CEMRACS,
July 19-23, 2021

Approximation and learning with tensor networks

Part II: Approximation theory of tree tensor networks

Anthony Nouy

Centrale Nantes, Laboratoire de Mathématiques Jean Leray

Outline

1. Approximation tools based on tree tensor networks
2. Universality, Proximinality and Expressivity
3. Choice of tensor formats
4. Approximation classes of tree tensor networks

Outline

1. Approximation tools based on tree tensor networks
2. Universality, Proximinality and Expressivity
3. Choice of tensor formats
4. Approximation classes of tree tensor networks

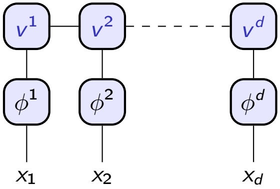
Approximation tools based on tree tensor networks

For the approximation of a target function $u(x_1, \ldots, x_d)$, a first approach is to introduce subspaces $V^\nu_{N_\nu}$ of finite dimension (e.g. polynomials, splines, wavelets...) and consider tensor networks $f \in \mathcal{T}^T_r(V_N)$ with

$$V_N = V^1_{N_1} \otimes \ldots \otimes V^d_{N_d}$$

e.g. with the tensor train format

$$f(x_1, \ldots, x_d) =$$



with ϕ^ν a feature map associated with $V^\nu_{N_\nu}$.

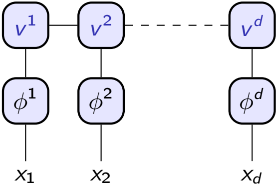
Approximation tools based on tree tensor networks

For the approximation of a target function $u(x_1, \ldots, x_d)$, a first approach is to introduce subspaces $V^\nu_{N_\nu}$ of finite dimension (e.g. polynomials, splines, wavelets...) and consider tensor networks $f \in \mathcal{T}_r^T(V_N)$ with

$$V_N = V^1_{N_1} \otimes \ldots \otimes V^d_{N_d}$$

e.g. with the tensor train format

$f(x_1, \ldots, x_d) =$



with ϕ^ν a feature map associated with $V^\nu_{N_\nu}$.

Spaces $V^\nu_{N_\nu}$ have to be well chosen, e.g. polynomials for analytic functions, splines with a degree adapted to the regularity of the target function...

An approximation tool $\Phi = (\Phi_n)_{n\in\mathbb{N}}$ is then defined by

$$\Phi_n = \{f \in \mathcal{T}_r^T(V_N) : N \in \mathbb{N}^d, r \in \mathbb{N}^T, \mathit{compl}(f) \leq n\}.$$

The dimensions N and the ranks r are free parameters, and $\mathit{compl}(\cdot)$ is some complexity measure.

Approximation tools based on tree tensor networks

An alternative is to rely on tensorization of functions. A d-variate function f is identified with a tensor

$$\boldsymbol{f} = T_{b,d}(f) \in (\mathbb{R}^b)^{\otimes Ld} \otimes (\mathbb{R}^{[0,1)})^{\otimes d}$$

such that

$$f(x_1, \dots, x_d) = \boldsymbol{f}(i_1^1, \dots, i_d^1, \dots, i_1^L, \dots, i_d^L, y_1, \dots, y_d) \quad \text{with} \quad x_\nu = b^{-L}([i_\nu^1 \dots i_\nu^L]_b + y_\nu).$$

Approximation tools based on tree tensor networks

An alternative is to rely on tensorization of functions. A d-variate function f is identified with a tensor

$$\boldsymbol{f} = T_{b,d}(f) \in (\mathbb{R}^b)^{\otimes Ld} \otimes (\mathbb{R}^{[0,1)})^{\otimes d}$$

such that

$$f(x_1, \dots, x_d) = \boldsymbol{f}(i_1^1, \dots, i_d^1, \dots, i_1^L, \dots, i_d^L, y_1, \dots, y_d) \quad \text{with} \quad x_\nu = b^{-L}([i_\nu^1 \dots i_\nu^L]_b + y_\nu).$$

Then we consider functions whose tensorization at resolution L are in the tensor space

$$\boldsymbol{V}_L = (\mathbb{R}^b)^{\otimes Ld} \otimes S^{\otimes d}$$

with $S \subset \mathbb{R}^{[0,1)}$ some subspace of univariate functions.

Approximation tools based on tree tensor networks

An alternative is to rely on tensorization of functions. A d-variate function f is identified with a tensor

$$\boldsymbol{f} = T_{b,d}(f) \in (\mathbb{R}^b)^{\otimes Ld} \otimes (\mathbb{R}^{[0,1)})^{\otimes d}$$

such that

$$f(x_1, \dots, x_d) = \boldsymbol{f}(i_1^1, \dots, i_d^1, \dots, i_1^L, \dots, i_d^L, y_1, \dots, y_d) \quad \text{with} \quad x_\nu = b^{-L}([i_\nu^1 \dots i_\nu^L]_b + y_\nu).$$

Then we consider functions whose tensorization at resolution L are in the tensor space

$$\boldsymbol{V}_L = (\mathbb{R}^b)^{\otimes Ld} \otimes S^{\otimes d}$$

with $S \subset \mathbb{R}^{[0,1)}$ some subspace of univariate functions.

If $S = \mathbb{P}_m$, $V_L = T_{b,d}^{-1}(\boldsymbol{V}_L)$ is identified with the space of multivariate splines of degree m over a uniform partition with b^{dL} elements, i.e.

$$V_L = V_{N_1}^1 \otimes \dots \otimes V_{N_d}^d$$

with $N_1 = \dots = N_d = b^L$ and $V_{N_\nu}^\nu$ a space of univariate splines of degree m over a uniform partition with $N_\nu = b^L$ intervals.

