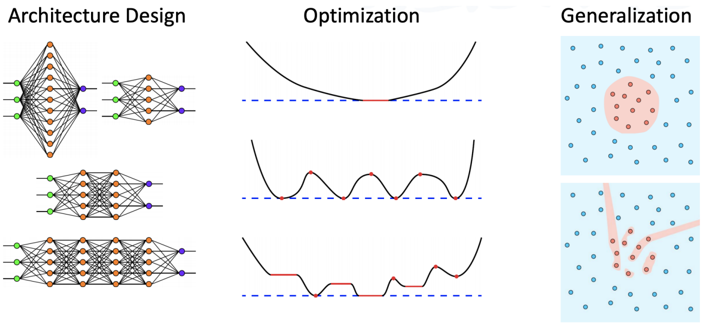
Neural Network Approximation Theory

Shpresim Sadiku

(Technische Universität Berlin & Zuse Institute Berlin)

9^{th} BMS Conference · March 4, 2021

Three Problems in Deep Learning



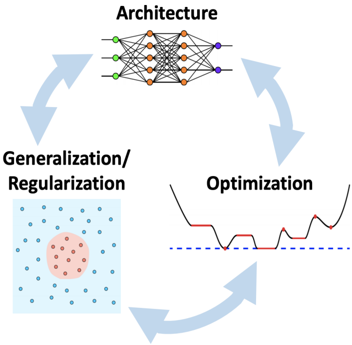
from: *Mathematics of Deep Learning*, René Vidal, DeepMath Plenary Lecture, 2020

The Three Problems are Interrelated

↪ It is easier to optimize some architectures than others (Haeffele et al., 2017)

↪ Generalization is strongly affected by architecture (Zhang et al., 2017)

↪ Optimization can impact generalization (Neyshabur et al., 2015, Zhou and Feng, 2017)

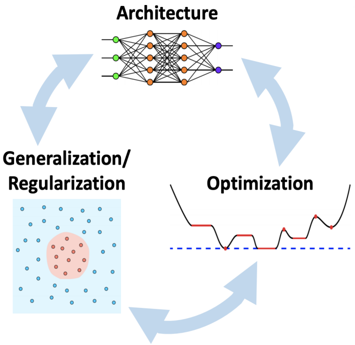


The Three Problems are Interrelated

↪ It is easier to optimize some architectures than others (Haeffele et al., 2017)

↪ Generalization is strongly affected by architecture (Zhang et al., 2017)

↪ Optimization can impact generalization (Neyshabur et al., 2015, Zhou and Feng, 2017)

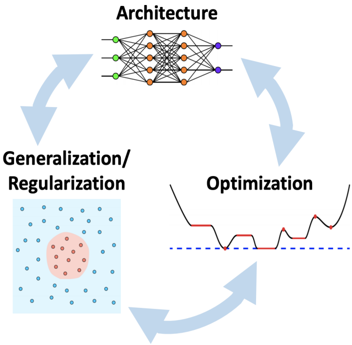


The Three Problems are Interrelated

↪ It is easier to optimize some architectures than others (Haeffele et al., 2017)

↪ Generalization is strongly affected by architecture (Zhang et al., 2017)

↪ Optimization can impact generalization (Neyshabur et al., 2015, Zhou and Feng, 2017)

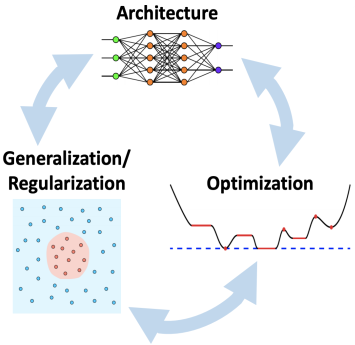


The Three Problems are Interrelated

↪ It is easier to optimize some architectures than others (Haeffele et al., 2017)

↪ Generalization is strongly affected by architecture (Zhang et al., 2017)

↪ Optimization can impact generalization (Neyshabur et al., 2015, Zhou and Feng, 2017)



Error Decomposition

$$R(f) - R^* = \underbrace{(R(f) - R(\hat{f}))}_{\text{optimization error}} + \underbrace{(R(\hat{f}) - R_{\mathcal{F}})}_{\text{estimation error}} + \underbrace{(R_{\mathcal{F}} - R^*)}_{\text{approximation error}}$$

for $R(f)$ the risk of a hypothesis f, $R^* = \inf_f R(f)$ the Bayes risk, $\hat{f}$ minimizer of the empirical risk $\hat{R}(f)$

Interplay of three areas

1. Learning
   ($\hookrightarrow$ Optimization, Optimal Control,...)
2. Generalization
   ($\hookrightarrow$ Statistics, Learning Theory, Stochastics,...)
3. Expressivity
   ($\hookrightarrow$ Approximation Theory, Applied Harmonic Analysis,...)

Error Decomposition

$$R(f) - R^* = \underbrace{(R(f) - R(\hat{f}))}_{\text{optimization error}} + \underbrace{(R(\hat{f}) - R_{\mathcal{F}})}_{\text{estimation error}} + \underbrace{(R_{\mathcal{F}} - R^*)}_{\text{approximation error}}$$

for $R(f)$ the risk of a hypothesis f, $R^* = \inf_f R(f)$ the Bayes risk, $\hat{f}$ minimizer of the empirical risk $\hat{R}(f)$

Interplay of three areas

1. Learning
   ($\hookrightarrow$ Optimization, Optimal Control,...)
2. Generalization
   ($\hookrightarrow$ Statistics, Learning Theory, Stochastics,...)
3. Expressivity
   ($\hookrightarrow$ Approximation Theory, Applied Harmonic Analysis,...)

Error Decomposition

$$R(f) - R^* = \underbrace{(R(f) - R(\hat{f}))}_{\text{optimization error}} + \underbrace{(R(\hat{f}) - R_{\mathcal{F}})}_{\text{estimation error}} + \underbrace{(R_{\mathcal{F}} - R^*)}_{\text{approximation error}}$$

for $R(f)$ the risk of a hypothesis f, $R^* = \inf_f R(f)$ the Bayes risk, $\hat{f}$ minimizer of the empirical risk $\hat{R}(f)$

Interplay of three areas

1. Learning
   ($\hookrightarrow$ Optimization, Optimal Control,...)
2. Generalization
   ($\hookrightarrow$ Statistics, Learning Theory, Stochastics,...)
3. Expressivity
   ($\hookrightarrow$ Approximation Theory, Applied Harmonic Analysis,...)

Error Decomposition

$$R(f) - R^* = \underbrace{(R(f) - R(\hat{f}))}_{\text{optimization error}} + \underbrace{(R(\hat{f}) - R_{\mathcal{F}})}_{\text{estimation error}} + \underbrace{(R_{\mathcal{F}} - R^*)}_{\text{approximation error}}$$

for $R(f)$ the risk of a hypothesis f, $R^* = \inf_f R(f)$ the Bayes risk, $\hat{f}$ minimizer of the empirical risk $\hat{R}(f)$

Interplay of three areas

1. Learning
   ($\hookrightarrow$ Optimization, Optimal Control,...)
2. Generalization
   ($\hookrightarrow$ Statistics, Learning Theory, Stochastics,...)
3. Expressivity
   ($\hookrightarrow$ Approximation Theory, Applied Harmonic Analysis,...)

Error Decomposition

$$R(f) - R^* = \underbrace{(R(f) - R(\hat{f}))}_{\text{optimization error}} + \underbrace{(R(\hat{f}) - R_{\mathcal{F}})}_{\text{estimation error}} + \underbrace{(R_{\mathcal{F}} - R^*)}_{\text{approximation error}}$$

for $R(f)$ the risk of a hypothesis f, $R^* = \inf_f R(f)$ the Bayes risk, $\hat{f}$ minimizer of the empirical risk $\hat{R}(f)$

Interplay of three areas

1. Learning
   ($\hookrightarrow$ Optimization, Optimal Control,...)
2. Generalization
   ($\hookrightarrow$ Statistics, Learning Theory, Stochastics,...)
3. Expressivity
   ($\hookrightarrow$ Approximation Theory, Applied Harmonic Analysis,...)

