Statistical theory for deep neural networks

Lecture 3

Johannes Schmidt-Hieber
outline

- statistical risk bounds
- theoretical comparison with other nonparametric methods
we observe $n$ i.i.d. copies $(X_1, Y_1), \ldots, (X_n, Y_n)$,

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

- $X_i \in \mathbb{R}^d$, $Y_i \in \mathbb{R}$,
- goal is to reconstruct the function $f : \mathbb{R}^d \to \mathbb{R}$
- has been studied extensively
  (kernel smoothing, wavelets, splines, ...)
choose network architecture \((L, p)\) and sparsity \(s\)

denote by \(\mathcal{F}(L, p, s)\) the class of all networks with

- architecture \((L, p)\)
- number of active (e.g. non-zero) parameters is \(s\)

our theory applies to any estimator \(\hat{f}_n\) taking values in \(\mathcal{F}(L, p, s)\)

prediction error

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

with \(X \overset{\mathcal{D}}{=} X_1\) being independent of the sample

study the dependence of \(n\) on \(R(\hat{f}_n, f)\)
function class

• classical idea: assume that regression function is $\beta$-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
• $\leadsto$ make a good structural assumption on $f$
hierarchical structure

- lines $\rightarrow$ letters $\rightarrow$ words $\rightarrow$ sentences

Important: Only few objects are combined on deeper abstraction level

- few letters in one word
- few words in one sentence
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 \( \beta_i \)-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

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

similar conditions have been proposed by Horowitz & Mammen (2007), Kohler & Kryzak (2017), Bauer & Kohler (2017), Kohler & Langer (2018).
example

\[ f_0(x_1, x_2, x_3) = g_{11}(g_{01}(x_3), g_{02}(x_2)) \]

- \( f_0 = g_1 \circ g_0 \)
- \( d_0 = 3, \ t_0 = 1, \ d_1 = t_1 = 2, \ d_2 = 1 \)
**main result**

**Theorem:** If

(i) depth $\preceq \log n$

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

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

prediction error $\preceq \phi_n + \Delta_n$

(up to log $n$-factors) with

$$\Delta_n := E \left[ \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{f}_n(X_i))^2 - \inf_{f \in \mathcal{F}(L,p,s)} \frac{1}{n} \sum_{i=1}^{n} (Y_i - f(X_i))^2 \right]$$

and

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

**Theorem:** If

(i) depth $\preceq \log n$

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

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

$$\text{prediction error} \preceq \phi_n + \Delta_n$$

(up to log $n$-factors) with

$$\Delta_n := E\left[\frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{f}_n(X_i))^2 - \inf_{f \in \mathcal{F}(L,p,s)} \frac{1}{n} \sum_{i=1}^{n} (Y_i - f(X_i))^2\right]$$

and

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

- empirical risk minimizer is optimal in this class
- problem is high-dimensional (no upper bound on the width)
- network sparsity induces regularization
- the assumption that depth $\approx \log n$ appears naturally
- in particular the depth scales with 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

~~> non-linearity is essential!!!
additive models

- functions are of the form

\[ f(x_1, \ldots, x_d) = f_1(x_1) + \ldots + f_d(x_d) \]

- \( f_i \) are \( \beta \)-smooth

- \( f = g_1 \circ g_0 \) with

\[ g_0(x) = (f_1(x_1), \ldots, f_d(x_d))^\top \quad \text{and} \quad g_1(y) = \sum_{j=1}^{d} y_j \]

- \( \sim \) \( d_0 = d, \ t_0 = 1, \ d_1 = t_1 = d, \ d_2 = 1 \)

rate achieved by a neural network

\[ R(\hat{f}_n, f_0) \lesssim n^{-\frac{2\beta}{2\beta+1}} \log^3 n + \Delta(\hat{f}_n, f_0). \]
on the proof

- oracle inequality (roughly)

\[ R(\hat{f}, f) \lesssim \inf_{f^* \in \mathcal{F}(L,p,s)} \| f^* - f \|_\infty^2 + \frac{\log \mathcal{N}_n}{n}. \]

