Predictive Risk Rates of Neural Networks

Louis Christie & Tobias Freidling

November 16, 2020

Contents

Non-Parametric Regression

Suppose we observe iid $(X_i, Y_i) \in [0,1]^d \times \mathbb{R}$, which we assume are of the form:

$$Y_i = f_0(X_i) + \epsilon_i$$

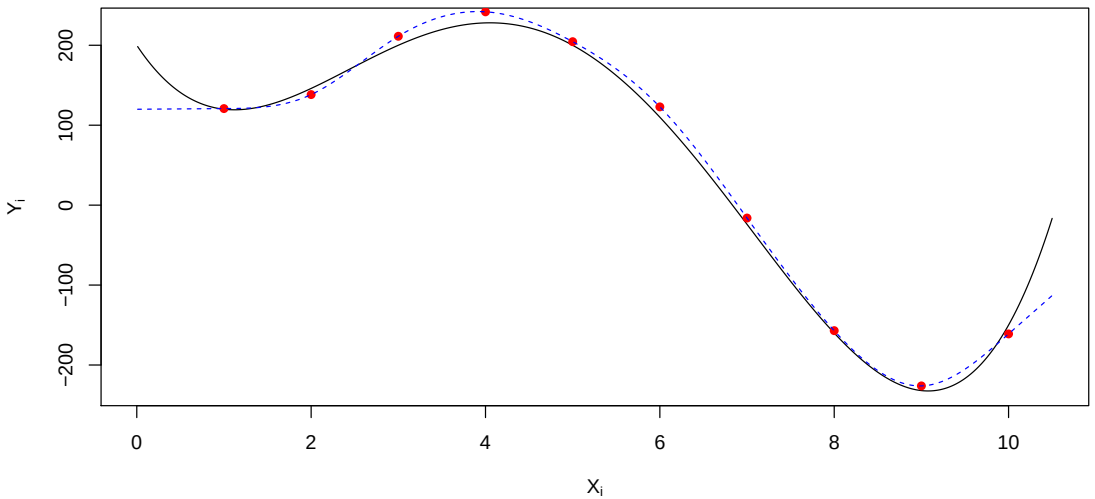
for some iid mean zero ϵ_i which are independent of all X_i.

Non-Parametric Regression

Suppose we observe iid $(X_i, Y_i) \in [0,1]^d \times \mathbb{R}$, which we assume are of the form:

$$Y_i = f_0(X_i) + \epsilon_i$$

for some iid mean zero ϵ_i which are independent of all X_i.



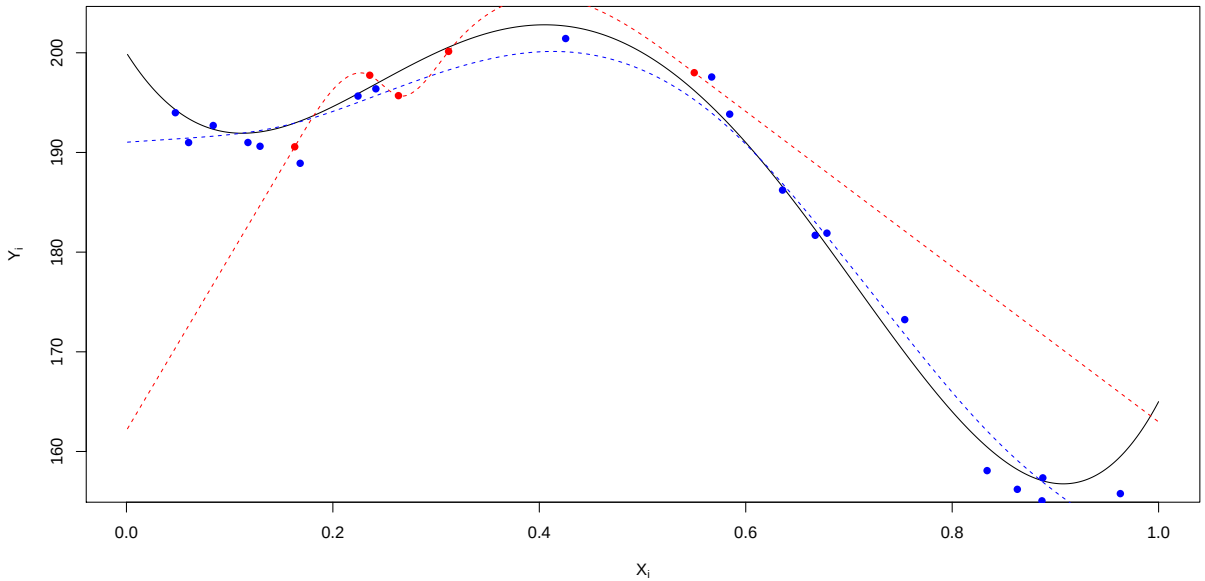
,

Non-Parametric Regression

How well can we estimate f_0?