Density in $C(\mathbb{R}^n)$

Approximation-theoretic results for the single hidden layer

- Density associated with the single hidden layer perceptron model

$$\Sigma(\sigma) = \mathrm{span}\{\sigma(w \cdot x - \theta) : \theta \in \mathbb{R}, w \in \mathbb{R}^n\}$$

  $\sigma : \mathbb{R} \to \mathbb{R}$ activation function, weights $w \in \mathbb{R}^n$, bias $\theta \in \mathbb{R}$

- Find conditions under which $\Sigma(\sigma)$ is dense in $C(K)$ for any compact set $K \subset \mathbb{R}^n$
- Consider *sigmoidal* activation functions satisfying $\lim_{x\to-\infty} \sigma(x) = 0$ and $\lim_{x\to\infty} \sigma(x) = 1$
- Extend density to other function spaces (L^p spaces, the space of measurable functions $\mathcal{M}$)

Approximation-theoretic results for the single hidden layer

- Density associated with the single hidden layer perceptron model

$$\Sigma(\sigma) = \text{span}\{\sigma(w \cdot x - \theta) : \theta \in \mathbb{R}, w \in \mathbb{R}^n\}$$

$\sigma : \mathbb{R} \to \mathbb{R}$ activation function, weights $w \in \mathbb{R}^n$, bias $\theta \in \mathbb{R}$

- Find conditions under which $\Sigma(\sigma)$ is dense in $C(K)$ for any compact set $K \subset \mathbb{R}^n$
- Consider *sigmoidal* activation functions satisfying $\lim_{x\to-\infty} \sigma(x) = 0$ and $\lim_{x\to\infty} \sigma(x) = 1$
- Extend density to other function spaces (L^p spaces, the space of measurable functions $\mathcal{M}$)

Approximation-theoretic results for the single hidden layer

- Density associated with the single hidden layer perceptron model

$$\Sigma(\sigma) = \operatorname{span}\{\sigma(w \cdot x - \theta) : \theta \in \mathbb{R}, w \in \mathbb{R}^n\}$$

  $\sigma : \mathbb{R} \to \mathbb{R}$ activation function, weights $w \in \mathbb{R}^n$, bias $\theta \in \mathbb{R}$

- Find conditions under which $\Sigma(\sigma)$ is dense in $C(K)$ for any compact set $K \subset \mathbb{R}^n$
- Consider *sigmoidal* activation functions satisfying $\lim_{x\to-\infty} \sigma(x) = 0$ and $\lim_{x\to\infty} \sigma(x) = 1$
- Extend density to other function spaces (L^p spaces, the space of measurable functions $\mathcal{M}$)

Approximation-theoretic results for the single hidden layer

- Density associated with the single hidden layer perceptron model

$$\Sigma(\sigma) = \operatorname{span}\{\sigma(w \cdot x - \theta) : \theta \in \mathbb{R}, w \in \mathbb{R}^n\}$$

  $\sigma : \mathbb{R} \to \mathbb{R}$ activation function, weights $w \in \mathbb{R}^n$, bias $\theta \in \mathbb{R}$

- Find conditions under which $\Sigma(\sigma)$ is dense in $C(K)$ for any compact set $K \subset \mathbb{R}^n$
- Consider *sigmoidal* activation functions satisfying $\lim_{x\to-\infty} \sigma(x) = 0$ and $\lim_{x\to\infty} \sigma(x) = 1$
- Extend density to other function spaces (L^p spaces, the space of measurable functions $\mathcal{M}$)

Approximation-theoretic results for the single hidden layer

- Density associated with the single hidden layer perceptron model

$$\Sigma(\sigma) = \operatorname{span}\{\sigma(w \cdot x - \theta) : \theta \in \mathbb{R}, w \in \mathbb{R}^n\}$$

  $\sigma : \mathbb{R} \to \mathbb{R}$ activation function, weights $w \in \mathbb{R}^n$, bias $\theta \in \mathbb{R}$

- Find conditions under which $\Sigma(\sigma)$ is dense in $C(K)$ for any compact set $K \subset \mathbb{R}^n$
- Consider *sigmoidal* activation functions satisfying $\lim_{x\to-\infty} \sigma(x) = 0$ and $\lim_{x\to\infty} \sigma(x) = 1$
- Extend density to other function spaces (L^p spaces, the space of measurable functions $\mathcal{M}$)

Density in $C(X)$ with continuous activation functions

Theorem (Cybenko, 1989)

If σ is continuous and sigmoidal, then $\Sigma(\sigma)$ is dense in $C(K)$.

- Density in $C(K)$ for any bounded, non-constant and monotonously increasing continuous activation function (Funahashi, 1989)
- Density in $C(K)$ for monotonic sigmoidal activation functions and potentially discontinuous at countably many points (Hornik et al., 1989)

$\hookrightarrow$ Density in $L^p(\mu)$ for finite μ and the measurable functions $\mathcal{M}$ for σ bounded and non-constant (Hornik, 1991)

Density in $C(X)$ with continuous activation functions

Theorem (Cybenko, 1989)

If σ is continuous and sigmoidal, then $\Sigma(\sigma)$ is dense in $C(K)$.

- Density in $C(K)$ for any bounded, non-constant and monotonously increasing continuous activation function (Funahashi, 1989)
- Density in $C(K)$ for monotonic sigmoidal activation functions and potentially discontinuous at countably many points (Hornik et al., 1989)

$\hookrightarrow$ Density in $L^p(\mu)$ for finite μ and the measurable functions $\mathcal{M}$ for σ bounded and non-constant (Hornik, 1991)

Density in $C(X)$ with continuous activation functions

Theorem (Cybenko, 1989)

If σ is continuous and sigmoidal, then $\Sigma(\sigma)$ is dense in $C(K)$.

- Density in $C(K)$ for any bounded, non-constant and monotonously increasing continuous activation function (Funahashi, 1989)
- Density in $C(K)$ for monotonic sigmoidal activation functions and potentially discontinuous at countably many points (Hornik et al., 1989)

$\hookrightarrow$ Density in $L^p(\mu)$ for finite μ and the measurable functions $\mathcal{M}$ for σ bounded and non-constant (Hornik, 1991)

Density in $C(X)$ with continuous activation functions

Theorem (Cybenko, 1989)

If σ is continuous and sigmoidal, then $\Sigma(\sigma)$ is dense in $C(K)$.

- Density in $C(K)$ for any bounded, non-constant and monotonously increasing continuous activation function (Funahashi, 1989)
- Density in $C(K)$ for monotonic sigmoidal activation functions and potentially discontinuous at countably many points (Hornik et al., 1989)

$\hookrightarrow$ Density in $L^p(\mu)$ for finite μ and the measurable functions $\mathcal{M}$ for σ bounded and non-constant (Hornik, 1991)

Density in $C(X)$ with continuous activation functions

Theorem (Cybenko, 1989)

If σ is continuous and sigmoidal, then $\Sigma(\sigma)$ is dense in $C(K)$.

- Density in $C(K)$ for any bounded, non-constant and monotonously increasing continuous activation function (Funahashi, 1989)
- Density in $C(K)$ for monotonic sigmoidal activation functions and potentially discontinuous at countably many points (Hornik et al., 1989)

$\hookrightarrow$ Density in $L^p(\mu)$ for finite μ and the measurable functions $\mathcal{M}$ for σ bounded and non-constant (Hornik, 1991)

The universal approximation theorem for potentially discontinuous σ

Theorem (Leshno et al., 1993)

$\Sigma(\sigma)$ is dense in $C(\mathbb{R}^n)$ iff $\sigma \in L^\infty_{loc}(\mathbb{R})$ is not a polynomial (a.e.) and the closure of its points of discontinuity is of zero Lebesgue measure.