Approximation tools based on tree tensor networks

An alternative is to rely on tensorization of functions. A d-variate function f is identified with a tensor

$$\boldsymbol{f} = T_{b,d}(f) \in (\mathbb{R}^b)^{\otimes Ld} \otimes (\mathbb{R}^{[0,1)})^{\otimes d}$$

such that

$$f(x_1, \ldots, x_d) = \boldsymbol{f}(i_1^1, \ldots, i_d^1, \ldots, i_1^L, \ldots, i_d^L, y_1, \ldots, y_d) \quad \text{with} \quad x_\nu = b^{-L}([i_\nu^1 \ldots i_\nu^L]_b + y_\nu).$$

Then we consider functions whose tensorization at resolution L are in the tensor space

$$\boldsymbol{V}_L = (\mathbb{R}^b)^{\otimes Ld} \otimes S^{\otimes d}$$

with $S \subset \mathbb{R}^{[0,1)}$ some subspace of univariate functions.

If $S = \mathbb{P}_m$, $V_L = T_{b,d}^{-1}(\boldsymbol{V}_L)$ is identified with the space of multivariate splines of degree m over a uniform partition with b^{dL} elements, i.e.

$$V_L = V_{N_1}^1 \otimes \ldots \otimes V_{N_d}^d$$

with $N_1 = ... = N_d = b^L$ and $V_{N_\nu}^\nu$ a space of univariate splines of degree m over a uniform partition with $N_\nu = b^L$ intervals.

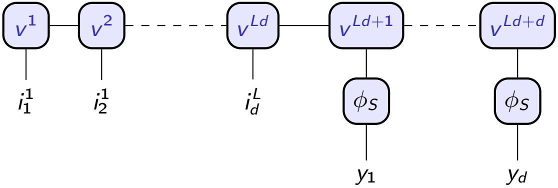
Note that different resolutions L_ν could be used to tensorize the different variables x_ν.

Approximation tools based on tree tensor networks

Then as an approximation tool, we consider functions f whose tensorization is a tensor network in $\mathcal{T}_r^{T_L}(\boldsymbol{V}_L)$, with T_L a dimension tree over $\{1, \ldots, Ld + d\}$.

Using the tensor train format, the corresponding function $f(x_1, \ldots, x_d)$ has the representation

$f(x_1, \ldots, x_d) =$



with ϕ_S the feature map associated with S. This is similar to the quantized tensor train (QTT) format [Kazeev, Khoromskij, Oseledets, Schwab, ...]

Later on, we consider $S = \mathbb{P}_m$ and $\phi_S(y) = (1, y, ..., y^{m+1})$ or any other polynomial basis.

Approximation tools based on tree tensor networks

An approximation tool $\Phi = (\Phi_n)_{n\in\mathbb{N}}$ is then defined by

$$\Phi_n = \{f \in \Phi_{L,T_L,r} : L \in \mathbb{N}_0, r \in \mathbb{N}^{T_L}, compl(f) \leq n\}$$

with $\Phi_{L,T_L,r}$ the functions whose tensorization at resolution L is in $\mathcal{T}_r^{T_L}(V_L)$.

The resolution L and ranks r are free parameters, and $compl(\cdot)$ is some complexity measure.

Complexity measures and corresponding approximation tools

The complexity $compl(f)$ of f is defined as the complexity of the associated tensor network $\boldsymbol{v} = \{v^\alpha\}_{\alpha \in T}$.

- Number of parameters (full tensors network)

$$compl_{\mathcal{F}}(f) = \sum_{\alpha} \text{number\_of\_entries}(v^\alpha)$$

- Number of non-zero parameters (sparse tensors network)

$$compl_{\mathcal{S}}(f) = \sum_{\alpha} \|v^\alpha\|_0$$

The complexity $compl(f)$ of f is defined as the complexity of the associated tensor network $\boldsymbol{v} = \{v^\alpha\}_{\alpha \in T}$.

- Number of parameters (full tensors network)

$$compl_{\mathcal{F}}(f) = \sum_{\alpha} \text{number\_of\_entries}(v^\alpha)$$

- Number of non-zero parameters (sparse tensors network)

$$compl_{\mathcal{S}}(f) = \sum_{\alpha} \|v^\alpha\|_0$$

Complexity measures $compl_{\mathcal{F}}$ and $compl_{\mathcal{S}}$ yield two different approximation tools

$$\Phi_n^{\mathcal{F}} \quad \text{and} \quad \Phi_n^{\mathcal{S}}$$

such that

$$\Phi_n^{\mathcal{F}} \subset \Phi_n^{\mathcal{S}}$$

Approximation with tree tensor networks

Given a function f from a Banach space X, the best approximation error of f by an element of Φ_n is

$$E(f, \Phi_n)_X := \inf_{g \in \Phi_n} \|f - g\|_X$$

Fundamental questions are:

- does $E(f, \Phi_n)_X$ converge to 0 for any f ?
  (universality)
- does a best approximation exist ?
  (proximinality)
- how fast does it converge for functions from classical function classes ?
  (expressivity)
- what are the functions for which $E(f, \Phi_n)_X$ converges with some given rate ?
  (characterization of approximation classes)

Approximation with tree tensor networks

Given a function f from a Banach space X, the best approximation error of f by an element of Φ_n is

$$E(f, \Phi_n)_X := \inf_{g \in \Phi_n} \|f - g\|_X$$

Fundamental questions are:

- does $E(f, \Phi_n)_X$ converge to 0 for any f ?
  (universality)
- does a best approximation exist ?
  (proximinality)
- how fast does it converge for functions from classical function classes ?
  (expressivity)
- what are the functions for which $E(f, \Phi_n)_X$ converges with some given rate ?
  (characterization of approximation classes)

Another fundamental problem (addressed later) is to provide algorithms to practically compute approximations using available information on the function (model equations, samples...)

Outline

Universality

First note that for any algebraic feature tensor space V, and any tree T,

$$\bigcup_r \mathcal{T}_r^T(V) = V.$$

so the question of universality of tree tensor networks boils down to conditions on the tensor feature spaces.

Universality

First note that for any algebraic feature tensor space V, and any tree T,

$$\bigcup_r \mathcal{T}_r^T(V) = V.$$

so the question of universality of tree tensor networks boils down to conditions on the tensor feature spaces.

