Aalto University
School of Science
Department of Applied Physics

Niko Savola

Machine learning with many-body tensor networks

Submitted for approval: 1st June 2022

Special exercise

PHYS-E0421 – Solid-State Physics

1 Introduction

Tensor networks have been originally developed for efficiently storing and manipulating high-dimensional quantum many-body states [1]. These variational families of wavefunctions emerge from low-entanglement representations of quantum states. Contemporary applications however include a wide range of fields, such as, machine learning [2], statistical mechanics [3], quantum chemistry [4], and cosmology [5]. This exercise aims to present the theory behind tensor networks and their use in machine learning. Then we present a numerical example for training a tensor network representing a quantum circuit, effectively pre-training a quantum computer for solving a problem.

2 Theory

2.1 Tensor networks

Tensor networks (TN) are a powerful tool for splitting a high-dimensional function into constituent parts of smaller tensors. Here a tensor refers to a multidimensional array, rank zero corresponding to a scalar, rank one to a vector, rank two to a matrix, and further ranks being referred to as rank n tensors. It is natural to use Tensor network notation (TNN), which can be considered a graphical generalisation of Einstein summation [6]. This notation maps tensors to nodes with n lines corresponding to the rank n of the tensor. The lines represent indices for the multidimensional array as demonstrated in Fig. 1(a).

The elegance of TNN arises from contracting tensors. Multiplication is done by linking the lines of the tensors. For example, matrix-vector multiplication consists of linking a node with one line to a node with two lines. The ensuing tensor will have one free line, resulting in a vector as expected. This logic is further demonstrated for a few examples in Fig. 1(b). Notice how multiplication of high-rank tensors is rendered graphically trivial. This is indeed useful for representing complex architectures of TN.

2.1.1 Machine learning

As popularity of deep learning has risen rapidly [8], interest in using TN as replacements or in conjunction with neural networks (NN) has been explored. Although TN are still an active research topic, some prominent TN architectures for machine learning have emerged, such as, Matrix Product States (MPS) and Tensor Renormalization Group (TRG).

TODO MPS

TODO TRG https://tensornetwork.org/trg/

One branch of research involves using a TN directly as machine learning model architecture. Another uses TNs to compress layers in neural network architectures or for other auxiliary

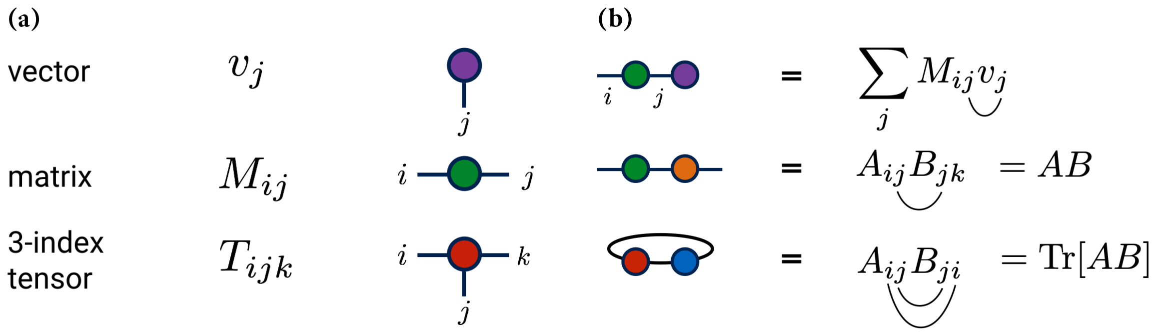

Figure 1: (a) Examples of low-rank tensors and their TNN representations. (b) Examples of tensor contractions for matrices. From top to bottom: matrix-vector product, matrix multiplication, and their trace. Note how the last example elegantly results in a scalar. Schematics from Ref. [7]

tasks.

a paradigm called differentiable programming.

In fact, it has been shown that there exists a mapping from generative neural networks referred to as Restricted Boltzmann machines to TNs [9], highlighting the link to deep learning.

2.2 Modelling many-body physics

todo work as Ansätze

3 Example: Training a quantum circuit

TODO EXPLAIN WHAT WE ARE DOING

3.1 Implementation

We use an ansatz circuit consisting of quantum U gates and Controlled-Z gates (CZ). The U-gates represent all possible single-qubit operations and are of the form

$$U(\theta, \phi, \lambda) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -e^{i\lambda}\sin\left(\frac{\theta}{2}\right) \\ e^{i\phi}\sin\left(\frac{\theta}{2}\right) & e^{i(\phi+\lambda)}\cos\left(\frac{\theta}{2}\right) \end{pmatrix}. \tag{1}$$