- Density in $C(\mathbb{R}^n)$ for any bounded and locally Riemann-integrable activation function (Pinkus, 1999)

$\hookrightarrow$ Density in $L^p(\mu)$ for a non-negative finite measure μ on $\mathbb{R}^n$ with compact support, which is absolutely continuous with respect to the Lebesgue measure (Leshno et al., 1993)

The universal approximation theorem for potentially discontinuous σ

Theorem (Leshno et al., 1993)

$\Sigma(\sigma)$ is dense in $C(\mathbb{R}^n)$ iff $\sigma \in L^\infty_{loc}(\mathbb{R})$ is not a polynomial (a.e.) and the closure of its points of discontinuity is of zero Lebesgue measure.

- Density in $C(\mathbb{R}^n)$ for any bounded and locally Riemann-integrable activation function (Pinkus, 1999)

$\hookrightarrow$ Density in $L^p(\mu)$ for a non-negative finite measure μ on $\mathbb{R}^n$ with compact support, which is absolutely continuous with respect to the Lebesgue measure (Leshno et al., 1993)

The universal approximation theorem for potentially discontinuous σ

Theorem (Leshno et al., 1993)

$\Sigma(\sigma)$ is dense in $C(\mathbb{R}^n)$ iff $\sigma \in L^\infty_{loc}(\mathbb{R})$ is not a polynomial (a.e.) and the closure of its points of discontinuity is of zero Lebesgue measure.

- Density in $C(\mathbb{R}^n)$ for any bounded and locally Riemann-integrable activation function (Pinkus, 1999)

$\hookrightarrow$ Density in $L^p(\mu)$ for a non-negative finite measure μ on $\mathbb{R}^n$ with compact support, which is absolutely continuous with respect to the Lebesgue measure (Leshno et al., 1993)

The universal approximation theorem for potentially discontinuous σ

Theorem (Leshno et al., 1993)

$\Sigma(\sigma)$ is dense in $C(\mathbb{R}^n)$ iff $\sigma \in L^\infty_{loc}(\mathbb{R})$ is not a polynomial (a.e.) and the closure of its points of discontinuity is of zero Lebesgue measure.

- Density in $C(\mathbb{R}^n)$ for any bounded and locally Riemann-integrable activation function (Pinkus, 1999)

$\hookrightarrow$ Density in $L^p(\mu)$ for a non-negative finite measure μ on $\mathbb{R}^n$ with compact support, which is absolutely continuous with respect to the Lebesgue measure (Leshno et al., 1993)

Order of Approximation

Order of Approximation

- A single hidden layer perceptron model can approximate arbitrarily well any continuous function of n variables on a compact domain

Questions:

↪ What is the complexity of the neural network needed to guarantee some specified error?

↪ Does the achievable error scale in favour of the input dimension?

↪ Does the achievable error depend on a parameter quantifying the smoothness of the target function?

Order of Approximation

- A single hidden layer perceptron model can approximate arbitrarily well any continuous function of n variables on a compact domain

Questions:

$\hookrightarrow$ What is the complexity of the neural network needed to guarantee some specified error?

$\hookrightarrow$ Does the achievable error scale in favour of the input dimension?

$\hookrightarrow$ Does the achievable error depend on a parameter quantifying the smoothness of the target function?

Order of Approximation

- A single hidden layer perceptron model can approximate arbitrarily well any continuous function of n variables on a compact domain

Questions:

$\hookrightarrow$ What is the complexity of the neural network needed to guarantee some specified error?

$\hookrightarrow$ Does the achievable error scale in favour of the input dimension?

$\hookrightarrow$ Does the achievable error depend on a parameter quantifying the smoothness of the target function?

Order of Approximation

- A single hidden layer perceptron model can approximate arbitrarily well any continuous function of n variables on a compact domain

Questions:

$\hookrightarrow$ What is the complexity of the neural network needed to guarantee some specified error?

$\hookrightarrow$ Does the achievable error scale in favour of the input dimension?

$\hookrightarrow$ Does the achievable error depend on a parameter quantifying the smoothness of the target function?

Order of Approximation

- A single hidden layer perceptron model can approximate arbitrarily well any continuous function of n variables on a compact domain

Questions:

$\hookrightarrow$ What is the complexity of the neural network needed to guarantee some specified error?

$\hookrightarrow$ Does the achievable error scale in favour of the input dimension?

$\hookrightarrow$ Does the achievable error depend on a parameter quantifying the smoothness of the target function?

Order of Approximation (cont.)

- Consider perceptron model with at most r units in the hidden layer

$$\Sigma_r(\sigma) = \left\{ \sum_{i=1}^{r} a_i \sigma(w_i \cdot x - \theta_i) : a_i, \theta_i \in \mathbb{R}, w_i \in \mathbb{R}^n \right\}$$

Definition (Pinkus, 1999)

For function f in a normed linear space X define the order of approximation by

$$E(f; \Sigma_r(\sigma); X) = \inf_{g \in \Sigma_r(\sigma)} \|f - g\|_X .$$

Problem: target f is unknown!

Order of Approximation (cont.)

- Consider perceptron model with at most r units in the hidden layer

$$\Sigma_r(\sigma) = \left\{ \sum_{i=1}^{r} a_i \sigma(w_i \cdot x - \theta_i) : a_i, \theta_i \in \mathbb{R}, w_i \in \mathbb{R}^n \right\}$$

Definition (Pinkus, 1999)

For function f in a normed linear space X define the order of approximation by

$$E(f; \Sigma_r(\sigma); X) = \inf_{g \in \Sigma_r(\sigma)} \|f - g\|_X .$$

Problem: target f is unknown!

Order of Approximation (cont.)

- Consider perceptron model with at most r units in the hidden layer

$$\Sigma_r(\sigma) = \left\{ \sum_{i=1}^{r} a_i \sigma(w_i \cdot x - \theta_i) : a_i, \theta_i \in \mathbb{R}, w_i \in \mathbb{R}^n \right\}$$

Definition (Pinkus, 1999)

For function f in a normed linear space X define the order of approximation by

$$E(f; \Sigma_r(\sigma); X) = \inf_{g \in \Sigma_r(\sigma)} \|f - g\|_X.$$

Problem: target f is unknown!

Order of Approximation (cont.)

- Consider perceptron model with at most r units in the hidden layer

$$\Sigma_r(\sigma) = \left\{ \sum_{i=1}^{r} a_i \sigma(w_i \cdot x - \theta_i) : a_i, \theta_i \in \mathbb{R}, w_i \in \mathbb{R}^n \right\}$$

Definition (Pinkus, 1999)

For function f in a normed linear space X define the order of approximation by

$$E(f; \Sigma_r(\sigma); X) = \inf_{g \in \Sigma_r(\sigma)} \|f - g\|_X.$$

Problem: target f is unknown!

