

Deep Learning

Lecture 1 – Introduction

Prof. Dr.-Ing. Andreas Geiger

Autonomous Vision Group

University of Tübingen

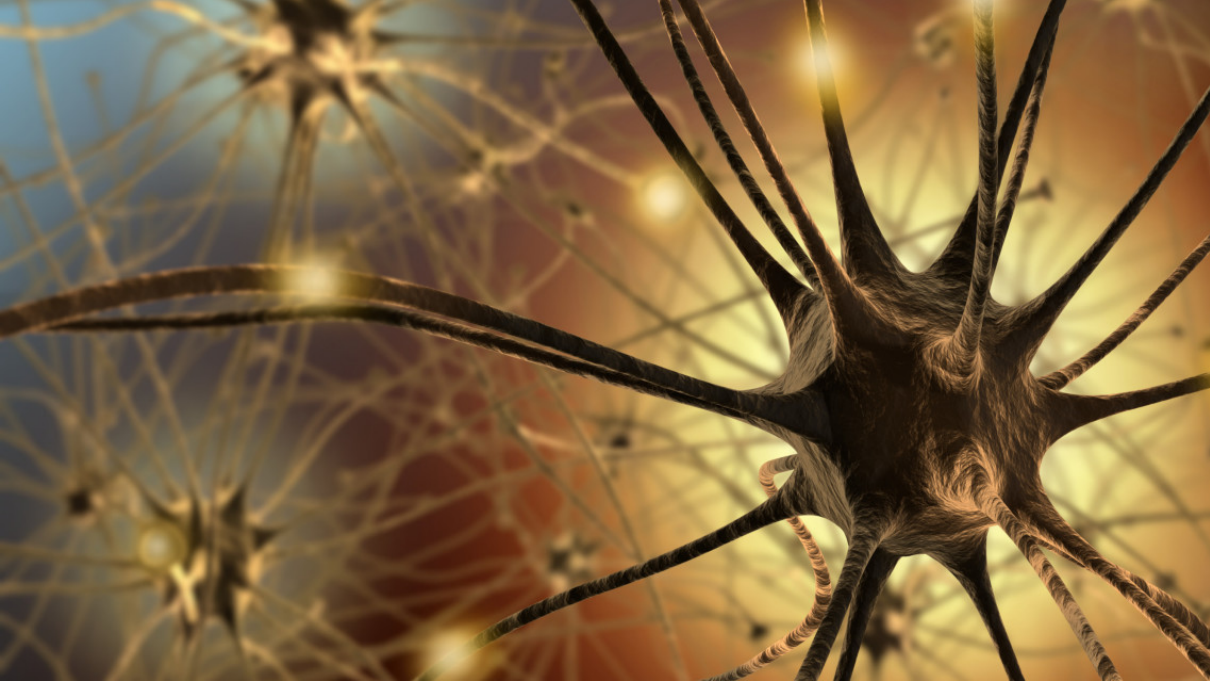
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN



Tübingen AI Center
tuebingen.ai



e l i a s
European Laboratory for Learning and Intelligent Systems

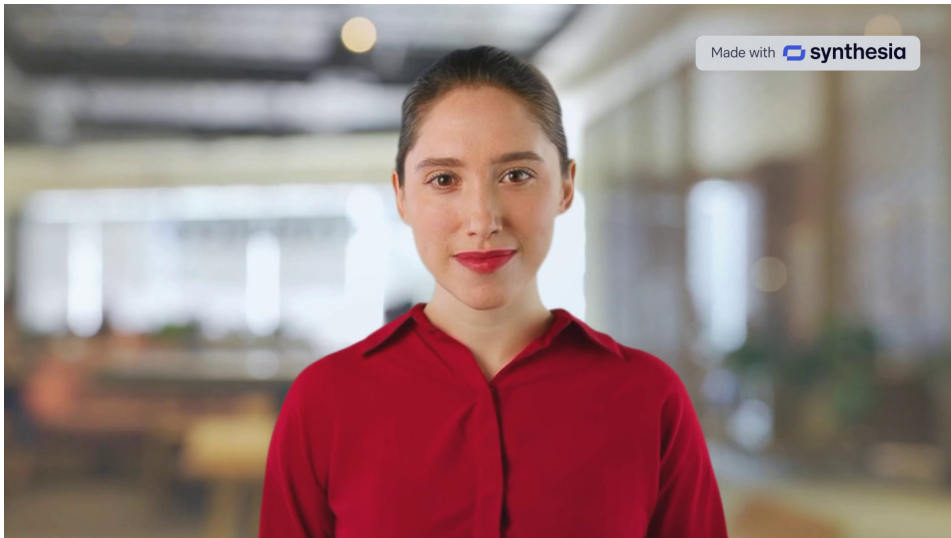




“Deep learning is just a buzzword for neural nets, and neural nets are just a stack of matrix-vector multiplications, interleaved with some non-linearities. No magic there.”

Ronan Collobert (Torch), 2011





Deep Learning for Natural Language



OpenAI ChatGPT **4.0**

“Write a joke about deep neural networks that the students in my deep learning class would find hilarious.”

Deep Learning for Audio

desc	MusicGen	MusicGen Stereo	MusicLM	Riffusion	Musai
Pop dance track with catchy melodies, tropical percussion, and upbeat rhythms, perfect for the beach	 0:00  	 0:00 / 0:30  	 0:00 / 0:19  	 0:00 / 0:30  	 0:00 / 0:30  
A grand orchestral arrangement with thunderous percussion, epic brass fanfares, and soaring strings, creating a cinematic atmosphere fit for a heroic battle.	 0:00 / 0:30  	 0:00 / 0:30  	 0:00 / 0:19  	 0:00 / 0:30  	 0:00 / 0:30  
classic reggae track with an electronic guitar solo	 0:00 / 0:30  	 0:00 / 0:30  	 0:00 / 0:19  	 0:00 / 0:30  	 0:00 / 0:30  
earthy tones, environmentally conscious, ukulele-infused, harmonic, breezy, easygoing, organic instrumentation, gentle grooves	 0:00 / 0:30  	 0:00 / 0:30  	 0:00 / 0:19  	 0:00 / 0:30  	 0:00 / 0:30  
lofi slow bpm electro chill with organic samples	 0:00 / 0:30  	 0:00 / 0:30  	 0:00 / 0:19  	 0:00 / 0:30  	 0:00 / 0:30  
drum and bass beat with intense percussions	 0:00 / 0:30  	 0:00 / 0:30  	 0:00 / 0:19  	 0:00 / 0:30  	 0:00 / 0:30  
A dynamic blend of hip-hop and orchestral elements, with sweeping strings and brass, evoking the vibrant energy of the city.	 0:00 / 0:30  	 0:00 / 0:30  	 0:00 / 0:19  	 0:00 / 0:30  	 0:00 / 0:30  
violins and synths that inspire awe at the finiteness of life and the universe	 0:00 / 0:30  	 0:00 / 0:30  	 0:00 / 0:19  	 0:00 / 0:30  	 0:00 / 0:30  
80s electronic track with melodic synthesizers, catchy beat and groovy bass	 0:00 / 0:30  	 0:00 / 0:30  	 0:00 / 0:19  	 0:00 / 0:30  	 0:00 / 0:30  

HDT: Hierarchical Document Transformer

Haoyu He^{1,2*}, Markus Flicke^{1,2*}, Jan Buchmann³, Iryna Gurevych³, Andreas Geiger^{1,2}

¹ University of Tübingen ² Tübingen AI Center

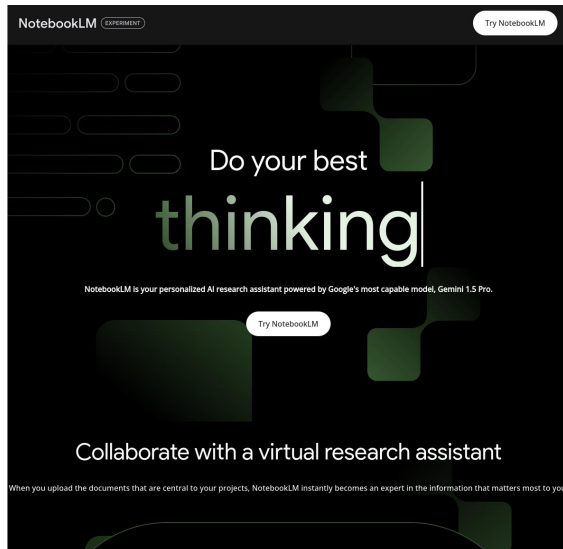
³ Technical University of Darmstadt and Hessian Center for AI (hessian.AI)

Abstract

In this paper, we propose the Hierarchical Document Transformer (HDT), a novel sparse Transformer architecture tailored for structured hierarchical documents. Such documents are extremely important in numerous domains, including science, law or medicine. However, most existing solutions are inefficient and fail to make use of the structure inherent to documents. HDT exploits document structure by introducing auxiliary anchor tokens and redesigning the attention mechanism into a sparse multi-level hierarchy. This approach facilitates information exchange between tokens at different levels while maintaining sparsity, thereby enhancing computational and memory efficiency while exploiting the document structure as an inductive bias. We address the technical challenge of implementing HDT's sample-dependent hierarchical attention pattern by developing a novel sparse attention kernel that considers the hierarchical structure of documents. As demonstrated by our experiments, utilizing structural information present in documents leads to faster convergence, higher sample efficiency and better performance on downstream tasks.

1 Introduction

Many natural language processing tasks including summarization and question answering require language models to encode long structured documents such as scientific papers or Wikipedia articles into meaningful representations. Since 2017, attention-based Transformer architectures Vaswani et al. (2017) have established themselves as the dominant modeling paradigm, demonstrating state-of-the-art performance on numerous NLP tasks. While dense attention yields context-rich representations, its computational complexity grows quadratically with the input. This is problematic when processing large documents, in particular on resource constrained hardware such as consumer GPUs.



Deep Learning for Images and Video

The screenshot shows a web browser interface with a navigation bar at the top containing links like 'Trending', 'Bookmarks', 'Collections', 'Interactions', 'Semantic Search', and 'Conferences'. Below the navigation bar, there is a red-bordered box containing a hiring announcement. The text in the box reads: 'www.scholar-inbox.com is hiring students (Hiwis)! If you have good coding skills and experience in Python, Flask, React, SQL, Celery, send your application (CV+skills+transcripts) to: a.geiger@uni-tuebingen.de'. Below the text, there is a row of seven circular profile pictures of researchers, each with their name and affiliation (University of Tübingen) listed underneath. The researchers are: Andreas Geiger, Markus Fiecko, Vitalii Protserenko, Madhav Iyer, Haozu He, Ilika Shukun, and Boris Kargi. Below the row of profile pictures, there is a paragraph of text describing a research project, mentioning 'Dust-GS' and '3DGS'.

www.scholar-inbox.com is hiring students (Hiwis)!
If you have good coding skills and experience in Python, Flask, React, SQL, Celery, send your application (CV+skills+transcripts) to: a.geiger@uni-tuebingen.de

Below the text, there are seven circular profile pictures of researchers, each with their name and affiliation (University of Tübingen) listed underneath:

- Andreas Geiger
- Markus Fiecko
- Vitalii Protserenko
- Madhav Iyer
- Haozu He
- Ilika Shukun
- Boris Kargi

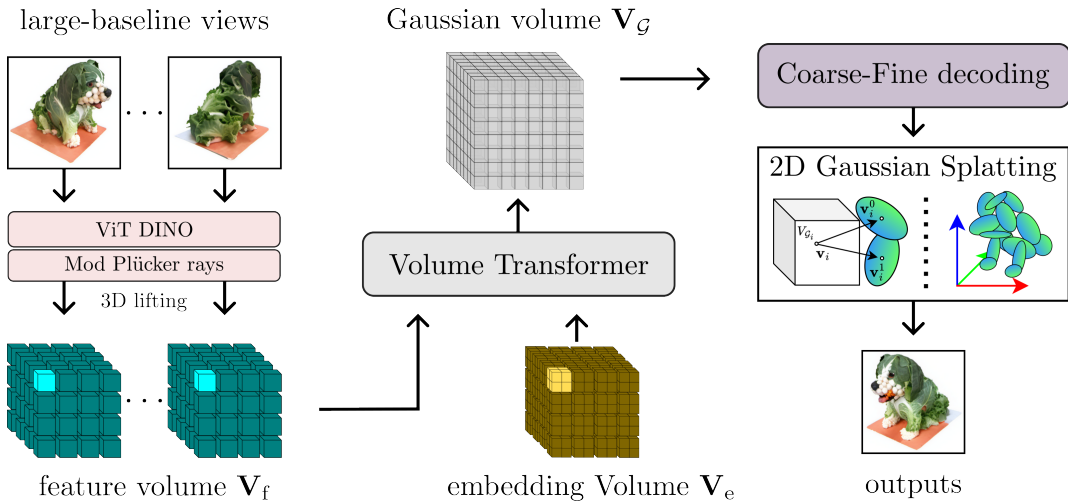
Below the row of profile pictures, there is a paragraph of text describing a research project, mentioning 'Dust-GS' and '3DGS'.



“Minecraft scenario: A researcher with a PhD hat sitting on a lot of very very large and high piles of research papers scattered across the ground of a university park, desperate and wondering which to read first. Typical minecraft scenario, with beautiful modern university buildings in Minecraft style. In the background a modern building with a big sign “Scholar Inbox” above the entrance.”

