FMAN45 MACHINE LEARNING SPRING 2019

LUND UNIVERSITY

L-1: Introduction and basics

Ted Kronvall, Cristian Sminchisescu

Computer Vision and Machine Learning Group

Cristian Sminchisescu

Prof. Dr.

Cristian Sminchisescu is a Professor in the Department of Mathematics, Faculty of Engineering at Lund University, working in computer vision and machine learning.

cristian.sminchisescu@math.lth.se

Ted Kronvall

Postdoc

Ted Kronvall is a postdoctoral researcher in machine learning working on deep generative modeling. He has earned his PhD in mathematical statistics, and his MSc in industrial engineering, from Lund University.

ted@maths.lth.se

Henning Petzka

Postdoc

Henning holds a Ph.D. in Mathematics from the University of Toronto, Canada. His research focuses on the mathematical understanding of deep learning techniques and generative models.

henning.petzka@math.lth.se

Erik Gärtner

Ph.D. student

Erik Gärtner is a doctoral student in computer vision and machine learning at the Department of Mathematics, Faculty of Engineering at Lund University. His work focuses on novel applications of reinforcement learning in computer vision tasks.

erik.gartner@math.lth.se

David Nilsson

Ph.D. student

David Nilsson is a PhD student at Lund University. His field of research is semantic segmentation applied to videos.

david.nilsson@math.lth.se

Aleksis Pirinen

Ph.D. student

Aleksis applies deep reinforcement learning to computer vision tasks such as object detection and active vision, aiming to make visual recognition systems more flexible and adaptive.

aleksis.pirinen@math.lth.se

Maria Priisalu

Ph.D. student

Maria Priisalu is a doctoral student at Lund University where is supervised by Prof. Sminchisescu. Her research is focused on machine learning in computer vision. In particular she is interested in scene understanding.

maria.priisalu@math.lth.se

Martin Trimmel

Ph.D. student

Martin Trimmel is a doctoral student at Lund University since October 2017 under the supervision of Professor Cristian Sminchisescu.

martin.trimmel@math.lth.se

LUND UNIVERSITY

Computer Vision and Machine Learning Group

Cristian Sminchisescu @Google in Zurich

Prof. Dr.

Cristian Sminchisescu is a Professor in the Department of Mathematics, Faculty of Engineering at Lund University, working in computer vision and machine learning.

cristian.sminchisescu@math.lth.se

Ted Kronvall

Postdoc

Ted Kronvall is a postdoctoral researcher in machine learning working on deep generative modeling. He has earned his PhD in mathematical statistics, and his MSc in industrial engineering, from Lund University.

ted@maths.lth.se

Henning Petzka

Postdoc

Henning holds a Ph.D. in Mathematics from the University of Toronto, Canada. His research focuses on the mathematical understanding of deep learning techniques and generative models.

henning.petzka@math.lth.se

Erik Gärtner

Ph.D. student

Erik Gärtner is a doctoral student in computer vision and machine learning at the Department of Mathematics, Faculty of Engineering at Lund University. His work focuses on novel applications of reinforcement learning in computer vision tasks.

erik.gartner@math.lth.se

David Nilsson

Ph.D. student

David Nilsson is a PhD student at Lund University. His field of research is semantic segmentation applied to videos.

david.nilsson@math.lth.se

Aleksis Pirinen

Ph.D. student

Aleksis applies deep reinforcement learning to computer vision tasks such as object detection and active vision, aiming to make visual recognition systems more flexible and adaptive.

aleksis.pirinen@math.lth.se

Maria Priisalu

Ph.D. student

Maria Priisalu is a doctoral student at Lund University where is supervised by Prof. Sminchisescu. Her research is focused on machine learning in computer vision. In particular she is interested in scene understanding.

maria.priisalu@math.lth.se

Martin Trimmel

Ph.D. student

Martin Trimmel is a doctoral student at Lund University since October 2017 under the supervision of Professor Cristian Sminchisescu.

martin.trimmel@math.lth.se

LUND UNIVERSITY

Admin

Course contents:

- Lectures (14), two per week.
- Computer sessions, approx. one per week.
- Office hours, TBD.

LUND UNIVERSITY

Admin

Examination:

- **Hand-in assignments**
  - Individual assessment
  - Approx. 2 weeks to submit report and solutions
  - Preferably solved using Matlab.
  - 70 % proficiency to pass
  - Two resubmissions allowed in total, provided a >= 30 % proficiency at initial submission.
  - No late submissions accepted!
- **Written exam**
  - Optional
  - Mandatory for higher grade (4-5).
  - Printed copies of course literature, lecture slides, and personal notes are ok to bring.

Admin

Assignments:

- Assignment 1
  - Handed out on Monday April 1st
  - To be submitted on Sunday April 14th
- Assignment 2
  - Handed out on Thursday April 11th
  - To be submitted on Wednesday May 8th
- Assignment 3
  - Handed out on Monday May 6th
  - To be submitted on Sunday May 19th
- Assignment 4
  - Handed out on Thursday May 17th
  - To be submitted on Wednesday May 30th

Easter break and re-exam period is April 18th - May 3rd

LUND UNIVERSITY

Admin

Assignments:

- Assignment 1
  - Handed out on Monday April 1st
  - To be submitted on Sunday April 14th
- Assignment 2
  - Handed out on Thursday April 11th
  - To be submitted on Wednesday May 8th
- Assignment 3
  - Handed out on Monday May 6th
  - To be submitted on Sunday May 19th
- Assignment 4
  - Handed out on Thursday May 17th
  - To be submitted on Wednesday May 30st

Some advice:

- The assignments will be hard work.
- Start working on them as soon as possible.
- There will be at least one computer session slot available for every student for every assignment - make use of these!
- Be concise in writing your reports!

LUND UNIVERSITY

Admin

Teaching platform:

- We'll be using Moodle, available through a link at the course webpage http://www.ctr.maths.lu.se/course/machinlearn/2019/.
- Register to the course ”Machine Learning 2019”
- Use enrollment key ”clustering2019”.

Usage:

- Find pdf prints of lecture slides.
- Fetch the assignment descriptions.
- Submit your assignment report and solutions
- Get grades and feedback.
- DO NOT USE MOODLE FOR MESSAGING!

LUND UNIVERSITY

Admin

Contact:

- For questions about registration, contact the course administrators: expedition@math.lth.se.
- … lecture content, contact the respective lecturer.
- … assignments, contact the respective TAs.
- For other questions, contact the course organizer Ted Kronvall, ted.kronvall@math.lth.se.

LUND UNIVERSITY

Admin

Course literature:

- Bishop, C. M.: Pattern Recognition and Machine Learning. Springer, 2006, Online version available at https:// www.microsoft.com/ en-us/research/people/cmbishop/ #!prml-book (March 2019).
- I. Goodfellow, Y. Bengio & A. Courville: Deep Learning. MIT press, 2016, HTLM version available at http:// www.deeplearningbook.org/ (March 2019).
- R. Sutton & A.G. Barto: Reinforcement Learning, An Introduction. MIT Press, 2017. Draft available at http:// incompleteideas.net/sutton/book/the-book-2nd.html (March 2019).

