Nonparametric regression using deep neural networks with ReLU activation function

Johannes Schmidt-Hieber

February 2018
Caltech

- Many impressive results in applications ...
- Lack of theoretical understanding ...

Algebraic definition of a deep net

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 $\sigma(x) = \max(x, 0)$.

Equivalence to graphical representation

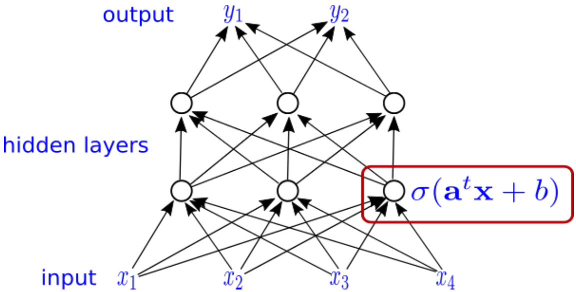


Figure: Representation as a direct graph of a network with two hidden layers $L = 2$ and width vector $\mathbf{p} = (4, 3, 3, 2)$.

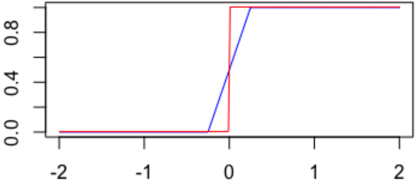
Characteristics of modern deep network architectures

- Networks are deep
  - version of ResNet with 152 hidden layers
  - networks become deeper
- Number of network parameters is larger than sample size
  - AlexNet uses 60 million parameters for 1.2 million training samples
- There is some sort of sparsity on the parameters
- ReLU activation function ($\sigma(x) = \max(x, 0)$)

The large parameter trick

- If we allow the network parameters to be arbitrarily large, then 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 our analysis, we **restrict all network parameters in absolute value by one**

Statistical analysis

- we want to study the **statistical** performance of a deep network
- $\rightsquigarrow$ do nonparametric regression
- 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

- 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
- choose network architecture $(L, \mathbf{p})$ and sparsity s
- least-squares estimator

$$\widehat{f}_n \in \operatorname*{argmin}_{f \in \mathcal{F}(L, \mathbf{p}, s)} \sum_{i=1}^n \big(Y_i - f(\mathbf{X}_i)\big)^2.$$

- this is the **global minimizer** [not computable]
- prediction error

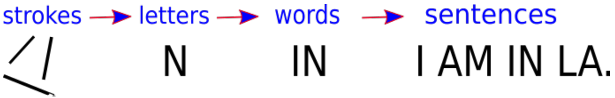
$$R(\widehat{f}_n, f) := E_f\big[\big(\widehat{f}_n(\mathbf{X}) - f(\mathbf{X})\big)^2\big],$$

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

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

Example

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

$$g_0(\mathbf{x}) = (f_i(x_i))_{i=1,\dots,d}, \quad g_2(\mathbf{y}) = \sum_{i=1}^{d} y_i.$$

  Hence, $t_0 = 1, d_1 = t_2 = d$.
- 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

The effective smoothness

For nonparametric regression,

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

Effective smoothness:

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

β_i^* is the smoothness induced on f by g_i

Main result

Theorem: If

(i) depth $\asymp \log n$

(ii) width $\asymp n^C$, with $C \geq 1$

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

Then,

$$R(\widehat{f}, f) \lesssim \max_{i=0,\ldots,q} n^{-\frac{2\beta_i^*}{2\beta_i^* + t_i}} \log^2 n.$$

Remarks on the rate

Rate:

$$R(\widehat{f}, f) \lesssim \max_{i=0,\ldots,q} n^{-\frac{2\beta_i^*}{2\beta_i^*+t_i}} \log^2 n.$$

Remarks:

- t_i can be seen as an effective dimension
- strong heuristic that this is the **optimal rate** (up to $\log^2 n$)
- other methods such as wavelets likely do not achieve these rates

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 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!!!**

On the proof

- Oracle inequality (roughly)

$$R(\widehat{f}, f) \lesssim \inf_{f^* \in \mathcal{F}(L,\mathbf{p},s,F)} \left\| f^* - f \right\|_\infty^2 + \frac{s \log n}{n}.$$

  - shows the trade-off between approximation and the number of active parameters s
- Approximation theory:
  - builds on work by Telgarsky (2016), Liang and Srikant (2016), Yarotski (2017)
  - network parameters bounded by one
  - explicit bounds on network architecture and sparsity

Additive models (ctd.)

- Consider again the additive model

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

- suppose that each function f_i is β-smooth
- the theorem gives the rate

$$R(\widehat{f}, f) \lesssim n^{-\frac{2\beta}{2\beta+1}} \log^2 n.$$

- this rate is known to be optimal up to the $\log^2 n$-factor

The function class considered here contains other structural constraints as a special case such a generalized additive models and it can be shown that the rates are optimal up to the $\log^2 n$-factor

Extensions

Some extensions are useful. To name a few

- high-dimensional input
- include stochastic gradient descent
- classification
- CNNs, recurrent neural networks, ...