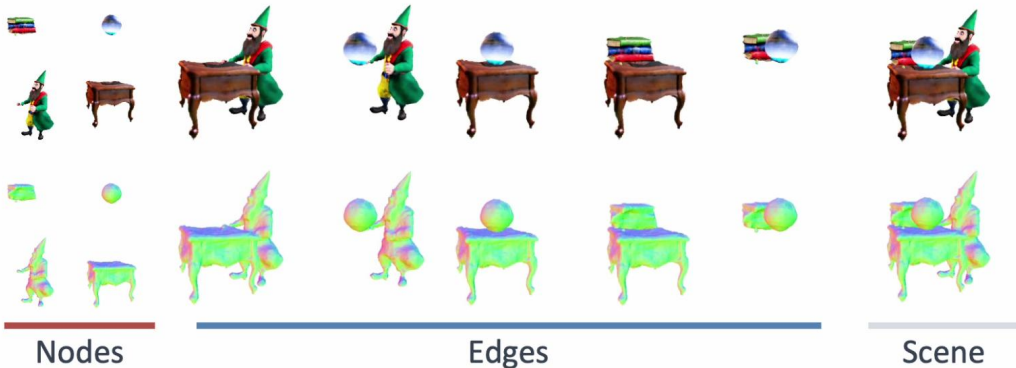


Deep Learning for 3D

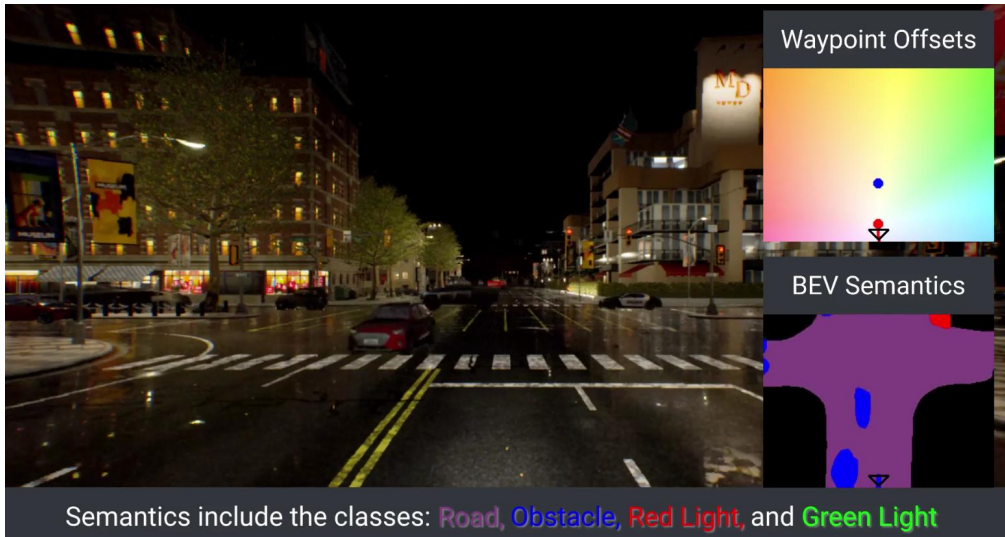




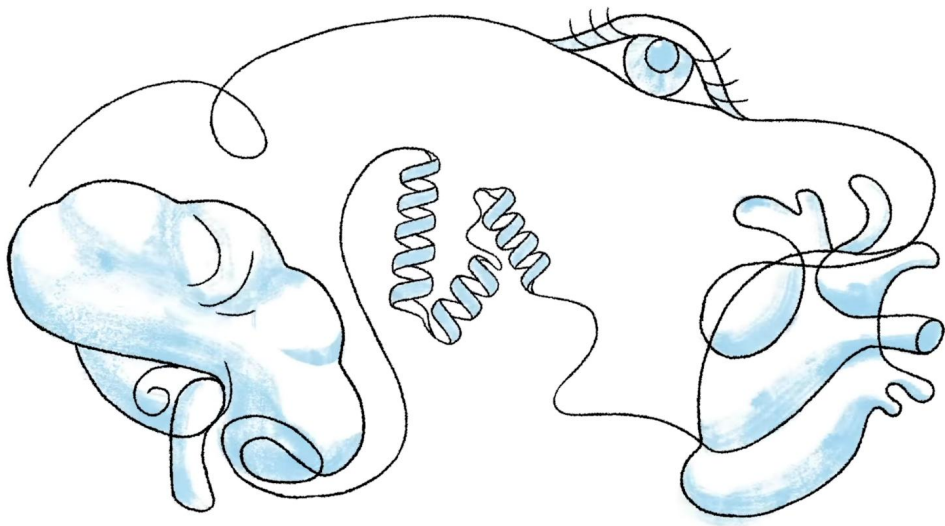
GraphDreamer

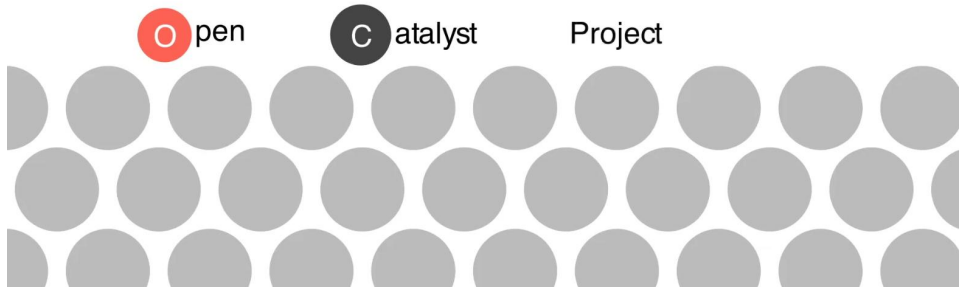


Deep Learning for Self-Driving



Deep Learning for Science





Agenda

1.1 Organization

1.2 History of Deep Learning

1.3 Machine Learning Basics

1.1

Organization

Team



Prof. Dr.-Ing.
Andreas Geiger



Bozidar
Antic



Christian
Reiser



Anpei
Chen



Daniel
Dauner



Takeru
Miyato

Lectures offered by our research group:

- ▶ Deep Learning (**1st semester MSc**, basic topics)
- ▶ Computer Vision (2nd semester MSc, basic & advanced topics)
- ▶ Self-Driving Cars (3rd semester MSc, **every other year**, basic & advanced topics)

Contents

Goal: Students gain an understanding of the theoretical and practical concepts of deep neural networks including, optimization, inference, architectures and applications. After this course, students should be able to develop and train deep neural networks, reproduce research results and conduct original research.

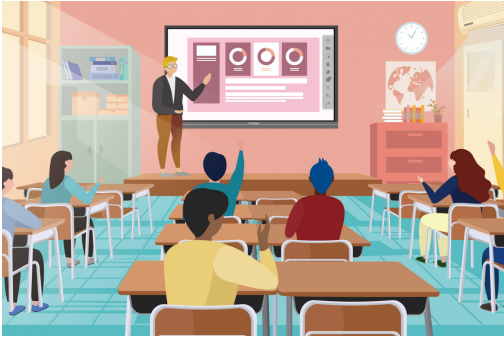
- ▶ History of deep learning
- ▶ Linear/logistic regression
- ▶ Multi-layer perceptrons
- ▶ Backpropagation
- ▶ Loss and Activation Functions
- ▶ Optimization and Regularization
- ▶ Convolutional Neural Networks
- ▶ Sequence Models
- ▶ Natural Language Processing
- ▶ Graph Neural Networks
- ▶ Autoencoders
- ▶ Generative Adversarial Networks

Related Lectures

- ▶ Introduction to ML (BSc): 4 lectures on DL basics, lighter treatment
- ▶ Mathematics for ML: Repetition of BSc math and probability / statistics
- ▶ Mathematics for DL (online): 4h, only the basics necessary to follow this course
- ▶ Data Literacy: Hands-on data science, classic tools, complementary to DL
- ▶ Statistical ML: Little overlap (e.g., PCA), theoretical focus
- ▶ Probabilistic ML: 3 lectures on DL, but probabilistic and more theoretical focus
- ▶ Reinforcement Learning: Builds upon deep learning, modern RL applications
- ▶ Computer Vision: Builds upon deep learning, modern CV applications
- ▶ Self-Driving Cars: Builds upon deep learning, modern SDC applications

Flipped Classroom

Flipped Classroom



Regular Classroom

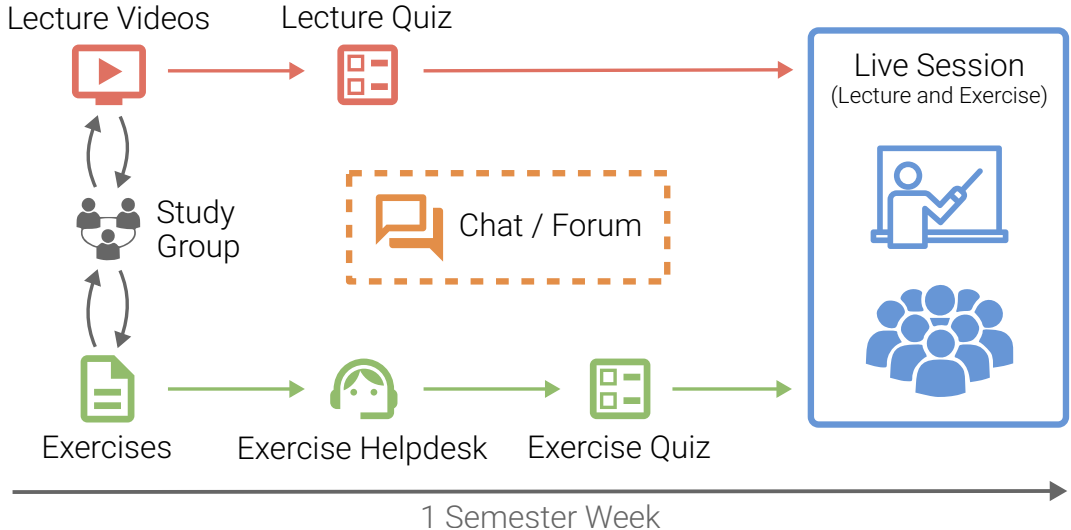
- ▶ Lecturer “reads” lecture in class
- ▶ Students digest content at home
- ▶ Little “quality time”, no fun



Flipped Classroom

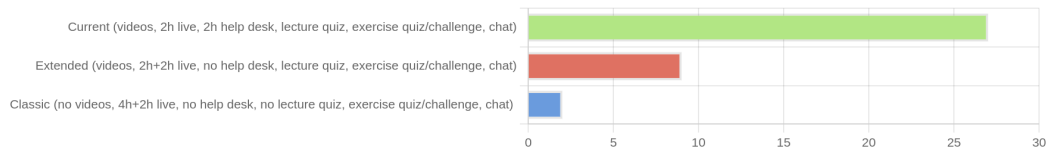
- ▶ Students watch lectures at home
- ▶ Content discussed during lecture
- ▶ Maximize interaction time

Flipped Classroom



Self-Driving Cars (ML-4340) | Winter 23/24 | Survey 1

Which lecture format do you prefer for AVG lectures (DL, CV, SDC)?



Organization

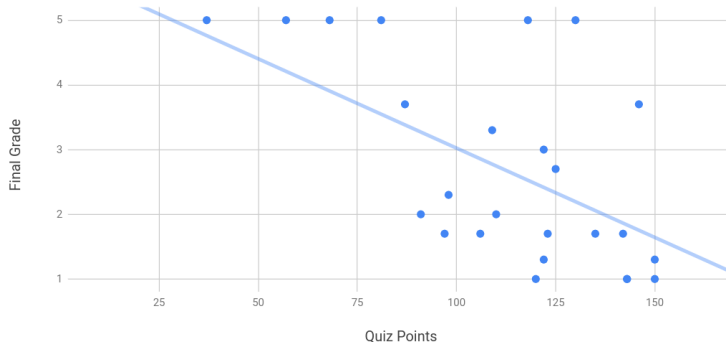
Organization

- ▶ **Lecture videos** provided in advance via the course website
 - ▶ Watch lecture before live session! (mark your calendar!)
 - ▶ Ask (many) questions via the quiz server chat room (dare to ask!)
- ▶ **Live sessions** (Wednesdays: 14:15-16:00, HS 21 Kupferbau)
 - ▶ Lecture Q&A and examples, exercise Q&A and task solving
 - ▶ Will not be hybrid and or recorded
- ▶ **No tutorials** in small groups, instead we provide:
 - ▶ Practice via **exercises** with quizzes and solutions (solve them!)
 - ▶ Plenary feedback via **live sessions** (join us!)
 - ▶ Individual help via weekly **helpdesk** (make use of it!)
 - ▶ Fast (<24h) feedback via **chat** (ask and reply!)
- ▶ **Exam:** Written (bonus upon quiz participation)
- ▶ **Course Website:** <https://uni-tuebingen.de/de/175884>
 - ▶ **Register** to ILIAS and Lecture Quiz Server today!

Exercises

- ▶ Exercises are offered every 2 weeks, **6 assignments** in total
- ▶ Exercises deepen the understanding with **pen & paper and coding tasks**
- ▶ Solving the exercises is not a prerequisite to participate in the exam
- ▶ Exercise tasks are **highly relevant** to the exam $\Rightarrow$ use this opportunity

[Slides / Exercises](#)



Exercises

- ▶ **Problems** handed out, introduced and discussed in the live sessions
- ▶ Can be conducted in groups of **up to 4 students**
- ▶ We offer a **helpdesk** every week via zoom to answer your questions
- ▶ At the end of each exercise, we will hand out and discuss the **solutions**
- ▶ But how do I stay motivated to watch the lectures and solve the exercises?

AVG Lecture Quiz Server

- ▶ We provide **quiz questions** to students during lectures 2-12 and exercises 1-6
- ▶ Collect exam **bonus points**
- ▶ Collect AI generated **Pokemon**
- ▶ Answers may **not** be **shared**
- ▶ Participation is **voluntary**
- ▶ Past quizzes only **visible to participants**
- ▶ Deadline: **Mondays, 9pm**
- ▶ **Register today**

Edit Assignment

Each correct answer yields +1 point. Each incorrect answer yields -1 point. Meaningful student questions yield +1 point. If you wish, you may answer only a subset of the questions. The minimum overall score that can be attained in this assignment is 0 points.

