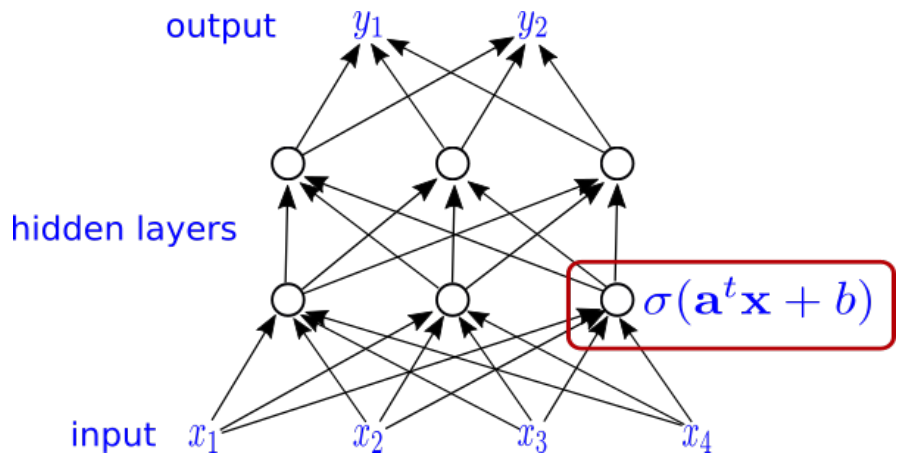
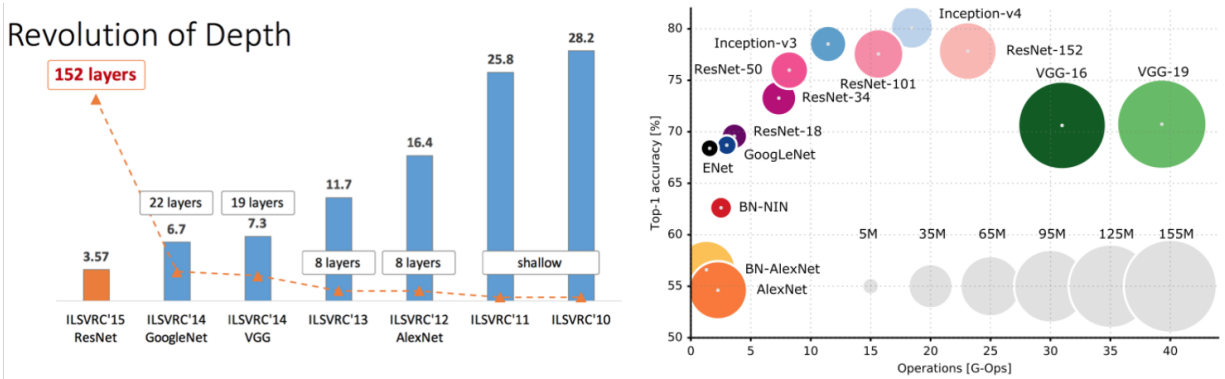
Statistical theory for deep neural networks



Johannes Schmidt-Hieber

Joint work with Konstantin Eckle

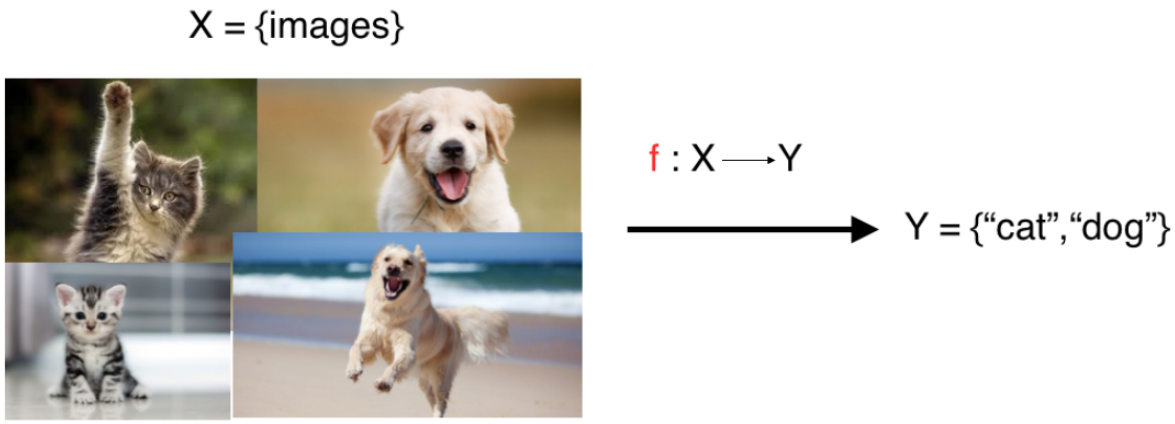
Characteristics of modern deep network architectures



Source: K. He, Deep Residual Networks and arxiv.org/pdf/1605.07678.pdf

- Networks are deep
- Number of network parameters is larger than sample size
- There is some sort of sparsity on the parameters
- ReLU activation function ($\sigma(x) = \max(x, 0)$)

Mathematical problem



The data are used to fit a network, i.e. estimate the network parameters.