- Consider the first family of approximation tools with variable feature spaces V_N, $N \in \mathbb{N}^d$.

  If $\bigcup_N V_N$ is dense in X, then the tools are universal for functions in X.

  In particular, this is true for $X = L^p((0,1)^d)$, $p < \infty$, and for polynomial or splines spaces V_N.

Universality

First note that for any algebraic feature tensor space V, and any tree T,

$$\bigcup_r \mathcal{T}_r^T(V) = V.$$

so the question of universality of tree tensor networks boils down to conditions on the tensor feature spaces.

- Consider the first family of approximation tools with variable feature spaces V_N, $N \in \mathbb{N}^d$.

  If $\bigcup_N V_N$ is dense in X, then the tools are universal for functions in X.

  In particular, this is true for $X = L^p((0,1)^d)$, $p < \infty$, and for polynomial or splines spaces V_N.

- Consider the second family of approximation tools using tensorization.

  If $\bigcup_L V_L$ is dense in X, then the tools are universal for functions in X.

  In particular, this is true for $X = L^p((0,1)^d)$, $p < \infty$, assuming that S contains the function one.

Proximinality

For any tree T, any T-rank r, and any finite dimensional tensor space V of X, $\mathcal{T}_r^T(V)$ is a closed set in V.

Φ_n is a finite union of such sets, all contained in a single finite dimensional space V^*. Then Φ_n is a closed set of a finite dimensional space V^* and is therefore proximinal in X.

Expressivity

Different ways to analyse the expressivity of tree tensor networks

- Exploit known results on other approximation tools and estimate the complexity to encode these tools using tree tensor networks.
- Directly encode a function using tree tensor networks (with controlled errors)
- Analyse the convergence of bilinear approximations

$$u(x_\alpha, x_{\alpha^c}) \approx \sum_{k=1}^{r_\alpha} u_k^\alpha(x_\alpha) u_k^{\alpha^c}(x_{\alpha^c})$$

  or the approximability of partial evaluations $u(\cdot, x_{\alpha^c})$ by linear approximation spaces of dimension r_α

Approximation of functions from smoothness classes

We consider approximation tools based on tensorization and functions from classical smoothness classes:

- Sobolev and Besov functions
- Analytic functions
- Analytic functions with singularities

Approximation of functions from Besov spaces $B^\alpha_q(L^p)$

From results on spline approximation and their encoding with tensor networks, we obtain

Theorem

Let $f \in B^\alpha_\infty(L^p)$ with $\alpha > 0$ and $0 < p \leq \infty$. Then

$$E(f, \Phi_n^{\mathcal{F}})_{L^p} \leq C n^{-\tilde{\alpha}/d} |f|_{B^\alpha_\infty(L^p)}$$

for arbitrary $\tilde{\alpha} < \alpha$.

- Tensor networks achieve (near to) optimal performance for any Besov regularity order (measured in L^p norm).
- They perform as well as optimal linear approximation tools (e.g. splines), without requiring to adapt the tool to the regularity order α.
- The depth (resolution L) of the network is crucial to capture extra regularity.

Approximation of functions from Besov spaces $B_q^\alpha(L^\tau)$

Now consider the much harder problem of approximating functions from Besov spaces $B_q^\alpha(L^\tau)$ where regularity is measured in a L^τ-norm weaker than L^p-norm.

From results on best n-term approximation using dilated splines, we obtain

Theorem

Let $f \in B_q^\alpha(L^\tau)$ with $\alpha > 0$, $0 < q \le \tau < p < \infty$, $1 \le p < \infty$ and

$$\frac{\alpha}{d} > \frac{1}{\tau} - \frac{1}{p}.$$

Then

$$E(f, \Phi_n^{\mathcal{S}})_{L^p} \le C n^{-\alpha'/d} |f|_{B_q^\alpha(L^\tau)}, \quad E(f, \Phi_n^{\mathcal{F}})_{L^p} \le C n^{-\alpha'/(2d)} |f|_{B_q^\alpha(L^\tau)},$$

for arbitrary $\alpha' < \alpha$.

Approximation of functions from Besov spaces $B_q^\alpha(L^\tau)$

Now consider the much harder problem of approximating functions from Besov spaces $B_q^\alpha(L^\tau)$ where regularity is measured in a L^τ-norm weaker than L^p-norm.

From results on best n-term approximation using dilated splines, we obtain

Theorem

Let $f \in B_q^\alpha(L^\tau)$ with $\alpha > 0$, $0 < q \le \tau < p < \infty$, $1 \le p < \infty$ and

$$\frac{\alpha}{d} > \frac{1}{\tau} - \frac{1}{p}.$$

Then

$$E(f, \Phi_n^{\mathcal{S}})_{L^p} \le C n^{-\alpha'/d} |f|_{B_q^\alpha(L^\tau)}, \quad E(f, \Phi_n^{\mathcal{F}})_{L^p} \le C n^{-\alpha'/(2d)} |f|_{B_q^\alpha(L^\tau)},$$

for arbitrary $\alpha' < \alpha$.

Approximation of functions from Besov spaces $B_q^\alpha(L^\tau)$

Now consider the much harder problem of approximating functions from Besov spaces $B_q^\alpha(L^\tau)$ where regularity is measured in a L^τ-norm weaker than L^p-norm.

From results on best n-term approximation using dilated splines, we obtain

Theorem

Let $f \in B_q^\alpha(L^\tau)$ with $\alpha > 0$, $0 < q \leq \tau < p < \infty$, $1 \leq p < \infty$ and

$$\frac{\alpha}{d} > \frac{1}{\tau} - \frac{1}{p}.$$

Then

$$E(f, \Phi_n^{\mathcal{S}})_{L^p} \leq Cn^{-\alpha'/d} |f|_{B_q^\alpha(L^\tau)}, \quad E(f, \Phi_n^{\mathcal{F}})_{L^p} \leq Cn^{-\alpha'/(2d)} |f|_{B_q^\alpha(L^\tau)},$$

for arbitrary $\alpha' < \alpha$.