Computer Vision | Summer 22 | Lecture 2

Opening: 21 Apr. 2022 (21:00) - Closing: 28 Apr. 2022 (21:00)

1. What is an ideal point?

- ☐ An inhomogeneous point
- ☐ A point at infinity

[Clear](#)

2. What is the dimensionality of a 4D point represented in homogeneous coordinates?

3. How many degrees of freedom has a 3D line?

4. Affine transformations preserve

- ☐ parallel lines
- ☐ lengths
- ☐ angles

[Clear](#)

5. The diameter of a telecentric lens should be

- ☐ about as large as the object/scene
- ☐ significantly larger than the object
- ☐ significantly smaller than the object

[Clear](#)

6. Increasing the depth of a 3D point in perspective projection

- ☐ increases the (x,y) pixel coordinates
- ☐ decreases the (x,y) pixel coordinates
- ☐ does not affect the (x,y) pixel coordinates

[Clear](#)

7. Perspective projection

- ☐ is linear when using homogeneous coordinates
- ☐ is linear when using inhomogeneous coordinates

[Clear](#)

8. Assume a point light source. Diffuse materials

- ☐ scatter light uniformly in all directions at any given surface point
- ☐ scatter more light into the mirror direction than into other directions at any given surface point
- ☐ scatter the same amount of light independent of the surface normal

[Clear](#)





OpenAI ChatGPT 4.0

- ▶ **Spending time** on the exercises is the **most effective** way to prepare for the exam
- ▶ Using **ChatGPT** or solutions from previous years will effectively **harm you**
- ▶ ChatGPT will not help you in the exam: you need to deeply understand the topic
- ▶ **Simple rule to success:** Don't use ChatGPT, don't copy solution
- ▶ The main goal of our quizzes is to **motivate you** to engage with the materials

Exercise Helpdesk

- ▶ We offer a **weekly helpdesk** where our TAs provide individual support:
 - ▶ Mo, 10-12, MvL6 (3rd floor)
 - ▶ Fri, 14-16, Zoom
- ▶ **Just show up** (no registration)
- ▶ Ask **any question** about the exercise
- ▶ Share your screen to show a problem
- ▶ Make use of this opportunity
- ▶ Start working on your exercise **early**

Chat



Andreas Geiger | 14 Apr. 2023 (17:16)

This is the chatroom for our lecture.



Andreas Geiger | 14 Apr. 2023 (17:18)

It supports multiple threads which can be associated with specific lectures or quiz questions.
It also supports **markdown** and latex:



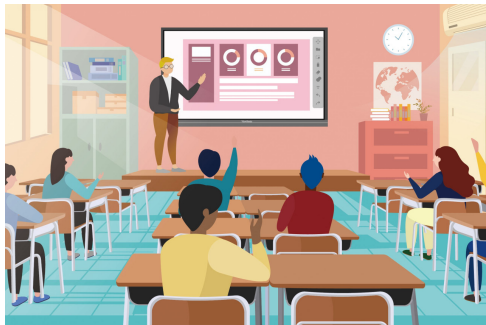
$$x = \int_0^1 \frac{5}{20} x \, dx$$

Send Message (Ctrl+Enter)

► **Ask** (many) questions! **Answer** questions if you think you know the answer!

Live Sessions

1. **Lecture Q&A** (50 min)
 - ▶ Live lecture and example walk throughs
 - ▶ Discussion of quiz questions
 - ▶ Discussion of student questions
2. Break (5 min)
3. **Exercise Q&A** (50 min)
 - ▶ Exercise introduction
 - ▶ Exercise live solving and Q&A
 - ▶ Start solving before joining!



What will the exam look like?

- ▶ **Written** main and make-up exam
- ▶ You may **choose freely** (but no 3rd exam)
- ▶ Registration via Quiz Server
- ▶ Only **pen and ruler** allowed (no notes)
- ▶ Duration: 90 minutes (can be solved in 60)
- ▶ 5 Tasks, 10 points each, 25 points will pass
- ▶ Bonus added only to passed exams
- ▶ Tasks cover both **lectures** and **exercises**
- ▶ Mix of knowledge, calculation, multiple choice
- ▶ **Old exams** available on ILIAS



Deep Learning

Exam, Winter 2020/2021

Tübingen, March 5, 2021

Student Number:	
Seat Number:	
First Name:	
Last Name:	
Date of Birth:	
Program of Study (e.g., "Informatik"):	
Master ☐ Bachelor ☐	

Task	1	2	3	4	5	Σ
Maximal Score	10	10	10	10	10	50
Attained Score						
Correction						

Remarks:

- Leave your bag and jacket at the front desk. Only take your **pen, ruler, student ID** with you.
- Notes, books, printouts, smartphones or calculators are not permitted.
- Have your student ID ready for inspection.
- Write your full name at the top of **every** page.
- Use the space provided for your answers. You can use the back side if you need more space.
- Additional paper can be obtained at the front desk.
- Use black or blue pen. Pencil or red pens are not allowed and will lead to a score of 0.
- A total score of **25** is sufficient for passing the exam. The duration of the exam is **90 minutes**.

Good Luck!

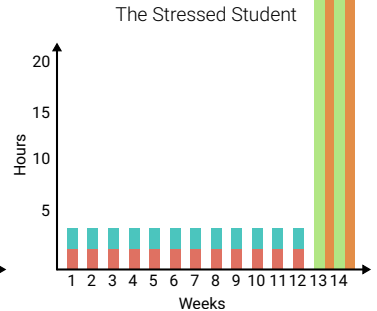
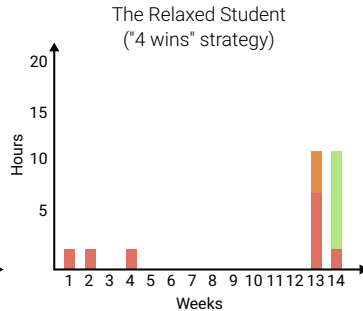
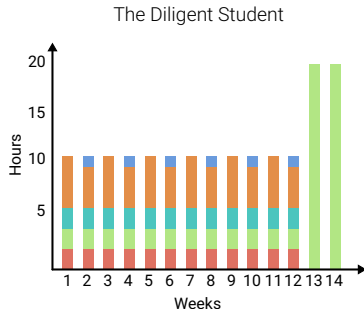
Bonus Points

- ▶ **11 lecture quizzes** (10 points per quiz)
- ▶ **6 exercise quizzes** (10 points per quiz)
- ▶ 110 lec. points + 60 ex. points in total
- ▶ Bonus Points = $\left\lfloor \frac{\text{lec. points}}{36} \right\rfloor + \left\lfloor \frac{\text{ex. points}}{20} \right\rfloor$
- ▶ Maximum: $3 + 3 = 6$ bonus points
- ▶ Bonus added only to **passed exams**

Work Ethics

This lecture has **6 ECTS**, corresponding to a total workload of **180 hours** (MHB)

Lecture Videos Self-Study Live Sessions Exercises Quizzes



Course Materials and Prerequisites

Course Materials

Books:

- ▶ Goodfellow, Bengio, Courville: Deep Learning
<http://www.deeplearningbook.org>
- ▶ Bishop: Pattern Recognition and Machine Learning
<http://www.springer.com/gp/book/9780387310732>
- ▶ Zhang, Lipton, Li, Smola: Dive into Deep Learning
<http://d2l.ai>
- ▶ Deisenroth, Faisal, Ong: Mathematics for Machine Learning
<https://mml-book.github.io>
- ▶ Petersen, Pedersen: The Matrix Cookbook
http://cs.toronto.edu/~bonner/courses/2018s/csc338/matrix_cookbook.pdf
- ▶ Inoffical lecture notes written by students in winter 2020/21
<https://uni-tuebingen.de/de/175884>

Course Materials

Courses:

- ▶ McAllester (TTI-C): Fundamentals of Deep Learning
<http://mcallester.github.io/ttic-31230/Fall12020/>
- ▶ Kolterl, Chen (CMU): Deep Learning Systems
<https://dlsyscourse.org/lectures/>
- ▶ Leal-Taixe, Niessner (TUM): Introduction to Deep Learning
<http://niessner.github.io/I2DL/>
- ▶ Grosse (UoT): Intro to Neural Networks and Machine Learning
http://www.cs.toronto.edu/~rgrosse/courses/csc321_2018/
- ▶ Li (Stanford): Convolutional Neural Networks for Visual Recognition
<http://cs231n.stanford.edu/>
- ▶ Abbeel, Chen, Ho, Srinivas (Berkeley): Deep Unsupervised Learning
<https://sites.google.com/view/berkeley-cs294-158-sp20/home>

Course Materials

Tutorials:

- ▶ The Python Tutorial

<https://docs.python.org/3/tutorial/>

- ▶ NumPy Quickstart

<https://numpy.org/devdocs/user/quickstart.html>

- ▶ PyTorch Tutorial

<https://pytorch.org/tutorials/>

Frameworks / IDEs:

- ▶ Visual Studio Code

<https://code.visualstudio.com/>

- ▶ Google Colab

<https://colab.research.google.com>

Prerequisites

Computer Science:

- ▶ Variables, functions, loops, classes, algorithms

Python:

- ▶ `https://docs.python.org/3/tutorial/`

Math:

- ▶ Linear algebra, probability and information theory. If unsure, have a look at:
Luxburg: [Mathematics for Machine Learning \(Lecture\)](#)
Deisenroth: [Mathematics for Machine Learning \(Book\)](#)
Geiger: [Mathematics for Deep Learning \(4h Micro Lecture\)](#)

Prerequisites

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN



$$\mathbf{A} = \begin{pmatrix} 3 & 1 \\ 2 & 1 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 1 & 2 \\ 3 & 1 \end{pmatrix} \Rightarrow \mathbf{AB} = \begin{pmatrix} 6 & 7 \\ 5 & 5 \end{pmatrix}$$

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{A}^{-1}\mathbf{Ax} = \mathbf{A}^{-1}\mathbf{b}$$

$$\mathbf{I}_N \mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

$$\hat{\mathbf{w}}_{ML} = \underset{\mathbf{w}}{\operatorname{argmax}} \underbrace{\sum_{i=1}^N \log p_{\text{model}}(y_i | \mathbf{x}_i, \mathbf{w})}_{\text{Log-Likelihood}}$$

$$\sum_{i=1}^M \frac{\partial f}{\partial g_i} \frac{dg_i}{dx}$$

$$\mathbf{x}^T \mathbf{y} = (\mathbf{x}^T \mathbf{y})^T = \mathbf{y}^T \mathbf{x} = \mathbf{x}^T \mathbf{y}$$

Math for Deep Learning

Prof. Dr. Andreas Geiger

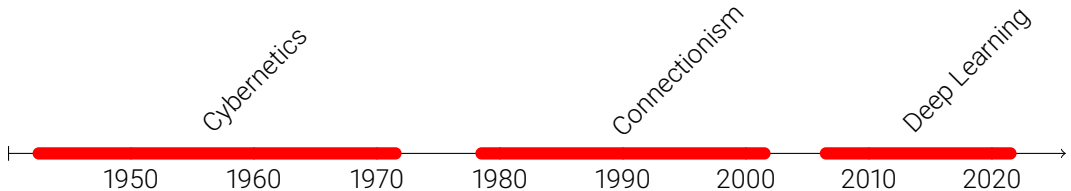
Stretching Break

5 Minutes

A Brief History of Deep Learning

Three waves of development:

- ▶ 1940-1970: “Cybernetics” (Golden Age)
 - ▶ Simple computational models of biological learning, simple learning rules
- ▶ 1980-2000: “Connectionism” (Dark Age)
 - ▶ Intelligent behavior through large number of simple units, Backpropagation
- ▶ 2006-now: “Deep Learning” (Revolution Age)
 - ▶ Deeper networks, larger datasets, more computation, state-of-the-art in many areas



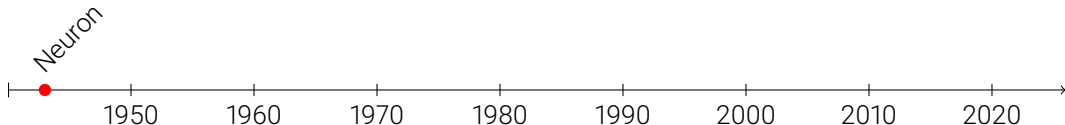
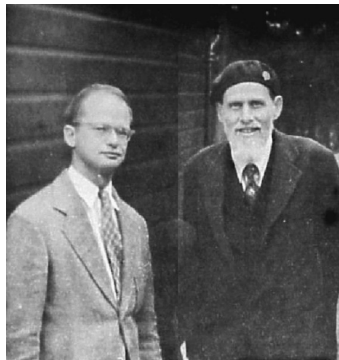
A Brief History of Deep Learning

1943: McCulloch and Pitts

- ▶ Early model for neural activation
- ▶ Linear threshold neuron (binary):

$$f_{\mathbf{w}}(\mathbf{x}) = \begin{cases} +1 & \text{if } \mathbf{w}^T \mathbf{x} \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

- ▶ More powerful than AND/OR gates
- ▶ But no procedure to learn weights



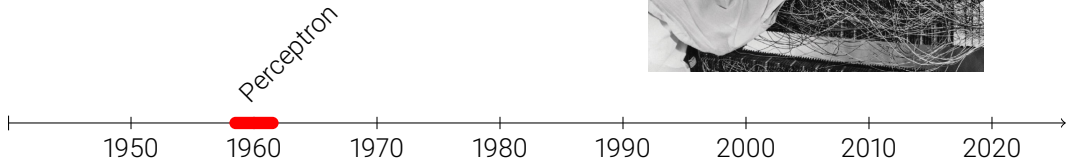
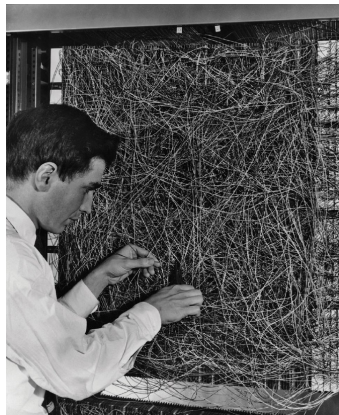
A Brief History of Deep Learning

