PAC)'
  - Given M as number of hypothesis of hypothesis set $\mathcal{H}$
  - 'Tolerance parameter' in learning ϵ
  - Mathematically established via 'modified Hoeffdings Inequality':
    (original Hoeffdings Inequality doesn't apply to multiple hypothesis)

'Approximately' 'Probably'

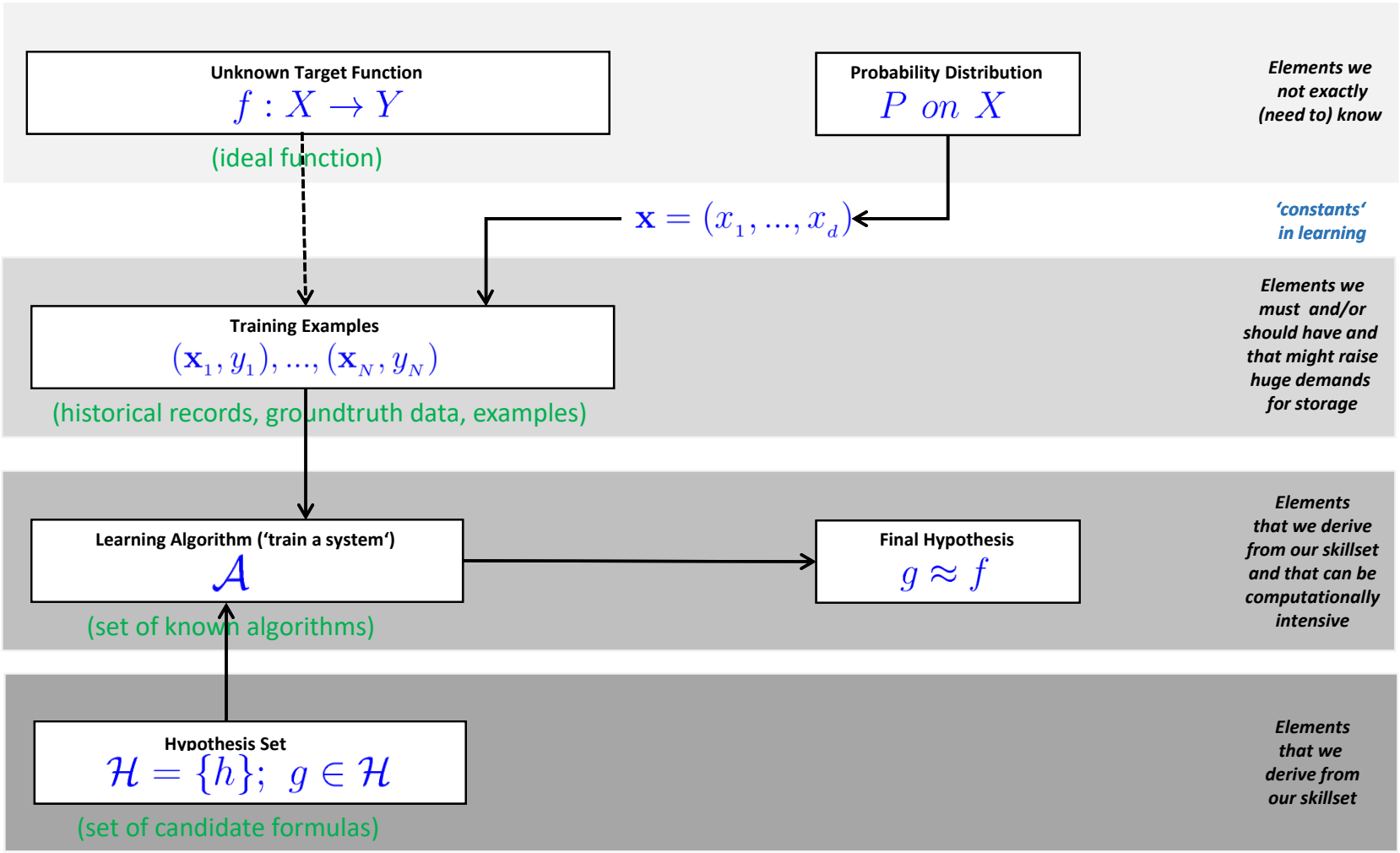
$$\Pr\left[\,|\,E_{in}(g) - E_{out}(g)\,|\,> \epsilon\,\right] <= 2Me^{-2\epsilon^2 N}$$

'Probability that E_{in} deviates from E_{out} by more than the tolerance Є is a small quantity depending on M and N'

[1] Valiant, 'A Theory of the Learnable', 1984

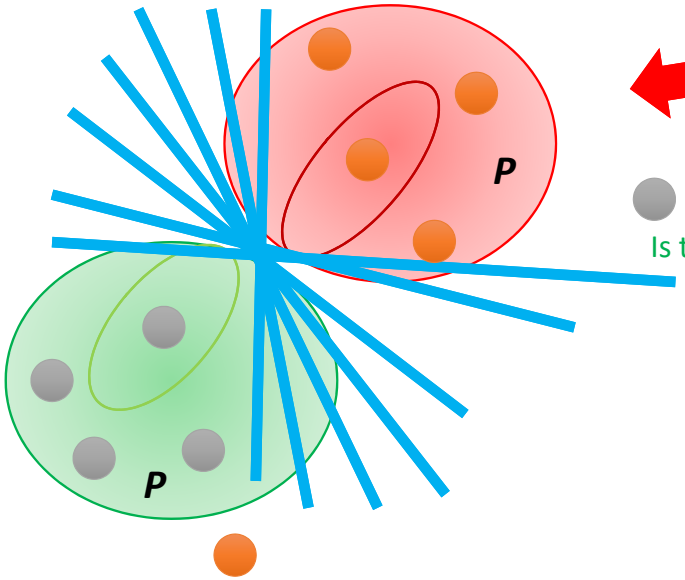
- **Statistical Learning Theory part describing the Probably Approximately Correct (PAC) learning**

- Theoretical 'Big Data' Impact → more N → better learning
  - The more samples N the more reliable will track $E_{in}(g)$ $E_{out}(g)$ well
  - (But: the 'quality of samples' also matter, not only the number of samples)
  - For supervised learning also the 'label' has a major impact in learning (later)


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

Mathematical Building Blocks – Our Linear Example

(infinite M decision boundaries depending on f)



Probability Distribution

$P \ on \ X$

$\mathbf{x} = (x_1, ..., x_d)$

Is this point very likely from the same distribution or just noise?