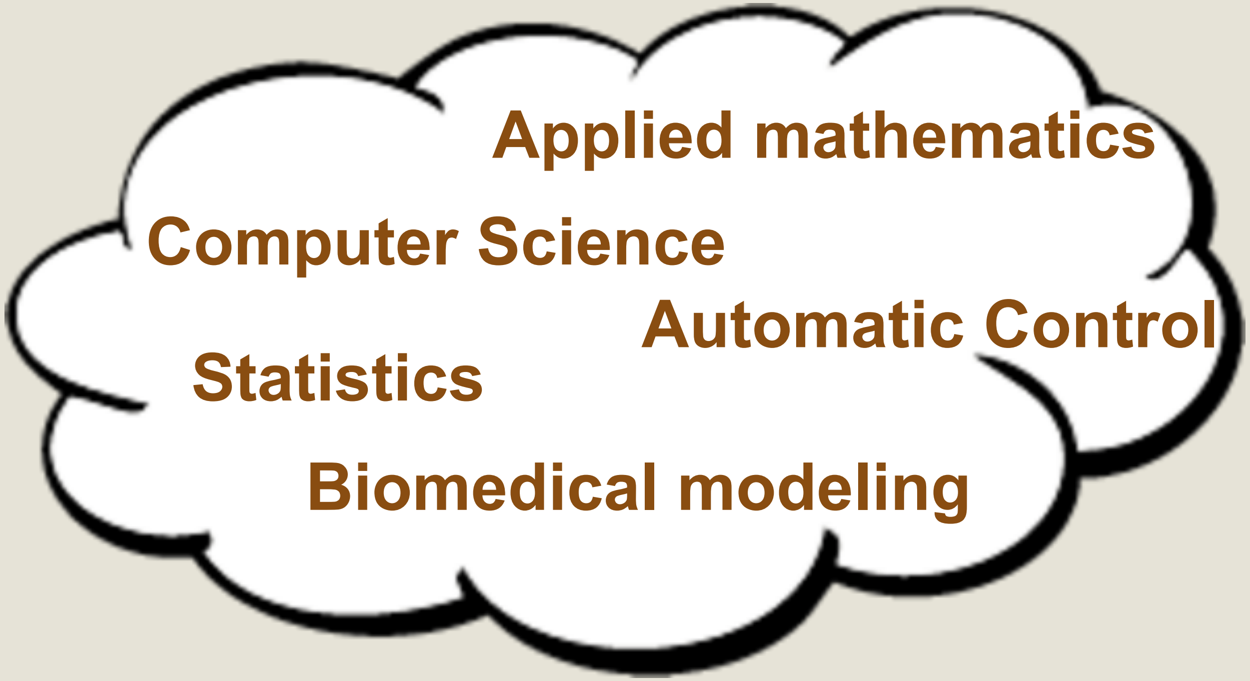
Additional references:

- T. Hastie, R. Tibshirani, J. Friedman: The elements of statistical learning, data mining, inference and prediction, 2nd edition, Springer, 2009, online version available at https:// web.stanford.edu/~hastie/Papers/ESLII.pdf (March 2019).

LUND UNIVERSITY

What is Machine Learning?

What is Machine Learning?



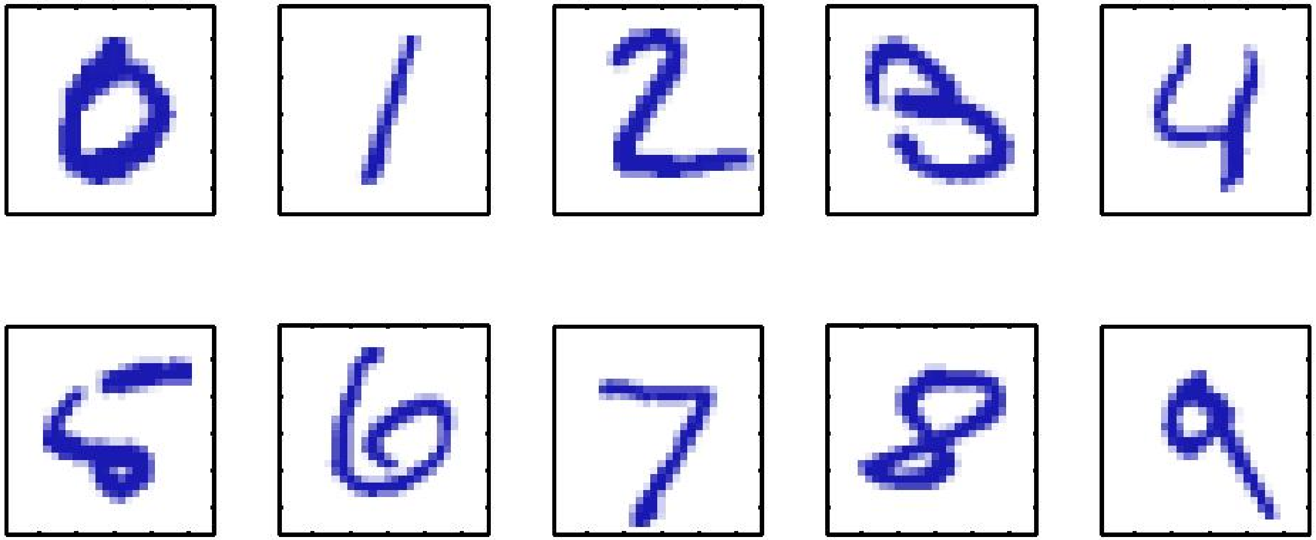
LUND UNIVERSITY

What is Machine Learning?

- Algorithms that allow machines to ”learn”.
- Extract information automatically
- Performing tasks not specifically programmed for.
- Allowing machines to find their own patterns in data.

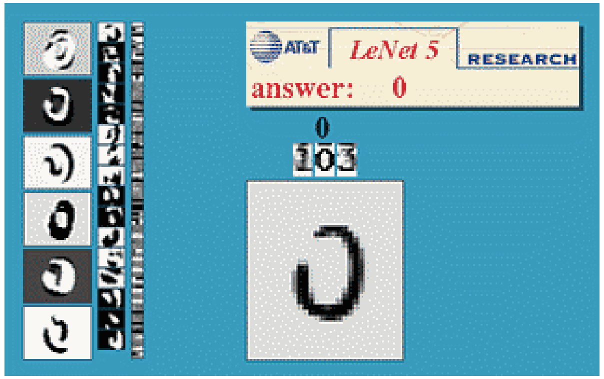
Task: Handwritten Digit Recognition

Difficult to characterize what is specific to each digit



Optical character recognition (OCR)

Technology to convert scanned images/docs to text



Digit recognition, AT&T labs
http://yann.lecun.com/exdb/lenet/

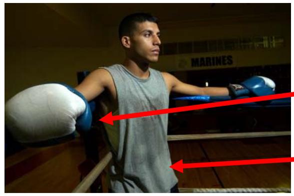
Automatic license plate reading

Visual Recognition of People in Images

General poses, high-dimensional *(30-100 dof)*

Self-occlusions

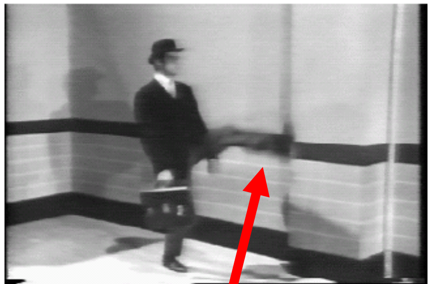
Difficult to segment the individual limbs



Loss of Information in the perspective projection

Partial views

Different body sizes



Accidental allignments
Motion blur

Several people, occlusions

Reduced observability of body parts due to loose fitting clothing

LUND UNIVERSITY

Machine Learning Approach

- It is difficult to explicitly design programs that can recognize people or digits in images
  - Modeling the object structure, the physics, the variability, and the image formation process can be very difficult
- Instead of designing the programs by hand, we collect many examples that specify the correct outputs for different inputs
- A machine learning algorithm takes these examples and produces a program trained to give the answers we want
  - If designed properly, the program may work in new situations, not just the ones it was trained on

LUND
UNIVERSITY

When to use Machine Learning?

- A machine learning approach is most effective when the structure of the task is not well understood - or too difficult to model explicitly -, but can be characterized by a dataset with strong statistical regularity

- Machine learning is also useful in dynamic situations, when the task is constantly changing
  - e.g. a robot operating in an unknown environment

A spectrum of machine learning tasks

Statistics----------------------------Artificial Intelligence

- Low-dimensional data (e.g. less than 100 dimensions)
- Lots of noise in the data
- There is not much structure in the data, and what structure there is, can be represented by a fairly simple model.
- The main problem is distinguishing true structure from noise.

