Machine Learning for Stochastics
A short intro to ML

Thorsten Schmidt

Abteilung für Mathematische Stochastik

www.stochastik.uni-freiburg.de
thorsten.schmidt@stochastik.uni-freiburg.de

SS 2025

Week 6

In Week 6 we cover the following topics:

- Introduction (AI, Machine Learning, Examples, Deep Networks)
- Deep learning - Introduction and definition
- Universal approximation theorems
- Kolmogorov/Arnold networks

Motivation

- **Machine Learning** is nowadays used at many places (Google, Amazon, etc.) with a great variety of applications.
- It is a great job opportunity ! It needs maths and probability !
- Many applications are surprisingly successful (speech / face recognition / robotic / autonomous cars / medicine / chemistry / chat GPT) and currently people are seeking further applications
- Here we want to learn about the foundations, discuss implications and what can be done by ML and what not.

Topics include:

- Foundations and deeper understanding
- Optimization under Uncertainty
- Regulation and Fairness

Slides and code of the previous lectures are available on github. But - proofs and theorems will mostly remain on blackboard only.

- Code and slides are on `https://github.com/tschmidtfreiburg/ML_lecture_2024` (see homepage).

Overview[1]

- Artificial intelligence is the field where computers solve problems.
- It is easy for a computer to solve tasks which can be described formally (Chess, Tic-Tac-Toe). The challenge is to solve a tasks which are hard to describe formally (but are easy for humans: walk, drive a car, speak, recognize people ...)
- The solution is to allow computers to learn from experience and to understand the world by a hierarchy of concepts, each concept defined in terms of its relation to simpler concepts.
- A fixed knowledge-base would be somehow limiting such that we are interested in such attempts where the systems acquire their own knowledge, which we call **Machine Learning**.

[1] This introduction follows somehow Goodfellow et.al. (2016) and my previous lectures.

Artificial Intelligence

- First examples of machine learning are **logistic regression** or **naive Bayes** → standard statistical procedures (E.g. the recognition of spam, more examples to follow)
- Problems become simpler with a nice representation. Of course it would be nice if the system itself could find such a representation, which we call **representation learning**.
- An example is the so-called **auto-encoder**. This is a combination of an encoder and a decoder. The encoder converts the input to a certain representation and the decoder converts it back again, such that the result has nice properties.
- Speech for example might be influenced by many factors of variation (age, sex, origin, ...) and it needs nearly human understanding to disentangle the variation from the content we are interested in.
- **Deep Learning** solves this problem by introducing hierarchical representations.

- This leads to the following hierarchy:
- AI → machine learning → representation learning → deep learning.

Examples of Machine Learning

Some of the most prominent examples:

- LeCun et.al.[2] recognition of handwritten digits. The MNIST Database[3] provides 60.000 samples for testing algorithms. The NIST database is of increased size[4]
- The Viola & Jones face recognition,[5]. This path-breaking work proposed a procedure to combine existing tools with machine-learning algorithms. One key is the use of approx. 5000 learning pictures to train the routine. We will revisit this procedure shortly.
- Imagenet is an image database containing many images classified (cats, cars, etc.)[6]
- Various twitter datasets are available, for example for learning to detect hate speach.
- `Kaggle`[7] is a platfrom where computational competitions are hosted. It also provides many many data examples with it.
- Datasets for machine-learning research on Wikipedia[8].

[2] Y. LeCun et al. (1998). „Gradient-based learning applied to document recognition“. In: **Proceedings of the IEEE** 86.11, pp. 2278–2324.

[3] `http://yann.lecun.com/exdb/mnist/`

[4] `https://www.nist.gov/srd/nist-special-database-19`

[5] P. Viola and M. Jones (2001). „Robust Real-time Object Detection“. In: **International Journal of Computer Vision**. Vol. 4. 34–47.