We assume future points are taken from the same probability distribution as those that we have in our training examples

Training Examples

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

Is this point very likely from the same distribution or just noise?

(we help here with the assumption for the samples)

(we do not solve the M problem here)

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ > \epsilon\ ]\ <=\ 2Me^{-2\epsilon^2 N}$$

(counter example would be for instance a random number generator, impossible to learn this!)

Statistical Learning Theory – Error Measure & Noisy Targets

- Question: How can we learn a function from (noisy) data?
- 'Error measures' to quantify our progress, the goal is: $h \approx f$
  - Often user-defined, if not often 'squared error':

$$e(h(\mathbf{x}), f(\mathbf{x})) = (h(\mathbf{x}) - f(\mathbf{x}))^2$$

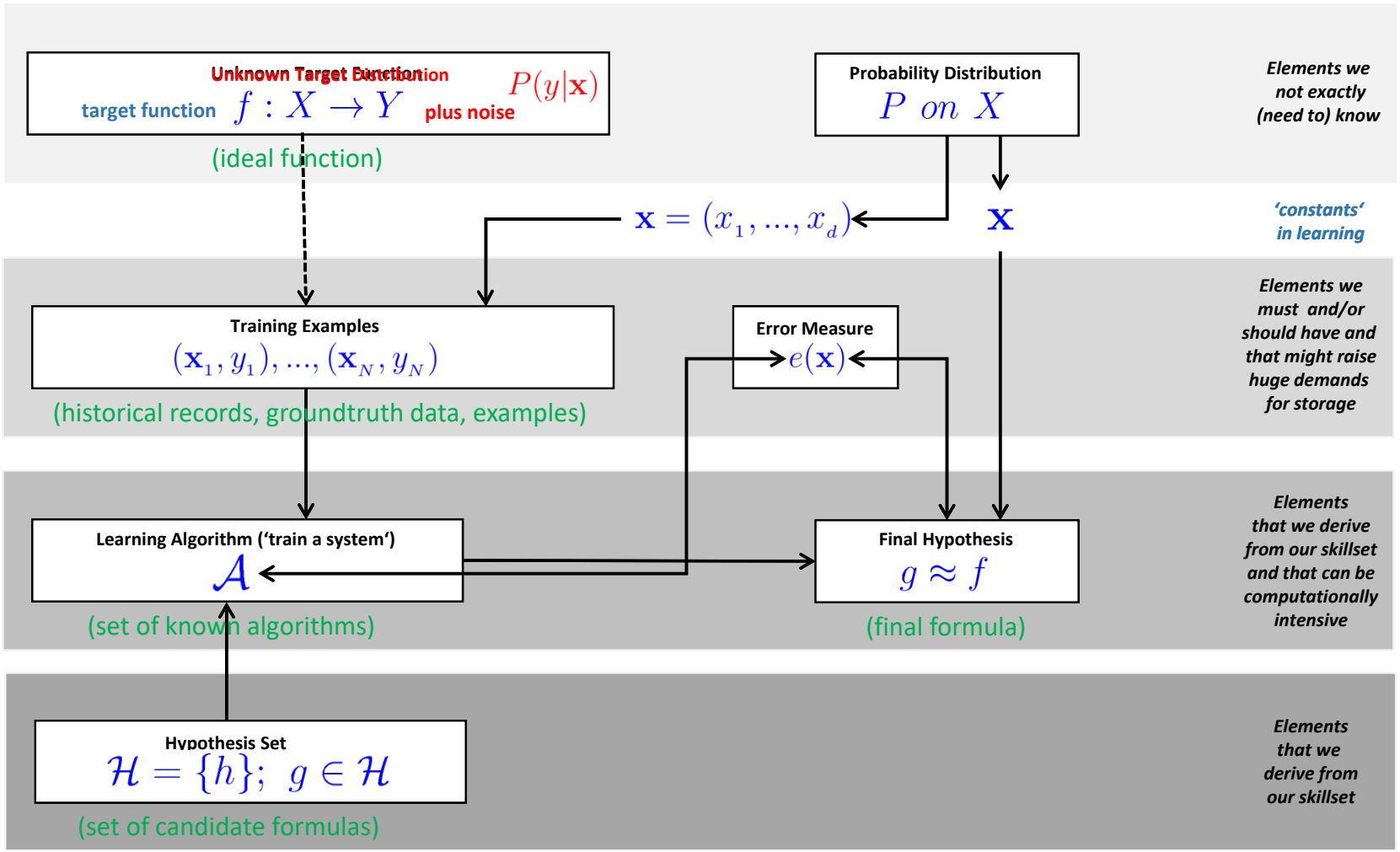
  - E.g. 'point-wise error measure'

Error Measure
α

- '(Noisy) Target function' is not a (deterministic) function (e.g. think movie rated now and in 10 years from now)
  - Getting with 'same x in' the 'same y out' is not always given in practice
  - Problem: 'Noise' in the data that hinders us from learning
  - Idea: Use a 'target distribution' instead of 'target function'
  - E.g. credit approval (yes/no)

Unknown Target Distribution
target function $f : X \to Y$ plus noise $P(y|\mathbf{x})$
(ideal function)

- **Statistical Learning Theory refines the learning problem of learning an unknown target distribution**