- High-dimensional data (e.g. more than 100 dimensions)
- The noise is not sufficient to obscure the structure in the data if we process it right.
- There is a huge amount of structure in the data, but the structure is too complicated to be represented by a simple model.
- The main problem is figuring out a way to represent the complicated structure that allows it to be learned.

slide from MLG@Toronto

Machine Learning Applications

- natural language processing
- speech and handwriting recognition
- object recognition in computer vision
- image retrieval
- robotics
- medical diagnosis
- bioinformatics
- classifying DNA sequences
- detecting credit card fraud
- stock market analysis
- recommender systems
- analyzing social patterns and networks
- visualization, game playing and many more…

In this course:

- Basic concepts; tasks, training, validation, regularization.
- Probability, statistics, and optimization

Classical

- Linear regression
- Methods for classification
- Clustering methods
- Neural networks

Modern

- Convolutional and recurrent NNs
- Generative modeling
- Reinforcement learning
- Natural language processing
- Deep learning for computer vision

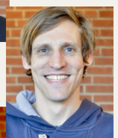
In this course:

- Basic concepts; tasks, training, validation, regularization.

- Probability theory and optimization
- Linear regression
- Methods for classification

- Clustering methods
- Neural networks

- Convolutional and recurrent NNs



- Generative modeling

- Reinforcement learning
- Natural language processing
- Deep learning for computer vision

Selected (and some outdated) examples:

Face detection

Most digital cameras now detect faces

LUND UNIVERSITY

Face detection

~~Most~~ digital cameras ~~now~~ detect faces

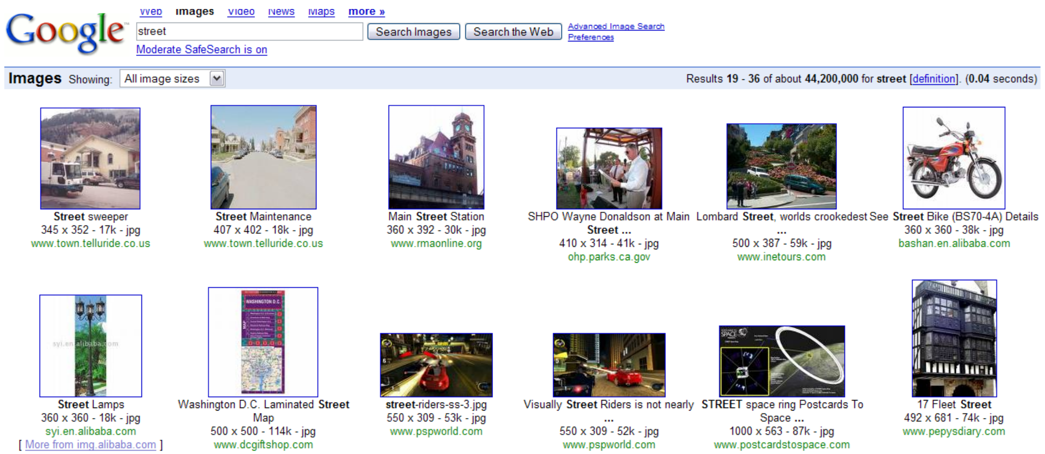
But also:

- Who's in the image?
- Where are they?
- When are they?
- What are they doing?
- How long has it been since they met last time?
- How to update their personal marketing messages from this new information.
- etc.

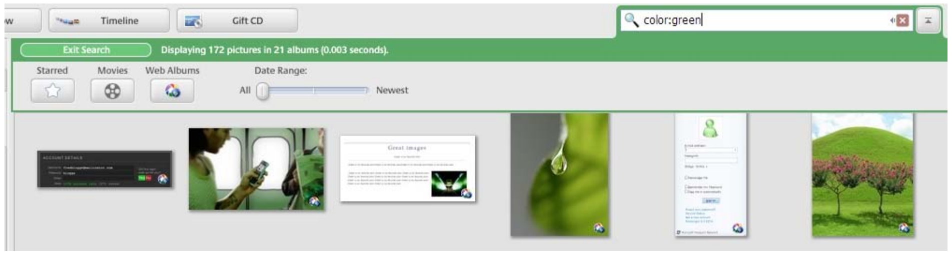
LUND UNIVERSITY

Online search engines

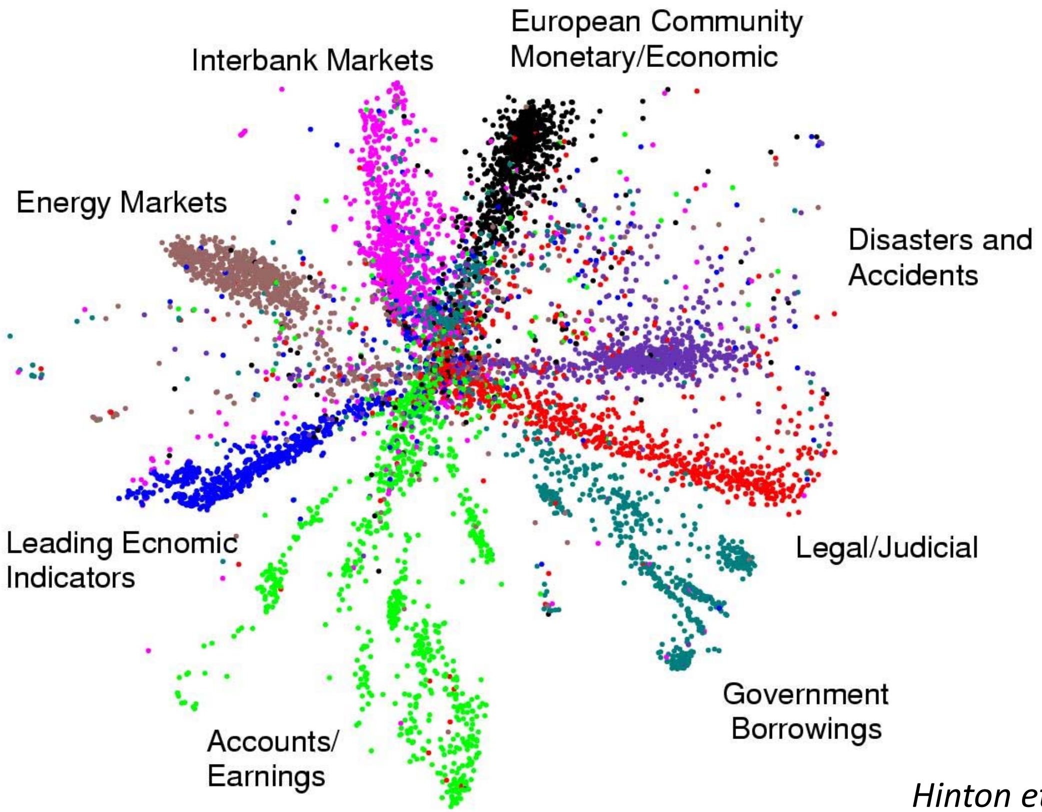
Query: STREET



Organizing photo collections



Displaying the structure of a set of documents using a deep neural network



Hinton et al., 2007

IBM's Watson (not) in Jeopardy (2011)

90 IBM servers, 2880 cores (@3.55Gb), 15TB RAM

Cities question: *Its largest airport was named for a World War II hero; its second largest, for a World War II battle*

Champions *answer 2/3 questions with 85-95% accuracy*

LUND UNIVERSITY

3D Human Pose – Microsoft's Kinect (2011)