- Sparse tensor networks achieve arbitrarily close to optimal rates in $O(n^{-\alpha/d})$ for functions with any Besov smoothness α (measured in L^τ norm), without the need to adapt the tool to the regularity order α.
- Here depth and sparsity are crucial for obtaining near to optimal performance.
- Full tensor networks have slightly lower performance in $O(n^{-\alpha/(2d)})$.

Analytic functions

For function $f : [0, 1]$ with analytic extension on an open complex domain

$$D_\rho = \{z \in \mathbb{C} : dist(z, [0, 1])) < \frac{\rho - 1}{2}\}, \quad \rho > 1,$$

we obtain an exponential convergence

$$E(f, \Phi_n^{\mathcal{F}})_{L^\infty} \leq C\gamma^{-n^{1/3}},$$

with $\gamma = \min\{\rho, b^{(m+1)/b}\}$.

Analytic functions

For function $f : [0,1]$ with analytic extension on an open complex domain

$$D_\rho = \{z \in \mathbb{C} : dist(z,[0,1])) < \frac{\rho - 1}{2}\}, \quad \rho > 1,$$

we obtain an exponential convergence

$$E(f, \Phi_n^{\mathcal{F}})_{L^\infty} \leq C\gamma^{-n^{1/3}},$$

with $\gamma = \min\{\rho, b^{(m+1)/b}\}$.

The proof relies on the approximation of analytic functions with polynomials and the encoding of polynomials with tree tensor networks: a chebychev polynomial p of deree $\bar{m}$ is such that

$$\|f - p\|_{L^\infty} \leq \frac{2}{\rho - 1} \|f\|_{L^\infty(D_\rho)} \rho^{-\bar{m}}$$

A polynomial of degree $\bar{m}$ can be approximated by φ in $\Phi_{L,r,m}$ with an error in $O(b^{-L(m+1)})$, so that

$$\|f - \varphi\|_{L^\infty} \lesssim \rho^{-\bar{m}} + b^{-L(m+1)}$$

We obtain the result by choosing $\bar{m} \sim n^{1/3}$ and $L \sim b^{-1} n^{1/3}$, so that $compl_{\mathcal{F}}(\varphi) \leq n$.

Functions with singularities

Consider the approximation $u(x) = x^\alpha$, $0 < \alpha \leq 1$, in L^∞.

- Piecewise constant linear approximation.

$$u \in B^\alpha_\infty(L^\infty), \quad u \notin B^\beta_\infty(L^\infty) \quad \text{for } \beta > \alpha,$$

  and a piecewise constant approximation on a uniform mesh with n elements gives a convergence in $O(n^{-\alpha})$ in L^∞,

Functions with singularities

Consider the approximation $u(x) = x^\alpha$, $0 < \alpha \leq 1$, in L^∞.

- Piecewise constant linear approximation.

$$u \in B^\alpha_\infty(L^\infty), \quad u \notin B^\beta_\infty(L^\infty) \quad \text{for } \beta > \alpha,$$

  and a piecewise constant approximation on a uniform mesh with n elements gives a convergence in $O(n^{-\alpha})$ in L^∞,

- Piecewise constant nonlinear approximation.

$$u \in BV \subset B^1_\infty(L^1),$$

  and a piecewise constant approximation on an optimal mesh with n elements gives a convergence in $O(n^{-1})$ in L^∞,

Functions with singularities

Consider the approximation $u(x) = x^\alpha$, $0 < \alpha \leq 1$, in L^∞.

- Piecewise constant linear approximation.

$$u \in B^\alpha_\infty(L^\infty), \quad u \notin B^\beta_\infty(L^\infty) \quad \text{for } \beta > \alpha,$$

  and a piecewise constant approximation on a uniform mesh with n elements gives a convergence in $O(n^{-\alpha})$ in L^∞,

- Piecewise constant nonlinear approximation.

$$u \in BV \subset B^1_\infty(L^1),$$

  and a piecewise constant approximation on an optimal mesh with n elements gives a convergence in $O(n^{-1})$ in L^∞,

- Piecewise constant approximation and tensor networks.
  A piecewise constant approximation on a uniform mesh with 2^d elements exploiting low-rank structures gives an exponential convergence in $O(\beta^{-n})$, where n is the complexity of the representation. Achieves the performance of h-p methods.

High-dimensional approximation

- For Besov spaces $B^{\alpha}_{q}(L^p)$, tensor networks achieve (near to) optimal rate in $O(n^{-\alpha/d})$ which deteriorates with d, that is the curse of dimensionality.

High-dimensional approximation

- For Besov spaces $B^{\alpha}_{q}(L^{p})$, tensor networks achieve (near to) optimal rate in $O(n^{-\alpha/d})$ which deteriorates with d, that is the curse of dimensionality.
- For Besov spaces with mixed smoothness $MB^{\alpha}_{q}(L^{p})$, sparse tensor networks achieve near to optimal performance in $O(n^{-\alpha}\log(n)^{d})$. But still the curse of dimensionality.

High-dimensional approximation

- For Besov spaces $B^\alpha_q(L^p)$, tensor networks achieve (near to) optimal rate in $O(n^{-\alpha/d})$ which deteriorates with d, that is the curse of dimensionality.
- For Besov spaces with mixed smoothness $MB^\alpha_q(L^p)$, sparse tensor networks achieve near to optimal performance in $O(n^{-\alpha}\log(n)^d)$. But still the curse of dimensionality.
- For Besov spaces with anisotropic smoothness $AB^{\boldsymbol{\alpha}}_q(L^p)$, sparse tensor networks also achieve near to optimal rates in $O(n^{-s(\boldsymbol{\alpha})/d})$ with

$$s(\boldsymbol{\alpha})/d = (\alpha_1^{-1} + \ldots + \alpha_d^{-1})^{-1}$$

  the aggregated smoothness. Curse of dimensionality can be circumvented with sufficient anisotropy.