The Sobolev space

- Assume f is a member of the Sobolev space

$$\mathcal{W}_s^p(B^n) := \{f \in L^p(B^n) : \partial^\alpha f \in L^p(B^n), \forall |\alpha| \leq s\},$$

for $1 \leq p \leq \infty, s \in \mathbb{N}$, and B^n the unit ball in $\mathbb{R}^n$

- $\mathcal{W}_s^p(B^n)$ may be defined as the completion of $C^s(B^n)$ w.r.t. norm

$$\|f\|_{s,p,\mu} := \begin{cases} \left[\sum_{|\alpha| \leq s} \int_{\mathbb{R}^n} |\partial^\alpha f|^p d\mu \right]^{1/p}, & 1 \leq p < \infty \\ \max_{|\alpha| \leq s} \sup_{x \in K} |\partial^\alpha f(x)|, & p = \infty \end{cases}$$

for compact $K \subset \mathbb{R}^n$

- Consider norm one Sobolev classes

$$\mathcal{B}_p^s(B^n) := \{f : f \in \mathcal{W}_s^p(B^n), \|f\|_{s,p,\mu} \leq 1\}$$

The Sobolev space

- Assume f is a member of the Sobolev space

$$\mathcal{W}_s^p(B^n) := \{f \in L^p(B^n) : \partial^\alpha f \in L^p(B^n), \forall |\alpha| \leq s\},$$

  for $1 \leq p \leq \infty, s \in \mathbb{N}$, and B^n the unit ball in $\mathbb{R}^n$

- $\mathcal{W}_s^p(B^n)$ may be defined as the completion of $C^s(B^n)$ w.r.t. norm

$$\|f\|_{s,p,\mu} := \begin{cases} \left[\sum_{|\alpha| \leq s} \int_{\mathbb{R}^n} |\partial^\alpha f|^p d\mu \right]^{1/p}, & 1 \leq p < \infty \\ \max_{|\alpha| \leq s} \sup_{x \in K} |\partial^\alpha f(x)|, & p = \infty \end{cases}$$

  for compact $K \subset \mathbb{R}^n$

- Consider norm one Sobolev classes

$$\mathcal{B}_p^s(B^n) := \{f : f \in \mathcal{W}_s^p(B^n), \|f\|_{s,p,\mu} \leq 1\}$$

The Sobolev space

- Assume f is a member of the Sobolev space

$$\mathcal{W}_s^p(B^n) := \{f \in L^p(B^n) : \partial^\alpha f \in L^p(B^n), \forall |\alpha| \leq s\},$$

  for $1 \leq p \leq \infty$, $s \in \mathbb{N}$, and B^n the unit ball in $\mathbb{R}^n$
- $\mathcal{W}_s^p(B^n)$ may be defined as the completion of $C^s(B^n)$ w.r.t. norm

$$\|f\|_{s,p,\mu} := \begin{cases} \left[\sum_{|\alpha| \leq s} \int_{\mathbb{R}^n} |\partial^\alpha f|^p d\mu \right]^{1/p}, & 1 \leq p < \infty \\ \max_{|\alpha| \leq s} \sup_{x \in K} |\partial^\alpha f(x)|, & p = \infty \end{cases}$$

  for compact $K \subset \mathbb{R}^n$
- Consider norm one Sobolev classes

$$\mathcal{B}_p^s(B^n) := \{f : f \in \mathcal{W}_s^p(B^n), \|f\|_{s,p,\mu} \leq 1\}$$

The Sobolev space

- Assume f is a member of the Sobolev space

$$\mathcal{W}_s^p(B^n) := \{f \in L^p(B^n) : \partial^\alpha f \in L^p(B^n), \forall |\alpha| \leq s\},$$

  for $1 \leq p \leq \infty$, $s \in \mathbb{N}$, and B^n the unit ball in $\mathbb{R}^n$
- $\mathcal{W}_s^p(B^n)$ may be defined as the completion of $C^s(B^n)$ w.r.t. norm

$$\|f\|_{s,p,\mu} := \begin{cases} \left[\sum_{|\alpha| \leq s} \int_{\mathbb{R}^n} |\partial^\alpha f|^p d\mu \right]^{1/p}, & 1 \leq p < \infty \\ \max_{|\alpha| \leq s} \sup_{x \in K} |\partial^\alpha f(x)|, & p = \infty \end{cases}$$

  for compact $K \subset \mathbb{R}^n$
- Consider norm one Sobolev classes

$$\mathcal{B}_p^s(B^n) := \{f : f \in \mathcal{W}_s^p(B^n), \|f\|_{s,p,\mu} \leq 1\}$$

Order of Approximation (cont.)

- Consider the more general class of *ridge* functions

$$\mathcal{R}_r(\sigma) = \left\{ \sum_{i=1}^{r} \sigma_i(w_i \cdot x) : \sigma_i \in C(\mathbb{R}), w_i \in \mathbb{R}^n, i = 1, ..., r \right\}$$

- Since $\Sigma_r(\sigma) \subset \mathcal{R}_r$ for every $\sigma \in C(\mathbb{R})$

$$E(f; \Sigma_r(\sigma); X) = \inf_{g \in \Sigma_r(\sigma)} \|f - g\|_X \geq \inf_{g \in \mathcal{R}_r(\sigma)} \|f - g\|_X = E(f; \mathcal{R}_r(\sigma); X)$$

Theorem (Maiorov, 1999)

For each $n \geq 2$ and $s \geq 1$,

$$E(\mathcal{B}_2^s; \mathcal{R}_r; L_2) = \sup_{f \in \mathcal{B}_2^s} E(f; \mathcal{R}_r; L_2) \asymp r^{-s/(n-1)}$$

Order of Approximation (cont.)

- Consider the more general class of *ridge* functions

$$\mathcal{R}_r(\sigma) = \left\{ \sum_{i=1}^{r} \sigma_i(w_i \cdot x) : \sigma_i \in C(\mathbb{R}), w_i \in \mathbb{R}^n, i = 1, \dots, r \right\}$$

- Since $\Sigma_r(\sigma) \subset \mathcal{R}_r$ for every $\sigma \in C(\mathbb{R})$

$$E(f; \Sigma_r(\sigma); X) = \inf_{g \in \Sigma_r(\sigma)} \|f - g\|_X \geq \inf_{g \in \mathcal{R}_r(\sigma)} \|f - g\|_X = E(f; \mathcal{R}_r(\sigma); X)$$

Theorem (Maiorov, 1999)

For each $n \geq 2$ and $s \geq 1$,

$$E(\mathcal{B}_2^s; \mathcal{R}_r; L_2) = \sup_{f \in \mathcal{B}_2^s} E(f; \mathcal{R}_r; L_2) \asymp r^{-s/(n-1)}$$

Order of Approximation (cont.)

- Consider the more general class of *ridge* functions

$$\mathcal{R}_r(\sigma) = \left\{ \sum_{i=1}^{r} \sigma_i(w_i \cdot x) : \sigma_i \in C(\mathbb{R}), w_i \in \mathbb{R}^n, i = 1, ..., r \right\}$$

- Since $\Sigma_r(\sigma) \subset \mathcal{R}_r$ for every $\sigma \in C(\mathbb{R})$

$$E(f; \Sigma_r(\sigma); X) = \inf_{g \in \Sigma_r(\sigma)} \|f - g\|_X \geq \inf_{g \in \mathcal{R}_r(\sigma)} \|f - g\|_X = E(f; \mathcal{R}_r(\sigma); X)$$

Theorem (Maiorov, 1999)

For each $n \geq 2$ and $s \geq 1$,

$$E(\mathcal{B}_2^s; \mathcal{R}_r; L_2) = \sup_{f \in \mathcal{B}_2^s} E(f; \mathcal{R}_r; L_2) \asymp r^{-s/(n-1)}$$

Order of Approximation (cont.)

- Consider the more general class of *ridge* functions

$$\mathcal{R}_r(\sigma) = \left\{ \sum_{i=1}^{r} \sigma_i(w_i \cdot x) : \sigma_i \in C(\mathbb{R}), w_i \in \mathbb{R}^n, i = 1, ..., r \right\}$$

- Since $\Sigma_r(\sigma) \subset \mathcal{R}_r$ for every $\sigma \in C(\mathbb{R})$

$$E(f; \Sigma_r(\sigma); X) = \inf_{g \in \Sigma_r(\sigma)} \|f - g\|_X \geq \inf_{g \in \mathcal{R}_r(\sigma)} \|f - g\|_X = E(f; \mathcal{R}_r(\sigma); X)$$

Theorem (Maiorov, 1999)

For each $n \geq 2$ and $s \geq 1$,

$$E(\mathcal{B}_2^s; \mathcal{R}_r; L_2) = \sup_{f \in \mathcal{B}_2^s} E(f; \mathcal{R}_r; L_2) \asymp r^{-s/(n-1)}$$

Order of Approximation (cont.)