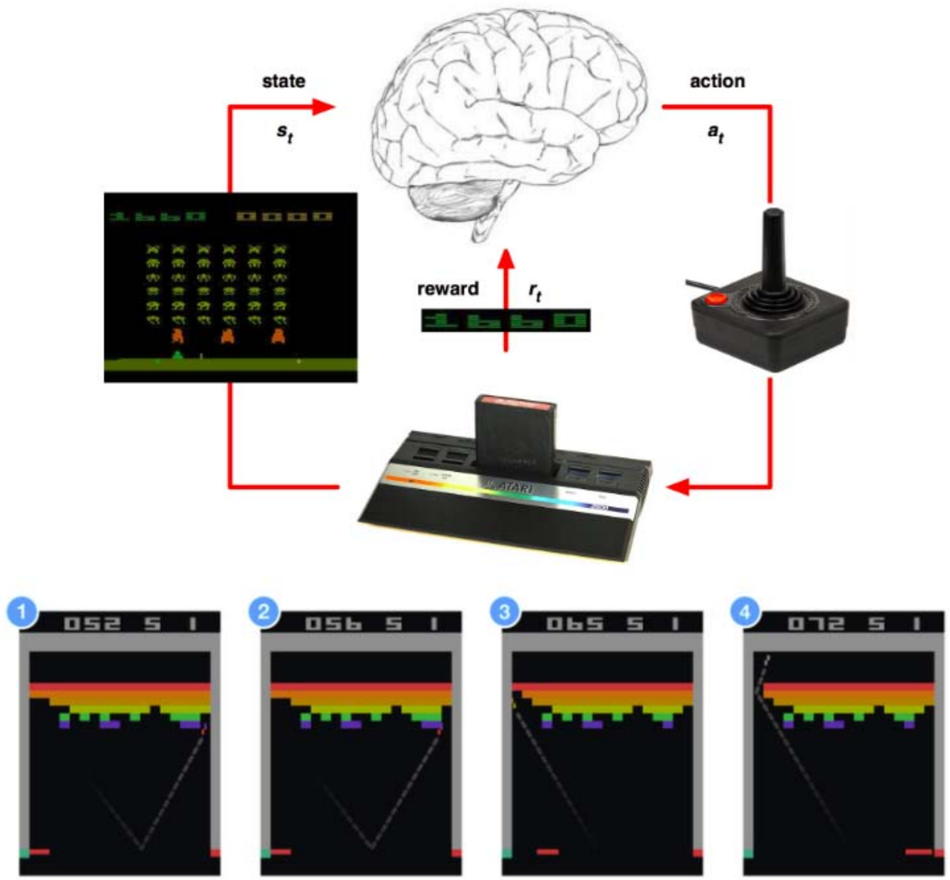
Shotton et al., CVPR 2011

LUND UNIVERSITY

Autonomous driving (2014)

LUND UNIVERSITY

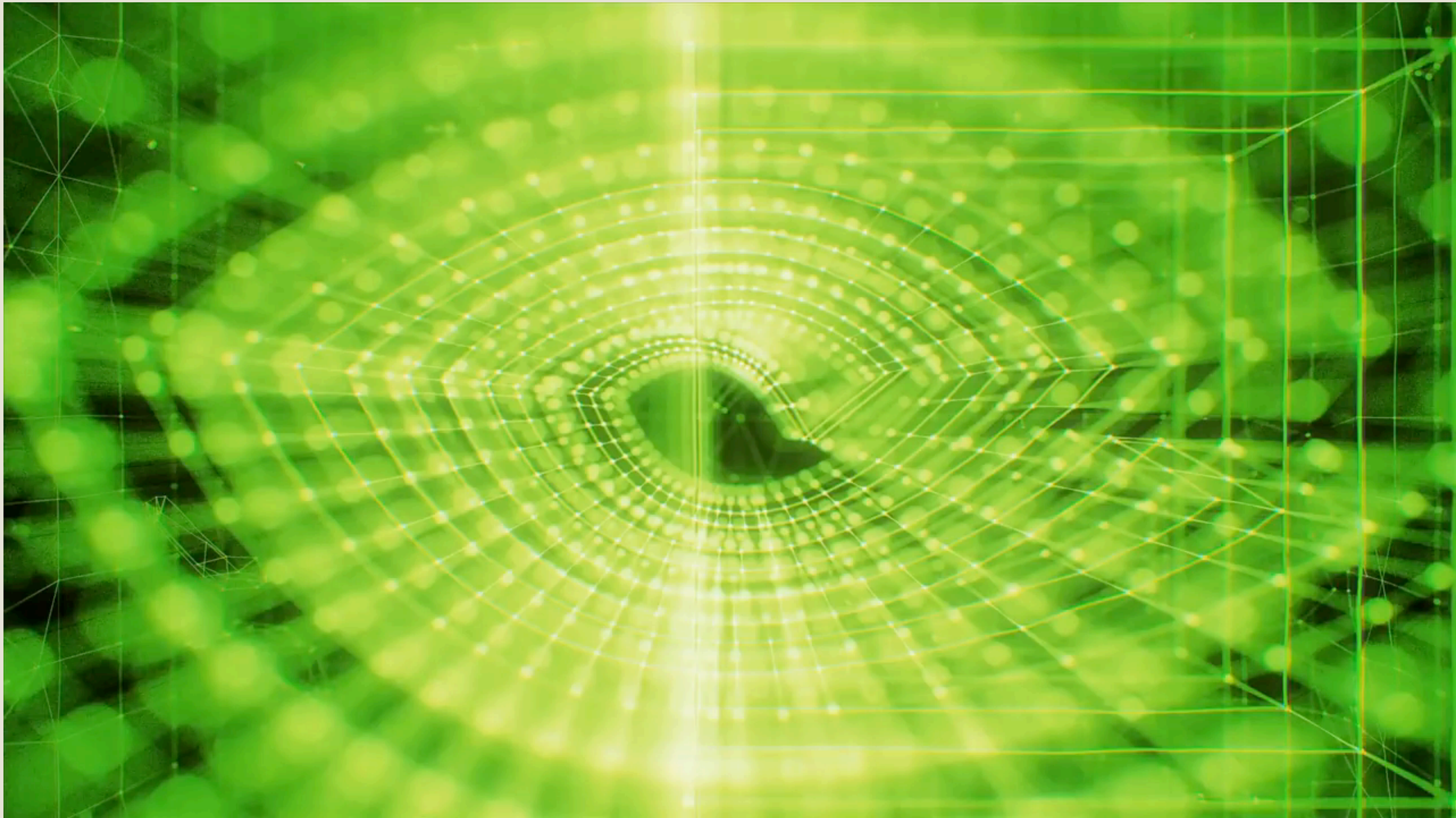
Game Playing, Reinforcement Learning
Atari



Reinforcement learning

Reinforcement learning (2017)

Generative modeling (2019)



Generative modeling (2018)

LUND UNIVERSITY

Is Machine Learning Solved?

Lots of success already but…

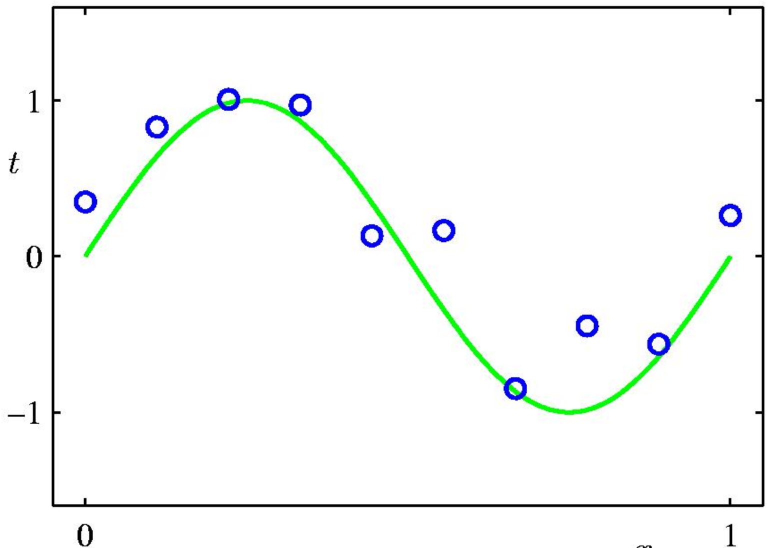
- Many existing systems lag behind human performance
  - Comparatively, see how fast children learn
- Handling large, unorganized data repositories, under uncertain and indirect supervision is an open problem
- Designing complex systems (e.g. a robot with sophisticated function and behavior) is an open problem
- Fundamental advances necessary at all levels
  - computational, algorithmic, representational, implementation and integration