Unknown Target Function
target function $f : X \to Y$
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
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
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
Hypothesis Set
$\mathcal{H} = \{h\}; \ g \in \mathcal{H}$
(set of candidate formulas)
Elements that we derive from our skillset

Mathematical Building Blocks – Our Linear Example – Error Measures

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

(one point at a time is picked)

1. Pick one misclassified training point where:

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

Error Measure α

2. Update the weight vector:

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

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

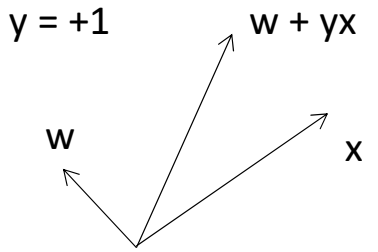
(y_n is either +1 or -1)

Error Measure α

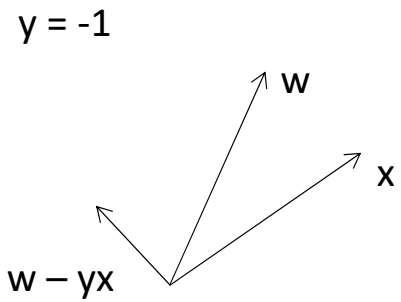
- Terminates when there are no misclassified points

(converges only with linearly seperable data)

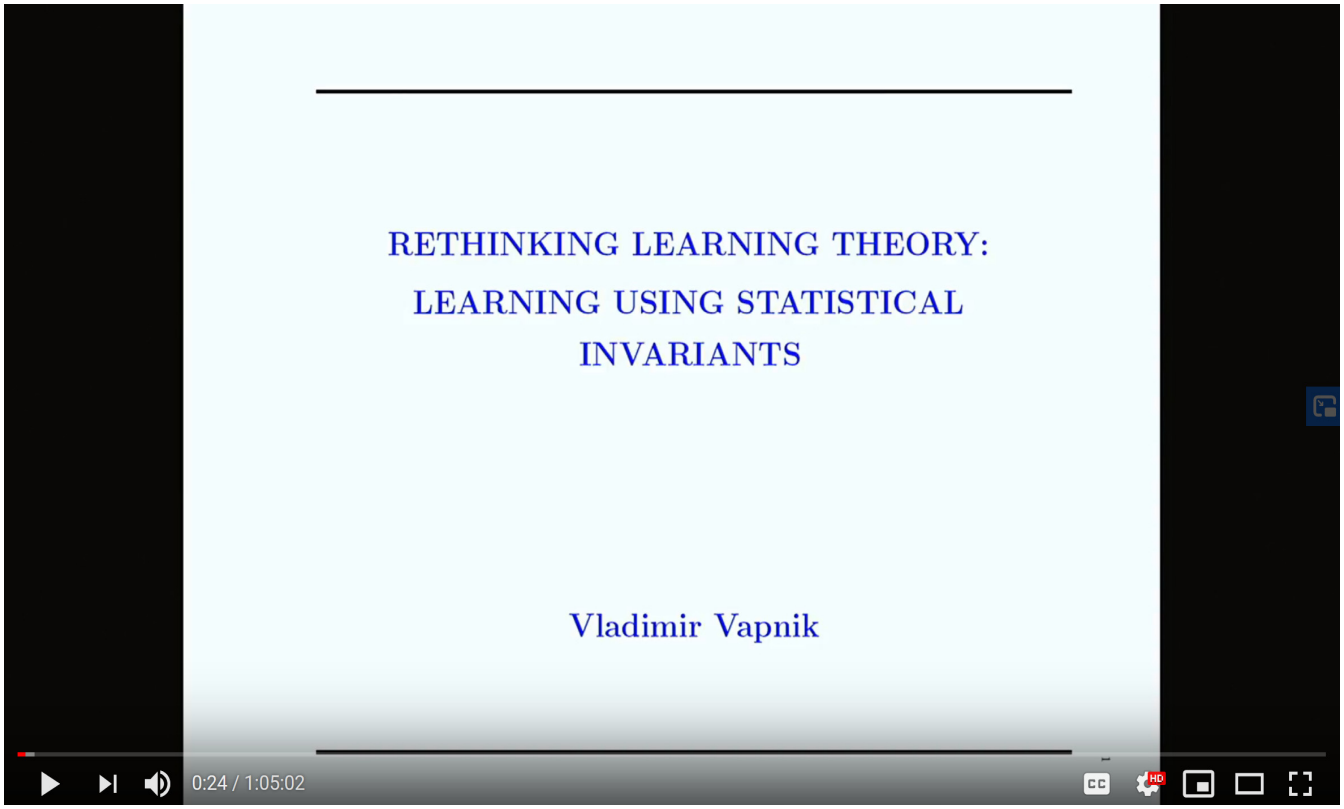
(a)



(b)