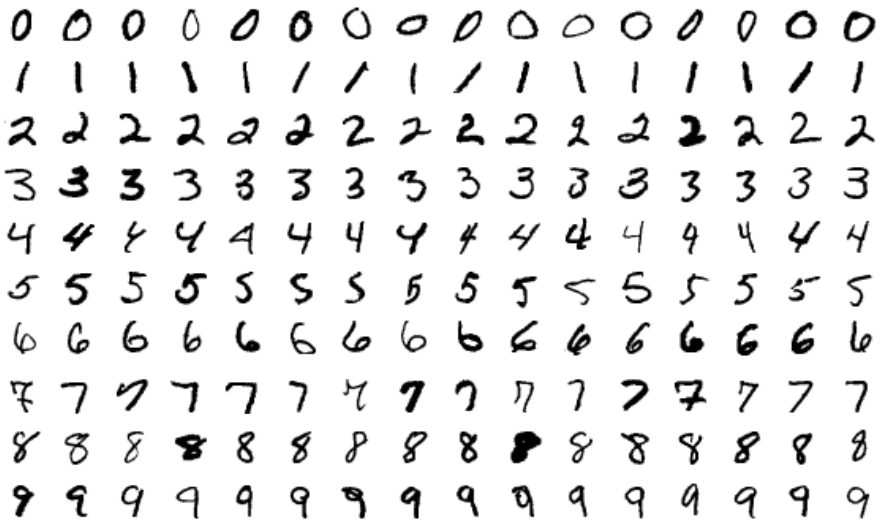
[6] `http://image-net.org`

[7] www.kaggle.com

[8] `https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research`

MNIST

We will see this data set in more detail.



- Speech recognition has long been a difficult problem for computers (first works date to the 50's) and only recently been solved with high computer power. It may seem surprising, that mathematical tools are at the core of these solutions. Let us quote Hinton et.al.[9]

  > *Most current speech recognition systems use hidden Markov models (HMMs) to deal with the temporal variability of speech and Gaussian mixture models (GMMs) to deter- mine how well each state of each HMM fits a frame or a short window of frames of coefficients that repre- sents the acoustic input. (...)*
  > *Deep neural networks (DNNs) that have many hidden layers and are trained using new methods have been shown to outperform GMMs on a variety of speech recognition benchmarks, sometimes by a large margin*

  So, one of our tasks will be to develop a little bit of mathematical tools which we will need later. Most notably, some of the mathematical parts can be replaced by deep learning, which will be of high interest to us.

[9] Geoffrey Hinton et al. (2012). „Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups". In: **IEEE Signal Processing Magazine** 29.6, pp. 82–97.