1958-1962: Rosenblatt's Perceptron

- ▶ First algorithm and implementation to train single linear threshold neuron
- ▶ Optimization of perceptron criterion:

$$\mathcal{L}(\mathbf{w}) = - \sum_{n \in \mathcal{M}} \mathbf{w}^T \mathbf{x}_n y_n$$

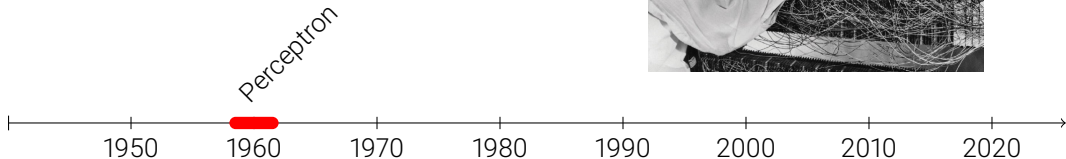
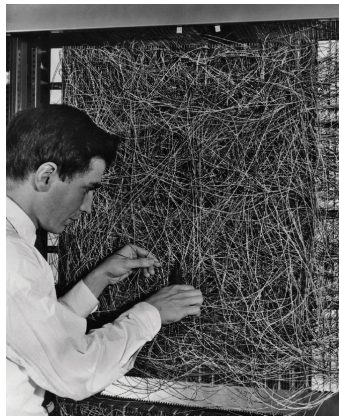
- ▶ Novikoff proved convergence



A Brief History of Deep Learning

1958-1962: Rosenblatt's Perceptron

- ▶ First algorithm and implementation to train single linear threshold neuron
- ▶ Overhyped: Rosenblatt claimed that the perceptron will lead to computers that walk, talk, see, write, reproduce and are conscious of their existence

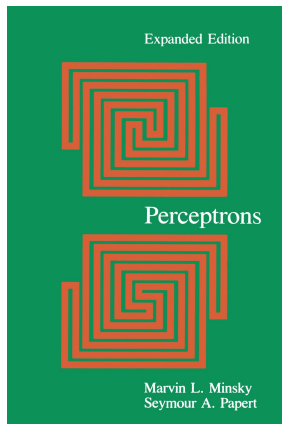
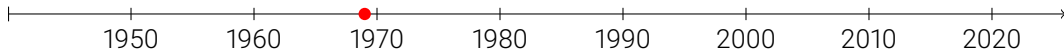


A Brief History of Deep Learning

1969: Minsky and Papert publish book

- ▶ Several discouraging results
- ▶ Showed that single-layer perceptrons cannot solve some very simple problems (XOR problem, counting)
- ▶ Symbolic AI research dominates 70s

Minsky/Papert



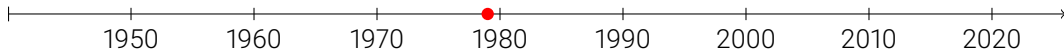
A Brief History of Deep Learning

1979: Fukushima's Neocognitron

- ▶ Inspired by Hubel and Wiesel experiments in the 1950s
- ▶ Study of visual cortex in cats
- ▶ Found that cells are sensitive to orientation of edges but insensitive to their position (simple vs. complex)
- ▶ H&W received Nobel prize in 1981



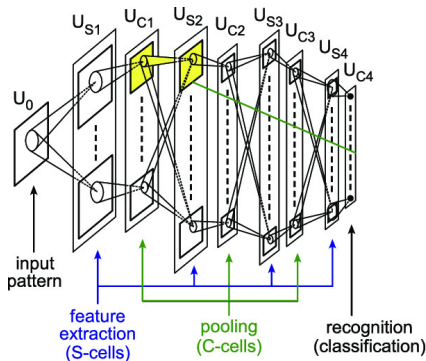
Neocognitron



A Brief History of Deep Learning

1979: Fukushima's Neocognitron

- ▶ Multi-layer processing to create intelligent behavior
- ▶ Simple (S) and complex (C) cells implement convolution and pooling
- ▶ Reinforcement based learning
- ▶ Inspiration for modern ConvNets



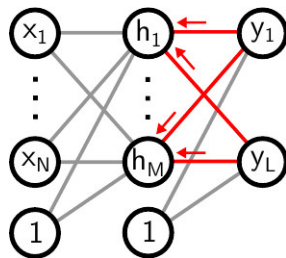
Neocognitron



A Brief History of Deep Learning

1986: Backpropagation Algorithm

- ▶ Efficient calculation of gradients in a deep network wrt. network weights
- ▶ Enables application of gradient based learning to deep networks
- ▶ Known since 1961, but first empirical success in 1986
- ▶ Remains main workhorse today



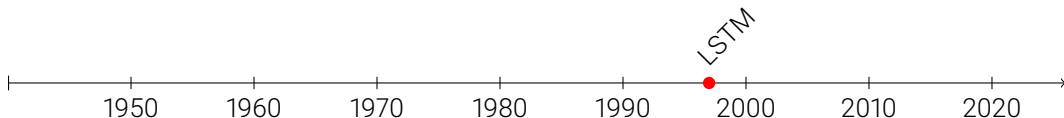
Backpropagation



A Brief History of Deep Learning

1997: Long Short-Term Memory

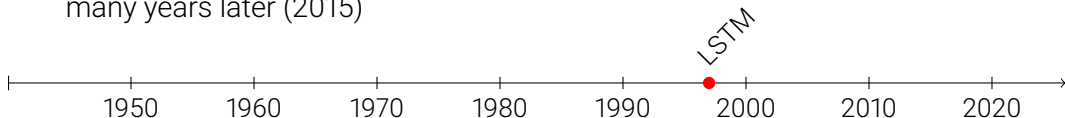
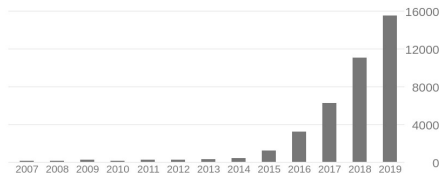
- ▶ In 1991, Hochreiter demonstrated the problem of vanishing/exploding gradients in his Diploma Thesis
- ▶ Led to development of long-short term memory for sequence modeling
- ▶ Uses feedback and forget/keep gate



A Brief History of Deep Learning

1997: Long Short-Term Memory

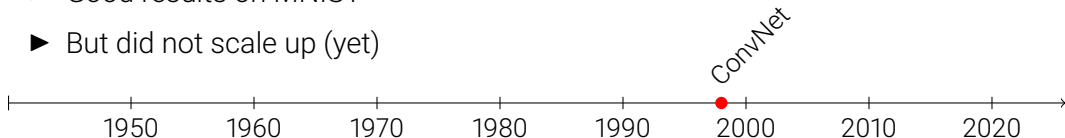
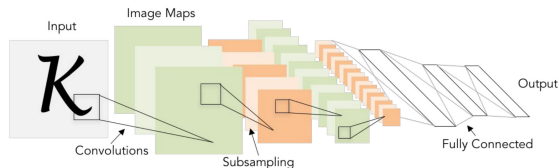
- ▶ In 1991, Hochreiter demonstrated the problem of vanishing/exploding gradients in his Diploma Thesis
- ▶ Led to development of long-short term memory for sequence modeling
- ▶ Uses feedback and forget/keep gate
- ▶ Revolutionized NLP (e.g. at Google) many years later (2015)



A Brief History of Deep Learning

1998: Convolutional Neural Networks

- ▶ Similar to Neocognitron, but trained end-to-end using backpropagation
- ▶ Implements spatial invariance via convolutions and max-pooling
- ▶ Weight sharing reduces parameters
- ▶ Tanh/Softmax activations
- ▶ Good results on MNIST
- ▶ But did not scale up (yet)



A Brief History of Deep Learning

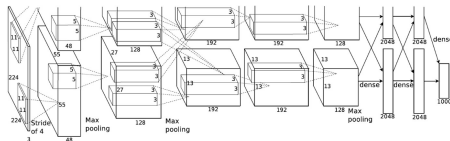
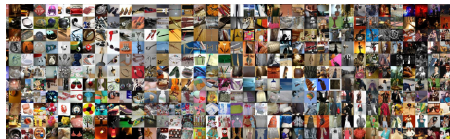
2009-2012: ImageNet and AlexNet

ImageNet

- Recognition benchmark (ILSVRC)
- 1 million annotated images
- 1000 categories

AlexNet

- First neural network to win ILSVRC
via **GPU training, deep models, data**



Image/AlexNet



A Brief History of Deep Learning

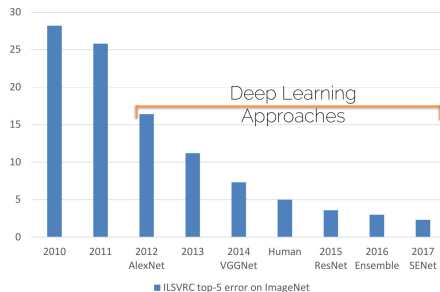
2009-2012: ImageNet and AlexNet

ImageNet

- Recognition benchmark (ILSVRC)
- 1 million annotated images
- 1000 categories

AlexNet

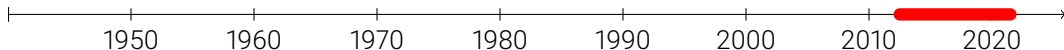
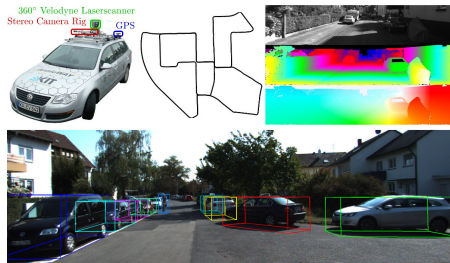
- First neural network to win ILSVRC via **GPU training, deep models, data**
- Sparked deep learning revolution



A Brief History of Deep Learning

2012-now: Golden Age of Datasets

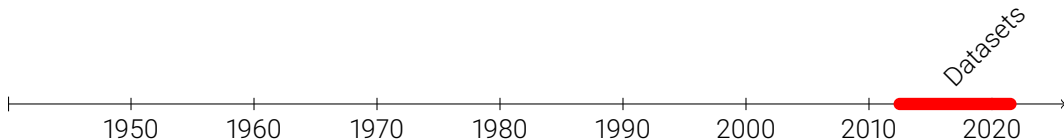
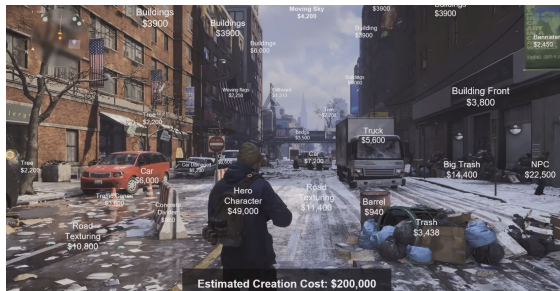
- ▶ KITTI, Cityscapes: Self-driving
- ▶ PASCAL, MS COCO: Recognition
- ▶ ShapeNet, ScanNet: 3D DL
- ▶ GLUE: Language understanding
- ▶ Visual Genome: Vision/Language
- ▶ VisualQA: Question Answering
- ▶ MITOS: Breast cancer



A Brief History of Deep Learning

2012-now: Synthetic Data

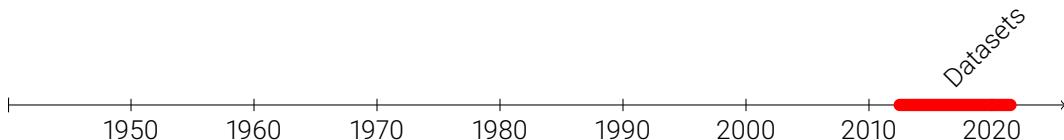
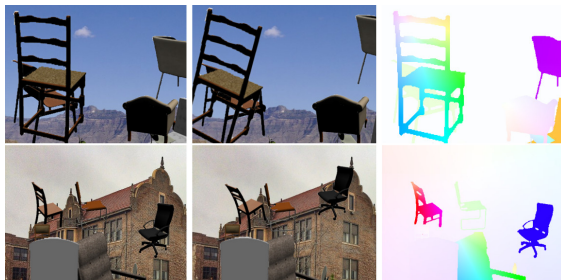
- ▶ Annotating real data is expensive
- ▶ Led to surge of synthetic datasets
- ▶ Creating 3D assets is also costly



A Brief History of Deep Learning

2012-now: Synthetic Data

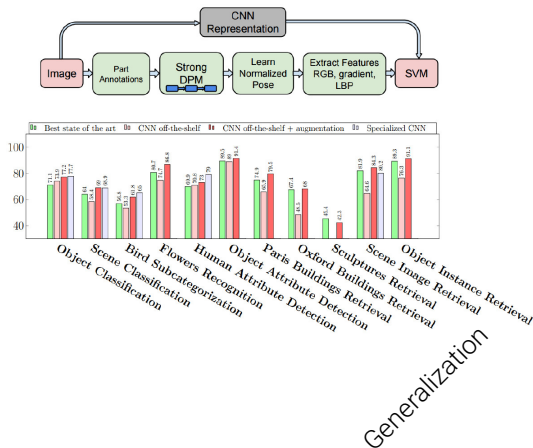
- ▶ Annotating real data is expensive
- ▶ Led to surge of synthetic datasets
- ▶ Creating 3D assets is also costly
- ▶ But even very simple 3D datasets proved tremendously useful for pre-training (e.g., in optical flow)



A Brief History of Deep Learning

2014: Generalization

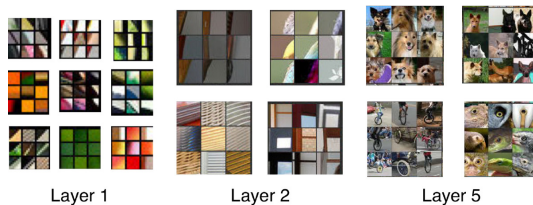
- ▶ Empirical demonstration that deep representations generalize well despite large number of parameters
- ▶ Pre-train CNN on large amounts of data on generic task (e.g., ImageNet)
- ▶ Fine-tune (re-train) only last layers on few data of a new task
- ▶ State-of-the-art performance