LUND
UNIVERSITY

Case Study
Polynomial Curve Fitting

LUND UNIVERSITY

Polynomial Curve Fitting



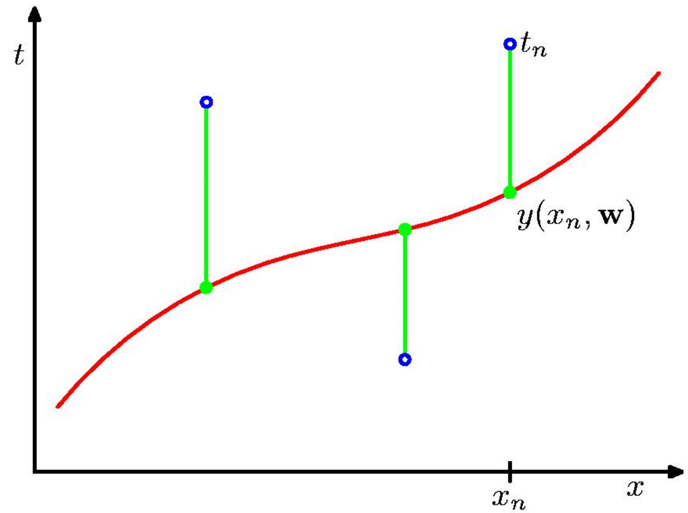
Synthetic data generated from:

$t(x) = sin2\pi x + \epsilon,$
$\epsilon \sim N\ (0, 0.3)$

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Model is **linear** in the parameters w and **non-linear** in the input x

Sum-of-Squares Error Function

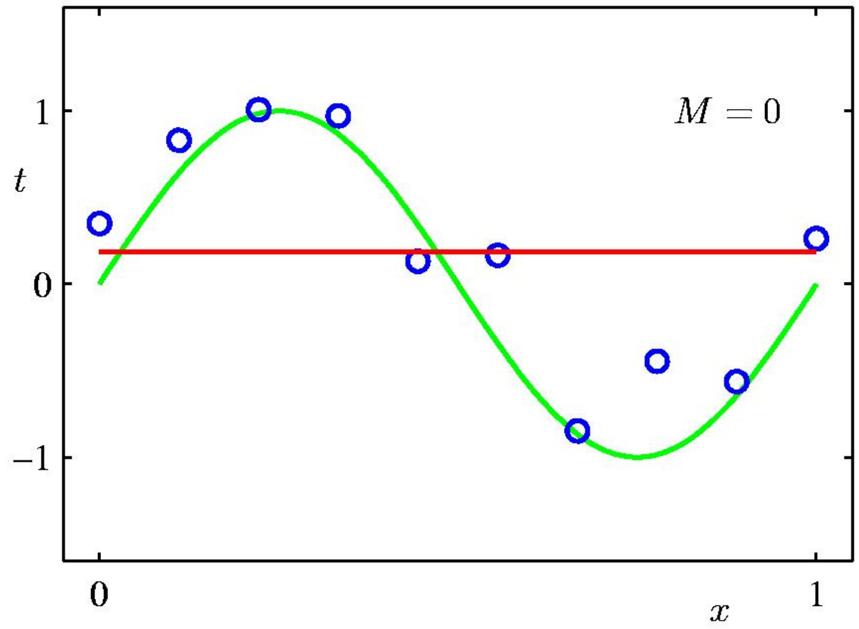


$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2$$

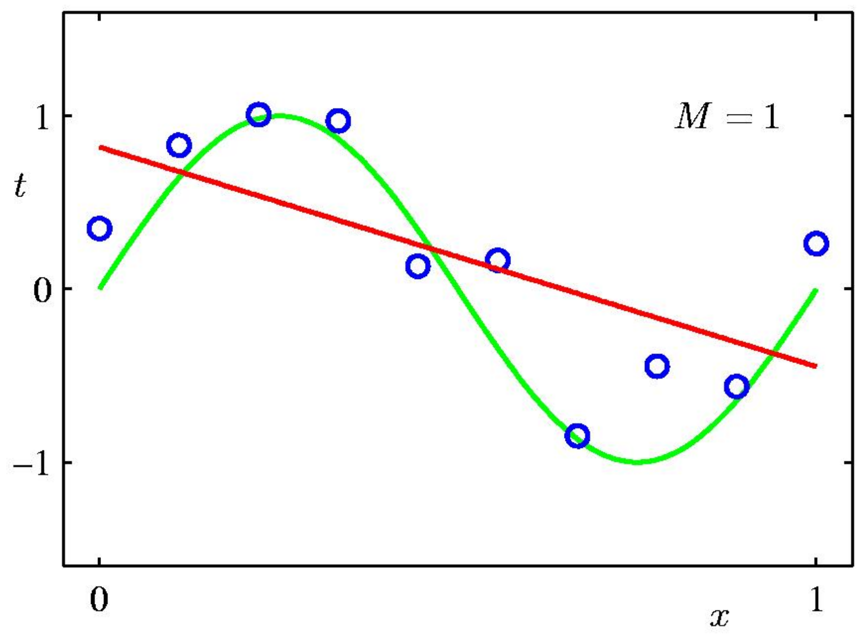
$$\sum_{n=1}^{N}\left(\sum_{j=0}^{M} w_j x_n^j - t_n\right) x_n^i = 0.$$

Closed-form solution to minimize *E* (take derivative w.r.t *w*, and set to 0 to find *w**)