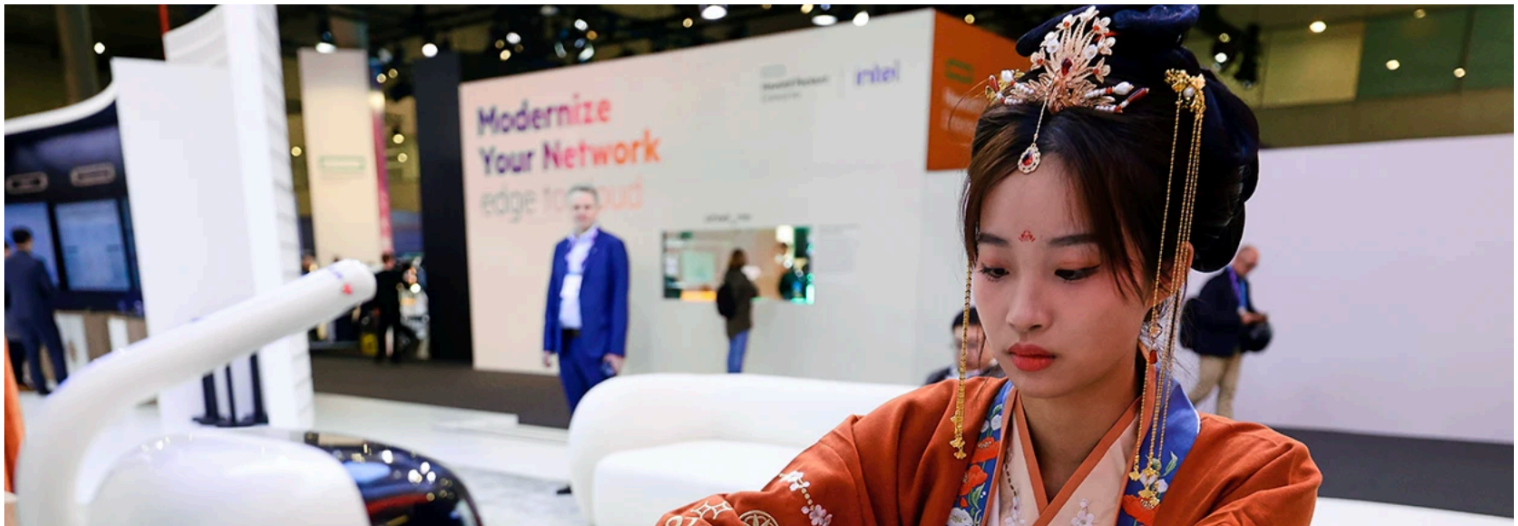
Recent advantages

NEWS | 15 April 2024

AI now beats humans at basic tasks — new benchmarks are needed, says major report

Stanford University's 2024 AI Index charts the meteoric rise of artificial-intelligence tools.

By Nicola Jones



Context

If we want to give this a little bit more context we could compare these results.

- The human brain has approximately 86 billion neurons
- GPT4 is announced to have approx. 175 trillion parameters
- The cost of training the human brain is .... (compare this to 80 Million dollar to train GPT4). But ...
- How many parameters does a neural network with n neurons have? Think of a simple fully connected NN with L layers and n_i neurons per layer.
- For each neuron in layer i we connect to n_{i+1} neurons, which makes $n_i \cdot n_{i+1}$ parameters. Additionally we include one parameter for a shift in mean which makes

$$\sum_{i=1}^{L-1} \left(n_i \cdot n_{i+1} + n_{i+1} \right)$$

- So for a single-layer NN with 100 input nodes and 100 output nodes we have $100 \cdot 100 + 100 \ \cdot 100 + 100 = 20.100$ parameters.

The AI Index Report 2024

1. AI beats humans on some tasks, but not on all.

AI has surpassed human performance on several benchmarks, including some in image classification, visual reasoning, and English understanding. Yet it trails behind on more complex tasks like competition-level mathematics, visual commonsense reasoning and planning.

2. Industry continues to dominate frontier AI research.

In 2023, industry produced 51 notable machine learning models, while academia contributed only 15. There were also 21 notable models resulting from industry-academia collaborations in 2023, a new high.

3. Frontier models get way more expensive.

According to AI Index estimates, the training costs of state-of-the-art AI models have reached unprecedented levels. For example, OpenAI's GPT-4 used an estimated $78 million worth of compute to train, while Google's Gemini Ultra cost $191 million for compute.

4. The United States leads China, the EU, and the U.K. as the leading source of top AI models.

In 2023, 61 notable AI models originated from U.S.-based institutions, far outpacing the European Union's 21 and China's 15.

5. Robust and standardized evaluations for LLM responsibility are seriously lacking.

New research from the AI Index reveals a significant lack of standardization in responsible AI reporting. Leading developers, including OpenAI, Google, and Anthropic, primarily test their models against different responsible AI benchmarks. This practice complicates efforts to systematically compare the risks and limitations of top AI models.

6. Generative AI investment skyrockets.

Despite a decline in overall AI private investment last year, funding for generative AI surged, nearly octupling from 2022 to reach $25.2 billion. Major players in the generative AI space, including OpenAI, Anthropic, Hugging Face, and Inflection, reported substantial fundraising rounds.

7. The data is in: AI makes workers more productive and leads to higher quality work.

In 2023, several studies assessed AI's impact on labor, suggesting that AI enables workers to complete tasks more quickly and to improve the quality of their output. These studies also demonstrated AI's potential to bridge the skill gap between low- and high-skilled

8. Scientific progress accelerates even further, thanks to AI.

In 2022, AI began to advance scientific discovery. 2023, however, saw the launch of even more significant science-related AI applications—from AlphaDev, which makes algorithmic sorting more efficient, to GNoME, which facilitates the process of materials discovery.

