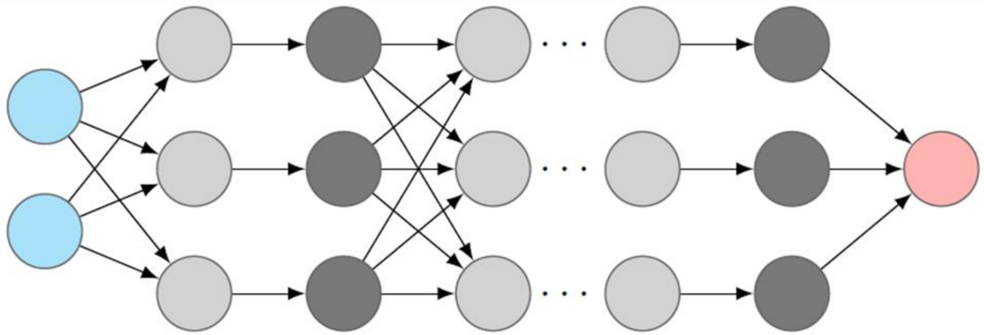
Deep Learning

Introductory course for Master students in computer science and mechatronics

Lecturer: Michael Möller – michael.moeller@uni-siegen.de
Exercises: Hartmut Bauermeister – hartmut.bauermeister@uni-siegen.de



Organizational Things

Necessary prior knowledge

- Linear Algebra
- Calculus (ideally with multiple variables, but we'll repeat this)
- Programming (we will introduce Python, NumPy, and PyTorch, but will not be able to repeat loops, conditions, data-types, classes…)

Nice to know but not necessary

- Image processing
- Optimization
- Basic machine learning, e.g. statistical learning theory, pattern recognition

Organizational Things

Exercises

- Will start next Monday 15th of October
- Will be in **room H-A 7118**
- We have 12 computers for the exercises, you may work in groups of 2 people
- There will be homework to prepare things we do in the exercises
- I will not collect and grade the homework, but **it is your responsibility to be prepared and actively work on the course material**
- You will get access to the exercise room and from there to a server with four NVIDIA GTX 1080Ti - for this we need your names, immatriculation number, and a signature that you respect certain rules of the room and computers.
- Hartmut Bauermeister, hartmut.bauermeister@uni-siegen.de, will lead the exercises.

Organizational Things

- My office: H-A 7106
- Hartmut's office: H-A 7116
- For appointments, please email us or contact us during the lecture/exercises
- The lecture and exercise start at quarter past.
- Course website: http://www.vsa.informatik.uni-siegen.de/en/deep-learning
- Username: student Passwort: 100%brain

This lecture is worth 5 credits. Discuss final exam!

Please do not be shy to say something and ask questions during the lecture!

The more we discuss, the more interesting the lecture is!

More deep learning discussions: http://www.uni-siegen.de/zess/kombibox/imr_kolloquium.html

Deep Learning

"So one way to think about all three of these ideas is that Machine Learning is the cutting edge of Artificial Intelligence. And Deep Learning is the cutting edge of the cutting edge."

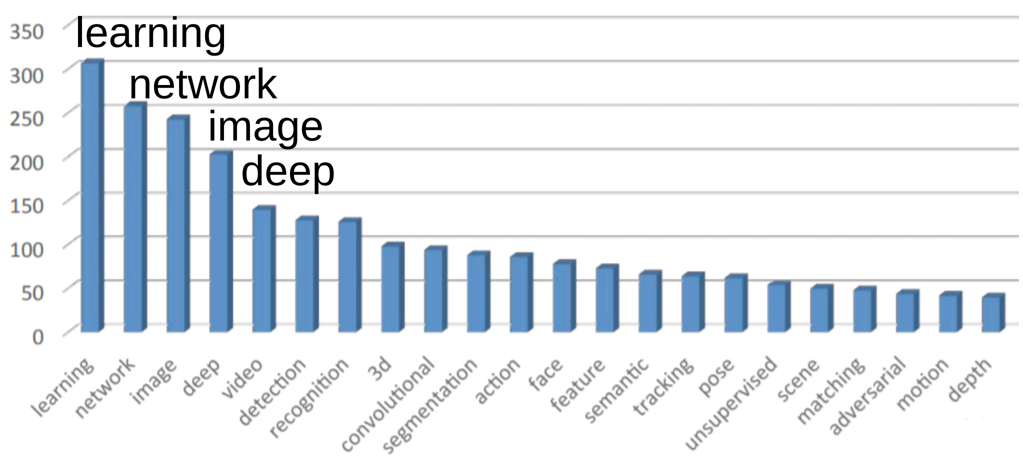
https://medium.com/machinevision/overview-of-artificial-intelligence-buzz-adb7a5487ac8, Oct. 4th