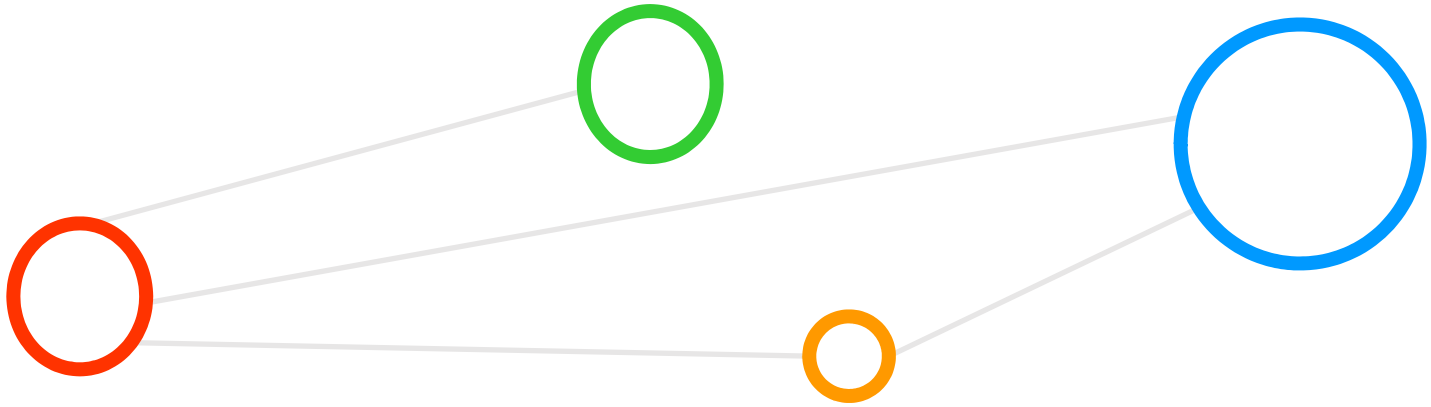


[Video] Rethinking Statistical Learning by Prof. Vladimir Vapnik (2018)



[7] YouTube Video, Rethinking Learning Theory

Vapnik – Chervonenkis (VC) Inequality & Dimension



Training and Testing – Influence on Learning

- Mathematical notations
  - Testing follows: (hypothesis clear) $\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ > \epsilon\ ]\ <=\ 2\quad e^{-2\epsilon^2 N}$
  - Training follows: (hypothesis search) $\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ > \epsilon\ ]\ <=\ 2Me^{-2\epsilon^2 N}$
- Practice on 'training examples'
  - Create two disjoint datasets
  - One used for training only (aka training set)
  - Another used for testing only (aka test set)

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

Training Examples

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

(historical records, groundtruth data, examples)

- Training & Testing are different phases in the learning process
  - Concrete number of samples in each set often influences learning

Theory of Generalization – Initial Generalization & Limits

- Learning is feasible in a probabilistic sense
  - Reported final hypothesis – using a 'generalization window' on $E_{out}(g)$
  - Expecting 'out of sample performance' tracks 'in sample performance'
  - Approach: $E_{in}(g)$ acts as a 'proxy' for $E_{out}(g)$

$$E_{out}(g) \approx E_{in}(g)$$

This is not full learning – rather 'good generalization' since the quantity $E_{out}(g)$ is an unknown quantity

- Reasoning
  - Above condition is not the final hypothesis condition:
  - More similiar like $E_{out}(g)$ approximates 0
    (out of sample error is close to 0 if approximating f)
  - $E_{out}(g)$ measures how far away the value is from the 'target function'
  - Problematic because $E_{out}(g)$ is an unknown quantity (cannot be used…)
  - The learning process thus requires 'two general core building blocks'

Final Hypothesis

$$g \approx f$$

Theory of Generalization – Learning Process Reviewed

- ‘Learning Well‘
  - Two core building blocks that achieve $E_{out}(g)$ approximates 0
- First core building block
  - Theoretical result using Hoeffdings Inequality $E_{out}(g) \approx E_{in}(g)$
  - Using $E_{out}(g)$ directly is not possible – it is an unknown quantity
- Second core building block (try to get the ‘in-sample‘ error lower)
  - Practical result using tools & techniques to get $E_{in}(g) \approx 0$
  - e.g. linear models with the Perceptron Learning Algorithm (PLA)
  - Using $E_{in}(g)$ is possible – it is a known quantity – ‘so lets get it small‘
  - Lessons learned from practice: in many situations ‘close to 0‘ impossible

- **Full learning means that we can make sure that $E_{out}(g)$ is close enough to $E_{in}(g)$ [from theory]**
- **Full learning means that we can make sure that $E_{in}(g)$ is small enough [from practical techniques]**

Complexity of the Hypothesis Set – Infinite Spaces Problem

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ > \epsilon\ ]\ <=\ 2Me^{-2\epsilon^2 N}$$

theory helps to find a way to deal with infinite M hypothesis spaces

- Tradeoff & Review
  - Tradeoff between Є, M, and the 'complexity of the hypothesis space H'
  - Contribution of detailed learning theory is to 'understand factor M'
- M Elements of the hypothesis set $\mathcal{H}$ M elements in H here
  - Ok if N gets big, but problematic if M gets big → bound gets meaningless
  - E.g. classification models like perceptron, support vector machines, etc.
  - Challenge: those classification models have continous parameters
  - Consequence: those classification models have infinite hypothesis spaces
  - Aproach: despite their size, the models still have limited expressive power

- **Many elements of the hypothesis set H have continous parameter with infinite M hypothesis spaces**

Factor M from the Union Bound & Hypothesis Overlaps

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ >\epsilon\ ]\ <= \Pr\ [\ |\ E_{in}(h_1) - E_{out}(h_1)\ |\ >\epsilon$$
$$\text{or}\ \ |\ E_{in}(h_2) - E_{out}(h_2)\ |\ >\epsilon\ \ ...$$
$$\text{or}\ \ |\ E_{in}(h_M) - E_{out}(h_M)\ |\ >\epsilon\ \ ]$$

assumes no overlaps, all probabilities happen disjointly

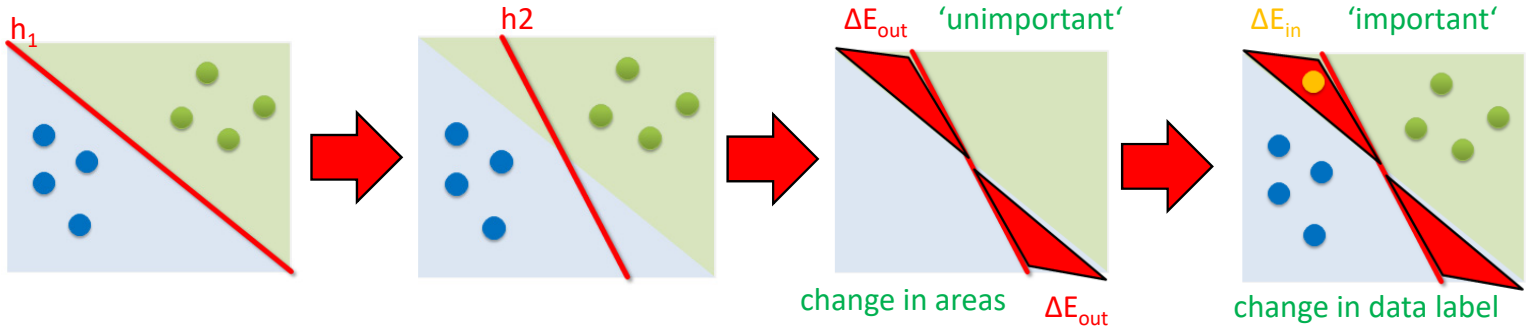
$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ >\epsilon\ ]\ <=\ 2Me^{-2\epsilon^2 N}$$