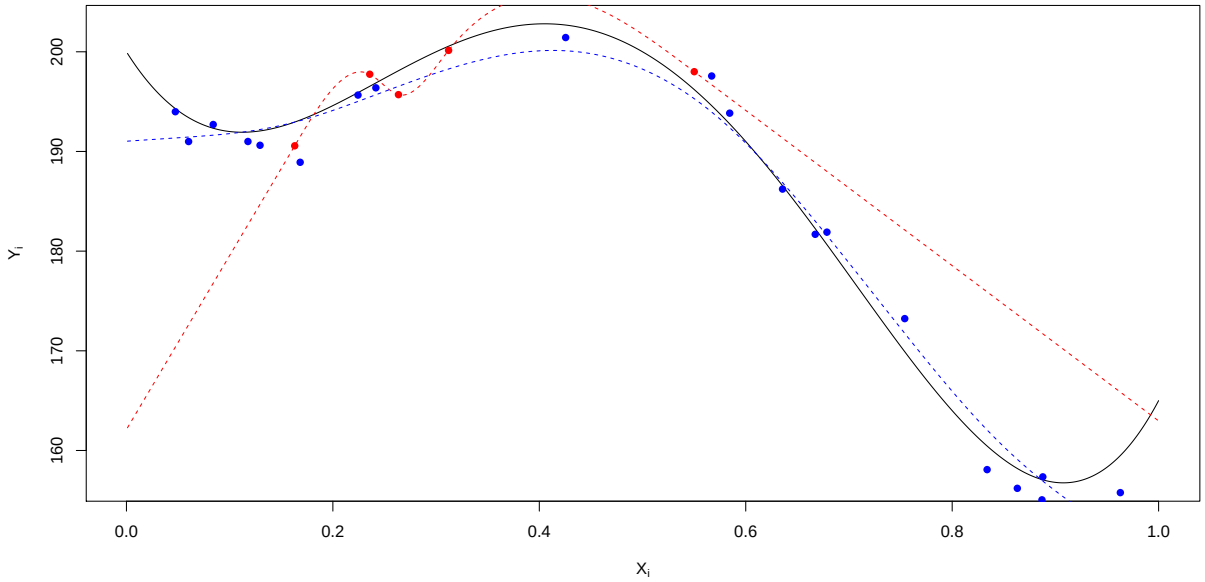
Non-Parametric Regression

How well can we estimate f_0?



Non-Parametric Regression

How well can we estimate f_0?

