Assignment #9

Course: *Machine learning*
Due date: *January 19th, 2023*

Assignment

In this assignment, you learn how to use matrix product states (MPS) to compress fully connected layers and perform simple MPS contractions.

Instead of compressing a weight matrix of a fully connected layer, we will consider the compression of a large image, which can also be nicely visualized.

Download a large image `https://www.publicdomainpictures.net/pictures/540000/velka/seamless-flowers-pattern-16952868310zL.jpg`, convert it to a grayscale, and resize it such that the height will be 4096 by keeping the aspect ratio constant. Finally, crop the image to the size 4096×4096 and convert it to a NumPy array A.

Write a method that compresses the image by transforming it to a matrix product state with a fixed bond dimension D.

- Reshape the 2D array A of size 4096×4096 representing the image to a tensor of size $\bigotimes^{24} 2$, i.e., $A_{\alpha,\beta} \rightarrow A_{i_1,\ldots,i_{12},j_1\ldots,j_{12}}$.
- Transpose the tensor obtained in the previous step such that the indices i and j will be alternating, i.e., $A_{i_1,\ldots,i_{12},j_1,\ldots,j_{12}} \rightarrow A_{i_1,j_1,i_2,j_2,\ldots,i_{12},j_{12}}$.
- Reshape the tensor obtained in the previous step by combining the even and odd indices, i.e., $A_{i_1,j_1,i_2,j_2\ldots,i_{12},j_{12}} \rightarrow A_{\alpha_1,\ldots,\alpha_{12}}$, where $\alpha_k = (i_k, j_k)$.
- Finally, compress the tensor obtained in the previous step by using successive SVD decompositions. In each decomposition keep at most D largest singular values. Use $D \leq 512$.

Write a function that transforms the MPS back to the original dimensions.

You have to perform the reshape and transpose operations in the reverse order as in the previous exercise.

Calculate the number of parameters in the compressed MPS. Visually compare the obtained MPS compressed image with the simple SVD compressed image with the same number of parameters.

Analyze the compression factor and the error of the compression as a function of the bond dimension, namely the size of the matrices in the MPS. Compare the errors obtained with the MPS and SVD compressed images as a function of the compression factor. Why is the MPS compression better? Isn't SVD an optimal decomposition concerning the 2-norm?

BONUS
Write an efficient sequential method to calculate the 2-norm of an MPS and compare it to `np.linalg.norm` method for the original matrix.

The following NumPy methods could be useful in the implementation of the assignment:

- `https://numpy.org/doc/stable/reference/generated/numpy.einsum.html`,
- `https://numpy.org/doc/stable/reference/generated/numpy.transpose.html`,
- `https://numpy.org/doc/stable/reference/generated/numpy.reshape.html`.