- The upper bound $r^{-s/(n-1)}$ valid for $E(\mathcal{B}_p^s; \Sigma_r(\sigma); L_p)$ for a $\sigma \in C^\infty$, sigmoidal and strictly increasing (Pinkus, 1999)

1. Denote H_k the linear space of homogeneous polynomials of degree k (in $\mathbb{R}^n$) and $P_k = \cup_{s=0}^k H_s$ the linear space of polynomials of degree at most k
2. Set $\dim H_k = r = \binom{n-1+k}{k} \asymp k^{n-1}$
3. But $E(\mathcal{B}_p^s; P_k; L_p) \le Ck^{-s} \le Cr^{-s/(n-1)}$

Question:

$\hookrightarrow$ Why is it worth using neural networks?

Order of Approximation (cont.)

- The upper bound $r^{-s/(n-1)}$ valid for $E(\mathcal{B}_p^s; \Sigma_r(\sigma); L_p)$ for a $\sigma \in C^\infty$, sigmoidal and strictly increasing (Pinkus, 1999)

1. Denote H_k the linear space of homogeneous polynomials of degree k (in $\mathbb{R}^n$) and $P_k = \cup_{s=0}^k H_s$ the linear space of polynomials of degree at most k
2. Set $\dim H_k = r = \binom{n-1+k}{k} \asymp k^{n-1}$
3. But $E(\mathcal{B}_p^s; P_k; L_p) \leq Ck^{-s} \leq Cr^{-s/(n-1)}$

Question:

$\hookrightarrow$ Why is it worth using neural networks?

Order of Approximation (cont.)

- The upper bound $r^{-s/(n-1)}$ valid for $E(\mathcal{B}_p^s; \Sigma_r(\sigma); L_p)$ for a $\sigma \in C^\infty$, sigmoidal and strictly increasing (Pinkus, 1999)

1. Denote H_k the linear space of homogeneous polynomials of degree k (in $\mathbb{R}^n$) and $P_k = \cup_{s=0}^k H_s$ the linear space of polynomials of degree at most k
2. Set $\dim H_k = r = \binom{n-1+k}{k} \asymp k^{n-1}$
3. But $E(\mathcal{B}_p^s; P_k; L_p) \leq Ck^{-s} \leq Cr^{-s/(n-1)}$

Question:

$\hookrightarrow$ Why is it worth using neural networks?

Order of Approximation (cont.)

- The upper bound $r^{-s/(n-1)}$ valid for $E(\mathcal{B}_p^s; \Sigma_r(\sigma); L_p)$ for a $\sigma \in C^\infty$, sigmoidal and strictly increasing (Pinkus, 1999)

1. Denote H_k the linear space of homogeneous polynomials of degree k (in $\mathbb{R}^n$) and $P_k = \cup_{s=0}^k H_s$ the linear space of polynomials of degree at most k
2. Set $\dim H_k = r = \binom{n-1+k}{k} \asymp k^{n-1}$
3. But $E(\mathcal{B}_p^s; P_k; L_p) \leq Ck^{-s} \leq Cr^{-s/(n-1)}$

Question:

$\hookrightarrow$ Why is it worth using neural networks?

Order of Approximation (cont.)

- The upper bound $r^{-s/(n-1)}$ valid for $E(\mathcal{B}_p^s; \Sigma_r(\sigma); L_p)$ for a $\sigma \in C^\infty$, sigmoidal and strictly increasing (Pinkus, 1999)

1. Denote H_k the linear space of homogeneous polynomials of degree k (in $\mathbb{R}^n$) and $P_k = \cup_{s=0}^k H_s$ the linear space of polynomials of degree at most k
2. Set $\dim H_k = r = \binom{n-1+k}{k} \asymp k^{n-1}$
3. But $E(\mathcal{B}_p^s; P_k; L_p) \leq Ck^{-s} \leq Cr^{-s/(n-1)}$

Question:

$\hookrightarrow$ Why is it worth using neural networks?

Continuous Methods of Approximation

- The approximation error in practice does not depend only on the order of approximation, but also on other factors (i.e., the method of approximation)
- Consider networks with parameters which depend continuously on the target function

Theorem (Maiorov, 1999)

Let $Q_r : L_p \to \Sigma_r(\sigma)$ be an approximating method where the network parameters c_i, θ_i and w_i are continuously dependent on the target function $f \in L_p$. Then

$$\sup_{f \in \mathcal{B}_p^s} \|f - Q_r(f)\|_p \geq C r^{-s/n}$$

Continuous Methods of Approximation

- The approximation error in practice does not depend only on the order of approximation, but also on other factors (i.e., the method of approximation)
- Consider networks with parameters which depend continuously on the target function

Theorem (Maiorov, 1999)

Let $Q_r : L_p \to \Sigma_r(\sigma)$ be an approximating method where the network parameters c_i, θ_i and w_i are continuously dependent on the target function $f \in L_p$. Then

$$\sup_{f \in \mathcal{B}_p^s} \|f - Q_r(f)\|_p \geq C r^{-s/n}$$

Continuous Methods of Approximation

- The approximation error in practice does not depend only on the order of approximation, but also on other factors (i.e., the method of approximation)
- Consider networks with parameters which depend continuously on the target function

Theorem (Maiorov, 1999)

Let $Q_r : L_p \to \Sigma_r(\sigma)$ be an approximating method where the network parameters c_i, θ_i and w_i are continuously dependent on the target function $f \in L_p$. Then

$$\sup_{f \in \mathcal{B}_p^s} \|f - Q_r(f)\|_p \geq C r^{-s/n}$$

Continuous Methods of Approximation

- The approximation error in practice does not depend only on the order of approximation, but also on other factors (i.e., the method of approximation)
- Consider networks with parameters which depend continuously on the target function

Theorem (Maiorov, 1999)

Let $Q_r : L_p \to \Sigma_r(\sigma)$ be an approximating method where the network parameters c_i, θ_i and w_i are continuously dependent on the target function $f \in L_p$. Then

$$\sup_{f \in \mathcal{B}_p^s} \|f - Q_r(f)\|_p \geq C r^{-s/n}$$

Curse of Dimensionality

- Relax the continuity assumption on the approximation procedure for specific σ (e.g. logistic sigmoid) (Maiorov et al., 2000)

Theorem (Petrushev, 1998)

For σ the ReLU function,

$$E(\mathcal{B}_2^s; \Sigma_r(\sigma); L_2) \leq C r^{-s/n}$$

for $s = 1, ..., \frac{n+3}{2}$.

- Curse of Dimensionality - the number of units in the hidden layer necessary for fixed accuracy ϵ is in the order $\mathcal{O}(\epsilon^{-n/s})$

Curse of Dimensionality

- Relax the continuity assumption on the approximation procedure for specific σ (e.g. logistic sigmoid) (Maiorov et al., 2000)

Theorem (Petrushev, 1998)

For σ the ReLU function,

$$E(\mathcal{B}_2^s; \Sigma_r(\sigma); L_2) \leq Cr^{-s/n}$$

for $s = 1, ..., \frac{n+3}{2}$.