A Brief History of Deep Learning

2014: Visualization

- ▶ Goal: provide insights into what the network (black box) has learned
- ▶ Visualized image regions that most strongly activate various neurons at different layers of the network
- ▶ Found that higher levels capture more abstract semantic information



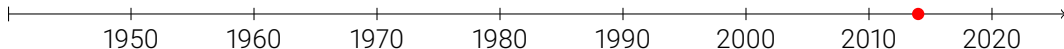
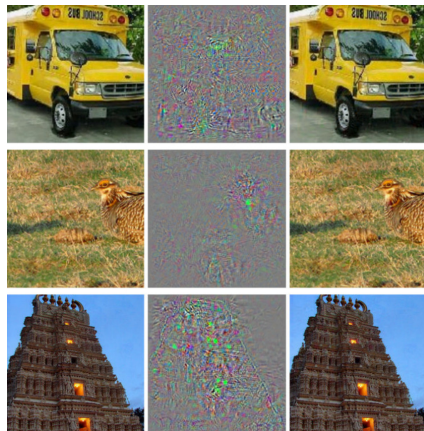
A Brief History of Deep Learning

2014: Adversarial Examples

- ▶ Accurate image classifiers can be fooled by imperceptible changes
- ▶ Adversarial example:

$$x + \underset{\Delta x}{\operatorname{argmin}} \{ \|\Delta x\|_2 : f(x + \Delta x) \neq f(x) \}$$

- ▶ All images classified as “ostrich”

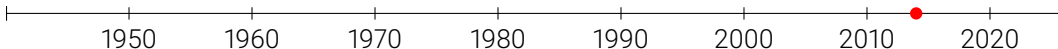


A Brief History of Deep Learning

2014: Domination of Deep Learning

► Machine translation (Seq2Seq)

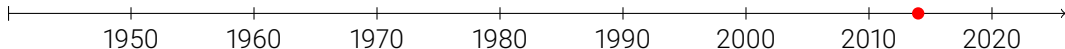
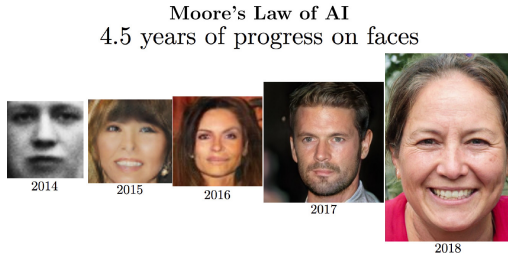
Type	Sentence
Our model	Ulrich UNK , membre du conseil d' administration du constructeur automobile Audi , affirme qu' il s' agit d' une pratique courante depuis des années pour que les téléphones portables puissent être collectés avant les réunions du conseil d' administration afin qu' ils ne soient pas utilisés comme appareils d' écoute à distance .
Truth	Ulrich Hackenberg , membre du conseil d' administration du constructeur automobile Audi , déclare que la collecte des téléphones portables avant les réunions du conseil , afin qu' ils ne puissent pas être utilisés comme appareils d' écoute à distance , est une pratique courante depuis des années .
Our model	" Les téléphones cellulaires , qui sont vraiment une question , non seulement parce qu' ils pourraient potentiellement causer des interférences avec les appareils de navigation , mais nous savons , selon la FCC , qu' ils pourraient interférer avec les tours de téléphone cellulaire lorsqu' ils sont dans l' air " , dit UNK .
Truth	" Les téléphones portables sont véritablement un problème , non seulement parce qu' ils pourraient éventuellement créer des interférences avec les instruments de navigation , mais parce que nous savons , d' après la FCC , qu' ils pourraient perturber les antennes-relais de téléphonie mobile s' ils sont utilisés à bord " , a déclaré Rosenker .
Our model	Avec la crémation , il y a un " sentiment de violence contre le corps d' un être cher " , qui sera " réduit à une pile de cendres " en très peu de temps au lieu d' un processus de décomposition " qui accompagnera les étapes du deuil " .
Truth	Il y a , avec la crémation , " une violence faite au corps aimé " , qui va être " réduit à un tas de cendres " en très peu de temps , et non après un processus de décomposition , qui " accompagnerait les phases du deuil " .



A Brief History of Deep Learning

2014: Domination of Deep Learning

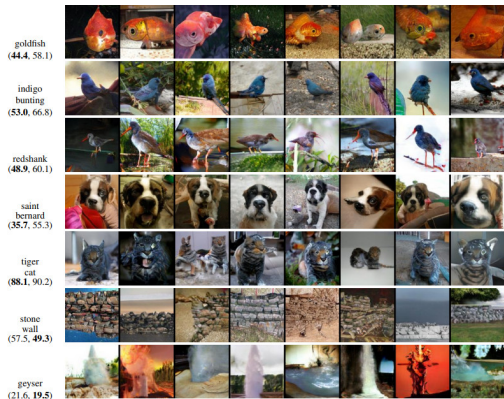
- ▶ Machine translation (Seq2Seq)
- ▶ Deep generative models (VAEs, GANs) produce compelling images



A Brief History of Deep Learning

2014: Domination of Deep Learning

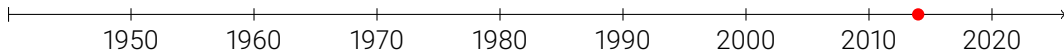
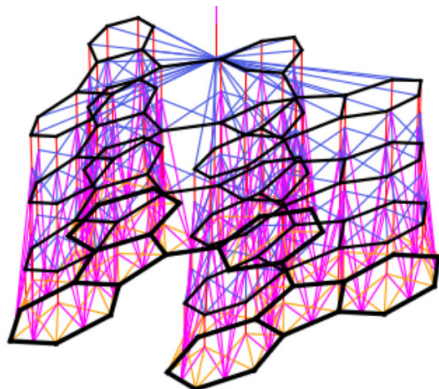
- ▶ Machine translation (Seq2Seq)
- ▶ Deep generative models (VAEs, GANs) produce compelling images



A Brief History of Deep Learning

2014: Domination of Deep Learning

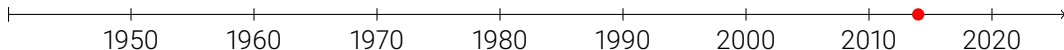
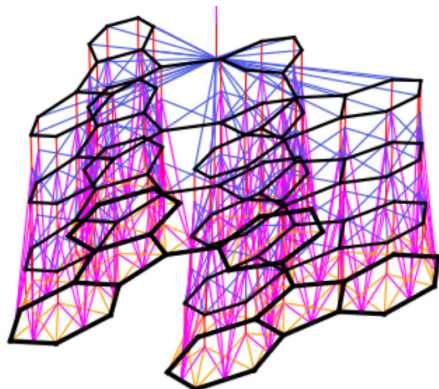
- ▶ Machine translation (Seq2Seq)
- ▶ Deep generative models (VAEs, GANs) produce compelling images
- ▶ Graph Neural Networks (GNNs) revolutionize the prediction of molecular properties



A Brief History of Deep Learning

2014: Domination of Deep Learning

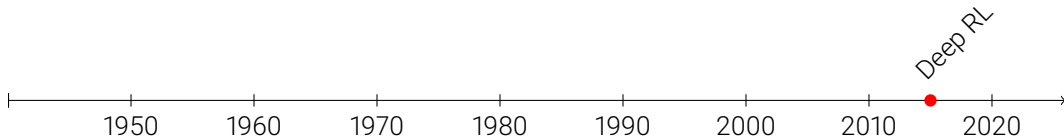
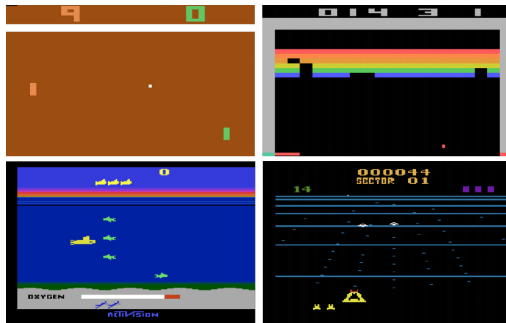
- ▶ Machine translation (Seq2Seq)
- ▶ Deep generative models (VAEs, GANs) produce compelling images
- ▶ Graph Neural Networks (GNNs) revolutionize the prediction of molecular properties
- ▶ Dramatic gains in vision and speech (Moore's Law of AI)



A Brief History of Deep Learning

2015: Deep Reinforcement Learning

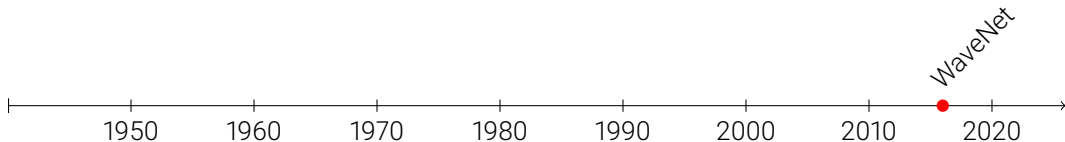
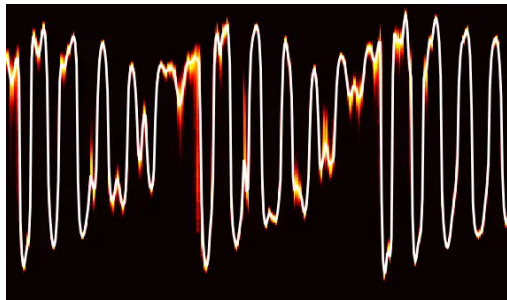
- ▶ Learning a policy (state→action) through random exploration and reward signals (e.g., game score)
- ▶ No other supervision
- ▶ Success on many Atari games
- ▶ But some games remain hard



A Brief History of Deep Learning

2016: WaveNet

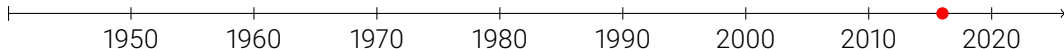
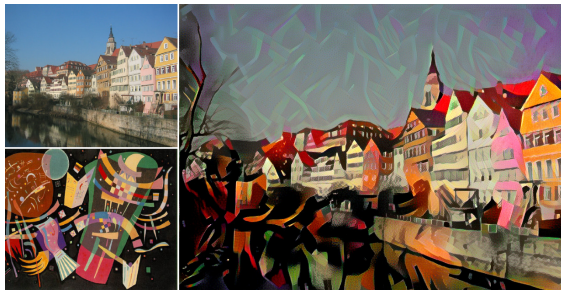
- ▶ Deep generative model of raw audio waveforms
- ▶ Generates **speech** which mimics human voice
- ▶ Generates **music**



A Brief History of Deep Learning

2016: Style Transfer

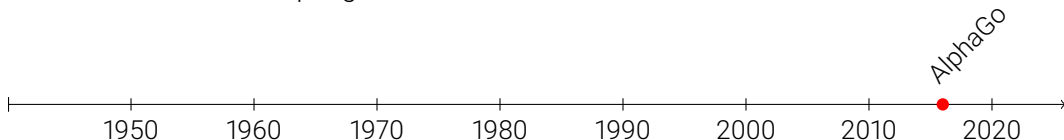
- ▶ Manipulate photograph to adopt style of a another image (painting)
- ▶ Uses deep network pre-trained on ImageNet for disentangling content from style
- ▶ It is fun! Try yourself:
<https://deepart.io/>



A Brief History of Deep Learning

2016: AlphaGo defeats Lee Sedol

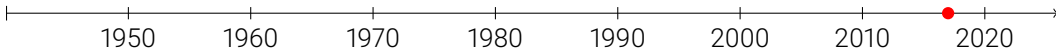
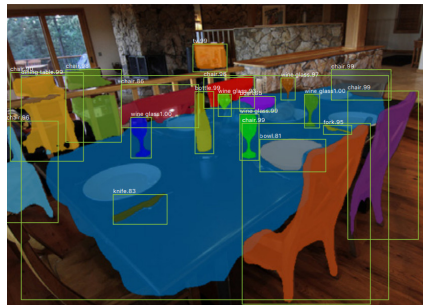
- ▶ Developed by DeepMind
- ▶ Combines deep learning with Monte Carlo tree search
- ▶ First computer program to defeat professional player
- ▶ AlphaZero (2017) learns via self-play and masters multiple games



A Brief History of Deep Learning

2017: Mask R-CNN

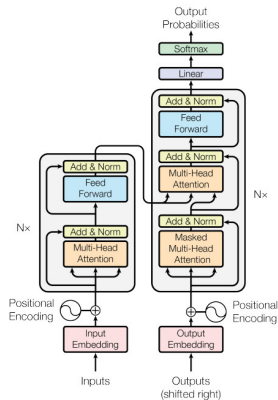
- ▶ Deep neural network for joint object detection and instance segmentation
- ▶ Outputs “structured object”, not only a single number (class label)
- ▶ State-of-the-art on MS-COCO



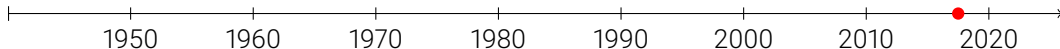
A Brief History of Deep Learning

2017-2018: Transformers and BERT

- Transformers: Attention replaces recurrence and convolutions



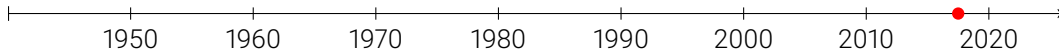
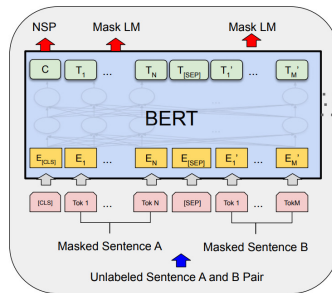
BERT/GLUE



A Brief History of Deep Learning

2017-2018: Transformers and BERT

- ▶ Transformers: Attention replaces recurrence and convolutions
- ▶ BERT: Pre-training of language models on unlabeled text