How fast does the estimated network converge to the truth f as sample size increases?

Statistical analysis

- we observe n i.i.d. copies $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$,

$$Y_i = f(\mathbf{X}_i) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, 1)$$

  - $\mathbf{X}_i \in \mathbb{R}^d$, $Y_i \in \mathbb{R}$,
  - goal is to reconstruct the function $f : \mathbb{R}^d \to \mathbb{R}$
- has been studied extensively (kernel smoothing, wavelets, splines, ...)

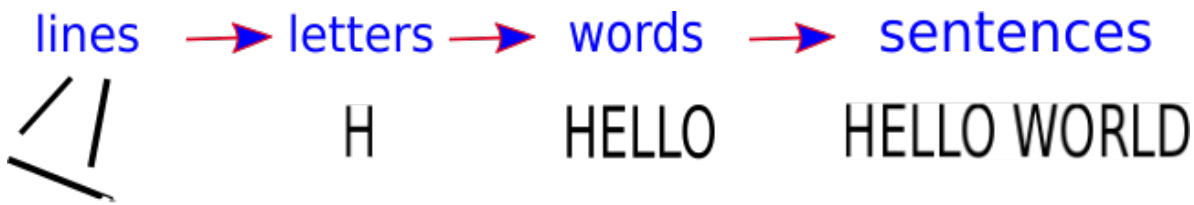
Estimator

- denote by $\mathcal{F}(L, \mathbf{p}, s)$ the class of all networks with
  - number of layers L,
  - width vector $\mathbf{p}$
  - number of non-zero weights s
- our theory applies to any estimator $\widehat{f}_n$ taking values in $\mathcal{F}(L, \mathbf{p}, s)$
- study the dependence of n on prediction error

Function class

- classical idea: assume that regression function is β-smooth
- optimal nonparametric estimation rate is $n^{-2\beta/(2\beta+d)}$
- suffers from curse of dimensionality
- to understand deep learning this setting is therefore useless
- $\rightsquigarrow$ make a good structural assumption on f

Hierarchical structure



- Important: Only few objects are combined on deeper abstraction level
  - few letters in one word
  - few words in one sentence

Function class

- We assume that

$$f = g_q \circ \ldots \circ g_0$$

with

- $g_i : \mathbb{R}^{d_i} \to \mathbb{R}^{d_{i+1}}$.
- each of the d_{i+1} components of g_i is β_i-smooth and depends only on t_i variables
- t_i can be much smaller than d_i
- effective smoothness

$$\beta_i^* := \beta_i \prod_{\ell=i+1}^{q} (\beta_\ell \wedge 1).$$

- we show that **the rate depends on the pairs**

$$(t_i, \beta_i^*), \quad i = 0, \ldots, q.$$

Main result

Theorem: If

(i) depth $\asymp \log n$

(ii) width $\geq$ network sparsity $\asymp \max_{i=0,\ldots,q} n^{\frac{t_i}{2\beta_i^* + t_i}} \log n$

Then, for any network reconstruction method $\widehat{f}_n$,

$$\text{predicition error} \asymp \phi_n + \Delta_n$$

(up to $\log n$-factors) with

$$\Delta_n := E\Big[\frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{f}_n(\mathbf{X}_i))^2 - \inf_{f \in \mathcal{F}(L,\mathbf{p},s)} \frac{1}{n}\sum_{i=1}^{n}(Y_i - f(\mathbf{X}_i))^2\Big]$$

and

$$\phi_n := \max_{i=0,\ldots,q} n^{-\frac{2\beta_i^*}{2\beta_i^* + t_i}}.$$

Consequences

- the assumption that depth $\asymp \log n$ appears naturally
- $\rightsquigarrow$ more data need deeper networks
- the networks can have much more parameters than the sample size
- **important for statistical performance is not the size of the network but the amount of regularization**
  - here the number of active parameters

Consequences (ctd.)

paradox:

- good rate for all smoothness indices
- existing piecewise linear methods only give good rates up to smoothness two
- Here the non-linearity of the function class helps

$\rightsquigarrow$ **non-linearity is essential!!!**

Suboptimality of wavelet estimators

- $f(\mathbf{x}) = h(x_1 + \ldots + x_d)$
- h is α-smooth
- deep neural networks achieve optimal rate $n^{-\alpha/(2\alpha+1)}$ (up to logarithmic factors)
- best wavelet thresholding estimator achieves only the rate $n^{-\alpha/(2\alpha+d)}$
- Reason: The low-dimensional structure does not affect the decay of the wavelet coefficients

Comparison with other piecewise linear methods

- there are several other popular methods in statistics/machine learning that fit piecewise linear functions to data
- how do they compare to DNNs?
- we prove that functions that can be represented by s parameters with respect to the other function systems can be represented by $s \log(1/\varepsilon)$-sparse DNNs up to sup-norm error ε
- the opposite is not true, one counterexample is $f(x_1, x_2) = (x_1 + x_2 - 1)_+$
- $\rightsquigarrow$ conclusion is that DNNs work better for correlated design