9. The number of AI regulations in the United States sharply increases.

The number of AI-related regulations in the U.S. has risen significantly in the past year and over the last five years. In 2023, there were 25 AI-related regulations, up from just one in 2016. Last year alone, the total number of AI-related regulations grew by 56.3%.

Questions

We repeatedly state questions after some slides. These allow you to reflect on the content and also invite you to research / experiment with some topics yourself.

- Was is artificial intelligence ?
- Was is machine learning ?
- Do you know what a neural network is (look for the history in the internet)?
- What are shallow / deep networks ?
- What are the applications which you find most exciting ?
- What are the applications that you think will have the largest impact on our future?
- Research a bit yourself: look for datasets, look for latest applications etc.

Deep Learning

- "Deep" learning contrasts "shallow" learning: such algorithms are for example linear regression, SVMs...: they have an input layer and an output layer. We have experienced the kernel trick: inputs may be transformed once before application of the algorithm.
- In deep learning there are one ore more **hidden layers** between input and output. Intuitively, at each layer we take the input, make a transformation and generate the output for the next layer.

Definition (Deep network)

A **neural network** is an n-fold composition of simple functions

$$f(x) = f^n \circ \cdots \circ f^1(x) = f^n(f^{n-1}(\cdots f^2(f^1(x)) \cdots)).$$

It is called **deep**, if $n \geq 2$. f^k is called the k-the layer of the network.
Each layer is a composition of a non-linear **activation function** σ and an affine function $a + Bx$,

$$f^k(\cdot) = \sigma^k(a^k + B^k \cdot)$$

In this case the network has one input layer, $n-1$ hidden layers and one (f^n) output layers.

A bit of the history

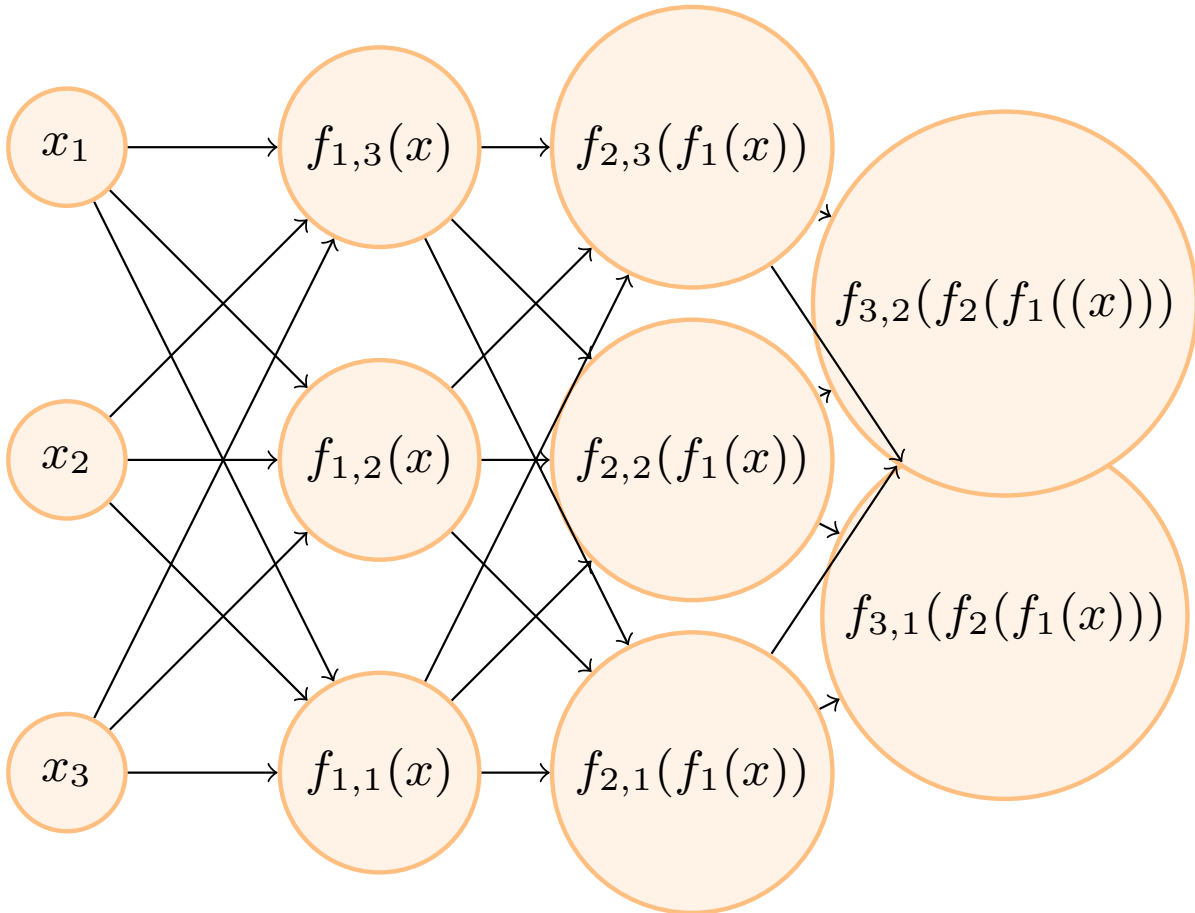
- The terminology of deep learning stems from early research on artifical intelligence and we dive shortly into this exciting subject. Two aspects were important at those times: to be inspired by the human brain and, on the other side, to try to understand the brain better through the construction of similar algorithms. Nowadays, we are more pragmatic and generalize the earlier ideas in several respects.
- A **neuron**[10] takes several inputs, say $x_1, \ldots, x_n$ and gives

