swissQuant
Mastering Complexity

# Mathematical Challenge April 2019

## Tensor network algorithms

## References


[1] Andrzej Cichocki. Tensor networks for big data analytics and large-scale optimization problems. *CoRR*, abs/1407.3124, 2014.

[2] Ulrich Schollwoeck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics - ANN PHYS N Y*, 326, 2010.


## Description

### Motivation

Tensor network algorithms first emerged in computational condensed-matter physics in the 90's, where they quickly established themselves as the leading method for simulation of strongly correlated one-dimensional quantum lattice systems. They have since found application in neighboring fields, such as quantum information and quantum chemistry, and are since recently being explored in the context of machine learning and big data analytics [1].

Loosely speaking, these methods seek to compress big, high-dimensional data into a representation consisting of a network of weakly coupled, small, low-dimensional data. In this challenge we will try and get a feel for what this looks like using as an example the case of Tensor Trains, aka Matrix Product States. [2] is a great introduction to this subject.

### Technical Details

#### Tensor-Train decomposition of an arbitrary tensor

Consider an $L$ dimensional real-valued tensor $C_{\sigma_1,\sigma_2,\sigma_3\ldots,\sigma_L}$ where every index $\sigma_i$ can take values between $1$ and $d$.

To represent it as a tensor-train, we can start by reshaping it into a matrix by merging indices $\sigma_2, ..., \sigma_L$ and perform an SVD:

$$C_{\sigma_1,\sigma_2,...,\sigma_L} = C_{\sigma_1,(\sigma_2,...,\sigma_L)} = \sum_{a_1,a_2} U_{\sigma_1,a_1} S_{a_1,a_1'} V^\dagger_{a_1',(\sigma_2,...,\sigma_L)}$$

From here-on we will drop summation signs and use a summation convention: whenever an index appears twice in a term, a summation over that index is implied.

$$\sum_{a_1,a_2} U_{\sigma_1,a_1} S_{a_1,a_1'} V^\dagger_{a_1',(\sigma_2,...,\sigma_L)} \rightarrow U_{\sigma_1,a_1} S_{a_1,a_1'} V^\dagger_{a_1',(\sigma_2,...,\sigma_L)}$$

We continue with reshapings and SVD's:

$$\begin{aligned}
C_{\sigma_1,\sigma_2,...,\sigma_L} &= U_{\sigma_1,a_1} S_{a_1,a_2} V^\dagger_{a_2,(\sigma_2,...,\sigma_L)} \\
&= U_{\sigma_1,a_1} C_{a_1,(\sigma_2,...,\sigma_L)} \\
&= U_{\sigma_1,a_1} C_{(a_1,\sigma_2)(\sigma_3,...,\sigma_L)} \\
&= U_{\sigma_1,a_1} U_{(a_1,\sigma_2),a_2} S_{a_2,a_2'} V^\dagger_{a_2',(\sigma_3,...,\sigma_L)} \\
&= U_{\sigma_1,a_1} U_{(a_1,\sigma_2),a_2} C_{a_2,(\sigma_3,...,\sigma_L)} \\
&= A^{\sigma_1}_{a_1} A^{\sigma_2}_{a_1,a_2} C_{a_2,(\sigma_3,...,\sigma_L)}
\end{aligned} \tag{1}$$

Where we have defined the left-normalized tensors $A^{\sigma_i}_{a_{i-1},a_i} = U_{(a_{i-1},\sigma_i),a_i}$

---

- **Q1:** *From the definition of the SVD the U matrices have the property $U^\dagger U = I$. Re-express this property for the $A^{\sigma_i}_{a_{i-1},a_i}$ tensors.*

---

If we continue this reshaping and SVD dance until the last index we eventually get

$$\begin{aligned}
C_{\sigma_1,\sigma_2,...,\sigma_L} &= A^{\sigma_1}_{a_1} A^{\sigma_2}_{a_1,a_2} ... A^{\sigma_{L-1}}_{a_{L-2},a_{L-1}} C_{a_{L-1},\sigma_L} \\
&= A^{\sigma_1}_{a_1} A^{\sigma_2}_{a_1,a_2} ... A^{\sigma_{L-1}}_{a_{L-2},a_{L-1}} A^{\sigma_L}_{a_{L-1}} C
\end{aligned}$$

Where we have replaced the last term $C_{a_{L-1},\sigma_L}$ by left-normalized tensor $A^{\sigma_L}_{a_{L-1}}$ and a single real number $C$.
We refer to the $a_i$ indices as the bond indices: they couple the individual $A^{\sigma_i}_{a_{i-1},a_i}$ tensors to represent the original $C_{\sigma_1,...,\sigma_L}$ tensor.

