Low-rank and sparse methods for high-dimensional approximation and model order reduction

—–

Lecture 10

Quantized tensor formats

From vectors to tensors

Let consider a vector

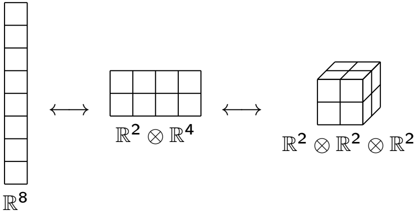
$$v \in \mathbb{R}^{2^d}.$$

By introducing the binary representation $(i_1, \ldots, i_d) \in \{0,1\}^d$ of an integer $i \in \{0, \ldots, 2^d - 1\}$, the vector v can be identified with an order-d tensor

$$\mathbf{v} \in \underbrace{\mathbb{R}^2 \otimes \ldots \otimes \mathbb{R}^2}_{d \text{ times}} := \mathbb{R}^{2^{\otimes d}},$$

such that

$$v(i) = \mathbf{v}(i_1, \ldots, i_d) \quad \text{for} \quad i = \sum_{\nu=1}^{d} 2^{\nu-1} i_\nu.$$



From vectors to tensors

It there exists a low-rank representation of the tensor $\mathbf{v}$ with representation ranks bounded by R, then the vector v of dimension $N = 2^d$ has a storage complexity

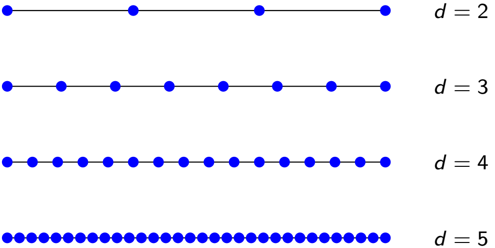
$$N_d = \text{storage}(\mathbf{v}) \lesssim dR^k = \log_2(N)R^k.$$

A representation of $\mathbf{v}$ in tensor train format is called a Quantized Tensor Train (QTT) representation.

Quantized representation of functions

Let $f(x)$ be a function defined on the interval $(0,1)$ and let v be the vector of evaluations of $f(x)$ on a uniform grid $\{x_i = ih\}_{i=0}^{N-1}$ with $N = 2^d$ points,

$$v(i) = f(x_i), \quad v \in \mathbb{R}^{2^d}.$$



The vector $v \in \mathbb{R}^{2^d}$ can be identified with an order-d tensor $\mathbf{v} \in (\mathbb{R}^2)^{\otimes d}$

Quantized representation of functions

Example 1

The function $f(x) = \exp(ax)$ is such that

$$v(i) = \exp(ax_i) = \exp(ahi) = \exp(ah \sum_{\nu=1}^{d} 2^{\nu-1} i_\nu) = \prod_{\nu=1}^{d} \exp(ah2^{\nu-1} i_\nu).$$

Therefore, the associated tensor $\mathbf{v}$ has a rank one, with

$$\mathbf{v}(i_1, \dots, i_d) = \mathbf{v}^{(1)}(i_1) \dots \mathbf{v}^{(d)}(i_d), \quad \mathbf{v}^{(\nu)} = \begin{pmatrix} 1 \\ \exp(ah2^{\nu-1}) \end{pmatrix}.$$

The storage complexity of $\mathbf{v}$ is $N_d = 2d = 2\log_2(N)$.

Quantized representation of functions

Example 2

The function $f(x) = \sin(ax + b)$ on $(0, 1)$ is such that

$$v(i) = \sin(\phi + \omega x_i) = \sin(\phi + \sum_{\nu=1}^{d} 2^{\nu-1} \omega i_\nu).$$

The associated tensor $\mathbf{v}$ admits a representation in tensor-train format with TT-rank $(2, \ldots, 2)$ and storage complexity $N_d = 8d = 8 \log_2(N)$.

The discretization of the function f with $\omega = 2\pi * 2^{100}$ (which means 2^{100} periods on $(0, 1)$) with 16 points per period requires $N = 2^{104}$ points and a storage complexity of $N_d = 832$ in QTT format !!!

Quantized representation of multivariate functions

Let $f(x_1, \dots, x_n)$ be a n-dimensional function defined on $(0,1)^n$ and let v be the order-n tensor of its evaluations on a uniform tensorized grid

$$x_i = (i_1 h, \dots, i_n h), \quad i = (i_1, \dots, i_n) \in \{0, \dots, 2^d - 1\}^n$$

such that

$$f(x_i) = v(i_1, \dots, i_n).$$

Then, v can be identified with a tensor $\mathbf{v} \in \mathbb{R}^{2^{\otimes nd}}$ of order nd such that

$$\mathbf{v}(\underbrace{i_{1,1}, \dots, i_{1,d}}_{\text{dimension } 1}, \dots, \underbrace{i_{n,1}, \dots, i_{n,d}}_{\text{dimension } n}) = v(i_1, \dots, i_n),$$

where $(i_{k,1}, \dots, i_{k,d})$ is the binary representation of the integer i_k.