High-dimensional approximation

- For Besov spaces $B^{\alpha}_q(L^p)$, tensor networks achieve (near to) optimal rate in $O(n^{-\alpha/d})$ which deteriorates with d, that is the curse of dimensionality.
- For Besov spaces with mixed smoothness $MB^{\alpha}_q(L^p)$, sparse tensor networks achieve near to optimal performance in $O(n^{-\alpha}\log(n)^d)$. But still the curse of dimensionality.
- For Besov spaces with anisotropic smoothness $AB^{\boldsymbol{\alpha}}_q(L^p)$, sparse tensor networks also achieve near to optimal rates in $O(n^{-s(\boldsymbol{\alpha})/d})$ with

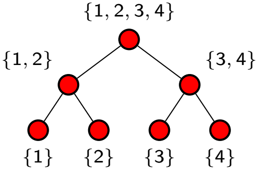
$$s(\boldsymbol{\alpha})/d = (\alpha_1^{-1} + \ldots + \alpha_d^{-1})^{-1}$$

  the aggregated smoothness. Curse of dimensionality can be circumvented with sufficient anisotropy.
- Curse of dimensionality can be circumvented for non usual function classes such as compositions of smooth functions (see Bachmayr, Nouy and Schneider 2021).

Compositional functions

Consider a tree-structured composition of smooth functions $\{f_\alpha : \alpha \in T\}$, see [Mhaskar, Liao, Poggio 2016] for deep neural networks.

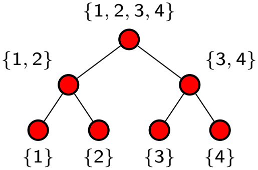
$$f_{1,2,3,4}\left(f_{1,2}\left(f_1(x_1), f_2(x_2)\right), f_{3,4}\left(f_3(x_3), f_4(x_4)\right)\right)$$



Compositional functions

Consider a tree-structured composition of smooth functions $\{f_\alpha : \alpha \in T\}$, see [Mhaskar, Liao, Poggio 2016] for deep neural networks.

$$f_{1,2,3,4}\left(f_{1,2}\left(f_1(x_1), f_2(x_2)\right), f_{3,4}\left(f_3(x_3), f_4(x_4)\right)\right)$$



Assuming that the functions $f_\alpha \in W^{k,\infty}$ with $\|f_\alpha\|_{L^\infty} \leq 1$ and $\|f_\alpha\|_{W^{k,\infty}} \leq B$, the complexity to achieve an accuracy ϵ

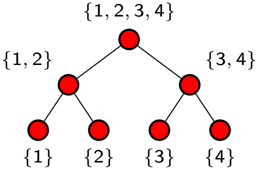
$$C(\epsilon) \lesssim \epsilon^{-3/k}(L+1)^3 B^{3L} d^{1+3/2k}$$

with $L = \log_2(d)$ for a balanced tree and $L + 1 = d$ for a linear tree.

Compositional functions

Consider a tree-structured composition of smooth functions $\{f_\alpha : \alpha \in T\}$, see [Mhaskar, Liao, Poggio 2016] for deep neural networks.

$$f_{1,2,3,4}\left(f_{1,2}\left(f_1(x_1), f_2(x_2)\right), f_{3,4}\left(f_3(x_3), f_4(x_4)\right)\right)$$



Assuming that the functions $f_\alpha \in W^{k,\infty}$ with $\|f_\alpha\|_{L^\infty} \leq 1$ and $\|f_\alpha\|_{W^{k,\infty}} \leq B$, the complexity to achieve an accuracy ϵ

$$C(\epsilon) \lesssim \epsilon^{-3/k}(L+1)^3 B^{3L} d^{1+3/2k}$$

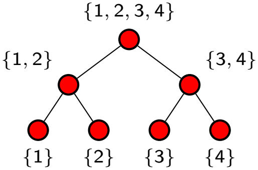
with $L = \log_2(d)$ for a balanced tree and $L+1 = d$ for a linear tree.

- Bad influence of the depth through the norm B of functions f_α (roughness).

Compositional functions

Consider a tree-structured composition of smooth functions $\{f_\alpha : \alpha \in T\}$, see [Mhaskar, Liao, Poggio 2016] for deep neural networks.

$$f_{1,2,3,4}\left(f_{1,2}\left(f_1(x_1), f_2(x_2)\right), f_{3,4}\left(f_3(x_3), f_4(x_4)\right)\right)$$



Assuming that the functions $f_\alpha \in W^{k,\infty}$ with $\|f_\alpha\|_{L^\infty} \leq 1$ and $\|f_\alpha\|_{W^{k,\infty}} \leq B$, the complexity to achieve an accuracy ϵ

$$C(\epsilon) \lesssim \epsilon^{-3/k}(L+1)^3 B^{3L} d^{1+3/2k}$$

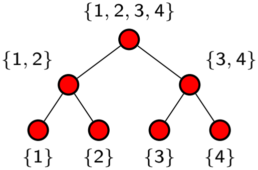
with $L = \log_2(d)$ for a balanced tree and $L + 1 = d$ for a linear tree.

- Bad influence of the depth through the norm B of functions f_α (roughness).
- For a balanced tree, complexity scales polynomially in d: no curse of dimensionality !

Compositional functions

Consider a tree-structured composition of smooth functions $\{f_\alpha : \alpha \in T\}$, see [Mhaskar, Liao, Poggio 2016] for deep neural networks.

$$f_{1,2,3,4}\left(f_{1,2}\left(f_1(x_1), f_2(x_2)\right), f_{3,4}\left(f_3(x_3), f_4(x_4)\right)\right)$$



Assuming that the functions $f_\alpha \in W^{k,\infty}$ with $\|f_\alpha\|_{L^\infty} \leq 1$ and $\|f_\alpha\|_{W^{k,\infty}} \leq B$, the complexity to achieve an accuracy ϵ

$$C(\epsilon) \lesssim \epsilon^{-3/k}(L+1)^3 B^{3L} d^{1+3/2k}$$

with $L = \log_2(d)$ for a balanced tree and $L + 1 = d$ for a linear tree.