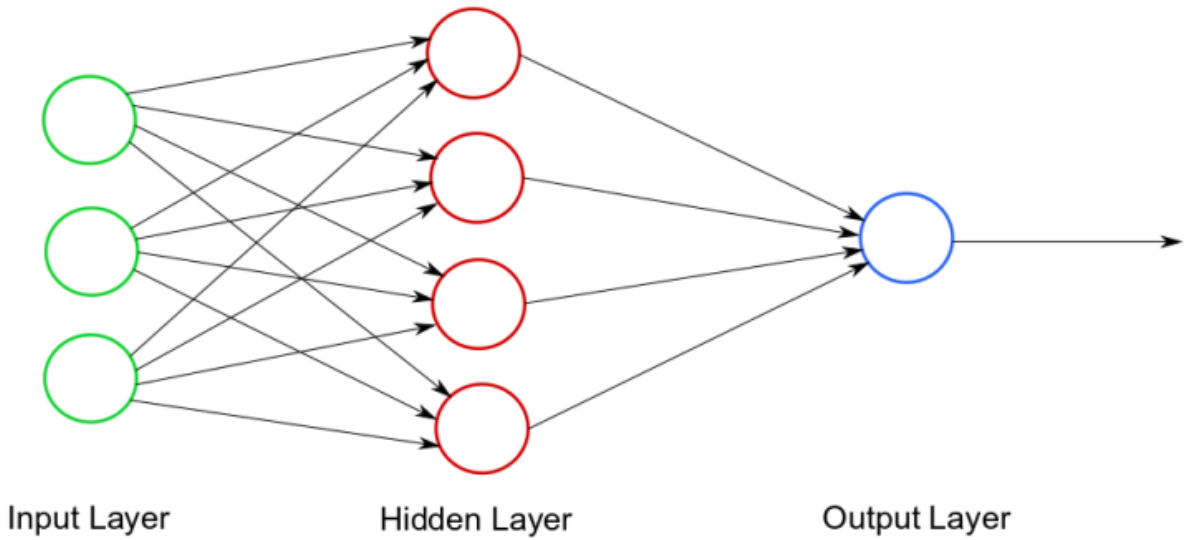


Theorem

If f_0 is β Hölder smooth then for any wavelet estimator $\hat{f}$:

$$\sup_{f_0} R(\hat{f}, f_0) = \sup_{f_0} \mathbb{E}((\hat{f}(X) - f_0(X))^2) \gtrsim n^{-2\beta/(2\beta+d)}$$

Neural Networks



Where the value at each node is given by $\sigma(a^T x + b)$

Neural Networks

Each node takes value $\sigma(a^T x + b)$, so we can represent the neural network as the composition:

$$f = W_L \sigma_{v_L} W_{L-1} \cdots W_1 \sigma_{v_1} W_0 \in \mathcal{N}(L, \mathrm{p})$$

Where $p \in \mathrm{N}^L$ are layer widths, $W_\ell \in \mathbb{R}^{p_\ell \times p_{\ell+1}}$ are weight matrices,

$$\sigma_v(x) = \begin{bmatrix} \sigma(x_1 + v_1) \\ \vdots \\ \sigma(x_m + v_m) \end{bmatrix}$$

and all coefficients in $[-1, 1]$.

Neural Networks

What we know:

- Universal approximation theorem: Any $f \in C(K, \mathbb{R})$, $K \subset \mathbb{R}^d$ compact, can be approximated by a 1-layer NN w.r.t. $\|\cdot\|_\infty$.
- Can show convergence in specific cases: e.g. $\frac{d}{n} \log n$ when considering a binomial classifier with smooth activation functions σ.
- They appear to break the curse of dimensionality in some cases.

Neural Networks

What we don't know:

- Convergence rates when $L > 2$
- What happens with non-smooth activation functions, e.g. ReLU $\sigma(x) = \max\{0, x\}$?

Neural Networks

What we don't know:

- Convergence rates when $L > 2$
- What happens with non-smooth activation functions, e.g. ReLU $\sigma(x) = \max\{0, x\}$?

What does this paper do?

Neural Networks

What we don't know:

- Convergence rates when $L > 2$
- What happens with non-smooth activation functions, e.g. ReLU $\sigma(x) = \max\{0, x\}$?

What does this paper do?

It gives dimension independent rates for ReLU on deep networks.

How well can Neural Networks to NPR?

If $\hat{f} \in \mathcal{N}(L, p)$ is a NN that approximates f_0, we want to find asymptotic bounds for the prediction error:

$$R(\hat{f}, f_0) = \mathbb{E}((\hat{f}(X) - f_0(X))^2)$$

How well can Neural Networks to NPR?

If $\hat{f} \in \mathcal{N}(L, p)$ is a NN that approximates f_0, we want to find asymptotic bounds for the prediction error:

$$R(\hat{f}, f_0) = \mathbb{E}((\hat{f}(X) - f_0(X))^2)$$

This is too hard *(as it stands)*

How well can Neural Networks to NPR?

If $\hat{f} \in \mathcal{N}(L, p)$ is a NN that approximates f_0, we want to find asymptotic bounds for the prediction error:

$$R(\hat{f}, f_0) = \mathbb{E}((\hat{f}(X) - f_0(X))^2)$$

This is too hard *(as it stands)*

So we restrict to "well-chosen" function classes:

$$\hat{f} \in \mathcal{F} \subseteq \mathcal{N}(L, \mathrm{p}) \qquad \text{and} \qquad f_0 \in \mathcal{G}$$

How well can Neural Networks to NPR?