Up to a permutation, v can also be identified with the tensor $\mathbf{v} \in \mathbb{R}^{2^{\otimes nd}}$ such that

$$\mathbf{v}(\underbrace{i_{1,1}, \dots, i_{n,1}}_{\text{level } 1}, \dots, \underbrace{i_{1,d}, \dots, i_{n,d}}_{\text{level } d}) = v(i_1, \dots, i_n),$$

These different choices yield different representation ranks for the tensor $\mathbf{v}$.

Approximation of functions in quantized tensor format

Let $f_N(x) = \sum_{i=0}^{N-1} v(i)\varphi_i(x)$ be an interpolation of $f(x)$ on the uniform grid with 2^d points and assume that

$$\|f - f_N\| \lesssim N^{-\alpha}.$$

If we assume that the tensor $\mathbf{v}$ has a representation in a certain low-rank format with a storage complexity growing not too fast with d, say

$$N_d \lesssim d^k.$$

Then

$$\|f - f_N\| \lesssim 2^{-\alpha N_d^{1/k}} = \exp(-\beta N_d^{1/k}),$$

which provides an exponential convergence of f_N with respect to the storage complexity N_d.

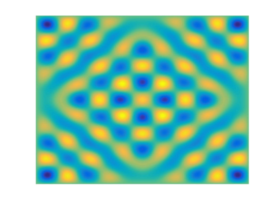
Approximation of functions in quantized tensor format

Example 3

Consider the partial differential equation

$$-\Delta u - ku = 1 \quad \text{on } \Omega = (0,1)^2,$$
$$u = 0 \quad \text{on } \partial\Omega,$$

with $k = 2.10^3$, and consider a piecewise linear finite element approximation on a uniform grid with $N = 2^d \times 2^d$ nodes.



For $d = 9$, $N = 260100$ and the discrete solution is identified with a tensor $\mathbf{u}$ of order $2d = 18$, which admits an approximate representation in tensor-train format with relative precision 10^{-3} (resp. 10^{-4}, 10^{-5}) and with storage complexity $N_d = 3532$ (resp. $N_d = 4656$, $N_d = 6170$).

Quantized tensor structure of PDEs

A partial differential equation

$$\mathcal{A}u = f,$$

defined on a product domain $(0,1)^n$ and discretized with a finite difference scheme on a uniform tensorized grid with 2^d nodes per dimension can be interpreted as a tensor structured equation

$$\mathbf{A}\mathbf{u} = \mathbf{b},$$

where $\mathbf{u} \in (\mathbb{R}^2)^{\otimes dn}$ represents the values of the approximation on the grid, where $\mathbf{b} \in (\mathbb{R}^2)^{\otimes dn}$ is the tensor of evaluations of f on the grid, and

$$\mathbf{A} \in (\mathbb{R}^{2\times 2})^{\otimes nd}$$

is the discrete differential operator.

For a partial differential operator $\mathcal{A} = \sum_\alpha a_\alpha D^\alpha$ with constant coefficients,

$$\mathbf{A} = \sum_\alpha a_\alpha \mathbf{D}^\alpha,$$

where $\mathbf{D}^\alpha$ is the discrete version of the operator D^α.

Quantized tensor structure of PDEs

Example 4 (QTT representation of the finite difference Laplace operator with Dirichlet boundary conditions)

The discrete Laplace operator on a uniform grid of $(0,1)^n$ with 2^d nodes per dimension is first identified with an order n tensor $\mathbf{\Delta} \in (\mathbb{R}^{2^d \times 2^d})^{\otimes n}$ with a representation in TT format with rank $(2, \ldots, 2)$,

$$\mathbf{\Delta} = \begin{pmatrix} \mathbf{\Delta}_1 & \mathbf{I} \end{pmatrix} \bowtie \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{\Delta}_1 & \mathbf{I} \end{pmatrix} \bowtie \ldots \bowtie \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{\Delta}_1 & \mathbf{I} \end{pmatrix} \bowtie \begin{pmatrix} \mathbf{I} \\ \mathbf{\Delta}_1 \end{pmatrix},$$

where the matrix $\mathbf{\Delta}_1 = h^{-2}\,\mathrm{diag}(-1, 2, -1) \in \mathbb{R}^{2^d \times 2^d}$ is the discrete Laplace operator in one dimension (with homogeneous Dirichlet boundary conditions).

The matrix $\mathbf{\Delta}_1$ can then be identified with a tensor $\mathbf{\Delta}_1 \in (\mathbb{R}^{2 \times 2})^{\otimes d}$ with a representation in tensor train format with ranks $(3, \ldots, 3)$

$$\mathbf{\Delta}_1 = \begin{pmatrix} \mathbf{I} & \mathbf{J}^T & \mathbf{J} \end{pmatrix} \bowtie \begin{pmatrix} \mathbf{I} & \mathbf{J}^T & \mathbf{J} \\ \mathbf{0} & \mathbf{J} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{J}^T \end{pmatrix} \bowtie \ldots \bowtie \begin{pmatrix} \mathbf{I} & \mathbf{J}^T & \mathbf{J} \\ \mathbf{0} & \mathbf{J} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{J}^T \end{pmatrix} \bowtie \begin{pmatrix} 2\mathbf{I} - \mathbf{J} - \mathbf{J}^T \\ -\mathbf{J} \\ -\mathbf{J}^T \end{pmatrix},$$

$$\mathbf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{J} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}.$$