- \( \log \mathcal{N}_n \) denotes the covering entropy
- shows the trade-off between approximation and model size
- for networks we obtain a bound of the type

\[ \log \mathcal{N}_n \lesssim sL \log(n) \]

- \( \leadsto \) trade-off between approximation and network sparsity
lower bounds on the network sparsity

the convergence theorem implies a deterministic lower bound on the network sparsity required to approximate $\beta$-smooth functions on $[0, 1]^d$

**Result:**

- if for $\varepsilon > 0$,

$$s \lesssim \frac{\varepsilon^{-d/\beta}}{L \log(1/\varepsilon)}$$

then

$$\sup_{f_0 \text{ is } \beta-\text{Hölder}} \inf_{f \text{ a } s\text{-sparse network}} \|f - f_0\|_\infty \geq \varepsilon.$$ 

- has been proved via a different technique in Bölcskei et al. '17
other statistical results

- piecewise smooth functions, Imaizumi and Fukumizu ’18
- binary classification with hinge loss, Kim, Ohn, Kim ’18
suboptimality of wavelet estimators

- \( f(x) = h(x_1 + \ldots + x_d) \)
- for some \( \alpha \)-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

\[ h_{l,t}(x_1, \ldots, x_d) = \prod_{j \in l} \left( \pm (x_j - t_j) \right)_+ \]

- 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 $\varepsilon$
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 $\varepsilon$-close with MARS functions
- $\leadsto$ conclusion: DNNs work better for correlated design
energy landscape

**Definition:**
- data \((X, Y) \in (\mathcal{X}, \mathcal{Y})\)
- class of functions \(F_\theta : \mathcal{X} \rightarrow \mathcal{Y}\)
- loss function \(L : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+\)

the energy landscape/loss surface is the function

\[
\theta \mapsto L(Y, F_\theta(X)).
\]
critical points of the energy landscape

- local/global minima
- saddle points
- (bad) saddle points (Hessian vanishes)
linear activation function

- fit a linear regression line $f(y) = ax$ to data $(X_i, Y_i)_{i=1,...,n}$
- $a, b$ parameters
- energy landscape for squared loss
  $$(a, b) \mapsto \sum_{i=1}^{n} (Y_i - abX_i)^2.$$ 
- saddle point for $a = b = 0$
- global minimum whenever $ab = \text{least squares solution}$
same setting as before but now we consider $f(x) = abc x$

$\bullet \ a = b = c = 0$ is a bad saddle point

**Kawaguchi ’16:**

$\bullet \ f(x) = W_L W_{L-1} \ldots W_0 x$

$\bullet \ $ every local minimum is a global minimum

$\bullet \ $ saddle points exist (if $L > 1$, there exist bad saddle points)

$\bullet \ $ proof is based on studying local perturbations
ReLU activation function

\[(ax + b)_+\]

- possible many local minima that are not global minima
- happens in practice
- very dependent on initialization
interpolation properties

- consider continuous activation function that is not a polynomial
- given data \((X_k, Y_k) \in \mathbb{R}^d \times \mathbb{R}\) with distinct design vectors \(X_k\)
- shallow networks: one can perfectly interpolate \(n\) data points with \(n\) units in the hidden layer
- related to the universal approximation theorem (therefore same condition appears)
vanishing training error

”in deep learning zero training error still generalizes well”

- it always depends on the problem
- many applications have little noise and interpolation is a good idea
- additive noise models are different and claim is probably false
theory for vanishing training error

smooth activation function

- Du et al. ’18 consider highly over-parametrized setting
- number of units in each layer has to be of some (unspecified ?) polynomial order in the sample size
- setup is regression with least-squares loss
- show that gradient descent with randomly initialization converges to zero training error

ReLU networks

- Allen-Zhu et al. ’18 shows a similar result
- one assumptions is that the network width scales at least with the 30-th power of the sample size
Deep networks are an exciting field with many open problems

- Classification, high-dimensional input, ...
- Energy landscape
- Network types: CNNs, RNNs, autoencoders, ...
- Generative adversarial networks (GANs)

Thank you for your attention!