$$\mathbb{1}_{\{\sum_{i=1}^n w_i x_i > \theta\}}$$

  as an output - $w_i \in \mathbb{R}$ are several weights and $\theta \in \mathbb{R}$ is a threshold. This description of a neuron was given 1943 by W. McCulloch and W. Pitts.
- It was the idea of F. Rosenblatt in 1958 to introduce a simple neural net, called **perceptron** (after perception) which takes (possibly several) neurons as inputs and generates a more complex decision mechanism.

[10] See the wikipedia article on perceptrons.

Feed-forward neural networks



- Networks of this type are called either **feed-forward neural networks** or **multi-layer perceptrons** (when the nodes are actually neurons).
- $x_1, \ldots, x_n$ constitute the input layer. They give the input to all connected neurons in the **first layer**. There are 3 **hidden units** in our case and **two hidden layers**.

- One problem which can not be achieved by a single layer perceptron is learning XOR ($\rightarrow$ Exercise).

The universal approximation theorem

- One important property of feed-forward neural networks is, that even in the single-layer case they can approximate arbitrary functions very well.
- The result is the so-called universal approximation theorem proved in Kurt Hornik (1991). „Approximation capabilities of multilayer feedforward networks“. In: **Neural networks** 4.2, pp. 251–257.
- We study the mathematical details of this results.

- We consider special classes of feed-forward neural networks, which can be thought of a small generalization of multi-layer perceptrons: in each step, a neuron transforms the input vector x in an affine form to $a^\top x + b$ and sends the output $\phi(a^\top x + b)$. The outputs are weighted and summed up by each connected neuron.
- If there is only one hidden layer and only one output unit, we arrive at the output

$$\sum_{i=1}^{n} c_i \phi(a_i^\top x + b_i).$$

- Hence, the functions implemented by such a network with n hidden units is

$$\mathscr{N}^{(n)} = \mathscr{N}^{(n)}(\phi) = \left\{ h : \mathbb{R}^d \to \mathbb{R} : h(x) = \sum_{i=1}^{n} c_i \phi(a_i^\top x + b_i) \right\}$$

  and for an arbitrary large number of units we set $\mathscr{N}(\phi) = \cup_n \mathscr{N}^{(n)}$.
- We consider functions in the $L^p(\mu)$-space with a finite measure μ. This are measurable functions $f : \mathbb{R}^d \to \mathbb{R}$, such that

$$\|f\|_p := \left(\int |f(x)|^p \mu(dx) \right)^{1/p} < \infty.$$

- A subset S of L^p is called **dense**, if for every $f \in L^p$ and $\varepsilon > 0$ there is a function $g \in S$, such that $\|f - g\|_p < \varepsilon$.