takes no overlaps of M hypothesis into account

- Union bound is a 'poor bound', ignores correlation between h
  - Overlaps are common: the interest is shifted to data points changing label

$$|\ E_{in}(h_1) - E_{out}(h_1)\ | \approx\ \ |\ E_{in}(h_2) - E_{out}(h_2)\ |$$

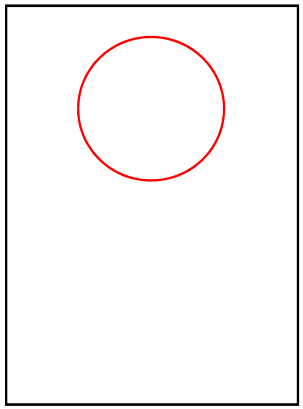
(at least very often, indicator to reduce M)



- **Statistical Learning Theory provides a quantity able to characterize the overlaps for a better bound**

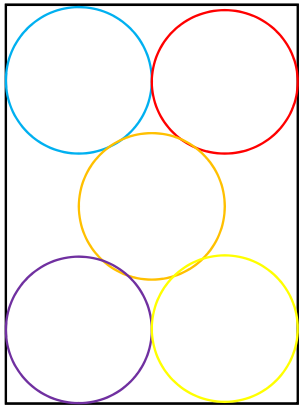
Replacing M & Large Overlaps

(Hoeffding Inequality)



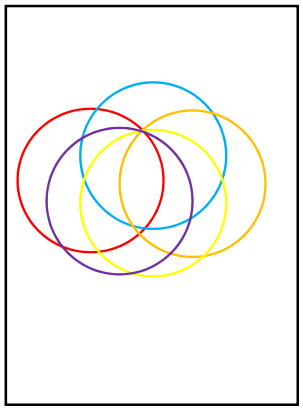
(valid for 1 hypothesis)

(Union Bound)



(valid for M hypothesis, worst case)

(towards Vapnik Chervonenkis Bound)



(valid for m (N) as growth function)

- Characterizing the overlaps is the idea of a 'growth function'
  - Number of dichotomies:
    Number of hypothesis but
    on finite number N of points

$$\mathbf{m}_{\mathcal{H}}(N) = \max_{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N} |\mathcal{H}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)|$$

  - Much redundancy: Many hypothesis will reports the same dichotomies

- **The mathematical proofs that $m_H(N)$ can replace M is a key part of the theory of generalization**

Complexity of the Hypothesis Set – VC Inequality

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ > \epsilon\ ]\ <=\ 2Me^{-2\epsilon^2 N}$$

$$\mathbf{m}_{\mathcal{H}}(N) = \max_{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N} |\mathcal{H}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)|$$

- Vapnik-Chervonenkis (VC) Inequality
  - Result of mathematical proof when replacing M with growth function m
  - 2N of growth function to have another sample (2 x $E_{in}(h)$, no $E_{out}(h)$)

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ > \epsilon\ ]\ <=\ 4m_{\mathcal{H}}(2N)e^{-1/8\epsilon^2 N}$$

(characterization of generalization)

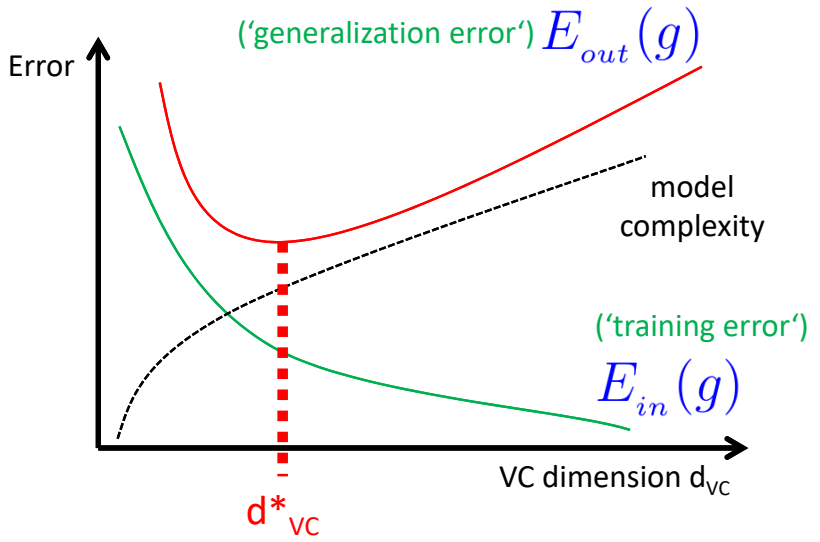
  - In Short – finally : We are able to learn and can generalize 'ouf-of-sample'

