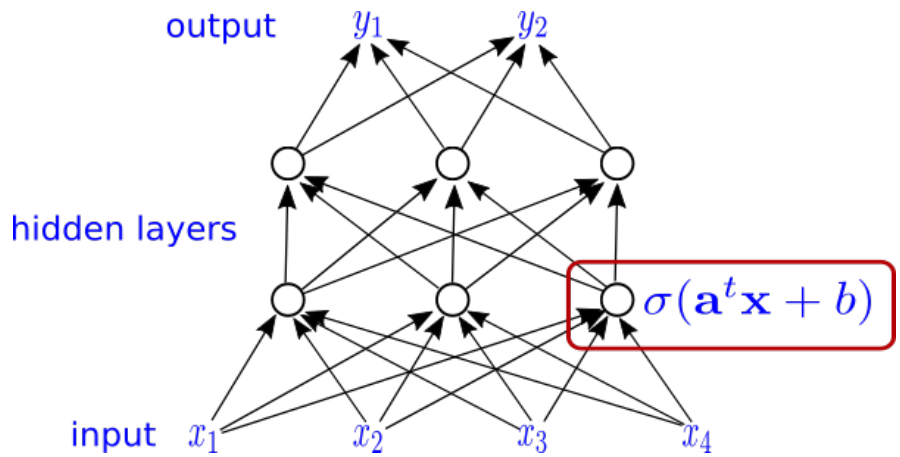
Some statistical theory for deep neural networks



Johannes Schmidt-Hieber

Deep neural networks

Network architecture $(L, \mathbf{p})$ consists of

- a positive integer L called the *number of hidden layers/depth*
- *width vector* $\mathbf{p} = (p_0, \ldots, p_{L+1}) \in \mathbb{N}^{L+2}$.

Neural network with network architecture $(L, \mathbf{p})$

$$f : \mathbb{R}^{p_0} \to \mathbb{R}^{p_{L+1}}, \quad \mathbf{x} \mapsto f(\mathbf{x}) = W_{L+1}\sigma_{\mathbf{v}_L} W_L \sigma_{\mathbf{v}_{L-1}} \cdots W_2 \sigma_{\mathbf{v}_1} W_1 \mathbf{x},$$

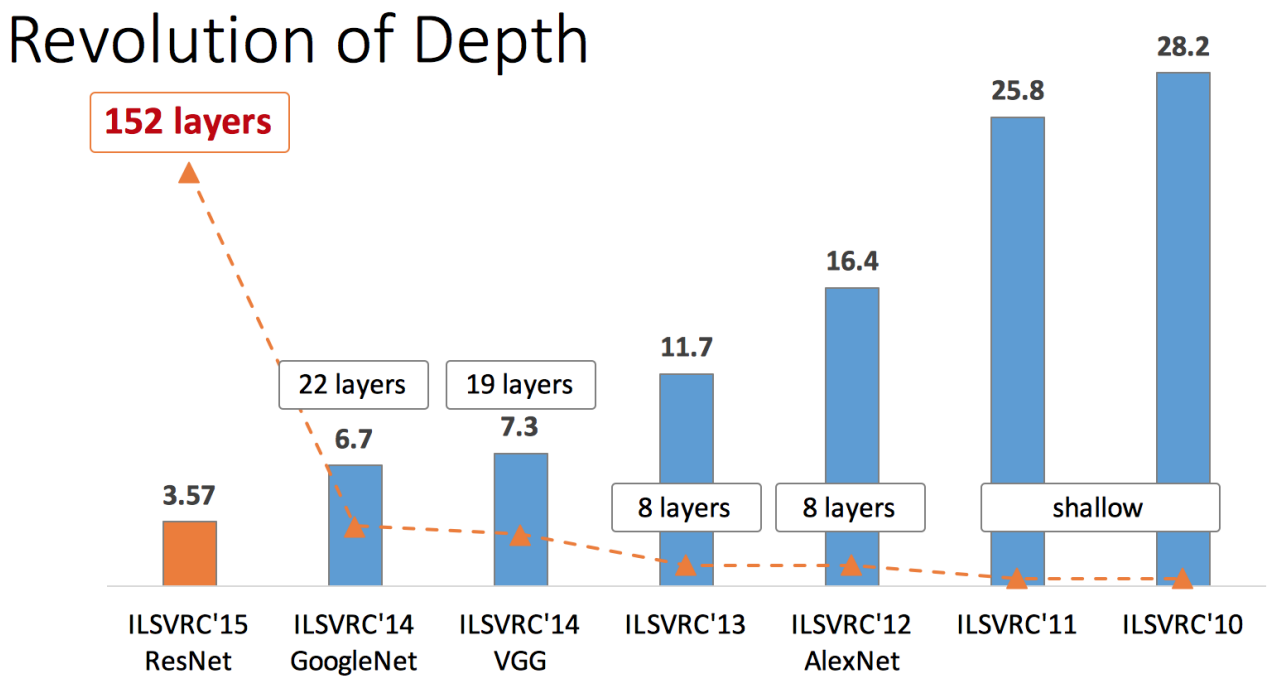
Network parameters:

- W_i is a $p_i \times p_{i-1}$ matrix
- $\mathbf{v}_i \in \mathbb{R}^{p_i}$

Activation function:

- We study the ReLU activation function $\sigma(x) = \max(x, 0)$.

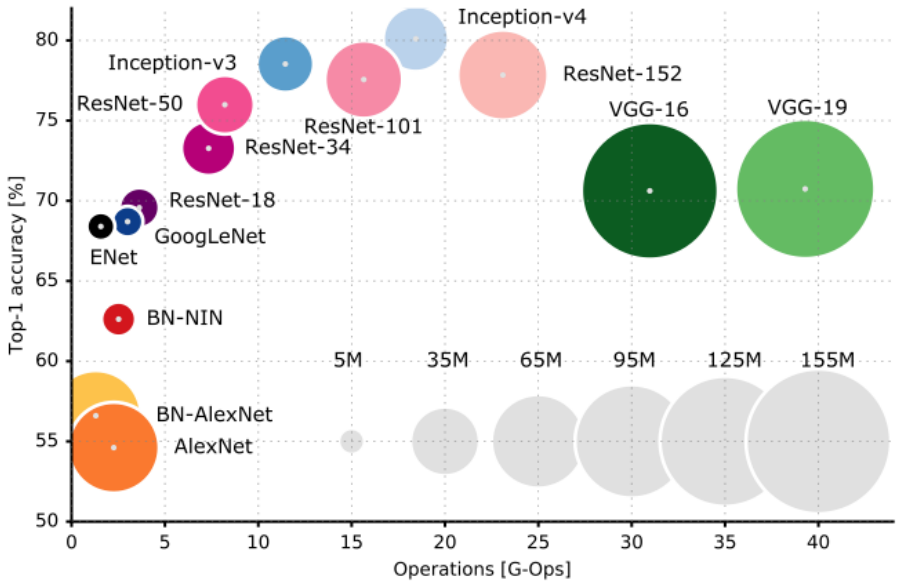
Depth



Source: Kaiming He, Deep Residual Networks

- Networks are deep
  - version of ResNet with 152 hidden layers
  - networks become deeper