- Bad influence of the depth through the norm B of functions f_α (roughness).
- For a balanced tree, complexity scales polynomially in d: no curse of dimensionality !
- For $B \leq 1$ (and even for 1-Lipschitz functions), the complexity only scales polynomially in d whatever the tree: no curse of dimensionality !

Outline

Canonical versus tree-based format

Consider a finite dimensional tensor space $V = V^1 \otimes \ldots \otimes V^d$ with $\dim(V_\nu) = \mathbb{R}^N$, which is identified with $\mathbb{R}^{N\times\ldots\times N}$. Denote by $\mathcal{T}_r^T = \{v : \mathrm{rank}_\alpha(v) \leq r, \alpha \in T\}$.

- From canonical format to tree-based format.
  For any v in V and any $\alpha \subset D$, the α-rank is bounded by the canonical rank:

$$\mathrm{rank}_\alpha(v) \leq \mathrm{rank}(v).$$

  Therefore, for any tree T,

$$\mathcal{R}_r \subset \mathcal{T}_r^T,$$

  so that an element in $\mathcal{R}_r$ with storage complexity $O(dNr)$ admits a representation in $\mathcal{T}_r^T$ with a storage complexity $O(dNr + dr^{s+1})$ where s is the arity of the tree T.

Canonical versus tree-based format

Consider a finite dimensional tensor space $V = V^1 \otimes \ldots \otimes V^d$ with $\dim(V_\nu) = \mathbb{R}^N$, which is identified with $\mathbb{R}^{N\times\cdots\times N}$. Denote by $\mathcal{T}_r^T = \{v : \text{rank}_\alpha(v) \leq r, \alpha \in T\}$.

- From canonical format to tree-based format.
  For any v in V and any $\alpha \subset D$, the α-rank is bounded by the canonical rank:

  $$\text{rank}_\alpha(v) \leq \text{rank}(v).$$

  Therefore, for any tree T,

  $$\mathcal{R}_r \subset \mathcal{T}_r^T,$$

  so that an element in $\mathcal{R}_r$ with storage complexity $O(dNr)$ admits a representation in $\mathcal{T}_r^T$ with a storage complexity $O(dNr + dr^{s+1})$ where s is the arity of the tree T.

- From tree-based format to canonical format. For a balanced or linear binary tree, the subset

  $$S = \{v \in \mathcal{T}_r^T : \text{rank}(v) < q^{d/2}\}, \quad q = \min\{N, r\},$$

  is of Lebesgue measure 0.

  Then a typical element $v \in \mathcal{T}_r^T$ with storage complexity of order $dNr + dr^3$ admits a representation in canonical format with a storage complexity of order $dNq^{d/2}$.

Influence of the tree

- For some functions, the choice of tree is not crucial. For example, an additive function

$$u_1(x_1) + \ldots + u_d(x_d)$$

has α-ranks equal to 2 whatever $\alpha \subset D$.

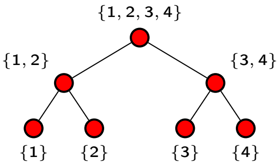
Influence of the tree

- For some functions, the choice of tree is not crucial. For example, an additive function

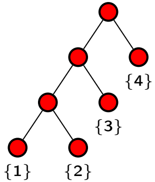
$$u_1(x_1) + \ldots + u_d(x_d)$$

has α-ranks equal to 2 whatever $\alpha \subset D$.

- But usually, different trees lead to different complexities of representations.



T^B (Balanced tree)



T^L (Linear tree)

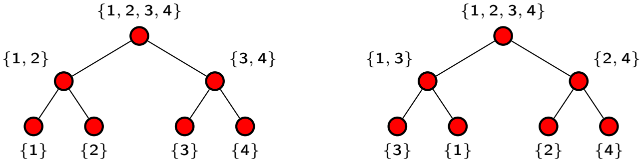
- If $\text{rank}_{T^L}(u) \leq r$ then $\text{rank}_{T^B}(u) \leq r^2$
- If $\text{rank}_{T^B}(u) \leq r$ then $\text{rank}_{T^L}(u) \leq r^{\log_2(d)/2}$

Influence of the tree

Given a tree T and a permutation σ of $D = \{1, \dots, d\}$, we define a tree T_σ

$$T_\sigma = \{\sigma(\alpha) : \alpha \in T\}$$

having the same structure as T but different nodes.



T $\qquad$ T_σ with $\sigma = (3, 1, 2, 4)$

If $\text{rank}_T(u) \leq r$ then $\text{rank}_{T_\sigma}(u)$ typically depends on d.

Influence of the tree

- Consider the Henon-Heiles potential

$$u(x) = \frac{1}{2}\sum_{i=1}^{d} x_i^2 + 0.2\sum_{i=1}^{d-1}(x_i x_{i+1}^2 - x_i^3) + \frac{0.2^2}{16}\sum_{i=1}^{d-1}(x_i^2 + x_{i+1}^2)^2$$

Using a linear tree $T = \{\{1\}, \{2\}, \dots, \{d\}, \{1,2\}, \{1,2,3\}, \dots, \{1,\dots,d-1\}, D\}$,

$$\mathrm{rank}_T(u) \leq 4, \quad storage(u) = O(d)$$

but for the permutation

$$\sigma = (1, 3, \dots, d-1, 2, 4, \dots, d) \tag{$\star$}$$

and the corresponding linear tree T_σ,

$$\mathrm{rank}_{T_\sigma}(u) \leq 2d + 1, \quad storage(u) = O(d^3).$$

Influence of the tree

- Consider the Henon-Heiles potential

$$u(x) = \frac{1}{2}\sum_{i=1}^{d} x_i^2 + 0.2\sum_{i=1}^{d-1}(x_i x_{i+1}^2 - x_i^3) + \frac{0.2^2}{16}\sum_{i=1}^{d-1}(x_i^2 + x_{i+1}^2)^2$$

  Using a linear tree $T = \{\{1\},\{2\},\dots,\{d\},\{1,2\},\{1,2,3\},\dots,\{1,\dots,d-1\},D\}$,

$$\mathrm{rank}_T(u) \le 4, \quad storage(u) = O(d)$$

  but for the permutation

$$\sigma = (1,3,\dots,d-1,2,4,\dots,d) \qquad (\star)$$

  and the corresponding linear tree T_σ,

$$\mathrm{rank}_{T_\sigma}(u) \le 2d+1, \quad storage(u) = O(d^3).$$

- For a typical tensor in $\mathcal{T}_r^T$ with T a binary tree, its representation in tree based format with tree T_σ, with σ as in $(\star)$, has a complexity scaling exponentially with d.