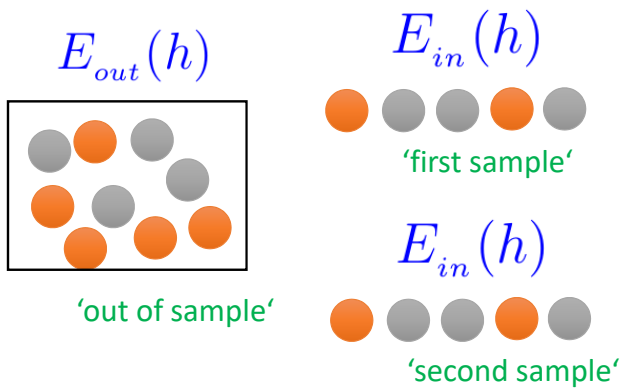
- **The Vapnik-Chervonenkis Inequality is the most important result in machine learning theory**
- **The mathematial proof brings us that M can be replaced by growth function (no infinity anymore)**
- **The growth function is dependent on the amount of data N that we have in a learning problem**

Complexity of the Hypothesis Set – VC Dimension & Model Complexity

- Vapnik-Chervonenkis (VC) Dimension over instance space X
  - VC dimension gets a 'generalization bound' on all possible target functions

Issue: unknown to 'compute' – VC solved this using the growth function on different samples





idea: 'first sample' frequency close to 'second sample' frequency

- **Complexity of Hypothesis set H can be measured by the Vapnik-Chervonenkis (VC) Dimension d_{VC}**
- **Ignoring the model complexity d_{VC} leads to situations where $E_{in}(g)$ gets down and $E_{out}(g)$ gets up**

Different Models – Hypothesis Set & Model Capacity

Hypothesis Set

$$\mathcal{H} = \{h\};\ g \in \mathcal{H}$$

$$\mathcal{H} = \{h_1, \ldots, h_m\};$$

(all candidate functions derived from models and their parameters)

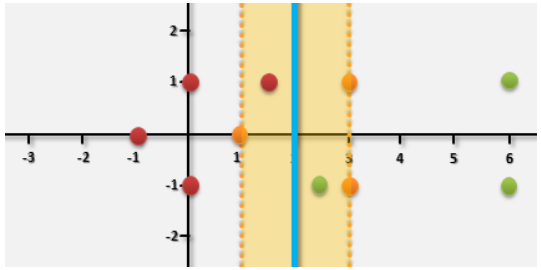
- **Choosing from various model approaches h_1, …, h_m is a different hypothesis**
- **Additionally a change in model parameters of h_1, …, h_m means a different hypothesis too**
- **The model capacity characterized by the VC Dimension helps in choosing models**
- **Occam's Razor rule of thumb: 'simpler model better' in any learning problem, not too simple!**

'select one function' that best approximates

Final Hypothesis

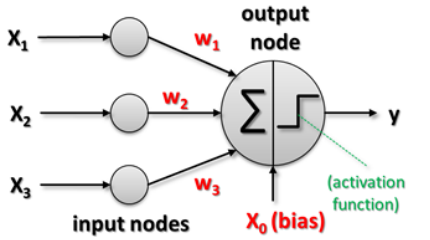
$$g \approx f$$

h_1



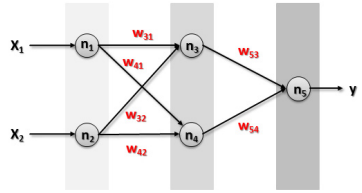
(e.g. support vector machine model)

h_2



(e.g. linear perceptron model)

h_m



(e.g. artificial neural network model)

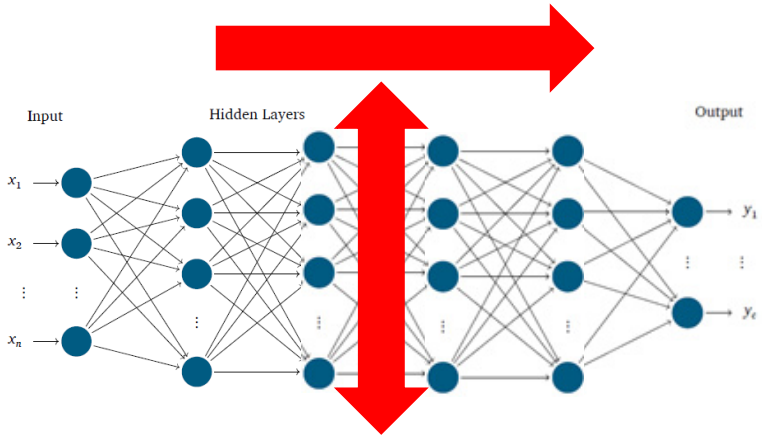
Exercises – ANN Four Hidden Layers & Increase Hidden Neurons (cf. Lecture 4)

```
# modeling step with no regularizer
model.add(Dense(N_HIDDEN, input_shape=(RESHAPED,)))
model.add(Activation('relu'))
model.add(Dense(N_HIDDEN))
model.add(Activation('relu'))
model.add(Dense(N_HIDDEN))
model.add(Activation('relu'))
model.add(Dense(N_HIDDEN))
model.add(Activation('relu'))
model.add(Dense(N_HIDDEN))
model.add(Activation('relu'))
model.add(Dense(N_HIDDEN))
model.add(Activation('relu'))
model.add(Dense(NB_CLASSES))
```

```
# parameter setup
NB_EPOCH = 200
BATCH_SIZE = 128
NB_CLASSES = 10 # number of outputs = number of digits
OPTIMIZER = SGD() # optimization technique
VERBOSE = 1
N_HIDDEN = 256 # number of neurons in one hidden layer, double for overfitting example
```

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 256)	200960
activation_1 (Activation)	(None, 256)	0
dense_2 (Dense)	(None, 256)	65792
activation_2 (Activation)	(None, 256)	0
dense_3 (Dense)	(None, 256)	65792
activation_3 (Activation)	(None, 256)	0
dense_4 (Dense)	(None, 256)	65792
activation_4 (Activation)	(None, 256)	0
dense_5 (Dense)	(None, 256)	65792
activation_5 (Activation)	(None, 256)	0
dense_6 (Dense)	(None, 256)	65792
activation_6 (Activation)	(None, 256)	0
dense_7 (Dense)	(None, 10)	2570
activation_7 (Activation)	(None, 10)	0