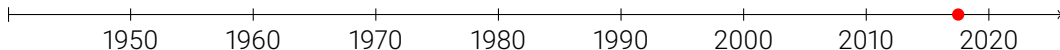
A Brief History of Deep Learning

2017-2018: Transformers and BERT

- ▶ Transformers: Attention replaces recurrence and convolutions
- ▶ BERT: Pre-training of language models on unlabeled text
- ▶ GLUE: Superhuman performance on some language understanding tasks (paraphrase, question answering, ..)
- ▶ But: Computers still fail in dialogue

Rank	Rank Name	Model	URL	Score
1	HFL FLYTEK	MacALBERT + DKM		90.7
+	2	Alibaba DAMO NLP	StructBERT + TAFT	90.6
+	3	PING-AN Omni-Sentic	ALBERT + QAAF + NAS	90.6
	4	ERNIE Team - Baidu	ERNIE	90.4
	5	TS Team - Google	TS	90.3
	6	Microsoft D365 AI & MSR AI & GATECH MIT-DNN-SMART		89.9
+	7	Zihang Dai	Funnel-Transformer (Ensemble B10-10H1024)	89.7
+	8	ELECTRA Team	ELECTRA-Large + Standard Tricks	89.4
+	9	Huawei Noah's Ark Lab	NEZHA-Large	89.1
+	10	Microsoft D365 AI & UWd	FreeLB-RoBERTa (ensemble)	88.4
	11	Jurjie Yang	HIRE-RoBERTa	88.3
	12	Facebook AI	RoBERTa	88.1
+	13	Microsoft D365 AI & MSR AI	MT-DNN-ensemble	87.6
	14	GLUE Human Baselines	GLUE Human Baselines	87.1

BERT/GLUE



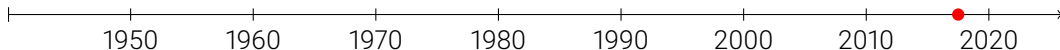
A Brief History of Deep Learning

2017-2018: Transformers and BERT

- ▶ Transformers: Attention replaces recurrence and convolutions
- ▶ BERT: Pre-training of language models on unlabeled text
- ▶ GLUE: Superhuman performance on some language understanding tasks (paraphrase, question answering, ..)
- ▶ But: Computers still fail in dialogue

Rank	Rank Name	Model	URL	Score
1	HFL FLYTEK	MacALBERT + DKM		90.7
+	2	Alibaba DAMO NLP	StructBERT + TAFT	90.6
+	3	PING-AN Omni-Sentic	ALBERT + QAAF + NAS	90.6
	4	ERNIE Team - Baidu	ERNIE	90.4
	5	TS Team - Google	TS	90.3
	6	Microsoft D365 AI & MSR AI & GATECH MIT-DNN-SMART		89.9
+	7	Zihang Dai	Funnel-Transformer (Ensemble B10-10H1024)	89.7
+	8	ELECTRA Team	ELECTRA-Large + Standard Tricks	89.4
+	9	Huawei Noah's Ark Lab	NEZHA-Large	89.1
+	10	Microsoft D365 AI & UWd	FreeLB-RoBERTa (ensemble)	88.4
	11	Junjie Yang	HIRE-RoBERTa	88.3
	12	Facebook AI	RoBERTa	88.1
+	13	Microsoft D365 AI & MSR AI	MT-DNN-ensemble	87.6
	14	GLUE Human Baselines	GLUE Human Baselines	87.1

BERT/GLUE

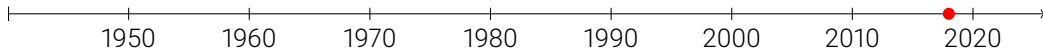


A Brief History of Deep Learning

2018: Turing Award

In 2018, the “nobel price of computing” has been awarded to:

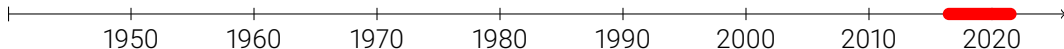
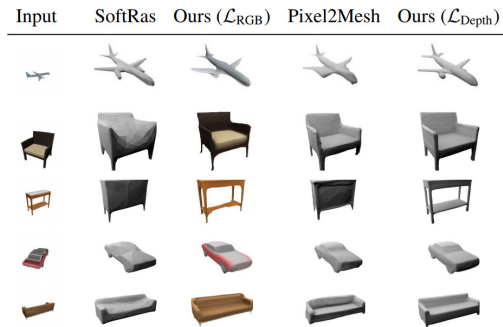
- ▶ Yoshua Bengio
- ▶ Geoffrey Hinton
- ▶ Yann LeCun



A Brief History of Deep Learning

2016-2020: 3D Deep Learning

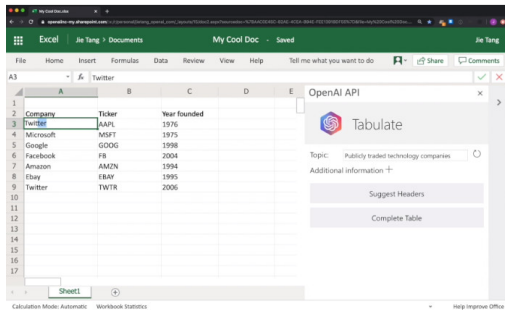
- ▶ First models to successfully output 3D representations
- ▶ Voxels, point clouds, meshes, implicit representations
- ▶ Prediction of 3D models even from a single image
- ▶ Geometry, materials, light, motion



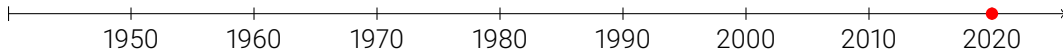
A Brief History of Deep Learning

2020: GPT-3

- ▶ Language model by OpenAI
- ▶ 175 Billion parameters
- ▶ Text-in / text-out interface
- ▶ Many use cases: coding, poetry, blogging, news articles, chatbots
- ▶ Controversial discussions
- ▶ Licensed exclusively to Microsoft on September 22, 2020



GPT-3



A Brief History of Deep Learning

Current Challenges

- ▶ Un-/Self-Supervised Learning
- ▶ Interactive learning
- ▶ Accuracy (e.g., self-driving)
- ▶ Robustness and generalization
- ▶ Inductive biases
- ▶ Understanding and mathematics
- ▶ Memory and compute
- ▶ Ethics and legal questions
- ▶ Does “Moore’s Law of AI” continue?



1.3

Machine Learning Basics

Goodfellow et al.: Deep Learning, Chapter 5

<http://www.deeplearningbook.org/contents/ml.html>

Learning Problems

► **Supervised learning**

- Learn model parameters using dataset of data-label pairs $\{(x_i, y_i)\}_{i=1}^N$
- Examples: Classification, regression, structured prediction

► **Unsupervised learning**

- Learn model parameters using dataset without labels $\{x_i\}_{i=1}^N$
- Examples: Clustering, dimensionality reduction, generative models

► **Self-supervised learning**

- Learn model parameters using dataset of data-data pairs $\{(x_i, x'_i)\}_{i=1}^N$
- Examples: Self-supervised stereo/flow, contrastive learning

► **Reinforcement learning**

- Learn model parameters using active exploration from sparse rewards
- Examples: deep q learning, gradient policy, actor critique

Supervised Learning

Classification, Regression, Structured Prediction

Classification / Regression:

$$f : \mathcal{X} \rightarrow \mathbb{N} \quad \text{or} \quad f : \mathcal{X} \rightarrow \mathbb{R}$$

- ▶ Inputs $x \in \mathcal{X}$ can be any kind of objects
 - ▶ images, text, audio, sequence of amino acids, ...
- ▶ Output $y \in \mathbb{N}/y \in \mathbb{R}$ is a discrete or real number
 - ▶ classification, regression, density estimation, ...

Structured Output Learning:

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

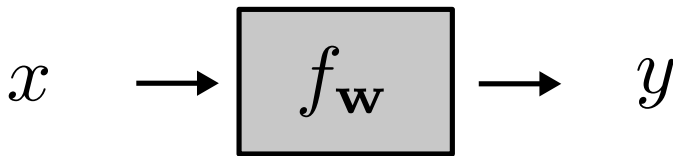
- ▶ Inputs $x \in \mathcal{X}$ can be any kind of objects
- ▶ Outputs $y \in \mathcal{Y}$ are complex (structured) objects
 - ▶ images, text, parse trees, folds of a protein, computer programs, ...

Supervised Learning

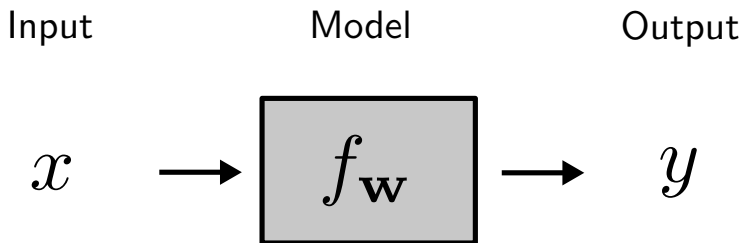
Input

Model

Output

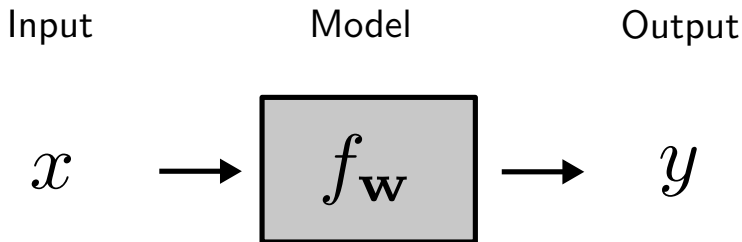


Supervised Learning



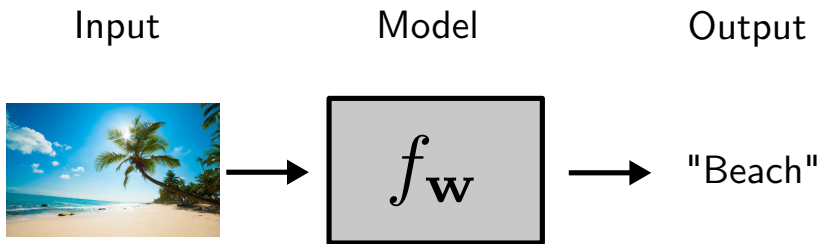
- **Learning:** Estimate parameters $\mathbf{w}$ from training data $\{(x_i, y_i)\}_{i=1}^N$

Supervised Learning



- **Learning:** Estimate parameters $\mathbf{w}$ from training data $\{(x_i, y_i)\}_{i=1}^N$
- **Inference:** Make novel predictions: $y = f_{\mathbf{w}}(x)$

Classification



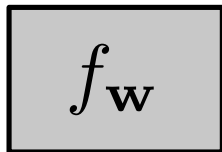
► **Mapping:** $f_{\mathbf{w}} : \mathbb{R}^{W \times H} \rightarrow \{\text{"Beach"}, \text{"No Beach"}\}$

Regression

Input



Model

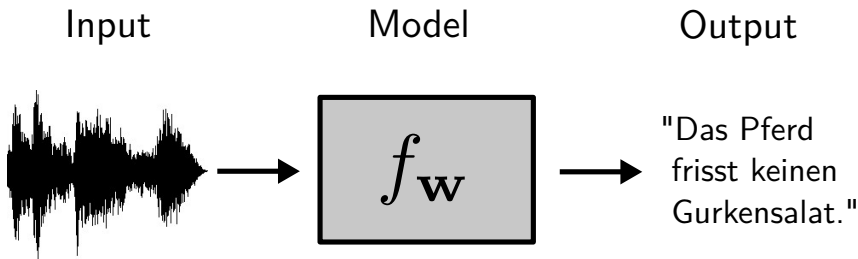


Output

143,52 €

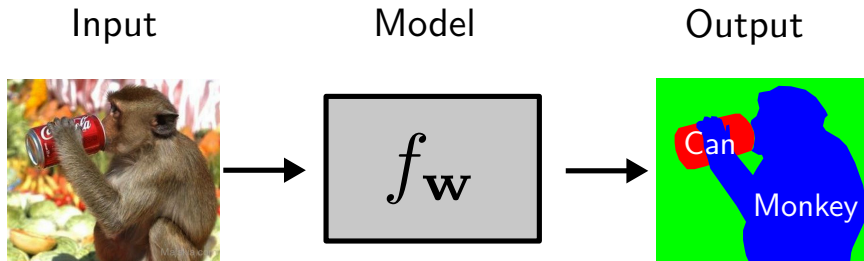
► Mapping: $f_{\mathbf{w}} : \mathbb{R}^N \rightarrow \mathbb{R}$

Structured Prediction



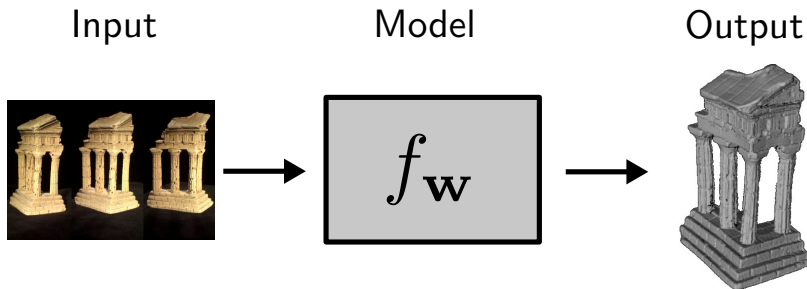
► **Mapping:** $f_{\mathbf{w}} : \mathbb{R}^N \rightarrow \{1, \dots, C\}^M$

Structured Prediction



► **Mapping:** $f_{\mathbf{w}} : \mathbb{R}^{W \times H} \rightarrow \{1, \dots, C\}^{W \times H}$

Structured Prediction



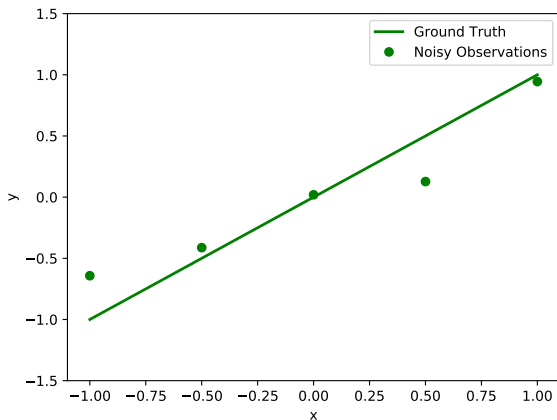
- **Mapping:** $f_{\mathbf{w}} : \mathbb{R}^{W \times H \times N} \rightarrow \{0, 1\}^{M^3}$
- **Suppose:** 32^3 voxels, binary variable per voxel (occupied/free)
- **Question:** How many different reconstructions? $2^{32^3} = 2^{32768}$
- **Comparison:** Number of atoms in the universe? $\sim 2^{273}$

Linear Regression

Linear Regression

Let $\mathcal{X}$ denote a dataset of size N and let $(\mathbf{x}_i, y_i) \in \mathcal{X}$ denote its elements ($y_i \in \mathbb{R}$).