Influence of the tree

- Consider the probability distribution $f(x) = \mathbb{P}(X = x)$ of a Markov chain $X = (X_1, \ldots, X_d)$ given by

$$f(x) = f_1(x_1) f_{2|1}(x_2|x_1) \ldots f_{d|d-1}(x_d|x_{d-1})$$

where bivariate functions $f_{i|i-1}$ have a rank r.

Influence of the tree

- Consider the probability distribution $f(x) = \mathbb{P}(X = x)$ of a Markov chain $X = (X_1, \ldots, X_d)$ given by

$$f(x) = f_1(x_1) f_{2|1}(x_2|x_1) \ldots f_{d|d-1}(x_d|x_{d-1})$$

where bivariate functions $f_{i|i-1}$ have a rank r.

  - With the linear tree T containing interior nodes $\{1,2\}, \{1,2,3\}, \ldots, \{1, \ldots, d-1\}$, f admits a representation in tree-based format with storage complexity in r^4.

Influence of the tree

- Consider the probability distribution $f(x) = \mathbb{P}(X = x)$ of a Markov chain $X = (X_1, \dots, X_d)$ given by

$$f(x) = f_1(x_1) f_{2|1}(x_2|x_1) \dots f_{d|d-1}(x_d|x_{d-1})$$

  where bivariate functions $f_{i|i-1}$ have a rank r.

  - With the linear tree T containing interior nodes $\{1,2\}, \{1,2,3\}, \dots, \{1, \dots, d-1\}$, f admits a representation in tree-based format with storage complexity in r^4.
  - The canonical rank of f is exponential in d.

Influence of the tree

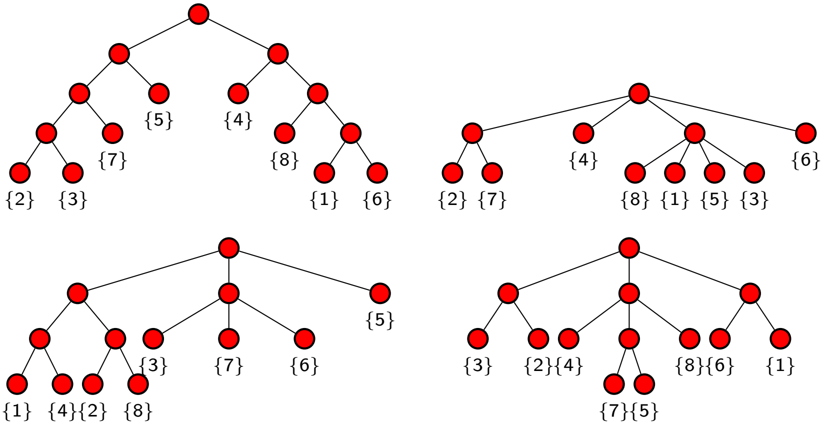
- Consider the probability distribution $f(x) = \mathbb{P}(X = x)$ of a Markov chain $X = (X_1, \dots, X_d)$ given by

$$f(x) = f_1(x_1) f_{2|1}(x_2|x_1) \dots f_{d|d-1}(x_d|x_{d-1})$$

  where bivariate functions $f_{i|i-1}$ have a rank r.
  - With the linear tree T containing interior nodes $\{1,2\}, \{1,2,3\}, \dots, \{1,\dots,d-1\}$, f admits a representation in tree-based format with storage complexity in r^4.
  - The canonical rank of f is exponential in d.
  - But when considering the linear tree T_σ obtained by applying permutation $\sigma = (1, 3, \dots, d-1, 2, 4, \dots, d)$ to the tree T, the storage complexity in tree-based format is also exponential in d.

How to choose a good tree ?

A combinatorial problem...



Outline

Properties of tree tensor networks

We here consider approximation tools based on tensor networks with tensorized functions (with or without sparsity).

They satisfy

(P1) $\Phi_0 = \{0\}$, $0 \in \Phi_n$

(P2) $a\Phi_n = \Phi_n$ for any $a \in \mathbb{R} \setminus \{0\}$ (cone)

(P3) $\Phi_n \subset \Phi_{n+1}$ (nestedness)

(P4) $\Phi_n + \Phi_n \subset \Phi_{cn}$ for some constant c (not too nonlinear)

Properties of tree tensor networks

We here consider approximation tools based on tensor networks with tensorized functions (with or without sparsity).

They satisfy

(P1) $\Phi_0 = \{0\}$, $0 \in \Phi_n$

(P2) $a\Phi_n = \Phi_n$ for any $a \in \mathbb{R} \setminus \{0\}$ (cone)

(P3) $\Phi_n \subset \Phi_{n+1}$ (nestedness)

(P4) $\Phi_n + \Phi_n \subset \Phi_{cn}$ for some constant c (not too nonlinear)

For $X = L^p$, they further satisfy

(P5) $\bigcup_n \Phi_n$ is dense in L^p for $0 < p < \infty$ (universality),

(P6) for each $f \in L^p$ for $0 < p \leq \infty$, there exists a best approximation in Φ_n (proximinal sets).