- Curse of Dimensionality - the number of units in the hidden layer necessary for fixed accuracy ϵ is in the order $\mathcal{O}(\epsilon^{-n/s})$

Curse of Dimensionality

- Relax the continuity assumption on the approximation procedure for specific σ (e.g. logistic sigmoid) (Maiorov et al., 2000)

Theorem (Petrushev, 1998)

For σ the ReLU function,

$$E(\mathcal{B}_2^s; \Sigma_r(\sigma); L_2) \leq Cr^{-s/n}$$

for $s = 1, ..., \frac{n+3}{2}$.

- Curse of Dimensionality - the number of units in the hidden layer necessary for fixed accuracy ϵ is in the order $\mathcal{O}(\epsilon^{-n/s})$

Curse of Dimensionality

- Relax the continuity assumption on the approximation procedure for specific σ (e.g. logistic sigmoid) (Maiorov et al., 2000)

Theorem (Petrushev, 1998)

For σ the ReLU function,

$$E(\mathcal{B}_2^s; \Sigma_r(\sigma); L_2) \leq Cr^{-s/n}$$

for $s = 1, ..., \frac{n+3}{2}$.

- Curse of Dimensionality - the number of units in the hidden layer necessary for fixed accuracy ϵ is in the order $\mathcal{O}(\epsilon^{-n/s})$

Circumventing the Curse of Dimensionality
with
Deep Neural Networks

Deep Neural Networks

- Deep neural networks - generalisation of shallow neural networks
- Theoretical accuracy achievable with deep or shallow networks is the same

Questions:

↪ Why are deep neural networks so widespread, even though it is harder to train them due to their depth?

↪ Does the multi-layer architecture of deep neural networks help in breaking the curse of dimensionality?

- (Poggio et al., 2017) succeed in beating the curse of dimensionality by assuming the target function is *compositional*

Deep Neural Networks

- Deep neural networks - generalisation of shallow neural networks
- Theoretical accuracy achievable with deep or shallow networks is the same

Questions:

↪ Why are deep neural networks so widespread, even though it is harder to train them due to their depth?

↪ Does the multi-layer architecture of deep neural networks help in breaking the curse of dimensionality?

- (Poggio et al., 2017) succeed in beating the curse of dimensionality by assuming the target function is *compositional*

Deep Neural Networks

- Deep neural networks - generalisation of shallow neural networks
- Theoretical accuracy achievable with deep or shallow networks is the same

Questions:

↪ Why are deep neural networks so widespread, even though it is harder to train them due to their depth?

↪ Does the multi-layer architecture of deep neural networks help in breaking the curse of dimensionality?

- (Poggio et al., 2017) succeed in beating the curse of dimensionality by assuming the target function is *compositional*

Deep Neural Networks

- Deep neural networks - generalisation of shallow neural networks
- Theoretical accuracy achievable with deep or shallow networks is the same

Questions:

$\hookrightarrow$ Why are deep neural networks so widespread, even though it is harder to train them due to their depth?

$\hookrightarrow$ Does the multi-layer architecture of deep neural networks help in breaking the curse of dimensionality?

- (Poggio et al., 2017) succeed in beating the curse of dimensionality by assuming the target function is *compositional*

Deep Neural Networks

- Deep neural networks - generalisation of shallow neural networks
- Theoretical accuracy achievable with deep or shallow networks is the same

Questions:

$\hookrightarrow$ Why are deep neural networks so widespread, even though it is harder to train them due to their depth?

$\hookrightarrow$ Does the multi-layer architecture of deep neural networks help in breaking the curse of dimensionality?

- (Poggio et al., 2017) succeed in beating the curse of dimensionality by assuming the target function is *compositional*

Deep Neural Networks

- Deep neural networks - generalisation of shallow neural networks
- Theoretical accuracy achievable with deep or shallow networks is the same

Questions:

$\hookrightarrow$ Why are deep neural networks so widespread, even though it is harder to train them due to their depth?

$\hookrightarrow$ Does the multi-layer architecture of deep neural networks help in breaking the curse of dimensionality?

- (Poggio et al., 2017) succeed in beating the curse of dimensionality by assuming the target function is *compositional*

$\mathcal{G} - function$

Definition (Poggio et al., 2017)

Let $\mathcal{G}$ be a directed acyclic graph (DAG) with the set of nodes V. Define a $\mathcal{G}-$function $f : \mathbb{R}^n \to \mathbb{R}$ with an architecture corresponding to $\mathcal{G}$, where each of the n source nodes of $\mathcal{G}$ represents a one dimensional input of f. Inner nodes of $\mathcal{G}$ represent constituent functions which get one real one-dimensional input from every incoming edge and the outgoing edges feed the one dimensional function value to the next node. There is only one sink node, whose output is the $\mathcal{G}$-function.

Definition (Poggio et al., 2017)

Define $\mathcal{B}_p^{s,2}$ to be the class of compositional functions $f : \mathbb{R}^n \to \mathbb{R}$ whose corresponding DAG $\mathcal{G}$ has a binary tree architecture and constituent functions h are in $\mathcal{B}_p^s(\mathbb{R}^2)$.

$\mathcal{G} - function$

Definition (Poggio et al., 2017)

Let $\mathcal{G}$ be a directed acyclic graph (DAG) with the set of nodes V. Define a $\mathcal{G}-$function $f : \mathbb{R}^n \to \mathbb{R}$ with an architecture corresponding to $\mathcal{G}$, where each of the n source nodes of $\mathcal{G}$ represents a one dimensional input of f. Inner nodes of $\mathcal{G}$ represent constituent functions which get one real one-dimensional input from every incoming edge and the outgoing edges feed the one dimensional function value to the next node. There is only one sink node, whose output is the $\mathcal{G}$-function.

Definition (Poggio et al., 2017)

Define $\mathcal{B}_p^{s,2}$ to be the class of compositional functions $f : \mathbb{R}^n \to \mathbb{R}$ whose corresponding DAG $\mathcal{G}$ has a binary tree architecture and constituent functions h are in $\mathcal{B}_p^s(\mathbb{R}^2)$.

$\mathcal{G} - function$

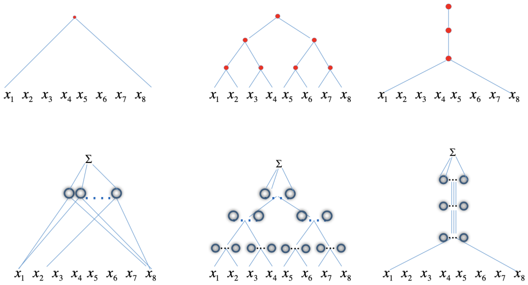
Definition (Poggio et al., 2017)

Let $\mathcal{G}$ be a directed acyclic graph (DAG) with the set of nodes V. Define a $\mathcal{G}-$function $f : \mathbb{R}^n \to \mathbb{R}$ with an architecture corresponding to $\mathcal{G}$, where each of the n source nodes of $\mathcal{G}$ represents a one dimensional input of f. Inner nodes of $\mathcal{G}$ represent constituent functions which get one real one-dimensional input from every incoming edge and the outgoing edges feed the one dimensional function value to the next node. There is only one sink node, whose output is the $\mathcal{G}$-function.

Definition (Poggio et al., 2017)

Define $\mathcal{B}_p^{s,2}$ to be the class of compositional functions $f : \mathbb{R}^n \to \mathbb{R}$ whose corresponding DAG $\mathcal{G}$ has a binary tree architecture and constituent functions h are in $\mathcal{B}_p^s(\mathbb{R}^2)$.

Compositional Functions



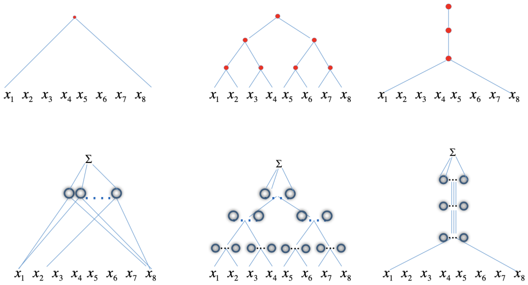
Graphs in the top row represent $\mathcal{G}$-functions of 8 variables. Each graph on the bottom row shows the optimal network architecture approximating the function above.