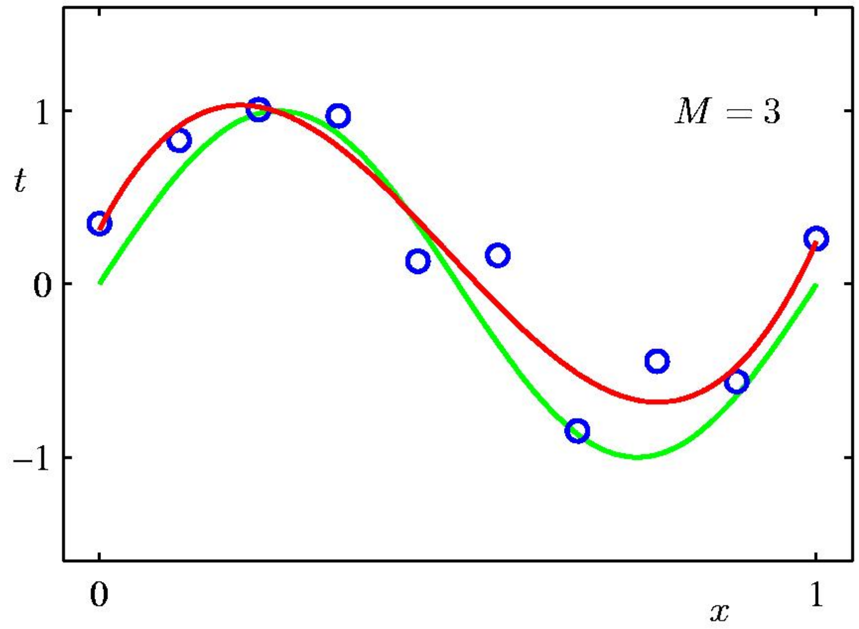
0^{th} Order Polynomial



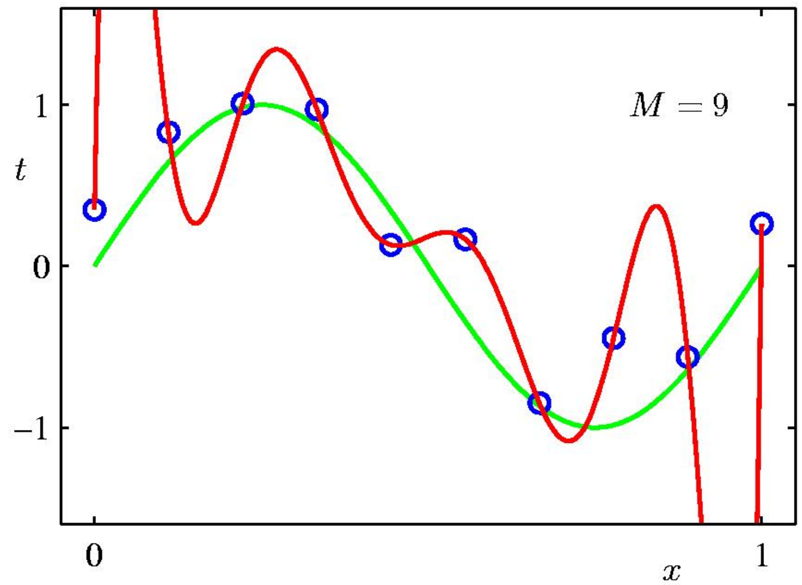
1st Order Polynomial



3rd Order Polynomial



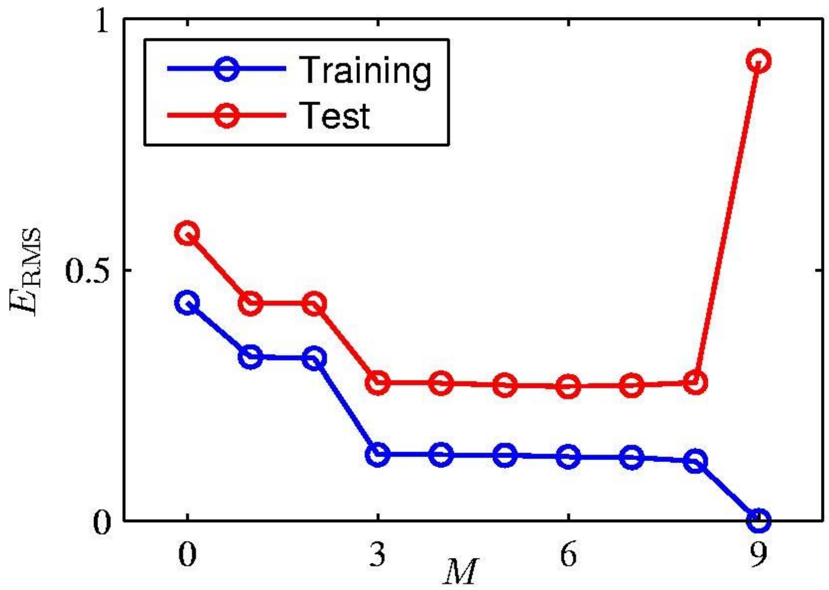
9th Order Polynomial (Over-fitting)



- Curve oscillates wildly
- Zero training error, but poor representation of data generating function
- Poor prediction on new inputs!

Power series expansions of the function sin(2πx) contain terms of all orders, so shouldn't results improve monotonically as we increase M?

Over-fitting



Unintuitive?

- Higher-order polynomials include lower order ones
- Power expansion of sin contains terms of all orders

What's wrong then?

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2$$

Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/N}$

Division by *N* allows us to compare different sizes of data sets on equal footing, square root ensures that E_{RMS} is measured on the same scale (and in the same units) as the target variable *t*.

LUND UNIVERSITY

Polynomial Coefficients

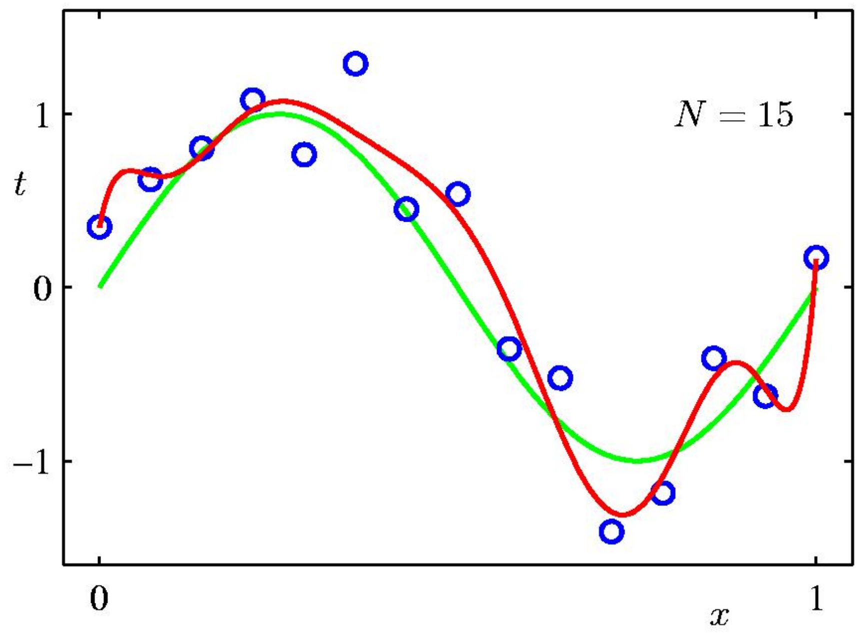
	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Higher-order polynomials tune to random noise

LUND UNIVERSITY

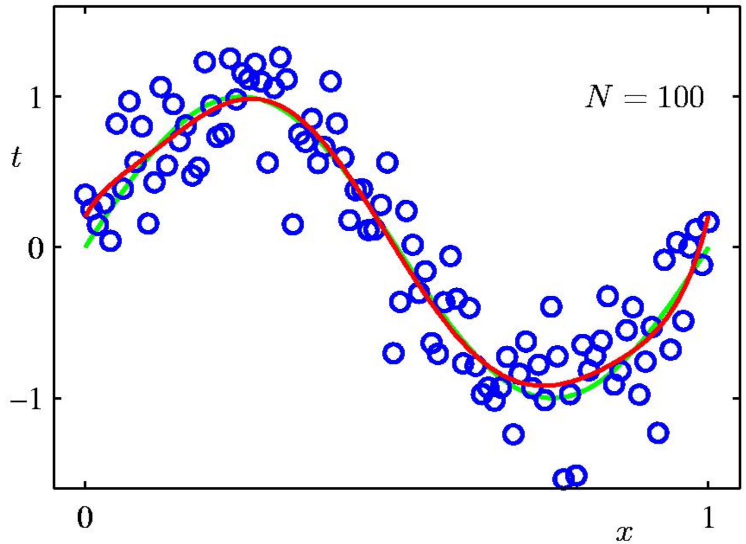
Data Set Size: $N = 15$

9^{th} Order Polynomial



Data Set Size: $N = 100$

9th Order Polynomial



The larger the data set, the more complex (flexible) the model that we can afford to fit. One heuristic is that the number of data points should be no less than some multiple (say 5 or 10) of the number of adaptive parameters in the model.

More data helps, but it should be better to adjust model capacity based on the intrinsic complexity of the problem to be solved, not just the datset size

