Stochastic Machine Learning
02 - Introduction to Deep Learning

Thorsten Schmidt

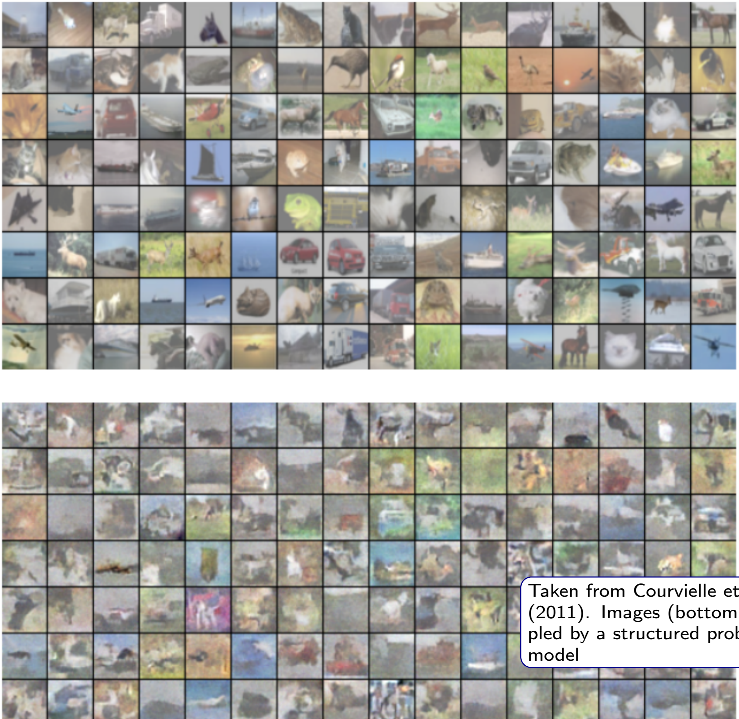
Abteilung für Mathematische Stochastik

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

WS 2020/21

Structured Probabilistic Models

Machine Learning often involves high-dimensional probability distributions and numerous interaction between the dimensions need to be specified. One possibility to achieve this are structured probabilistic models and we discuss two possibilities.



Taken from Courvielle et al (2011). Images (bottom) sampled by a structured probabilistic model

The goal of these probabilistic models is to describe the probability (or the density) of a n-dimensional random variable $\boldsymbol{x}$,

$$p(\boldsymbol{x}) = p(x_1, \dots, x_n).$$

Without any further assumption we can represent this probability by conditioning,

$$p(\boldsymbol{x}) = p(x_1) \cdot p(x_2|x_1) \cdot p(x_3|x_1, x_2) \cdots p(x_n|x_1, \dots, x_{n-1}).$$

It is the goal of these algorithms to find a parsimoniuous representation of $p(\boldsymbol{x})$.

Directed graphical models describe the **conditional factorization** via directed graphs. The density is factorized as follows:

$$p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i|x_j : j \in J_i)$$

with subsets $J_1, J_2, \dots$ of $\{1, \dots, n\}$.

The goal of these probabilistic models is to describe the probability (or the density) of a n-dimensional random variable $\boldsymbol{x}$,

$$p(\boldsymbol{x}) = p(x_1, \dots, x_n).$$

Without any further assumption we can represent this probability by conditioning,

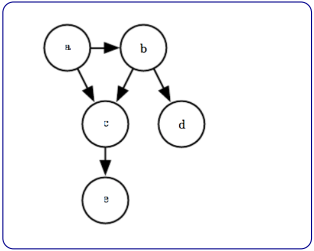
$$p(\boldsymbol{x}) = p(x_1) \cdot p(x_2|x_1) \cdot p(x_3|x_1, x_2) \cdots p(x_n|x_1, \dots, x_{n-1}).$$

It is the goal of these algorithms to find a parsimoniuous representation of $p(\boldsymbol{x})$.

Directed graphical models describe the **conditional factorization** via directed graphs. The density is factorized as follows:

$$p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i|x_j : j \in J_i)$$

with subsets $J_1, J_2, \dots$ of $\{1, \dots, n\}$.



The example (left[1]) describes the density

$$p(a, b, c, d, e) = p(a)p(b|a)p(c|a, b)p(d|b)p(e|c).$$

[1] Taken from Goodfellow e.a.(2016), Figure 3.7.

Undirected graphical models describe the **unconditional factorization** via undirected graphs. The density is factorized as follows:

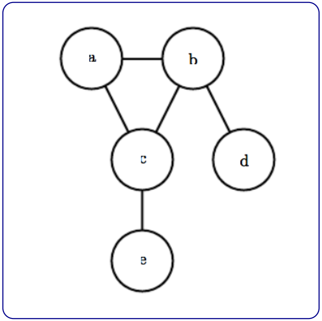
$$p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_j : j \in J_i)$$

with subsets $J_1, J_2, \ldots$ of $\{1, \ldots, n\}$.

Undirected graphical models describe the **unconditional factorization** via undirected graphs. The density is factorized as follows:

$$p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_j : j \in J_i)$$

with subsets $J_1, J_2, \ldots$ of $\{1, \ldots, n\}$.



The example (left[2]) describes the density

$$p(a, b, c, d, e) \propto \phi^1(a, b, c)\phi^2(b, d)\phi^3(c, e).$$

[2] Taken from Goodfellow e.a.(2016), Figure 3.8.

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

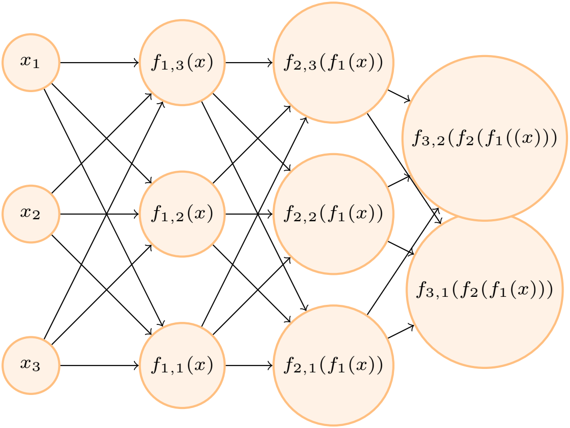
- The terminology of deep learning stems from early research on artifical intelligence and we dive shortly into this exciting subject. Two aspects were important at those times: to be inspired by the human brain and, on the other side, to try to understand the brain better through the construction of similar algorithms. Nowadays, we are more pragmatic and generalize the earlier ideas in several respects.
- A **neuron**[3] 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.

[3] 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 (→ 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

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

- 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 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*

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

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 shows the following result.

Theorem

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

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

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