Approximation classes

For an approximation tool $\Phi = (\Phi_n)_{n\in\mathbb{N}}$, we define for any $\alpha > 0$ the approximation class

$$A^\alpha_\infty(L^p) := A^\alpha_\infty(L^p, \Phi)$$

of functions $f \in L^p$ such that

$$E(f, \Phi_n)_{L^p} \le Cn^{-\alpha}$$

Approximation classes

For an approximation tool $\Phi = (\Phi_n)_{n\in\mathbb{N}}$, we define for any $\alpha > 0$ the approximation class

$$A^\alpha_\infty(L^p) := A^\alpha_\infty(L^p, \Phi)$$

of functions $f \in L^p$ such that

$$E(f, \Phi_n)_{L^p} \leq Cn^{-\alpha}$$

- Properties (P1)-(P4) of Φ imply that $A^\alpha_\infty(L^p)$ is a quasi-Banach spaces with quasi-semi-norm

$$|f|_{A^\alpha_\infty} := \sup_{n\geq 1} n^\alpha E(f, \Phi_n)_{L^p}$$

Approximation classes

For an approximation tool $\Phi = (\Phi_n)_{n\in\mathbb{N}}$, we define for any $\alpha > 0$ the approximation class

$$A^\alpha_\infty(L^p) := A^\alpha_\infty(L^p, \Phi)$$

of functions $f \in L^p$ such that

$$E(f, \Phi_n)_{L^p} \leq Cn^{-\alpha}$$

- Properties (P1)-(P4) of Φ imply that $A^\alpha_\infty(L^p)$ is a quasi-Banach spaces with quasi-semi-norm

$$|f|_{A^\alpha_\infty} := \sup_{n\geq 1} n^\alpha E(f, \Phi_n)_{L^p}$$

- Full and sparse complexity measures yield two different approximation spaces

$$\mathcal{F}^\alpha_\infty(L^p) = A^\alpha_\infty(L^p, \Phi^{\mathcal{F}}), \quad \mathcal{S}^\alpha_\infty(L^p) = A^\alpha_\infty(L^p, \Phi^{\mathcal{S}})$$

  such that

$$\mathcal{F}^\alpha_\infty(L^p) \hookrightarrow \mathcal{S}^\alpha_\infty(L^p) \hookrightarrow \mathcal{F}^{\alpha/2}_\infty(L^p)$$

Direct embeddings

From results on the approximation properties for Besov spaces, we have the following results.

- Let $\alpha > 0$ and $0 < p \leq \infty$. For arbitrary $\tilde{\alpha} < \alpha$,

  $$B_q^{\alpha}(L^p) \hookrightarrow \mathcal{F}_q^{\tilde{\alpha}/d}(L^p)$$

  and

  $$MB_q^{\alpha}(L^p) \hookrightarrow \mathcal{S}_q^{\tilde{\alpha}}(L^p).$$

  For arbitrary $\tilde{s} < s(\boldsymbol{\alpha}) := d(\alpha_1^{-1} + \ldots + \alpha_d^{-1})^{-1}$,

  $$AB_q^{\boldsymbol{\alpha}}(L^p) \hookrightarrow \mathcal{S}_q^{\tilde{s}/d}(L^p)$$

- For $\alpha > 0$, $1 \leq p < \infty$, $0 < q \leq \tau < p < \infty$ and $\frac{\alpha}{d} > \frac{1}{\tau} - \frac{1}{p}$,

  $$B_q^{\alpha}(L^{\tau}) \hookrightarrow \mathcal{S}_{\infty}^{\tilde{\alpha}/d}(L^p) \hookrightarrow \mathcal{F}_{\infty}^{\tilde{\alpha}/(2d)}(L^p)$$

  for arbitrary $\tilde{\alpha} < \alpha$, and similar results for anisotropic and mixed smoothness.

No inverse embedding

For any $\alpha > 0$, $q \leq \infty$, and any β,

$$\mathcal{F}^{\alpha}_{\infty}(L^p) \not\hookrightarrow B^{\beta}_{\infty}(L^p).$$

That means that approximation classes contain functions that have no smoothness in a classical sense.

Tensor networks may be useful for the approximation of functions beyond standard smoothness classes.

Open questions

- What are the properties of the approximation tool with free tree

$$\Phi_n = \{f \in \Phi_{L,T_L,r} : L \in \mathbb{N}_0, T_L \subset 2^{\{1,\dots,(L+1)d\}}, r \in \mathbb{N}^{\#T}, compl(f) \le n\}$$

Higher expressivity (or larger approximation classes) but how much higher ?

Open questions

- What are the properties of the approximation tool with free tree

$$\Phi_n = \{f \in \Phi_{L,T_L,r} : L \in \mathbb{N}_0, T_L \subset 2^{\{1,\dots,(L+1)d\}}, r \in \mathbb{N}^{\#T}, compl(f) \leq n\}$$

  Higher expressivity (or larger approximation classes) but how much higher ?

- What about expressivity and approximation classes of more general tensor networks ?

References I

M. Ali and A. Nouy.
Approximation with tensor networks. part i: Approximation spaces.
ArXiv, abs/2007.00118, 2020.

M. Ali and A. Nouy.
Approximation with tensor networks. part ii: Approximation rates for smoothness classes.
ArXiv, abs/2007.00128, 2020.

M. Ali and A. Nouy.
Approximation with tensor networks. part iii: Multivariate approximation.
ArXiv, abs/2007.00128, 2020.

M. Bachmayr, A. Nouy and R. Schneider.
Approximation power of tree tensor networks for compositional functions.
In preparation.

R. A. DeVore and G. G. Lorentz.
Constructive approximation, volume 303.
Springer Science & Business Media, 1993.

L. Grasedyck.
Polynomial approximation in hierarchical Tucker format by vector-tensorization.
Inst. für Geometrie und Praktische Mathematik, 2010.

References II

V. Kazeev and C. Schwab.
Approximation of singularities by quantized-tensor fem.
PAMM, 15(1):743–746, 2015.

V. Kazeev, I. Oseledets, M. Rakhuba, and C. Schwab.
Qtt-finite-element approximation for multiscale problems i: model problems in one dimension.
Advances in Computational Mathematics, 43(2):411–442, Apr 2017.

R. Schneider and A. Uschmajew.
Approximation rates for the hierarchical tensor format in periodic sobolev spaces.
Journal of Complexity, 30(2):56 – 71, 2014.