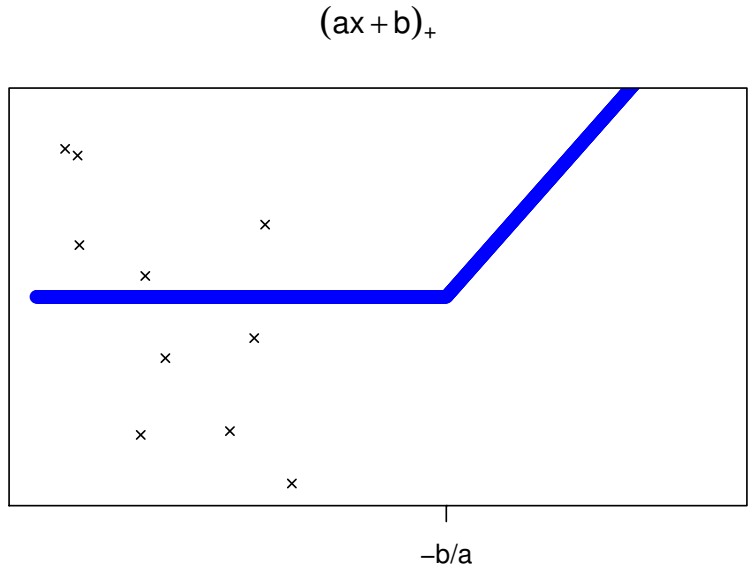
High-dimensionality



Source: arxiv.org/pdf/1605.07678.pdf

- Number of network parameters is larger than sample size
  - AlexNet uses 60 million parameters for 1.2 million training samples

Network sparsity

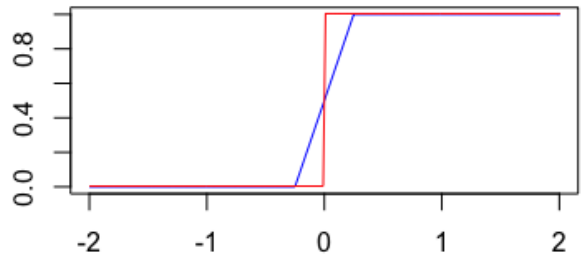


- There is some sort of sparsity on the parameters
- units get deactivated by initialization or by learning
- we assume instead sparsely connected networks

The large parameter trick

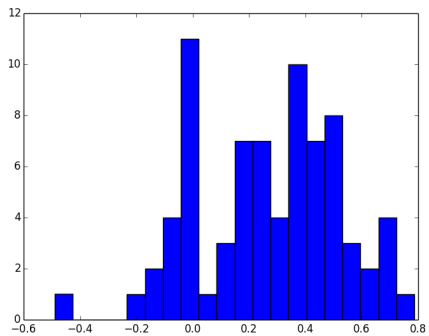
- with arbitrarily large network parameters we can approximate the indicator function via

$$x \mapsto \sigma(ax) - \sigma(ax - 1)$$



- it is common in approximation theory to use networks with network parameters tending to infinity

- in practice, network parameters are typically small



we restrict all network parameters in absolute value by one

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, ...)

The estimator

- choose network architecture $(L, \mathbf{p})$ and sparsity s
- denote by $\mathcal{F}(L, \mathbf{p}, s)$ the class of all networks with
  - architecture $(L, \mathbf{p})$
  - number of active (e.g. non-zero) parameters is s
- our theory applies to any estimator $\widehat{f}_n$ taking values in $\mathcal{F}(L, \mathbf{p}, s)$
- prediction error

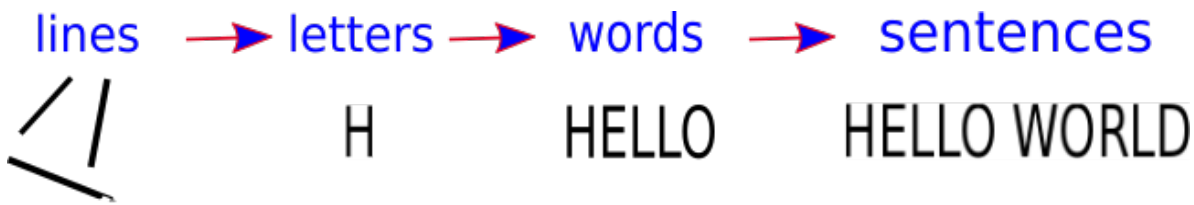
$$R(\widehat{f}_n, f) := E_f\left[\left(\widehat{f}_n(\mathbf{X}) - f(\mathbf{X})\right)^2\right],$$

  with $\mathbf{X} \stackrel{\mathcal{D}}{=} \mathbf{X}_1$ being independent of the sample
- study the dependence of n on $R(\widehat{f}_n, f)$