If $\hat{f} \in \mathcal{N}(L, p)$ is a NN that approximates f_0, we want to find asymptotic bounds for the prediction error:

$$R(\hat{f}, f_0) = \mathbb{E}((\hat{f}(X) - f_0(X))^2)$$

This is too hard *(as it stands)*

So we restrict to "well-chosen" function classes:

$$\hat{f} \in \mathcal{F} \subseteq \mathcal{N}(L, \mathrm{p}) \qquad \text{and} \qquad f_0 \in \mathcal{G}$$

Choosing Spaces to Optimise over

Our estimator space:

$$\mathcal{F}(L, \mathrm{p}, s, F) = \left\{ f \in \mathcal{N}(L, \mathrm{p}) : \sum_{j=0}^{L} \|W_j\|_0 + |v_j|_0 \leq s, \|f\|_\infty \leq F \right\}$$

I.e., s-sparse networks with bounded outputs.

Choosing Spaces to Optimise over

Our target space:

$$\mathcal{G}(q, \mathrm{d}, \mathrm{t}, \beta, K) = \{f = g_q \circ g_{q-1} \circ \cdots \circ g_0 : g_i \in \mathcal{G}_i(\mathrm{d}, \mathrm{t}_i, \beta_i, K)\}$$

Choosing Spaces to Optimise over

Our target space:

$$\mathcal{G}(q, \mathrm{d}, \mathrm{t}, \beta, K) = \{f = g_q \circ g_{q-1} \circ \cdots \circ g_0 : g_i \in \mathcal{G}_i(\mathrm{d}, \mathrm{t}_i, \beta_i, K)\}$$

$$\mathcal{G}_i(\mathrm{d}, \mathrm{t}_i, \beta_i, K) = \Big\{ g_i : [a_i, b_i]^{d_i} \to [a_{i+1}, b_{i+1}]^{d_{i+1}} : \\ g_{ij} \text{ depends on only } t_i \text{ variables }, \\ \text{has Hölder smoothness index } \beta_i, \textit{and} \\ \sum_{\alpha:|\alpha|<\beta} \|\partial^\alpha g_{ij}\|_\infty + \sum_{\alpha:|\alpha|=\lfloor\beta\rfloor} \sup_{x\neq y\in[a_i,b_i]} \frac{|\partial^\alpha g_{ij}(x) - \partial^\alpha g_{ij}(y)|}{|x-y|_\infty^{\beta-\lfloor\beta\rfloor}} \leq K \Big\}$$

i.e., compositions of β_i Hölder-smooth functions of radius K that depend on at most t_i variables.

Functions in $\mathcal{G}$

(Generalised) Additive Models:

$$f_0(x_1, \ldots, x_d) = h\left(\sum_{j=1}^{n} f_j(x_j)\right) = h \circ g_1 \circ g_0(x)$$

Where $g_0(x) = (f_1(x_1), \ldots, f_d(x_d))^T$ and $g_1(x) = \sum_{j=1}^{d} x_j$.

Functions in $\mathcal{G}$

(Generalised) Additive Models:

$$f_0(x_1, \ldots, x_d) = h\left(\sum_{j=1}^{n} f_j(x_j)\right) = h \circ g_1 \circ g_0(x)$$

Where $g_0(x) = (f_1(x_1), \ldots, f_d(x_d))^T$ and $g_1(x) = \sum_{j=1}^{d} x_j$.

If each f_i is β-smooth or radius K and h is γ-smooth of radius K then:

$$f_0 : [0,1] \xrightarrow{g_0} [-K, K]^d \xrightarrow{g_1} [-Kd, Kd] \xrightarrow{h} [-K, K]$$

and one can show:

$$f_0 \in \mathcal{G}\left(2, \begin{bmatrix} d \\ d \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ d \\ 1 \end{bmatrix}, \begin{bmatrix} \beta \\ (\beta \vee 2)d \\ \gamma \end{bmatrix}, (K+1)d\right)$$

Why did we choose $\mathcal{G}$?

Recall that if f_0 is Hölder-smoothness index β then the optimal rate of convergence is $n^{-\frac{2\beta}{2\beta+d}}$.

Why did we choose $\mathcal{G}$?

Recall that if f_0 is Hölder-smoothness index β then the optimal rate of convergence is $n^{-\frac{2\beta}{2\beta+d}}$.

What can be said about $f_0 \in \mathcal{G}$?

Why did we choose $\mathcal{G}$?

Recall that if f_0 is Hölder-smoothness index β then the optimal rate of convergence is $n^{-\frac{2\beta}{2\beta+d}}$.