Total params: 532,490
Trainable params: 532,490
Non-trainable params: 0




```
Epoch 84/200
60000/60000 [==============================] - 2s 27us/step - loss: 0.0030 - acc: 0.9997
Epoch 85/200
60000/60000 [==============================] - 2s 27us/step - loss: 0.0027 - acc: 0.9999
Epoch 86/200
60000/60000 [==============================] - 2s 27us/step - loss: 0.0027 - acc: 0.9998
Epoch 87/200
60000/60000 [==============================] - 2s 27us/step - loss: 0.0025 - acc: 0.9999
Epoch 88/200
60000/60000 [==============================] - 2s 27us/step - loss: 0.0024 - acc: 0.9999
Epoch 89/200
60000/60000 [==============================] - 2s 27us/step - loss: 0.0022 - acc: 0.9999
Epoch 90/200
60000/60000 [==============================] - 2s 27us/step - loss: 0.0022 - acc: 0.9999
Epoch 91/200
60000/60000 [==============================] - 2s 27us/step - loss: 0.0020 - acc: 0.9999
Epoch 92/200
60000/60000 [==============================] - 2s 27us/step - loss: 0.0020 - acc: 1.0000
Epoch 93/200
60000/60000 [==============================] - 2s 27us/step - loss: 0.0020 - acc: 0.9999
Epoch 94/200
60000/60000 [==============================] - 2s 27us/step - loss: 0.0018 - acc: 0.9999
Epoch 95/200
60000/60000 [==============================] - 2s 27us/step - loss: 0.0017 - acc: 1.0000
Epoch 96/200
60000/60000 [==============================] - 2s 27us/step - loss: 0.0017 - acc: 0.9999
Epoch 97/200
60000/60000 [==============================] - 2s 27us/step - loss: 0.0017 - acc: 1.0000
Epoch 98/200
60000/60000 [==============================] - 2s 28us/step - loss: 0.0016 - acc: 1.0000
Epoch 99/200
60000/60000 [==============================] - 2s 27us/step - loss: 0.0015 - acc: 1.0000
Epoch 100/200
60000/60000 [==============================] - 2s 28us/step - loss: 0.0016 - acc: 1.0000
Epoch 101/200
60000/60000 [==============================] - 2s 28us/step - loss: 0.0015 - acc: 1.0000
Epoch 102/200
60000/60000 [==============================] - 2s 27us/step - loss: 0.0014 - acc: 1.0000
Epoch 103/200
60000/60000 [==============================] - 2s 27us/step - loss: 0.0014 - acc: 1.0000
Epoch 104/200
60000/60000 [==============================] - 2s 27us/step - loss: 0.0013 - acc: 1.0000
Epoch 105/200
60000/60000 [==============================] - 2s 27us/step - loss: 0.0013 - acc: 1.0000

Epoch 189/200
60000/60000 [==============================] - 2s 28us/step - loss: 5.1711e-04 - acc: 1.0000
Epoch 190/200
60000/60000 [==============================] - 2s 28us/step - loss: 5.1452e-04 - acc: 1.0000
Epoch 191/200
60000/60000 [==============================] - 2s 27us/step - loss: 5.1261e-04 - acc: 1.0000
Epoch 192/200
60000/60000 [==============================] - 2s 27us/step - loss: 5.0608e-04 - acc: 1.0000
Epoch 193/200
60000/60000 [==============================] - 2s 27us/step - loss: 5.0765e-04 - acc: 1.0000
Epoch 194/200
60000/60000 [==============================] - 2s 27us/step - loss: 5.0340e-04 - acc: 1.0000
Epoch 195/200
60000/60000 [==============================] - 2s 27us/step - loss: 5.0246e-04 - acc: 1.0000
Epoch 196/200
60000/60000 [==============================] - 2s 27us/step - loss: 4.9839e-04 - acc: 1.0000
Epoch 197/200
60000/60000 [==============================] - 2s 27us/step - loss: 4.9692e-04 - acc: 1.0000
Epoch 198/200
60000/60000 [==============================] - 2s 27us/step - loss: 4.9328e-04 - acc: 1.0000
Epoch 199/200
60000/60000 [==============================] - 2s 27us/step - loss: 4.9209e-04 - acc: 1.0000
Epoch 200/200
60000/60000 [==============================] - 2s 27us/step - loss: 4.9030e-04 - acc: 1.0000
```

```
Epoch 200/200
60000/60000 [==============================] - 2s 27us/step - loss: 4.9030e-04 - acc: 1.0000
```

```
#### Model evaluation
```

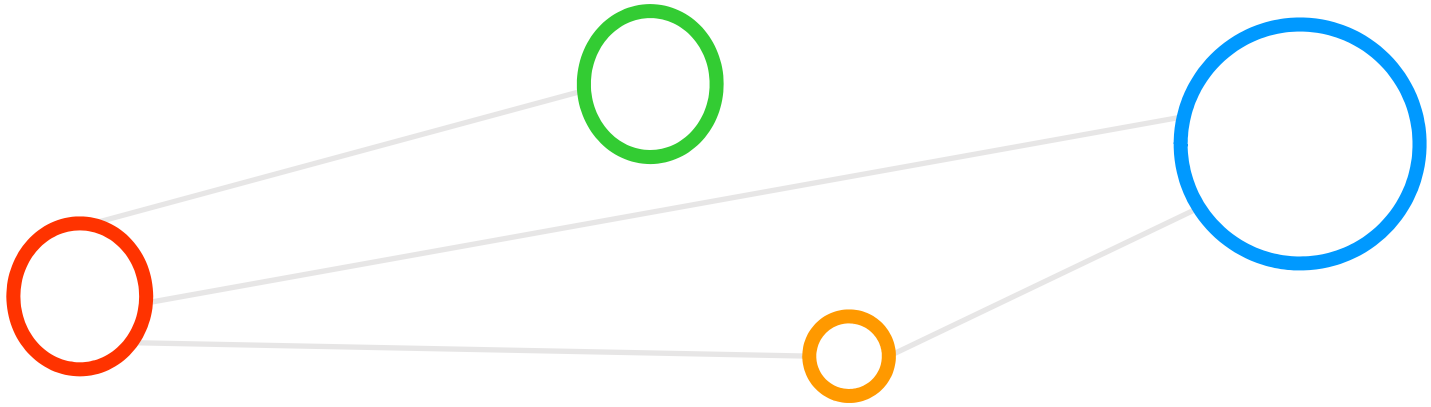
```
score = model.evaluate(X_test, Y_test, verbose=VERBOSE)
print("Test score:", score[0])
print('Test accuracy:', score[1])
```

```
10000/10000 [==============================] - 1s 50us/step
Test score: 0.13010296761920395
Test accuracy: 0.978
```