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**

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

Example: Additive models

- In an additive model

$$f(\mathbf{x}) = \sum_{i=1}^{d} f_i(x_i)$$

- This can be written as $f = g_1 \circ g_0$ with

$$g_0(\mathbf{x}) = (f_i(x_i))_{i=1,\dots,d}, \quad g_1(\mathbf{y}) = \sum_{i=1}^{d} y_i.$$

  Hence, $t_0 = 1$, $d_1 = t_1 = d$.
- Decomposes additive functions in
  - one function that can be non-smooth but every component is one-dimensional
  - one function that has high-dimensional input but the function is smooth

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
- in particular the depth scales with the sample size
- 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

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

- $f(\mathbf{x}) = h(x_1 + \ldots + x_d)$
- for some α-smooth function h
- Rate for DNNs $\lesssim n^{-\alpha/(2\alpha+1)}$ (up to logarithmic factors)
- Rate for best wavelet thresholding estimator $\gtrsim n^{-\alpha/(2\alpha+d)}$
- Reason: Low-dimensional structure does not affect the decay of the wavelet coefficients

MARS

- consider products of ramp functions

$$h_{I,\boldsymbol{t}}(x_1, \ldots, x_d) = \prod_{j \in I} \left(\pm (x_j - t_j) \right)_+$$

- piecewise constant in each component
- MARS (multivariate adaptive regression splines) fits linear combinations of such functions to data
- greedy algorithm
- has depth and width type parameters

Comparison with MARS

- how does MARS compare to ReLU networks?
- functions that can be represented by s parameters with respect to the MARS function system can be represented by $s \log(1/\varepsilon)$-sparse DNNs up to sup-norm error ε

Comparison with MARS (ctd.)

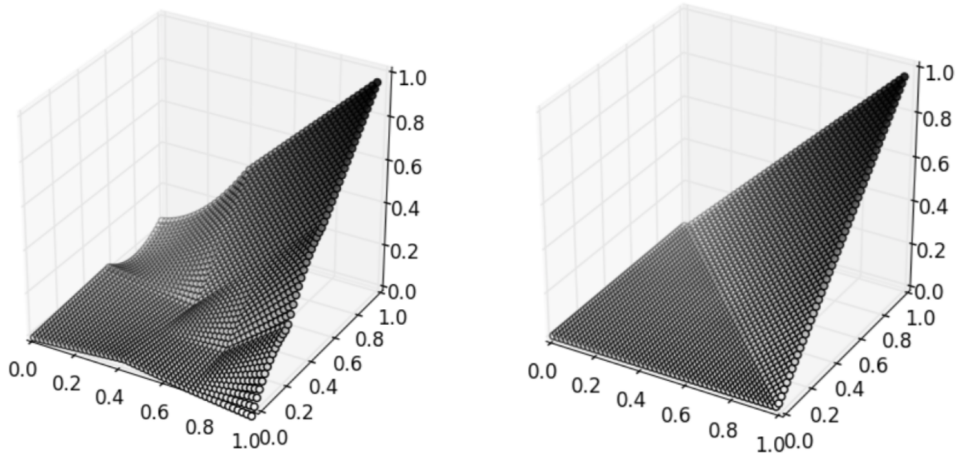


Figure: Reconstruction using MARS (left) and networks (right)

- the opposite is not true, one counterexample is

$$f(x_1, x_2) = (x_1 + x_2 - 1)_+$$

- we need $\gtrsim \varepsilon^{-1/2}$ many parameters to get ε-close with MARS functions
- $\rightsquigarrow$ conclusion: DNNs work better for correlated design

References:

- SH (2017). Nonparametric regression using deep neural networks with ReLU activation function. arxiv:1708.06633
- Eckle, SH (2018). A comparison of deep networks with ReLU activation function and linear spline-type methods. arXiv:1804.02253