"Deep learning may be one of the most overhyped of modern technologies, but there is a good chance that it will one day become the secret sauce in many different business processes."

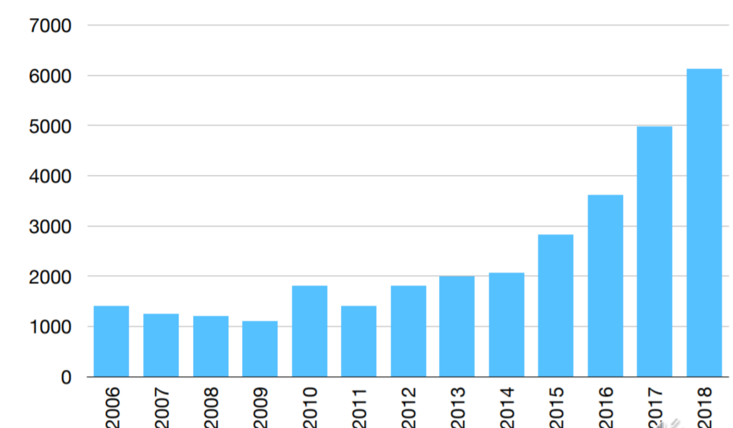
https://www.ft.com/content/0a879bec-48bd-11e8-8c77-ff51caedcde6, financial times, Oct. 4th

ICCV 2017

the most used words in the title of all submitted papers



CVPR Attendance



My perspective on what (supervised) “Deep Learning” is: A fancy word for function approximation

Assume there is an unknown function G that maps some kind of input data x to some kind of desired output y.

Space of all images

G

<u>Answer to the question if the image shows a giraffe</u>

NO!

YES!

Deep Learning

My perspective on what (supervised) “Deep Learning” is:
A fancy word for function approximation

Assume there is an unknown function G that maps some kind of input data x to some kind of desired output y.

Assume we are given some evaluations of this (unknown) function G. This is what we ill call ***training data***!

Giraffe

No giraffe

No giraffe

No giraffe

Giraffe

$G\Bigg($  $\Bigg) = 0,\quad G\Bigg($  $\Bigg) = 1$

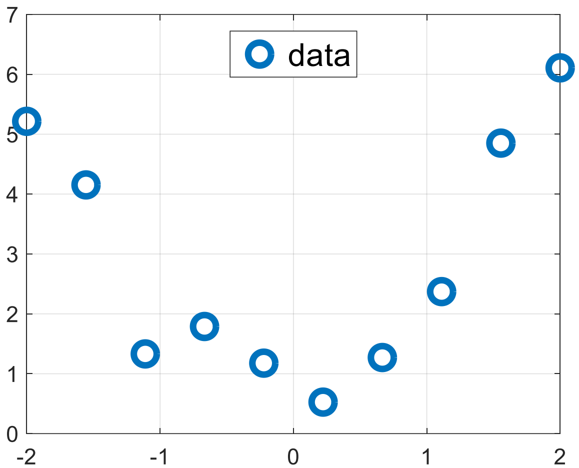
My perspective on what (supervised) "Deep Learning" is: A fancy word for function approximation

Assume there is an unknown function G that maps some kind of input data x to some kind of desired output y.