Goal: Predict y for a previously unseen input $\mathbf{x}$. The input $\mathbf{x}$ may be multidimensional.



Linear Regression

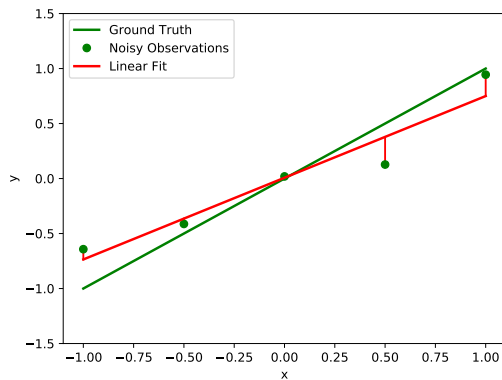
The **error function** $E(\mathbf{w})$ measures the displacement along the y dimension between the data points (green) and the model $f(\mathbf{x}, \mathbf{w})$ (red) specified by the parameters $\mathbf{w}$.

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^\top \mathbf{x}$$

$$E(\mathbf{w}) = \sum_{i=1}^N (f(\mathbf{x}_i, \mathbf{w}) - y_i)^2$$

$$= \sum_{i=1}^N (\mathbf{x}_i^\top \mathbf{w} - y_i)^2$$

$$= \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$$



Here: $\mathbf{x} = [1, x]^\top \Rightarrow f(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x$

Linear Regression

The **gradient of the error function** with respect to the parameters $\mathbf{w}$ is given by:

$$\begin{aligned}\nabla_{\mathbf{w}} E(\mathbf{w}) &= \nabla_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 \\ &= \nabla_{\mathbf{w}} (\mathbf{X}\mathbf{w} - \mathbf{y})^\top (\mathbf{X}\mathbf{w} - \mathbf{y}) \\ &= \nabla_{\mathbf{w}} \left(\mathbf{w}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w} - 2\mathbf{w}^\top \mathbf{X}^\top \mathbf{y} + \mathbf{y}^\top \mathbf{y} \right) \\ &= 2\mathbf{X}^\top \mathbf{X} \mathbf{w} - 2\mathbf{X}^\top \mathbf{y}\end{aligned}$$

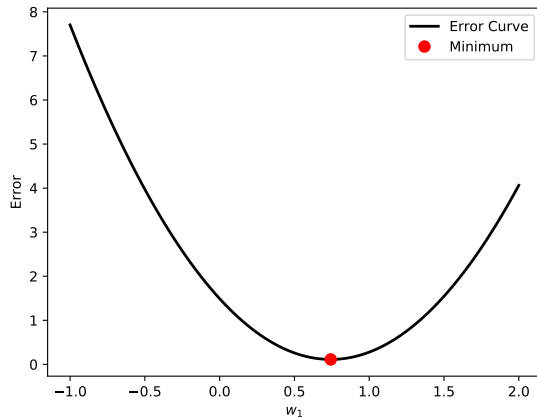
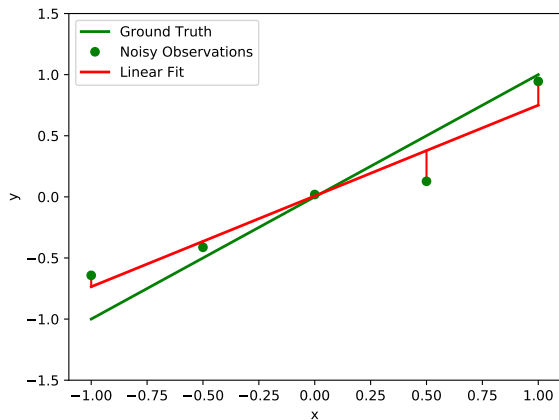
As $E(\mathbf{w})$ is quadratic and convex in $\mathbf{w}$, its minimizer (wrt. $\mathbf{w}$) is given in closed form:

$$\begin{aligned}\nabla_{\mathbf{w}} E(\mathbf{w}) &= 0 \\ \Rightarrow \mathbf{w} &= (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}\end{aligned}$$

The matrix $(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top$ is also called **Moore-Penrose inverse** or pseudoinverse.

Example: Line Fitting

Line Fitting



Linear least squares fit of model $f(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x$ (red) to data points (green). Errors are also shown in red. Right: Error function $E(\mathbf{w})$ wrt. parameter w_1 .

Example: Polynomial Curve Fitting

Polynomial Curve Fitting

Let us choose a **polynomial of order M** to model dataset $\mathcal{X}$:

$$f(x, \mathbf{w}) = \sum_{j=0}^M w_j x^j = \mathbf{w}^\top \mathbf{x} \quad \text{with features} \quad \mathbf{x} = (1, x^1, x^2, \dots, x^M)^\top$$

Tasks:

- ▶ **Training:** Estimate $\mathbf{w}$ from dataset $\mathcal{X}$
- ▶ **Inference:** Predict y for novel x given estimated $\mathbf{w}$

Note:

- ▶ Features can be anything, including multi-dimensional inputs (e.g., images, audio), radial basis functions, sine/cosine functions, etc. In this example: monomials.

Polynomial Curve Fitting

Let us choose a **polynomial of order M** to model the dataset $\mathcal{X}$:

$$f(x, \mathbf{w}) = \sum_{j=0}^M w_j x^j = \mathbf{w}^\top \mathbf{x} \quad \text{with features} \quad \mathbf{x} = (1, x^1, x^2, \dots, x^M)^\top$$

How can we estimate $\mathbf{w}$ from $\mathcal{X}$?

Polynomial Curve Fitting

The **error function** from above is quadratic in $\mathbf{w}$ but not in x :

$$E(\mathbf{w}) = \sum_{i=1}^N (f(x_i, \mathbf{w}) - y_i)^2 = \sum_{i=1}^N (\mathbf{w}^\top \mathbf{x}_i - y_i)^2 = \sum_{i=1}^N \left(\sum_{j=0}^M w_j x_i^j - y_i \right)^2$$

It can be rewritten in the **matrix-vector notation** (i.e., as linear regression problem)

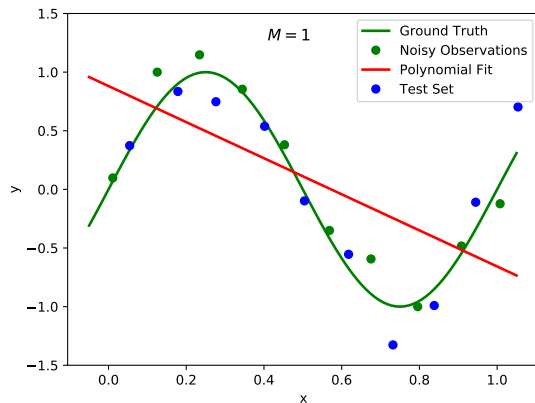
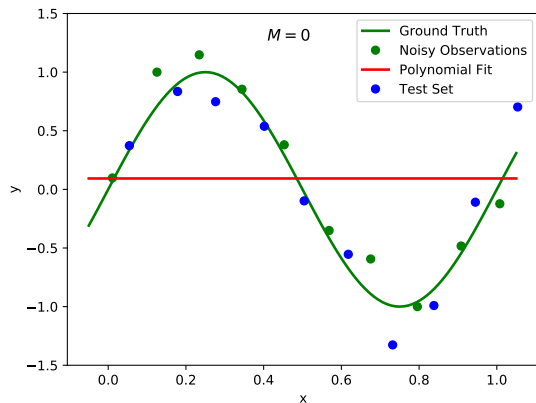
$$E(\mathbf{w}) = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$$

with feature matrix $\mathbf{X}$, observation vector $\mathbf{y}$ and weight vector $\mathbf{w}$:

$$\mathbf{X} = \begin{pmatrix} \vdots & \vdots & \vdots & & \vdots \\ 1 & x_i & x_i^2 & \dots & x_i^M \\ \vdots & \vdots & \vdots & & \vdots \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} \vdots \\ y_i \\ \vdots \end{pmatrix} \quad \mathbf{w} = \begin{pmatrix} w_0 \\ \vdots \\ w_M \end{pmatrix}$$

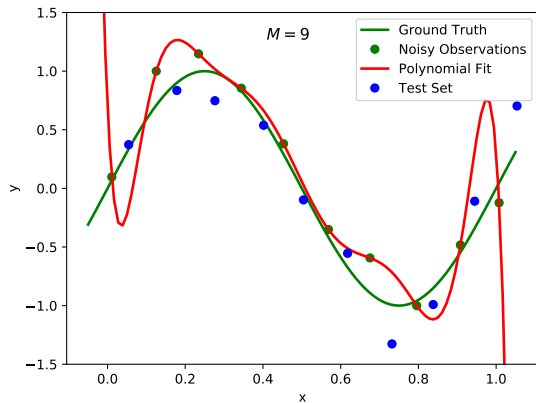
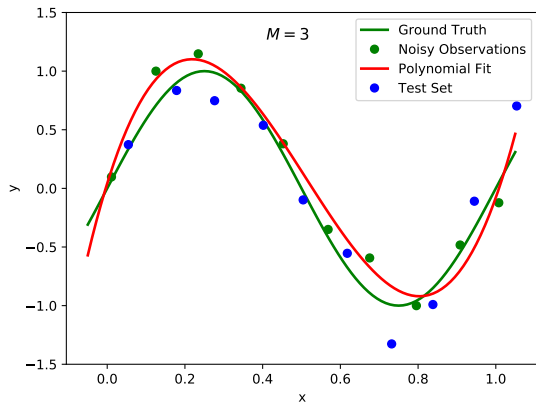
Polynomial Curve Fitting Results

Polynomial Curve Fitting



Plots of polynomials of various degrees M (red) fitted to the data (green). We observe underfitting ($M = 0/1$) and overfitting ($M = 9$). This is a model selection problem.

Polynomial Curve Fitting

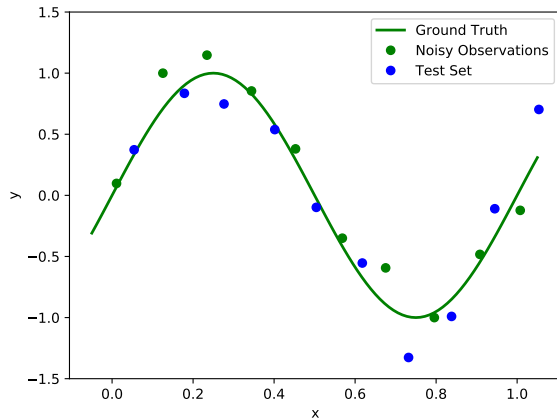


Plots of polynomials of various degrees M (red) fitted to the data (green). We observe underfitting ($M = 0/1$) and overfitting ($M = 9$). This is a model selection problem.

Capacity, Overfitting and Underfitting

Goal:

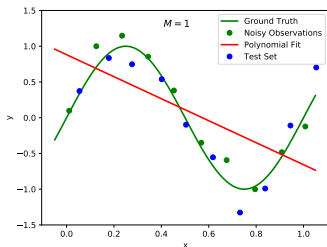
- ▶ Perform well on new, previously unseen inputs (test set, blue), not only on the training set (green)
- ▶ This is called **generalization** and separates ML from optimization
- ▶ Assumption: training and test data independent and identically (i.i.d.) drawn from distribution $p_{data}(x, y)$
- ▶ Here: $p_{data}(x) = \mathcal{U}(0, 1)$
 $p_{data}(y|x) = \mathcal{N}(\sin(2\pi x), \sigma)$



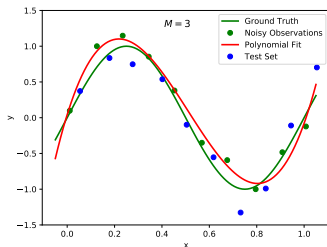
Capacity, Overfitting and Underfitting

Terminology:

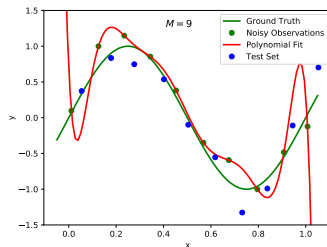
- **Capacity:** Complexity of functions which can be represented by model f
- **Underfitting:** Model too simple, does not achieve low error on training set
- **Overfitting:** Training error small, but test error (= generalization error) large



Capacity too low



Capacity about right

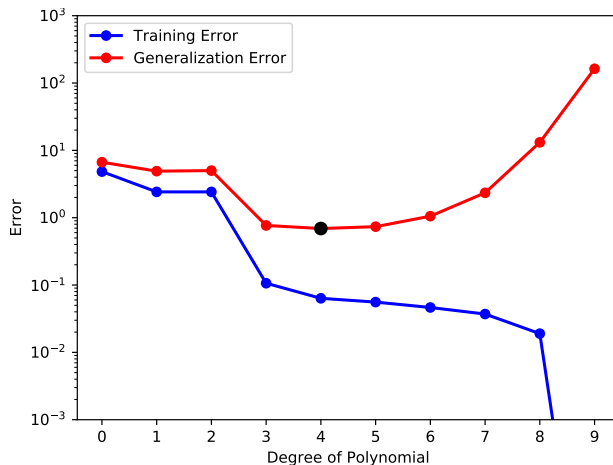


Capacity too high

Capacity, Overfitting and Underfitting

Example: Generalization error for various polynomial degrees M

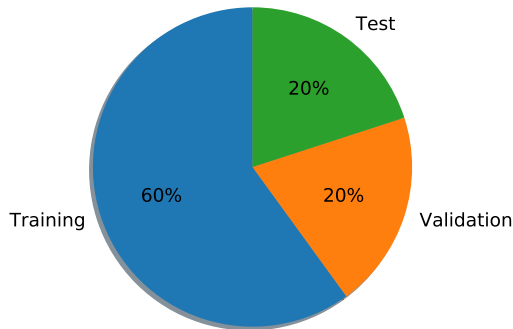
- Model selection: Select model with the smallest generalization error



Capacity, Overfitting and Underfitting

General Approach: Split dataset into training, validation and test set

- Choose hyperparameters (e.g., degree of polynomial, learning rate in neural net, ..) using validation set. Important: Evaluate once on test set (typically not available).