What can be said about $f_0 \in \mathcal{G}$?

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

These effective smoothness indices describe the optimal convergence rate via:

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

Main Result

How close can networks in $\mathcal{F}(L, p, s, F)$ get to the optimal rate?

[1] These are a lower bound on F, asymptotic bounds on L, an asymptotic lower bound on the widths p_i, and asymptotic bounds on s. All w.r.t. ϕ_n.

Main Result

How close can networks in $\mathcal{F}(L, p, s, F)$ get to the optimal rate?

Theorem

If $f_0 \in \mathcal{G}(q, \mathrm{d}, \mathrm{t}, \beta, K)$ and each estimator $\hat{f}_n \in \mathcal{F}(L, \mathrm{p}, s, F)$ where the classes satisfy some technical requirements[1] then there exist constants A and B (depending only on q, d, t, β and F) such that: if $\Delta_n(\hat{f}_n, f_0) \leq A\phi_n L \log^2(n)$ then

$$R(\hat{f}_n, f_0) \leq B\phi_n L \log^2(n)$$

and otherwise:

$$B^{-1}\Delta_n(\hat{f}_n, f_0) \leq R(\hat{f}_n, f_0) \leq B\Delta_n(\hat{f}_n, f_0)$$

[1] These are a lower bound on F, asymptotic bounds on L, an asymptotic lower bound on the widths p_i, and asymptotic bounds on s. All w.r.t. ϕ_n.

Main Result

Here $\Delta_n(\hat{f}, f_0)$ is given by:

$$\Delta_n(\hat{f}, f_0) = \mathbb{E}_{f_0}\left(\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{f}(X_i))^2 - \inf_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}(Y_i - f(X_i))^2\right)$$

I.e., how "close" are we to an empirical risk minimiser. Thus the result can be restated:

If we can minimise our empirical risk, then neural networks achieve (nearly) optimal convergence rates

Application to Generalised Linear Models

Recall $f_0(x) = h(\sum_{j=1}^{d} f_j(x_j)$ with each f_i β-smooth and h γ-smooth, so

$$f_0 \in \mathcal{G}\left(2, \begin{bmatrix} d \\ d \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ d \\ 1 \end{bmatrix}, \begin{bmatrix} \beta \\ (\beta \vee 2)d \\ \gamma \end{bmatrix}, (K+1)d\right)$$

If $\hat{f}$ is an estimator in $\mathcal{F}(L, p, s, F)$ (satisfying the technical requirements of the main result) then:

$$R(\hat{f}, f_0) \lesssim \left(n^{-\frac{2\beta(\gamma \wedge 1)}{2\beta(\gamma \wedge 1)+1}} + n^{-\frac{2\gamma}{2\gamma+1}}\right) \log^3(n) + \Delta(\hat{f}, f_0)$$

High-level proof idea

Risk and empirical risk:

$$R(\hat{f}, f_0) = \mathbb{E}[(\hat{f}(X) - f_0(X))^2], \quad \hat{R}_n(\hat{f}, f_0) = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(\hat{f}(X_i) - f_0(X_i))^2\right]$$

Relating R and $\hat{R}_n$:

$$R(\hat{f}, f_0) \lesssim \hat{R}_n(\hat{f}, f_0) + \tau_1(s, L)$$

Decomposition into approximation and estimation error:

$$\begin{aligned}\hat{R}_n(\hat{f}, f_0) &\lesssim \inf_{f \in \mathcal{F}} \mathbb{E}[(f(X) - f_0(X))^2] + \Delta_n(\hat{f}, f_0) + \tau_2(s, L) \\ &\lesssim \inf_{f \in \mathcal{F}} \|f - f_0\|_\infty^2 + \Delta_n(\hat{f}, f_0) + \tau_2(s, L)\end{aligned}$$

Function class $\mathcal{G}(q, \mathrm{d}, \mathrm{t}, \beta, K)$

Asssuming that $f_0 \in \mathcal{G}$ is a compromise between parametric, i.e. $f_0 \in \mathcal{F}$, and non-parametric, i.e. distribution-free, models.

Function class $\mathcal{G}(q, \mathrm{d}, \mathrm{t}, \beta, K)$

Asssuming that $f_0 \in \mathcal{G}$ is a compromise between parametric, i.e. $f_0 \in \mathcal{F}$, and non-parametric, i.e. distribution-free, models.