The CZ gates are two-qubit gates flipping the phase if one of the qubits is in the $|1\rangle$ state. It is a symmetric operation and is represented as

$$CZ(q_1, q_0) = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes \sigma_z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}, \tag{2}$$

where σ_z is a Pauli-Z gate. Our ansatz is implemented numerically in Python using the *quimb* library [10] as

```python
import quimb as qu
import quimb.tensor as qtn

n = 5
depth = 4

circ = qtn.Circuit(n)

for d in range(depth):
    for i in range(circ.N):
        params = qu.randn(3, dist='uniform') # initialize with random parameters
        circ.apply_gate('U3', *params, i, gate_round=d, parametrize=True)

    for i in (reversed(regs) if (d % 2 == 0) else regs):
        circ.apply_gate('CZ', i, i + 1, gate_round=d)

# final single qubit layer
for i in range(circ.N):
    params = qu.randn(3, dist='uniform') # initialize with random parameters
    circ.apply_gate('U3', *params, i, gate_round=r, parametrize=True)
```

The resulting ansatz consists of exclusively tensors. Thus, a graph complying with TNN presented in Fig. 1 can be generated. Similar graphs are shown in Fig. 2 with colouring displaying the quantum gates and qubit indices in (a) and (b), respectively.

(a) **(b)**

Figure 2: TNN graph of our ansatz quantum circuit with colouring for (a) quantum gates (b) qubit indices. The open ends of the TN, labelled with k and b, are consequently linked to a target problem.

The TN is then connected to TODO qu.ham_ising

3.2 Results

todo suprajohto qubit toteutus https://arxiv.org/pdf/2103.12305.pdf

but the motivation is that the architecture generalises to unknown problem

Figure 3: Result of the trained quantum ansatz circuit generated with Qiskit [11]. The purple blocks represent U-gates, while the blue lines with dots are Controlled-Z gates. The U-gates are the most general form of single-qubit gates and thus on a physical device might be implemented as a combination of different gates depending on the given rotations in the noisy intermediate-scale quantum era (NISQ) [12].

4 Summary

Todo

TODO https://github.com/google/TensorNetwork
https://arxiv.org/pdf/1905.01331.pdf

References

[1] S. R. White, 'Density matrix formulation for quantum renormalization groups,' *Phys. Rev. Lett.*, vol. 69, no. 19, pp. 2863–2866, November 1992, DOI: 10.1103/PhysRevLett.69.2863.

[2] C. Roberts, A. Milsted, M. Ganahl *et al.*, *TensorNetwork: A Library for Physics and Machine Learning*, 2019, arXiv: 1905.01330.

[3] M. Levin and C. P. Nave, 'Tensor Renormalization Group Approach to Two-Dimensional Classical Lattice Models,' *Physical Review Letters*, vol. 99, no. 12, p. 120 601, Sep. 2007, DOI: 10.1103/PhysRevLett.99.120601.

[4] S. R. White and R. L. Martin, 'Ab initio quantum chemistry using the density matrix renormalization group,' *The Journal of Chemical Physics*, vol. 110, no. 9, pp. 4127–4130, March 1999, DOI: 10.1063/1.478295.

[5] N. Bao, C. Cao, S. M. Carroll and A. Chatwin-Davies, 'de Sitter space as a tensor network: Cosmic no-hair, complementarity, and complexity,' *Physical Review D*, vol. 96, no. 12, p. 123 536, December 2017, DOI: 10.1103/PhysRevD.96.123536.

[6] J. C. Bridgeman and C. T. Chubb, 'Hand-waving and interpretive dance: an introductory course on tensor networks,' *Journal of Physics A: Mathematical and Theoretical*, vol. 50, no. 22, p. 223 001, May 2017, DOI: 10.1088/1751-8121/aa6dc3.

[7] E. M. Stoudenmire, *Tensor Diagram Notation*, April 2021, [Online]. Available: https://tensornetwork.org/diagrams (visited on 30th May 2022).

[8] M. M. Yapıcı, A. Tekerek and N. Topaloğlu, 'Literature Review of Deep Learning Research Areas,' *Gazi Journal of Engineering Sciences*, vol. 5, no. 3, pp. 188–215, December 2019, DOI: 10.30855/gmbd.2019.03.01.

[9] J. Chen, S. Cheng, H. Xie, L. Wang and T. Xiang, 'Equivalence of restricted Boltzmann machines and tensor network states,' *Physical Review B*, vol. 97, no. 8, p. 085 104, February 2018, DOI: 10.1103/PhysRevB.97.085104.

[10] J. Gray, 'quimb: A python package for quantum information and many-body calculations,' *Journal of Open Source Software*, vol. 3, no. 29, p. 819, 2018, DOI: 10.21105/joss.00819.

[11] M. S. ANIS, Abby-Mitchell, H. Abraham *et al.*, *Qiskit: An open-source framework for quantum computing*, 2021, DOI: 10.5281/zenodo.2573505.

[12] A. Li, S. Stein, S. Krishnamoorthy and J. Ang, 'QASMBench: A Low-level QASM Benchmark Suite for NISQ Evaluation and Simulation,' May 2020, arXiv: 2005.13018.