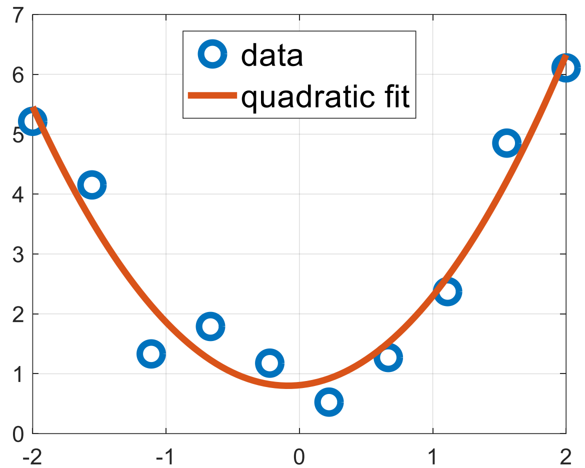
Assume we are given some evaluations of this (unknown) function G. This is what we will call ***training data***!

1. Choose a parameterized function $\mathcal{N}(x;\theta)$ in the hope that for the right choice of parameters θ it approximates the unknown function G well. We call $\mathcal{N}$ the ***network***, and sometimes refer to θ as the ***weights***.
2. Try to determine suitable weights θ in such a way that $\mathcal{N}(x_i;\theta) \approx y_i$ holds for all examples (x_i, y_i) from your training data set. This is referred to as ***training the network***.
3. Make try to ensure that both, the architecture as well as the training are chosen in such as way that the network makes good predictions during inference, i.e. on previously unseen data x: $\mathcal{N}(x;\theta) \approx G(x)$. We refer to this property as ***generalization***.

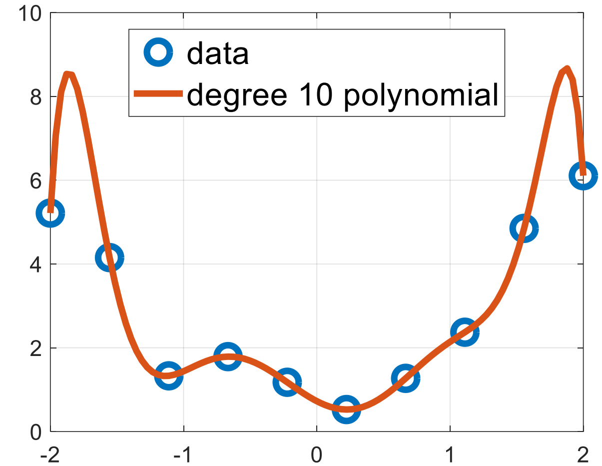
Shallow Example



Training data (x,y)



$$\mathcal{N}(x;\theta) = \theta_1 x^2 + \theta_2 x + \theta_3$$



$$\mathcal{N}(x;\theta) = \sum_{i=0}^{10} \theta_i x^i$$

Depending on the underlying function G, one or the other choice might be better!

This is a very simple 1d example! The power of deep learning, and the reason it receives a lot of attention are that similar concepts seem to work extremely well for incredibly complex functions G!!

Impressive Deep Examples

Predicting the sound objects make when you hit them:
https://www.youtube.com/watch?v=0FW99AQmMc8

Lip-synchronization from audio:
https://www.youtube.com/watch?v=9Yq67CjDqvw&t=268s

Lip-reading:

https://www.youtube.com/watch?v=5aogzAUPilE
https://www.youtube.com/watch?v=fa5QGremQf8&t=4s

Video reenactment:

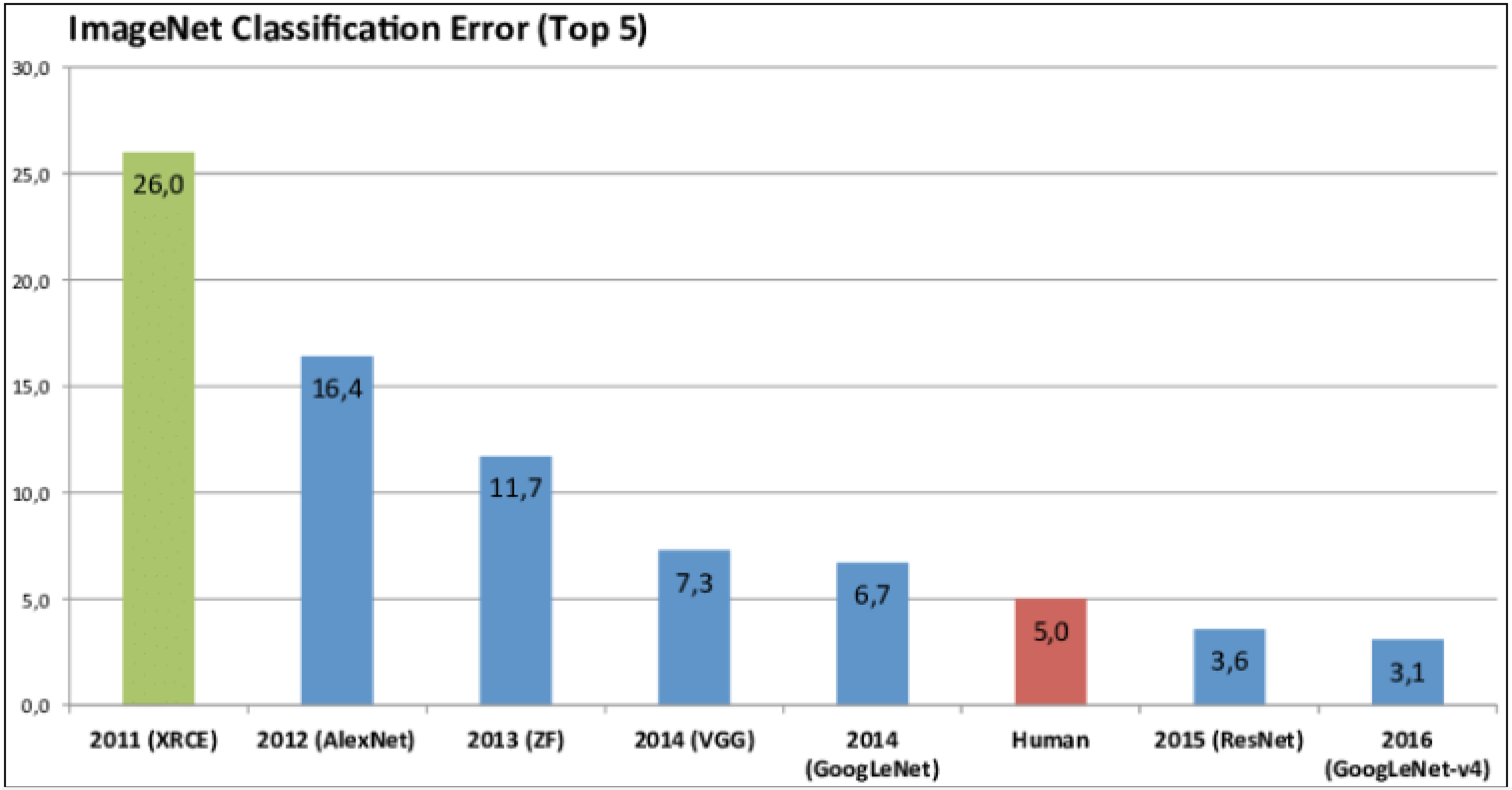
https://www.youtube.com/watch?v=qc5P2bvfl44

Image inpainting:
https://www.youtube.com/watch?v=gg0F5JjKmhA

How did we get there?

Milestones in the Development of Neural Networks

Today, deep networks are (partially) beating humans at image recognition!



https://www.embedded-vision.com/industry-analysis/blog/deep-learning-five-and-half-minutes

This course

Regression and classification using *fully connected* networks

- Learn main principles of deeply nested network architectures
- Implement fully connected networks yourself using NumPy
- Write your own optimization algorithm for training such networks
- Learn how to validate and test your performance

Advanced network architectures using PyTorch

- Learn how to work with images using *convolutional neural networks (CNNs)*
- Weight initialization, self-normalization, and skip-connections for improved training
- Regularization, early-stopping, dropout, and data augmentation for improved generalization

Your own project in your area of interest

- Apply your knowledge in a miniproject towards the end of the course