- Compositional function with a binary tree architecture

$$f(x_1, x_2.x_3, x_4) = h(h_1(x_1, x_2), h_2(x_3, x_4)) \tag{1}$$

- Dimensionality of constituent functions $\ll$ overall input dimension

Compositional Functions



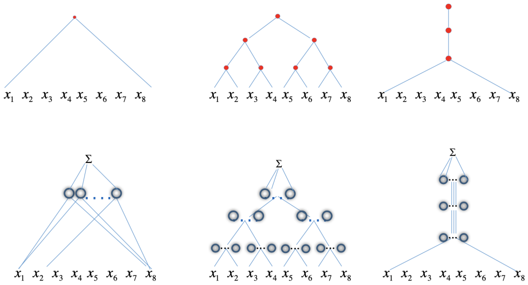
Graphs in the top row represent $\mathcal{G}$-functions of 8 variables. Each graph on the bottom row shows the optimal network architecture approximating the function above.

- Compositional function with a binary tree architecture

$$f(x_1, x_2.x_3, x_4) = h(h_1(x_1, x_2), h_2(x_3, x_4)) \quad (1)$$

- Dimensionality of constituent functions $\ll$ overall input dimension

Compositional Functions



Graphs in the top row represent $\mathcal{G}$-functions of 8 variables. Each graph on the bottom row shows the optimal network architecture approximating the function above.

- Compositional function with a binary tree architecture

$$f(x_1, x_2.x_3, x_4) = h(h_1(x_1, x_2), h_2(x_3, x_4)) \quad (1)$$

- Dimensionality of constituent functions $\ll$ overall input dimension

Effective dimension

Definition (Poggio et al., 2017)

The effective dimension of a function class X is said to be the smallest positive integer k if for every $\epsilon > 0$, any function in X can be approximated up to accuracy ϵ by a neural network of ϵ^{-k} parameters.

- $\mathcal{B}_p^s(\mathbb{R}^n)$ has effective dimension $\frac{n}{s}$

Theorem (Poggio et al., 2017)

For $f \in \mathcal{B}_2^{s,2}$ consider a deep network with the same compositional architecture and $\sigma \in C^\infty$ which is not a polynomial. The complexity of the network to achieve accuracy at least ϵ in the supremum norm is

$$\mathcal{O}\left((n-1)\epsilon^{-2/s}\right).$$

Effective dimension

Definition (Poggio et al., 2017)

The effective dimension of a function class X is said to be the smallest positive integer k if for every $\epsilon > 0$, any function in X can be approximated up to accuracy ϵ by a neural network of ϵ^{-k} parameters.

- $\mathcal{B}_p^s(\mathbb{R}^n)$ has effective dimension $\frac{n}{s}$

Theorem (Poggio et al., 2017)

For $f \in \mathcal{B}_2^{s,2}$ consider a deep network with the same compositional architecture and $\sigma \in C^\infty$ which is not a polynomial. The complexity of the network to achieve accuracy at least ϵ in the supremum norm is

$$\mathcal{O}\left((n-1)\epsilon^{-2/s}\right).$$

Effective dimension

Definition (Poggio et al., 2017)

The effective dimension of a function class X is said to be the smallest positive integer k if for every $\epsilon > 0$, any function in X can be approximated up to accuracy ϵ by a neural network of ϵ^{-k} parameters.

- $\mathcal{B}_p^s(\mathbb{R}^n)$ has effective dimension $\frac{n}{s}$

Theorem (Poggio et al., 2017)

For $f \in \mathcal{B}_2^{s,2}$ consider a deep network with the same compositional architecture and $\sigma \in C^\infty$ which is not a polynomial. The complexity of the network to achieve accuracy at least ϵ in the supremum norm is

$$\mathcal{O}\left((n-1)\epsilon^{-2/s}\right).$$

Effective dimension

Definition (Poggio et al., 2017)

The effective dimension of a function class X is said to be the smallest positive integer k if for every $\epsilon > 0$, any function in X can be approximated up to accuracy ϵ by a neural network of ϵ^{-k} parameters.

- $\mathcal{B}_p^s(\mathbb{R}^n)$ has effective dimension $\frac{n}{s}$

Theorem (Poggio et al., 2017)

For $f \in \mathcal{B}_2^{s,2}$ consider a deep network with the same compositional architecture and $\sigma \in C^\infty$ which is not a polynomial. The complexity of the network to achieve accuracy at least ϵ in the supremum norm is

$$\mathcal{O}\left((n-1)\epsilon^{-2/s}\right).$$

$\mathcal{B}_p^{s,2}(\mathbb{R}^n)$ has effective dimension $\frac{2}{s}$

1. Each constituent function is in $\mathcal{B}_p^{s}(\mathbb{R}^2)$, hence it can be approximated by an element of $\Sigma_r(\sigma)$ with accuracy $\epsilon = cr^{-s/2}$
2. $f \in \mathcal{B}_p^{s,2}$, hence each of the constituent functions is Lipschitz continuous
3. E.g. for the function f (1) and approximators to level ϵ p, p_1, p_2 of h, h_1, h_2

$$\begin{aligned} & \|h(h_1,h_2) - p(p_1,p_2)\| \\ = \; & \|h(h_1,h_2) - h(p_1,p_2) + h(p_1,p_2) - p(p_1,p_2)\| \\ \leq \; & \|h(h_1,h_2) - h(p_1,p_2)\| + \|h(p_1,p_2) - p(p_1,p_2)\| \\ \leq \; & c\epsilon \end{aligned}$$

4. There are $n-1$ such nodes

$\mathcal{B}_p^{s,2}(\mathbb{R}^n)$ has effective dimension $\frac{2}{s}$

1. Each constituent function is in $\mathcal{B}_p^s(\mathbb{R}^2)$, hence it can be approximated by an element of $\Sigma_r(\sigma)$ with accuracy $\epsilon = cr^{-s/2}$
2. $f \in \mathcal{B}_p^{s,2}$, hence each of the constituent functions is Lipschitz continuous
3. E.g. for the function f (1) and approximators to level ϵ p, p_1, p_2 of h, h_1, h_2

$$\begin{aligned} & \|h(h_1,h_2) - p(p_1,p_2)\| \\ = \; & \|h(h_1,h_2) - h(p_1,p_2) + h(p_1,p_2) - p(p_1,p_2)\| \\ \leq \; & \|h(h_1,h_2) - h(p_1,p_2)\| + \|h(p_1,p_2) - p(p_1,p_2)\| \\ \leq \; & c\epsilon \end{aligned}$$

4. There are $n-1$ such nodes

$\mathcal{B}_p^{s,2}(\mathbb{R}^n)$ has effective dimension $\frac{2}{s}$

1. Each constituent function is in $\mathcal{B}_p^{s}(\mathbb{R}^2)$, hence it can be approximated by an element of $\Sigma_r(\sigma)$ with accuracy $\epsilon = cr^{-s/2}$
2. $f \in \mathcal{B}_p^{s,2}$, hence each of the constituent functions is Lipschitz continuous
3. E.g. for the function f (1) and approximators to level ϵ p, p_1, p_2 of h, h_1, h_2

$$\begin{aligned} & \|h(h_1,h_2) - p(p_1,p_2)\| \\ = \; & \|h(h_1,h_2) - h(p_1,p_2) + h(p_1,p_2) - p(p_1,p_2)\| \\ \leq \; & \|h(h_1,h_2) - h(p_1,p_2)\| + \|h(p_1,p_2) - p(p_1,p_2)\| \\ \leq \; & c\epsilon \end{aligned}$$