Two approaches for choosing $\mathcal{G}$:

- realistic function class for the data
- function class for which neural networks perform well

Function class $\mathcal{G}(q, \mathrm{d}, \mathrm{t}, \beta, K)$

Asssuming that $f_0 \in \mathcal{G}$ is a compromise between parametric, i.e. $f_0 \in \mathcal{F}$, and non-parametric, i.e. distribution-free, models.

Two approaches for choosing $\mathcal{G}$:

- realistic function class for the data
- function class for which neural networks perform well

Open question: What is the largest possible function class for a given rate?

Sparsity

Number of network parameters in a fully connected NN:
$\mathcal{O}(\sum_{i=0}^{L} p_i p_{i+1})$
The main result requires

$$s \asymp n \, \phi_n \log(n) \lesssim \min_{i \in \{1,\ldots,L\}} p_i \log(n)$$

Hence, only sparse networks are considered.

Sparsity

Number of network parameters in a fully connected NN:
$\mathcal{O}(\sum_{i=0}^{L} p_i p_{i+1})$
The main result requires

$$s \asymp n \, \phi_n \log(n) \lesssim \min_{i \in \{1,\dots,L\}} p_i \log(n)$$

Hence, only sparse networks are considered.

Sparsity is a non-standard assumption.

- need for regularisation in over-parametrised problem
- empirical counterexamples, e.g. first layer of VGG-19
- related results for fully connected NNs

Sparsity

Number of network parameters in a fully connected NN:
$\mathcal{O}(\sum_{i=0}^{L} p_i p_{i+1})$
The main result requires

$$s \asymp n \, \phi_n \log(n) \lesssim \min_{i \in \{1,\dots,L\}} p_i \log(n)$$

Hence, only sparse networks are considered.

Sparsity is a non-standard assumption.

- need for regularisation in over-parametrised problem
- empirical counterexamples, e.g. first layer of VGG-19
- related results for fully connected NNs

Open question: Can we derive the same convergence rate without the sparsity assumption?

Optimisation

The computation of $\hat{f}_n$ is not addressed in the paper. In practice, we solve

$$\min_{(W_j, v_j)_{j=0}^{L}} \frac{1}{n} \sum_{i=1}^{n} L(Y_i, f(X_i))$$

by a variant of stochastic gradient descent (SGD):

Optimisation

The computation of $\hat{f}_n$ is not addressed in the paper. In practice, we solve

$$\min_{(W_j, v_j)_{j=0}^{L}} \frac{1}{n} \sum_{i=1}^{n} L(Y_i, f(X_i))$$

by a variant of stochastic gradient descent (SGD):

while not converged **do**
  random shuffling of $(Y_i, X_i)_{i=0}^{n}$
  for all $i \in \{1, \ldots, n\}$ **do**
    $W_j \leftarrow W_j - \eta \nabla_W L(Y_i, X_i), \qquad \forall j \in \{0, \ldots, L\}$
    $v_j \leftarrow v_j - \eta \nabla_v L(Y_i, X_i), \qquad \forall j \in \{0, \ldots, L\}$
  end for
end while

Optimisation

Typically, the optimisation problem is overparametrised and non-convex.

- Practice: SGD finds local minima with good generalisation behaviour, i.e. no overfit
- Theory: no guarantee for convergence to local minimum, only heuristics for relationship between local and global minima

Optimisation

Typically, the optimisation problem is overparametrised and non-convex.

- Practice: SGD finds local minima with good generalisation behaviour, i.e. no overfit
- Theory: no guarantee for convergence to local minimum, only heuristics for relationship between local and global minima

Therefore, there's no guarantee that we can find $\hat{f}$ such that $\Delta_n(\hat{f}, f_0)$ is sufficiently small.
The randomness of SGD possibly entails "implicit regularisation".

Open question: How does SGD effect the performance of deep neural networks?

Curse of dimensionality

The main result achieves a convergence rate independent of the dimension d. However,

- the constant B depends on d.
- the rate ϕ_n depends exponentially on $t_i, i \in \{0, \ldots, q\}$.

Conclusion

- class of neural networks $\mathcal{F}(L, \mathrm{p}, s, F)$, class of regression functions $\mathcal{G}(q, \mathrm{d}, \mathrm{t}, \beta, K)$
- nearly optimal convergence rates for deep neural networks with ReLU activation function
- overcoming curse of dimensionality
- empirical risk minimisation
- stochastic gradient descent (SGD)