Spline Fitting: Regularization

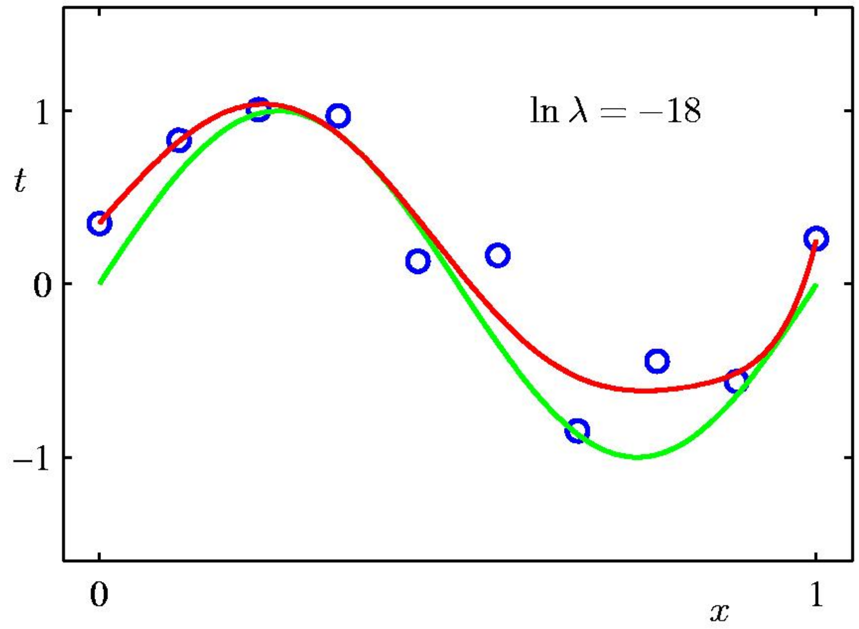
Penalize large coefficient values

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

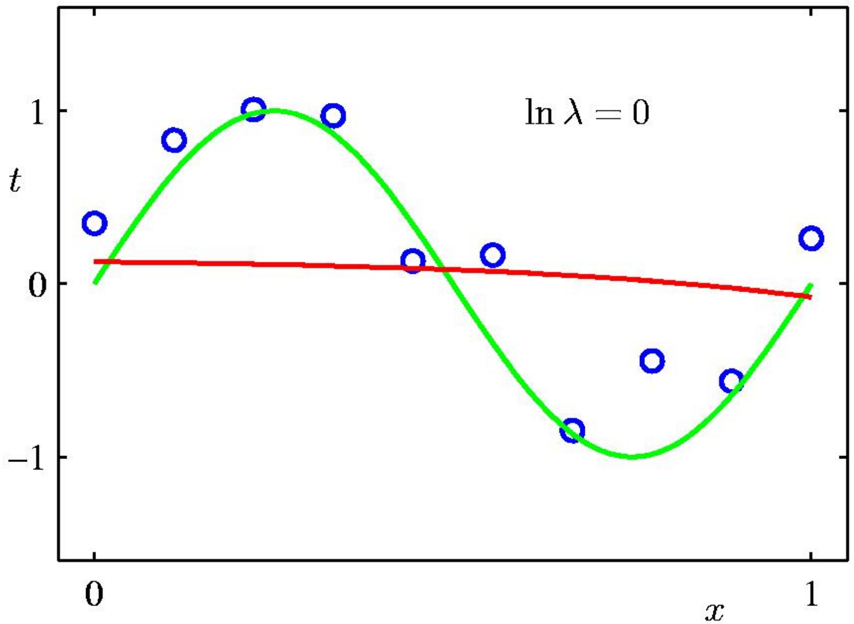
Shrinkage approaches

For quadratic regularizers: ridge regression, weight decay (neural networks)

Regularization ($M = 9$): $\ln \lambda = -18$



Regularization ($M = 9$) : $\ln \lambda = 0$



Polynomial Coefficients

	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

Machine Learning Basics

LUND
UNIVERSITY

Machine Learning Basics

- How to think about machine learning problems (and how to solve them).
- These approaches should permeate to all corners of the machine learning universe, including state-of-the-art deep learning methods.

LUND UNIVERSITY

Core terminology

THREE PILLARS OF MACHINE LEARNING PROBLEMS

The Task (T)	The Performance measure (P)	The Experience (E)
Which problem?	Which metric?	What data?

LUND UNIVERSITY

Task (T)

Example tasks:

- Classification
- Regression
- Transcription
- Anomaly detection
- Synthesis/sampling
- Denoising

LUND UNIVERSITY

Task (T)

Hypothesis space $\xrightarrow{\text{Learning}}$ **Model**

Task (T)

Hypothesis space

Hypothesis Spaces and Learning Machines

We can view learning as the process of exploring a *hypothesis space*

- We select functions from a specified set, the *hypothesis space*
- Hypothesis space is indexed by a set of *parameters w***,** which are variables we can adjust (search) to create different machines
- In supervised learning, each hypothesis is a function that maps inputs to outputs
- A challenge of machine learning is deciding how to represent inputs and outputs, and how to select the hypothesis space that would be appropriate for a given task
- The central trade-off is in selecting a hypothesis space that is powerful enough to represent the relationships between inputs and outputs, yet simple enough to be searched efficiently

LUND UNIVERSITY

Task (T)

CAPACITY

- A model's capacity is its ability to fit a wide variety of functions (data).
- May be controlled by choosing the hypothesis space (representational capacity)
- May be controlled by the number of trainable parameters
- In practice, other factors also affect a model's capacity, such as, e.g., optimization method and imperfection, and how much data is used for training (effective capacity)
- Difficult to quantify…
- In statistical learning, capacity is quantified by the Vapnik-Chervonenkis dimension (VC-dimension)

LUND UNIVERSITY

Task (T)

CAPACITY

”The central challenge in machine learning is that our algorithm must perform well on new, previously unseen inputs - not just those those on which our model was trained. The ability […] is called **generalization”**

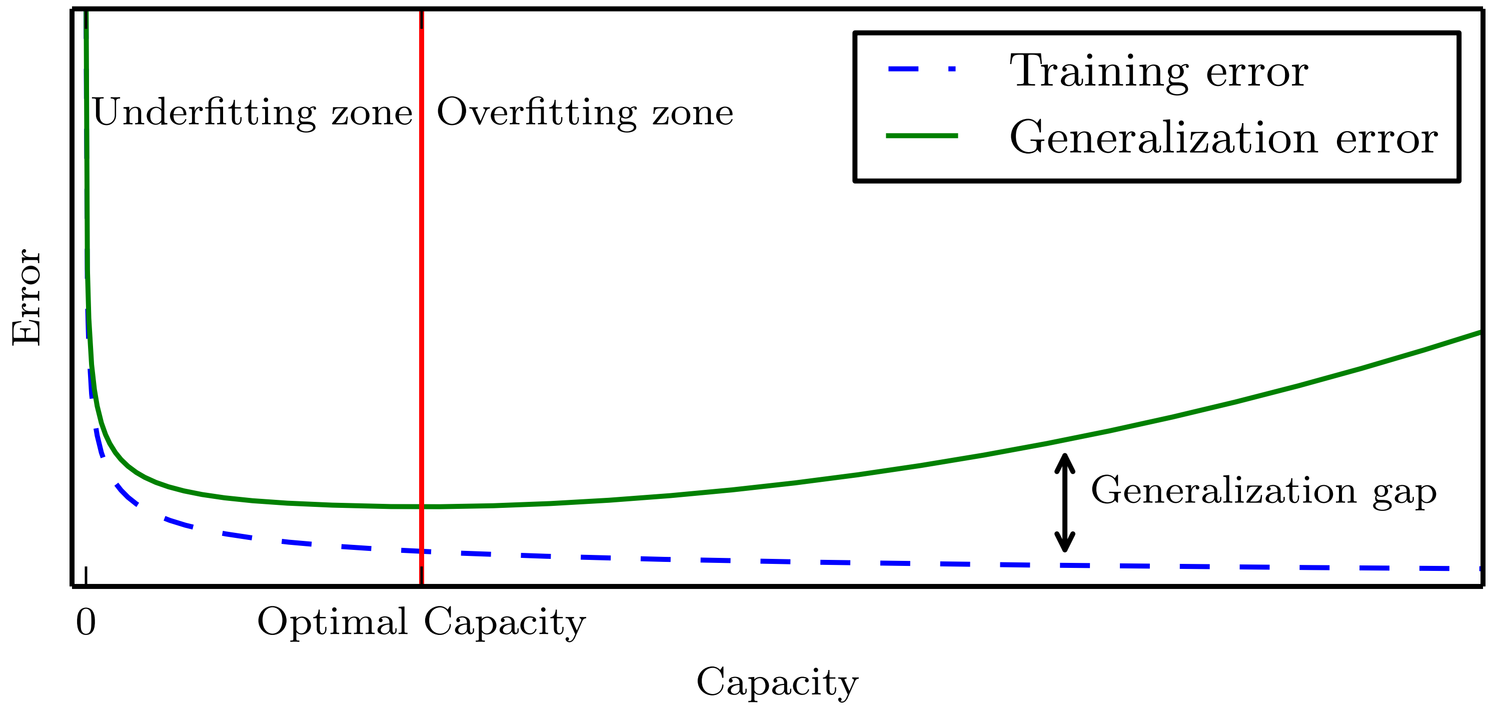


Figure 5.3

Task (T)
CAPACITY

Probabilistic Guarantees

$$E_{test} \leq E_{train} + \left(\frac{h + h\log(2N/h) - \log(p/4)}{N} \right)^{\frac{1}{2}}$$

where N = size of training set
h = VC dimension of the model class = complexity
p = upper bound on probability that this bound fails

If we train models of different complexities, we should ideally select the one that minimizes the type of bound above

We do not have such bounds for all models, and even when we do, the procedure would be effective only if the bounds are tight. But in general they are not. The theory offers insight, but using such results in practice will not be straightforward

adapted from MLG@Toronto

- The discrepancy between test and training error grows when the VC dimension increases.
- And decreases when the number of training examples increase.
- The K-I-S-S rule.

LUND UNIVERSITY

Task (T)

NO-FREE-LUNCH THEOREM

Averaged over all possible data-generating distributions, every classification algorithm has the same error rate when classifying previously unobserved points. (Wolpert, 1996)

- No ML algorithm is universally better than any other for all tasks.
- For specific data-generating distributions, some algorithms may be better than others.
- The key to good performance is to restrict a model's capacity suitably for the task at hand.