[Video] Prevent Overfitting for better Generalization

[6] YouTube Video, Stop Overfitting

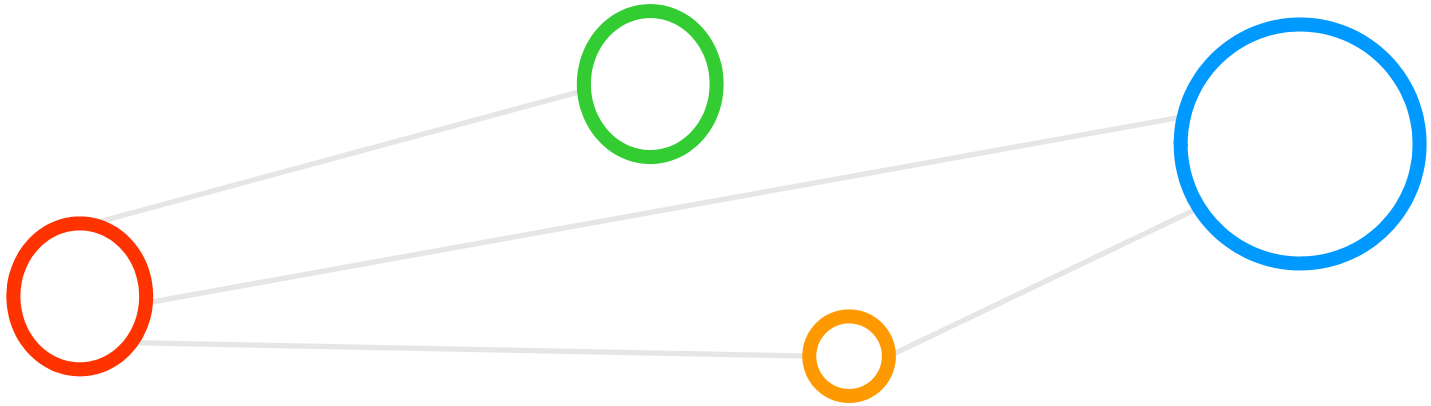
Lecture Bibliography



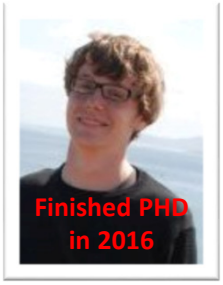
Lecture Bibliography

- [1] Multilayer Perceptron, Online: https://www.eecs.yorku.ca/course_archive/2012-13/F/4404-5327/lectures/10%20Multilayer%20Perceptrons.pdf
- [2] MIT 6.S191: Introduction to Deep Learning, Online: http://www.introtodeeplearning.com
- [3] Species Iris Group of North America Database, Online: http://www.signa.org
- [4] Cheng, A.C, Lin, C.H., Juan, D.C., InstaNAS: Instance-aware Neural Architecture Search, Online: https://arxiv.org/abs/1811.10201
- [5] Evgeniou T., Pontil M., Poggio T., 'Statistical Learning Theory: A Primer', International Journal of Computer Vision 38(1), 9–13, 2000, Online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.92.9504&rep=rep1&type=pdf
- [6] YouTube Video, Udacity, 'Overfitting', Online: https://www.youtube.com/watch?v=CxAxRCv9WoA
- [7] YouTube Video, 'Rethinking Learning Theory', Online: https://www.youtube.com/watch?v=LEYglsxKclo&t=2921s

Acknowledgements



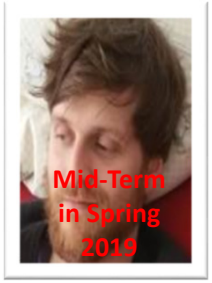
Acknowledgements – High Productivity Data Processing Research Group



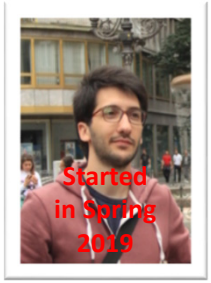
PD Dr.
G. Cavallaro

Senior PhD
Student A.S. Memon

Senior PhD
Student M.S. Memon



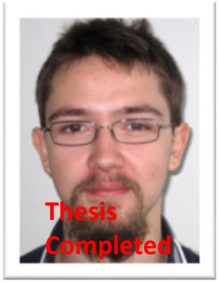
PhD Student
E. Erlingsson



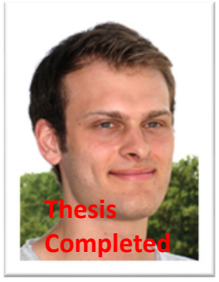
PhD Student
S. Bakarat

PhD Student
R. Sedona

Dr. M. Goetz
(now KIT)

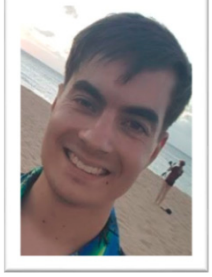


MSc M.
Richerzhagen
(now other division)



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)


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