---

- **Q2:** *Can you express $C$ in terms of $C_{\sigma_1,\sigma_2,...\sigma_L}$? The left-normalization property of the $A$ tensors should make this easier.*
- **Q3:** *What are the maximal dimensions of these $A$ tensors? Is this exact tensor-train representation more compact than the original $d^L$ tensor?*

---

**SVD compression of a tensor-train**

As we were performing our tensor-train decomposition of the $C_{\sigma_1,...\sigma_L}$ tensor we could have truncated the singular values at every SVD in order to keep the bond dimensions of our tensors small. This is however just an illustrative example: tensor-network algorithms are typically applied when the tensor we wish to represent is too large to be SVD'ed.
Instead a typical problem is to reduce the bond dimensions of an existing tensor train.
One starts by left-normalizing the tensors:

$$\begin{aligned}
M^{\sigma_1}_{a_1} M^{\sigma_2}_{a_1,a_2} ... M^{\sigma_L}_{a_{L-1}} &= U_{\sigma_1,s_1} S_{s_1,s'_1} V^\dagger_{s'_1,a_1} M^{\sigma_2}_{a_1,a_2} ... M^{\sigma_L}_{a_{L-1}} \\
&= A^{\sigma_1}_{a_1} M'^{\sigma_2}_{a_1,a_2} M^{\sigma_3}_{a_2,a_3} ... M^{\sigma_L}_{a_{L-1}} \\
&= A^{\sigma_1}_{a_1} U_{(\sigma_2,a_1),s_2} S_{s_2,s'_2} V^\dagger_{s'_2,a_2} M'^{\sigma_3}_{a_2,a_3} ... M^{\sigma_L}_{a_{L-1}} \\
(...) &= A^{\sigma_1}_{a_1} A^{\sigma_2}_{a_1,a_2} ... A^{\sigma_L}_{a_{L-1}}
\end{aligned}$$

One then starts again from the right:

$$\begin{aligned}
A^{\sigma_1}_{a_1} A^{\sigma_2}_{a_1,a_2} ... A^{\sigma_L}_{a_{L-1}} &= A^{\sigma_1}_{a_1} ... A^{\sigma_{L-1}}_{a_{L-2},a_{L-1}} U_{a_{L-1},s_{L-1}} S_{s_{L-1},s'_{L-1}} V^\dagger_{s'_{L-1},\sigma_L} \\
&\approx A^{\sigma_1}_{a_1} ... A^{\sigma_{L-1}}_{a_{L-2},a_{L-1}} \tilde{U}_{a_{L-1},s_{L-1}} \tilde{S}_{s_{L-1},s'_{L-1}} \tilde{V}^\dagger_{s'_{L-1},\sigma_L} \\
&= A^{\sigma_1}_{a_1} ... A^{\sigma_{L-2}}_{a_{L-3},a_{L-2}} M^{\sigma_{L-1}}_{a_{L-2},a_{L-1}} \tilde{B}^{\sigma_i}_{a_{L-1}} \\
&= A^{\sigma_1}_{a_1} ... A^{\sigma_{L-2}}_{a_{L-3},a_{L-2}} U_{a_{L-2},s_{L-2}} S_{s_{L-2},s'_{L-2}} V^\dagger_{s'_{L-2},(a_{L-1},\sigma_{L-1})} \tilde{B}^{\sigma_L}_{a_{L-1}} \\
&\approx A^{\sigma_1}_{a_1} ... A^{\sigma_{L-3}}_{a_{L-4},a_{L-3}} M^{\sigma_{L-2}}_{a_{L-3},a_{L-2}} \tilde{B}^{\sigma_{L-1}}_{a_{L-2},a_{L-1}} \tilde{B}^{\sigma_L}_{a_{L-1}} \\
(...) &\approx \tilde{B}^{\sigma_1}_{a_1} ... \tilde{B}^{\sigma_L}_{a_{L-1}}
\end{aligned}$$

Where a tilde (as in $S \rightarrow \tilde{S}$) indicates a truncation of the smallest singular values.
We have also introduced the right-normalized tensors $B^{\sigma_i}_{a_{i-1},a_i} = V^\dagger_{a_{i-1},(a_i,\sigma_i)}$

---

- **Q4:** *Question 1, but now for the $V^\dagger$ matrices and $B$ tensors*
- **Q5:** *Quantify the $l^2$ norm error associated with each truncation. Why did we bring the surrounding tensors into left and right-normalized form before truncating?*
- **Q6 (open):** *To achieve a low bond dimension representation with low truncation error the singular values at each bond index should be rapidly decaying. In the context of physics this can be tied to the notion of entanglement, giving some intuition as to which problems can be handled by these methods and which cannot.*
  *Are you aware of similar problems in other fields in which you suspect such a condition would hold?*

---

We look forward to your opinions and insights.

Best Quant Regards,

swissQuant Group Leadership Team