4. There are $n-1$ such nodes

$\mathcal{B}^{s,2}_p(\mathbb{R}^n)$ has effective dimension $\frac{2}{s}$

1. Each constituent function is in $\mathcal{B}^s_p(\mathbb{R}^2)$, hence it can be approximated by an element of $\Sigma_r(\sigma)$ with accuracy $\epsilon = cr^{-s/2}$
2. $f \in \mathcal{B}^{s,2}_p$, hence each of the constituent functions is Lipschitz continuous
3. E.g. for the function f (1) and approximators to level ϵ p, p_1, p_2 of h, h_1, h_2

$$
\begin{aligned}
& \|h(h_1,h_2) - p(p_1,p_2)\| \\
= \; & \|h(h_1,h_2) - h(p_1,p_2) + h(p_1,p_2) - p(p_1,p_2)\| \\
\leq \; & \|h(h_1,h_2) - h(p_1,p_2)\| + \|h(p_1,p_2) - p(p_1,p_2)\| \\
\leq \; & c\epsilon
\end{aligned}
$$

4. There are $n-1$ such nodes

$\mathcal{B}_p^{s,2}(\mathbb{R}^n)$ has effective dimension $\frac{2}{s}$

1. Each constituent function is in $\mathcal{B}_p^s(\mathbb{R}^2)$, hence it can be approximated by an element of $\Sigma_r(\sigma)$ with accuracy $\epsilon = cr^{-s/2}$
2. $f \in \mathcal{B}_p^{s,2}$, hence each of the constituent functions is Lipschitz continuous
3. E.g. for the function f (1) and approximators to level ϵ p, p_1, p_2 of h, h_1, h_2

$$
\begin{aligned}
& \|h(h_1, h_2) - p(p_1, p_2)\| \\
= \; & \|h(h_1, h_2) - h(p_1, p_2) + h(p_1, p_2) - p(p_1, p_2)\| \\
\leq \; & \|h(h_1, h_2) - h(p_1, p_2)\| + \|h(p_1, p_2) - p(p_1, p_2)\| \\
\leq \; & c\epsilon
\end{aligned}
$$

4. There are $n - 1$ such nodes

Breaking the Curse of Dimensionality with DNNs

Theorem (Poggio et al., 2017)

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a compositional $\mathcal{G}-$function corresponding to a DAG $\mathcal{G}$ with nodes V where each constituent function represented by node $v \in V$ is in $\mathcal{B}_p^{s_v}(\mathbb{R}^{n_v})$ for n_v the number of incoming edges of v. Then for $\sigma \in C^\infty$, the complexity of the shallow network to achieve accuracy at least ϵ in the supremum norm is

$$\mathcal{O}\left(\epsilon^{-n/min_{v\in V} s_v}\right)$$

while the complexity of a deep network represented by $\mathcal{G}$ in the supremum norm is

$$\mathcal{O}\left(\sum_{v\in V} \epsilon^{-n_v/s_v}\right).$$

Breaking the Curse of Dimensionality with DNNs

Theorem (Poggio et al., 2017)

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a compositional $\mathcal{G}-$function corresponding to a DAG $\mathcal{G}$ with nodes V where each constituent function represented by node $v \in V$ is in $\mathcal{B}_p^{s_v}(\mathbb{R}^{n_v})$ for n_v the number of incoming edges of v. Then for $\sigma \in C^\infty$, the complexity of the shallow network to achieve accuracy at least ϵ in the supremum norm is

$$\mathcal{O}\left(\epsilon^{-n/min_{v\in V} s_v}\right)$$

while the complexity of a deep network represented by $\mathcal{G}$ in the supremum norm is

$$\mathcal{O}\left(\sum_{v\in V} \epsilon^{-n_v/s_v}\right).$$

Breaking the Curse of Dimensionality (cont.)

- If the effective dimensionality of constituent functions is smaller than the effective dimensionality of the shallow network $\frac{n}{\min_{v \in V} s_v}$, then deep networks avoid the curse of dimensionality
- Extensions to the ReLU activation function (Bach, 2017)
- (Poggio et al., 2017) conjecture that compositional functions are common in nature and describe the structure of the brain (i.e., visual cortex)

Breaking the Curse of Dimensionality (cont.)

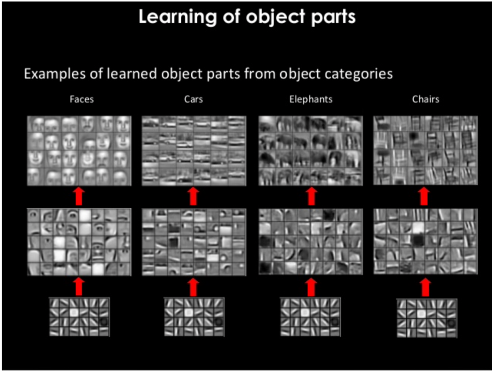
- If the effective dimensionality of constituent functions is smaller than the effective dimensionality of the shallow network $\frac{n}{\min_{v \in V} s_v}$, then deep networks avoid the curse of dimensionality
- Extensions to the ReLU activation function (Bach, 2017)
- (Poggio et al., 2017) conjecture that compositional functions are common in nature and describe the structure of the brain (i.e., visual cortex)

Breaking the Curse of Dimensionality (cont.)

- If the effective dimensionality of constituent functions is smaller than the effective dimensionality of the shallow network $\frac{n}{\min_{v \in V} s_v}$, then deep networks avoid the curse of dimensionality
- Extensions to the ReLU activation function (Bach, 2017)
- (Poggio et al., 2017) conjecture that compositional functions are common in nature and describe the structure of the brain (i.e., visual cortex)

Deep vs Shallow

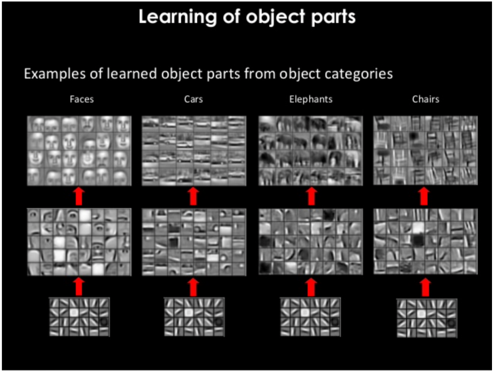
- Deep networks learn 'features' of 'features' - better generalization
- A shallow network tends to memorize the data



from: *Understanding and Improving Deep Learning Algorithms*, Yoshua Bengio, ML Google Distinguished Lecture, 2010

Deep vs Shallow

- Deep networks learn ’features’ of ’features’ - better generalization
- A shallow network tends to memorize the data



from: *Understanding and Improving Deep Learning Algorithms*, Yoshua Bengio, ML Google Distinguished Lecture, 2010

Related work

- The number of linear regions that can be synthesized by a deep network with ReLU nonlinearities is much larger than by a shallow network (Bengio et al., 2014)
- Examples of specific functions that cannot be represented efficiently by shallow networks (Telgarsky, 2015, Shamir et al., 2016)
- Approximation with sparsely connected deep networks (Bölcskei et al., 2019)

Related work

- The number of linear regions that can be synthesized by a deep network with ReLU nonlinearities is much larger than by a shallow network (Bengio et al., 2014)
- Examples of specific functions that cannot be represented efficiently by shallow networks (Telgarsky, 2015, Shamir et al., 2016)
- Approximation with sparsely connected deep networks (Bölcskei et al., 2019)

Related work

- The number of linear regions that can be synthesized by a deep network with ReLU nonlinearities is much larger than by a shallow network (Bengio et al., 2014)
- Examples of specific functions that cannot be represented efficiently by shallow networks (Telgarsky, 2015, Shamir et al., 2016)
- Approximation with sparsely connected deep networks (Bölcskei et al., 2019)