Loss Functions for Parameter Estimation

- Given an input and an output representation, and a hypothesis space, how do we set the machine parameters w?

- We will quantify performance on task defined by loss function $L(x, y, t, w)$
  - Inputs x, outputs of machine y, desired targets t, parameters w

- We will set the parameters w to minimize this average loss function measured empirically over some dataset

Possible Loss Functions

- Squared difference between predicted $y_n(x_n)$ and target (ground truth) real-valued outputs t_n

$$L = \sum_n \| t_n - y_n(x_n) \|^2$$

- Number of classification errors, makes optimization harder due to non-smooth derivative

$$L = \sum_n [t_n \neq y_n(x_n)]$$

- Negative log probability assigned to the correct answer
  - Principled, statistically justified
  - For some models, the same as squared error (regression with Gaussian output noise)
  - For other models, different expressions (cross-entropy error, for classification with discrete classes)

Practice: Use of a Validation Set

- Divide the available data into **three subsets**
  - **Training data** is used to learn the parameters of the model
  - **Validation data** is used to decide the type of model and the amount of regularization that works best
  - **Testing data** is used to get a final, unbiased estimate of how well the learning machine works
- We can re-divide the complete dataset and get another unbiased estimate of the true error rate. We can then average multiple estimates to decrease variance. This can be expensive

adapted from MLG@Toronto

Performance measure (P)

REGULARIZATION

”Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error” (Deep Learning)

$$\underset{\mathbf{w}}{\text{minimize}} \quad J(\mathbf{w}) = MSE_{\text{train}}(\mathbf{w}) + \lambda \mathbf{w}^\top \mathbf{w}$$

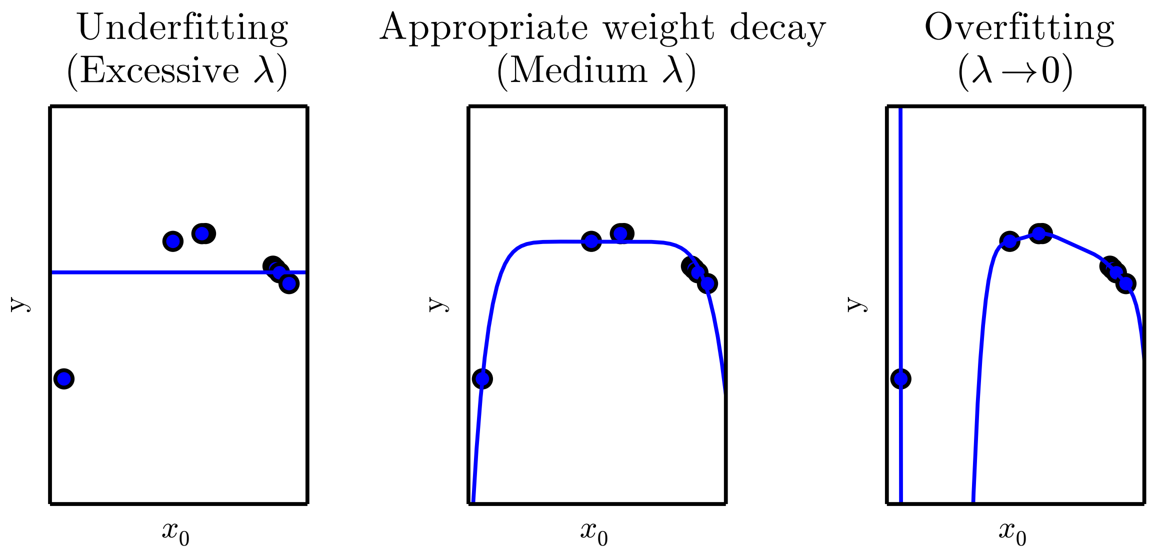


Figure 5.5

capacity hyperparameter

Performance measure (P)

REGULARIZATION

”Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error” (Deep Learning)

$$\underset{\mathbf{w}}{\text{minimize}} \quad J(\mathbf{w}) = MSE_{\text{train}}(\mathbf{w}) + \lambda \mathbf{w}^\top \mathbf{w}$$

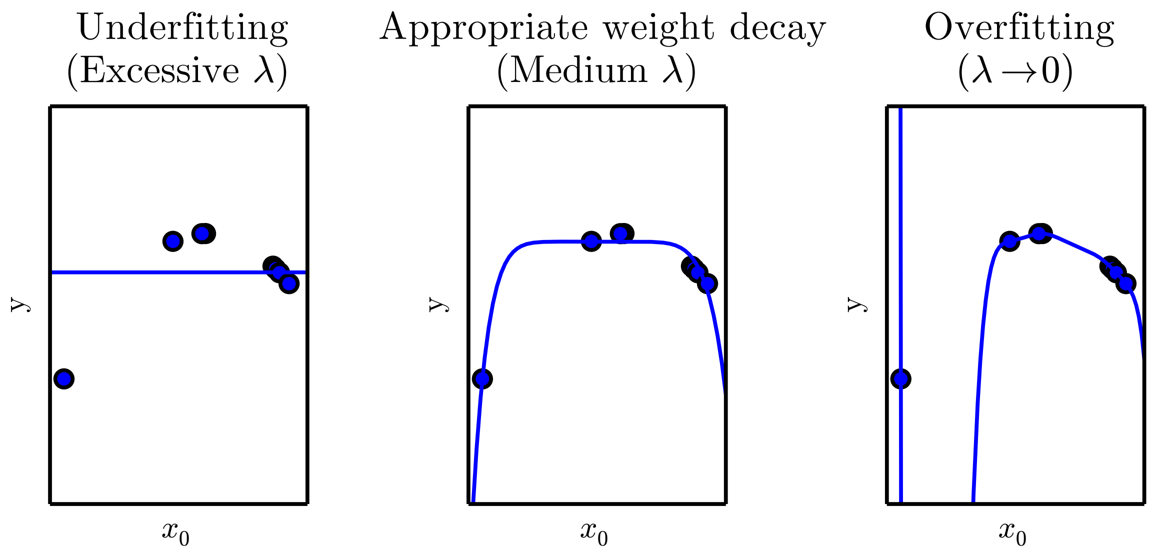


Figure 5.5

Performance measure (P)

Model Selection

Cross-Validation (rotated estimation)

One round of cross-validation involves partitioning a sample of the data into complementary subsets, estimate the model on one subset (called the *training/validation set*), ~~and test it on the other subset (called the *testing set*)~~. To reduce variance, multiple rounds of cross-validation are performed using different partitions, and results are averaged over all rounds.

The Experience (E)

1 • ***Supervised Learning:*** given examples of inputs and desired outputs (labeled data), predict outputs on future inputs
  – Ex: classification, regression, time series prediction
  – Who provides the correct outputs? Can these always be measured? Can the process scale?

2 • ***Unsupervised Learning*:** given only inputs (unlabeled data), automatically discover hidden representations, features, structure, etc.
  – Ex: clustering, outlier detection, compression
  – How can we know that representation is good?

• ***Semi-supervised Learning:*** given both labeled and unlabeled data, leverage information to improve both tasks
  – Somewhat practical compromise for data acquisition and modeling

Standard Learning Tasks

3 • ***Reinforcement Learning*:** given sequences of inputs, actions from a fixed set, and scalar rewards and punishments, learn to select action sequences in a way that maximizes expected reward
  - Not much information in the reward, which is often delayed
  - e.g. game playing, robot operating in a structured environment

Self study:

- **Goodfellow et al 2016 (Deep Learning):**
  - Chapter 5: up to 5.4
- **Bishop 2006 (Pattern Recognition and Machine Learning)**
  - Chapter 1: 1.1, 1.3 (1.4-1.5 are optional)
- **Hastie et al. 2019 (Elements of Statistical Learning)**
  - Chapter 7: 7.9 - 7.10

LUND UNIVERSITY

FMAN45 MACHINE LEARNING SPRING 2019

L-1: Introduction and basics

Ted Kronvall, Cristian Sminchisescu

LUND UNIVERSITY