Theorem (Hornik (1991))

If ϕ is bounded and not constant, then $\mathcal{N}(\phi)$ is dense in $L^p(\mu)$ for any finite measure μ on $\mathbb{R}^d$.

This result also holds on the Banach space $C(K)$, K compact, with respect to the sup-norm. Further results are found in Hornik (1991).

The proof

- We will not discuss all the details of the proof, but have a look at certain components.
- First, observe that $\mathscr{N}$ is a **linear** subspace of $L^p(\mu)$ (elements are bounded!)
- If $\mathscr{N}$ is **not** dense, then the Hahn-Banach theorem yields the existence of a (non-zero) continuos linear function Λ such that Λ vanishes on $\mathscr{N}$. The goal is to construct a contradiction by this.
- Currently, Λ seems not to be so tractable, but duality of Hilbert spaces actually gives a very good description of such functionals. In particular, in our case we know that
$$\Lambda f = \int f g \mu(dx)$$
with some $g \in L^q(\mu)$ and $q = p/(p-1)$.
- Now we can write
$$\Lambda f = \int f d\mu'$$
with (by Hölders inequality) some finite (but possibly signed) measure μ'.
- As Λ vanishes on $\mathscr{N}$,
$$\int \phi(a^\top x + b)\mu'(dx) = 0$$
for all $a \in \mathbb{R}^d$ and $b \in \mathbb{R}$.

We have

$$\int \phi(a^\top x + b)\mu'(dx) = 0 \tag{1}$$

for all $a \in \mathbb{R}^d$ and $b \in \mathbb{R}$. It is clear that this can not hold for any function ϕ. Hornik was able to show, that if ϕ is bounded and not constant, then (1) will not hold for any finite signed measure μ'.

- A first step is the transformation

$$\int \phi(a^\top x + b)\mu'(dx) = \int \phi(t + b)\mu_a(dt)$$

with the projection measure $\mu_a(B) = \mu'(x \in \mathbb{R}^d : a\top x \in B)$.

- For the next step, Hornik specializes to $L^1(\mathbb{R})$ with the Lesbesgue measure. In this case, $\mu_a(t)$ is dominated by the Lesbesgue-measure. Using Radon-Nikodym one arrives at

$$\int \phi(t)h(\alpha t + \beta)dt.$$

Now one can apply Fourier transform and arrives at $\mu_a = 0$ for all $a \in \mathbb{R}^d$.

Other versions of UAT

- Starting point of representation theorems is the famous Theorem from Andrej Kolmogorov and Vladimir Arnold (1957/58). It solved en passant a version (!) of Hilberts 13th problem. David Sprecher improved 1962 this result to the following version.

Theorem (Kolmogorov/Arnold)

Any continuous function $f : \mathbb{R}^n \mapsto \mathbb{R}$ *can be represented as*

$$f(x_1, \ldots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^{n} \lambda_p \phi(x_p + \eta q) + q \right)$$

with continuous functions Φ_q *and an increasing function* $\phi : [0, 1] \to [0, 1]$.

It is the first version of an universal approximation theorem.

There are various extensions. For example, Ludger Rüschendorf and Wolfgang Thomsen (1998). „Closedness of sum spaces and the generalized Schrödinger problem“. In: **Theory of Probability & Its Applications** 42.3, pp. 483–494 show the following result.

Theorem

Consider a Borel measure μ on $\mathbb{R}^n$ and a measurable function which is locally bounded. Then

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{2n+1} g\left(\sum_{j=1}^{n} \alpha_{ij}(x) \right)$$

μ-almost everywhere for some measurable functions g and piecewise Lipschitz-continuous α_{ij}.