- When dataset is small, use (k-fold) cross validation instead of fixed split.

Ridge Regression

Ridge Regression

Polynomial Curve Model:

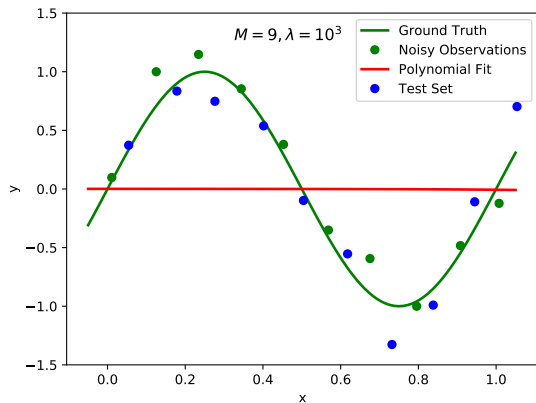
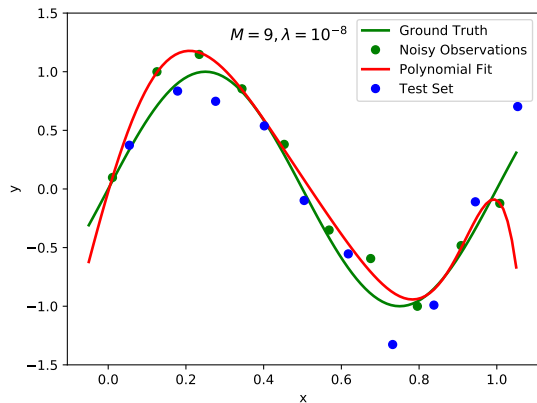
$$f(x, \mathbf{w}) = \sum_{j=0}^M w_j x^j = \mathbf{w}^\top \mathbf{x} \quad \text{with features} \quad \mathbf{x} = (1, x^1, x^2, \dots, x^M)^\top$$

Ridge Regression:

$$\begin{aligned} E(\mathbf{w}) &= \sum_{i=1}^N (f(x_i, \mathbf{w}) - y_i)^2 + \lambda \sum_{j=0}^M w^2 \\ &= \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|_2^2 \end{aligned}$$

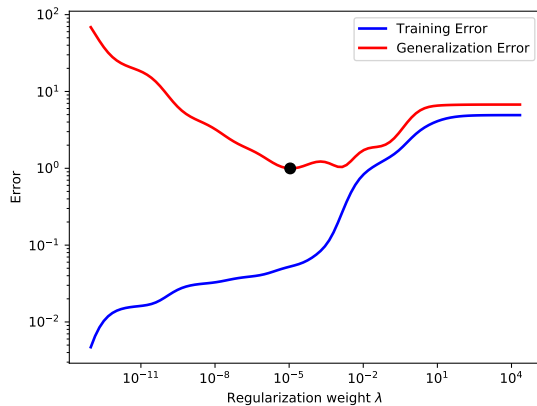
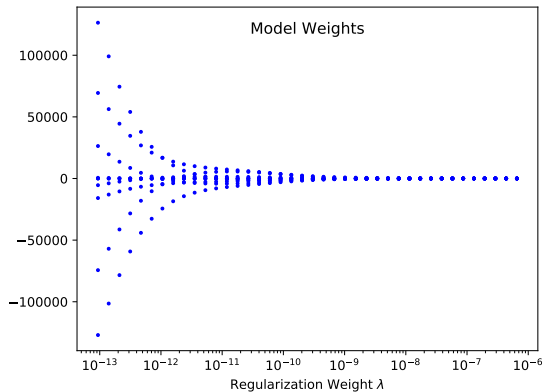
- Idea: Discourage large parameters by adding a regularization term with strength λ
- Closed form solution: $\mathbf{w} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}$

Ridge Regression



Plots of polynomial with degree $M = 9$ fitted to 10 data points using ridge regression. Left: weak regularization ($\lambda = 10^{-8}$). Right: strong regularization (right, $\lambda = 10^3$).

Ridge Regression



Left: With low regularization, parameters can become very large (ill-conditioning).

Right: Select model with the smallest generalization error on the validation set.

Estimators, Bias and Variance

Estimators, Bias and Variance

Point Estimator:

- ▶ A point estimator $g(\cdot)$ is function that maps a dataset $\mathcal{X}$ to model parameters $\hat{\mathbf{w}}$:

$$\hat{\mathbf{w}} = g(\mathcal{X})$$

- ▶ Example: Estimator of ridge regression model: $\hat{\mathbf{w}} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}$
- ▶ We use the hat notation to denote that $\hat{\mathbf{w}}$ is an estimate
- ▶ A good estimator is a function that returns a parameter set close to the true one
- ▶ The data $\mathcal{X} = \{(x_i, y_i)\}$ is drawn from a random process $(x_i, y_i) \sim p_{data}(\cdot)$
- ▶ Thus, any function of the data is random and $\hat{\mathbf{w}}$ is a random variable.

Estimators, Bias and Variance

Properties of Point Estimators:

Bias:

$$\text{Bias}(\hat{\mathbf{w}}) = \mathbb{E}(\hat{\mathbf{w}}) - \mathbf{w}$$

- ▶ Expectation over datasets $\mathcal{X}$
- ▶ $\hat{\mathbf{w}}$ is unbiased $\Leftrightarrow \text{Bias}(\hat{\mathbf{w}}) = 0$
- ▶ A good estimator has little bias

Variance:

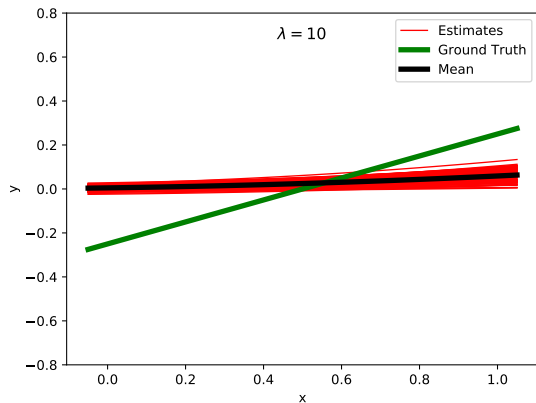
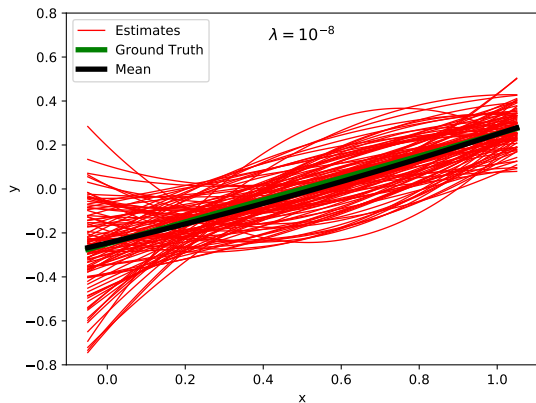
$$\text{Var}(\hat{\mathbf{w}}) = \mathbb{E}(\hat{\mathbf{w}}^2) - \mathbb{E}(\hat{\mathbf{w}})^2$$

- ▶ Variance over datasets $\mathcal{X}$
- ▶ $\sqrt{\text{Var}(\hat{\mathbf{w}})}$ is called “standard error”
- ▶ A good estimator has low variance

Bias-Variance Dilemma:

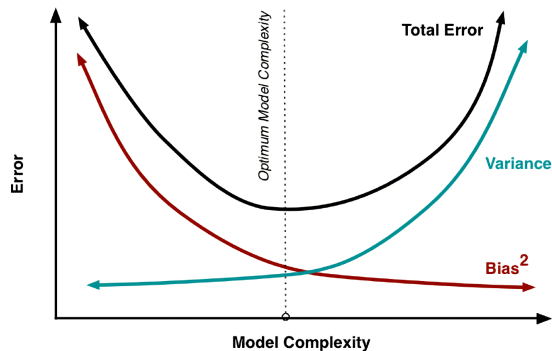
- ▶ Statistical learning theory tells us that we can't have both $\Rightarrow$ there is a trade-off

Estimators, Bias and Variance

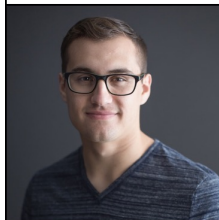


Ridge regression with weak ($\lambda = 10^{-8}$) and strong ($\lambda = 10$) regularization.
Green: True model. Black: Plot of model with mean parameters $\bar{\mathbf{w}} = \mathbb{E}(\mathbf{w})$.

Estimators, Bias and Variance



Statisticians Hate Him



Get low bias AND
LOW VARIANCE
with this one
WEIRD trick

LEARN THE TRUTH NOW

- ▶ There is a bias-variance tradeoff: $\mathbb{E}[(\hat{\mathbf{w}} - \mathbf{w})^2] = \text{Bias}(\hat{\mathbf{w}})^2 + \text{Var}(\hat{\mathbf{w}})$
- ▶ Or not? In deep neural networks the test error decreases with network width!
<https://www.bradyn Neal.com/bias-variance-tradeoff-textbooks-update>

Maximum Likelihood Estimation

Maximum Likelihood Estimation

- ▶ We now reinterpret our results by taking a **probabilistic viewpoint**
- ▶ Let $\mathcal{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ be a dataset with samples drawn i.i.d. from p_{data}
- ▶ Let the model $p_{model}(y|\mathbf{x}, \mathbf{w})$ be a parametric family of probability distributions
- ▶ The conditional **maximum likelihood estimator** for $\mathbf{w}$ is given by

$$\begin{aligned}\hat{\mathbf{w}}_{ML} &= \operatorname{argmax}_{\mathbf{w}} p_{model}(\mathbf{y}|\mathbf{X}, \mathbf{w}) \\ &\stackrel{\text{iid}}{=} \operatorname{argmax}_{\mathbf{w}} \prod_{i=1}^N p_{model}(y_i|\mathbf{x}_i, \mathbf{w}) \\ &= \operatorname{argmax}_{\mathbf{w}} \underbrace{\sum_{i=1}^N \log p_{model}(y_i|\mathbf{x}_i, \mathbf{w})}_{\text{Log-Likelihood}}\end{aligned}$$

Note: In this lecture we consider log to be the natural logarithm.

Maximum Likelihood Estimation

Example: Assuming $p_{model}(y|\mathbf{x}, \mathbf{w}) = \mathcal{N}(y|\mathbf{w}^\top \mathbf{x}, \sigma)$, we obtain

$$\begin{aligned}\hat{\mathbf{w}}_{ML} &= \operatorname{argmax}_{\mathbf{w}} \sum_{i=1}^N \log p_{model}(y_i|\mathbf{x}_i, \mathbf{w}) \\&= \operatorname{argmax}_{\mathbf{w}} \sum_{i=1}^N \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(\mathbf{w}^\top \mathbf{x}_i - y_i)^2} \right] \\&= \operatorname{argmax}_{\mathbf{w}} - \sum_{i=1}^N \frac{1}{2} \log(2\pi\sigma^2) - \sum_{i=1}^N \frac{1}{2\sigma^2} (\mathbf{w}^\top \mathbf{x}_i - y_i)^2 \\&= \operatorname{argmax}_{\mathbf{w}} - \sum_{i=1}^N (\mathbf{w}^\top \mathbf{x}_i - y_i)^2 \\&= \operatorname{argmin}_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2\end{aligned}$$

Maximum Likelihood Estimation

We see that choosing $p_{model}(y|\mathbf{x}, \mathbf{w})$ to be Gaussian causes maximum likelihood to yield exactly the same least squares estimator derived before:

$$\hat{\mathbf{w}} = \operatorname{argmin}_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$$

Variations:

- ▶ If we were choosing $p_{model}(y|\mathbf{x}, \mathbf{w})$ as a Laplace distribution, we would obtain an estimator that minimizes the ℓ_1 norm: $\hat{\mathbf{w}} = \operatorname{argmin}_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_1$
- ▶ Assuming a Gaussian distribution over the parameters $\mathbf{w}$ and performing a maximum a-posteriori (MAP) estimation yields ridge regression:

$$\operatorname{argmax}_{\mathbf{w}} p(\mathbf{w}|y, \mathbf{x}) = \operatorname{argmax}_{\mathbf{w}} p(y|\mathbf{x}, \mathbf{w})p(\mathbf{w})$$

Maximum Likelihood Estimation

We see that choosing $p_{model}(y|\mathbf{x}, \mathbf{w})$ to be Gaussian causes maximum likelihood to yield exactly the same least squares estimator derived before:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$$

Remarks:

- ▶ **Consistency:** As the number of training samples approaches infinity $N \rightarrow \infty$, the maximum likelihood (ML) estimate converges to the true parameters
- ▶ **Efficiency:** The ML estimate converges most quickly as N increases
- ▶ These theoretical considerations make ML estimators appealing