Tensor Networks and Deep Learning for Simulations in Quantum Mechanics

Moritz August, august@in.tum.de
Doctoral candidate at the Chair of Scientific Computing, Department of Informatics, TUM
Member of the doctoral programme *Exploring Quantum Matter*
Joint work with Thomas Huckle, Mari Carmen Bañuls and Xiaotong Ni

Tensor networks and (artificial) neural networks allow us to tackle previously unsolvable problems in quantum physics and computer science

Tensor Networks

- Developed simultaneously in quantum physics and numerical mathematics
- Provide a way to decompose tensors of high order or dimensionality
- Represent tensors by contractions of many small tensors
- Inherently linear, basic arithmetic operations can be expressed
- Known to represent ground states of certain systems faithfully
- Methods to perform time evolution and much more exist

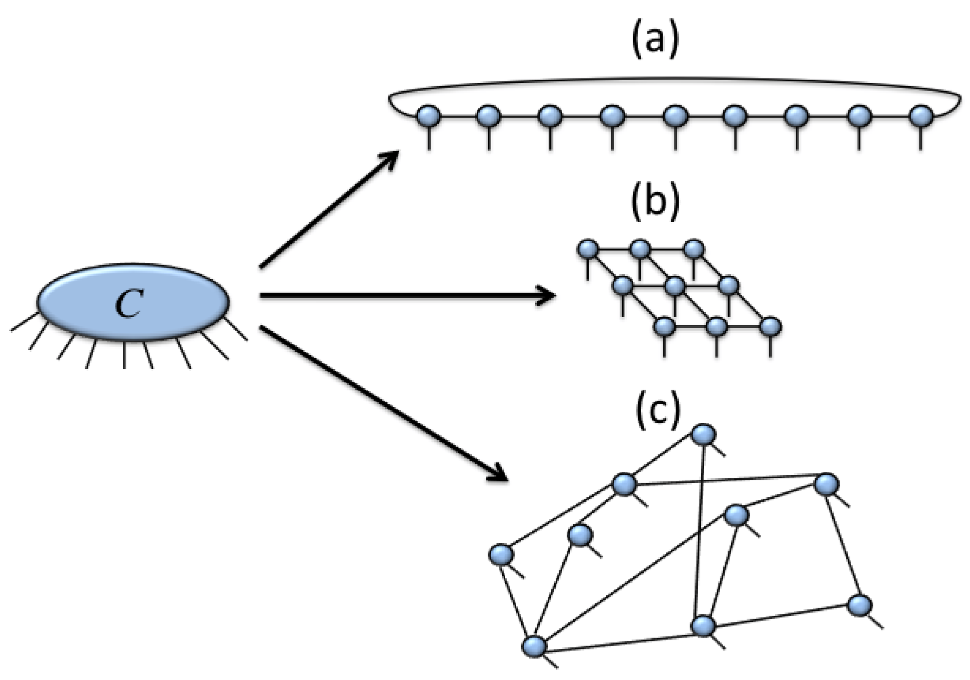


Figure taken from *A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States* by R. Orus

Neural Networks

- Approximate function by composition and summation of simple functions
- Based on connectionist paradigm, inspired by the brain
- Topology of the networks reflects causal structure of data
- Expressive power owed to composition of simple non-linearities
- Term *Deep Learning* refers to high number of compositions
- Special kinds of networks for images, time-series data, etc.
- Can be used for regression, classification, data generation, compression ...

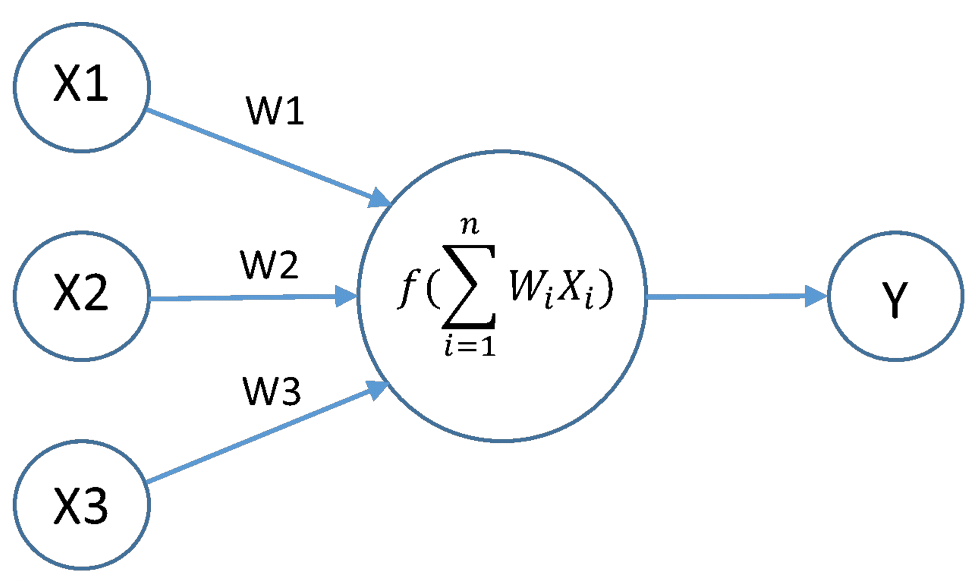


Figure taken from *2centsapiece.blogspot.de*

Our work focuses on the development of novel algorithms to solve challenging problems in quantum physics

Approximating Functionals of MPOs

Algorithm

Algorithm 1: Approximation Algorithm

Input : MPO $A[D_A] \in \mathbb{C}^{N\times N}$, Orthogonal MPO $U[D_{init}] \in \mathbb{C}^{N\times N}$, Number of Dimensions K, Maximal Bond-Dimension D_{max}, Stopping Criteria $\mathcal{S}$

$U_0 \leftarrow U$;
for $i \leftarrow 1; i \leq K$ **do**
  $U_i, T_i \leftarrow$ `orthonormalize`(A, U_{i-1}, D_{max}) ;
  $V\Lambda V^* \leftarrow$ `spectralDecomposition`(T_i) ;
  $\mathcal{G}f \leftarrow \beta_1^2 e_1^T V f(\Lambda) V^* e_1$;
  if `checkStop`$(\mathcal{G}f, \Lambda, \mathcal{S})$ **then**
    break ;
  end
end

Output: Approximation $\mathcal{G}f$ of Tr$f(A)$

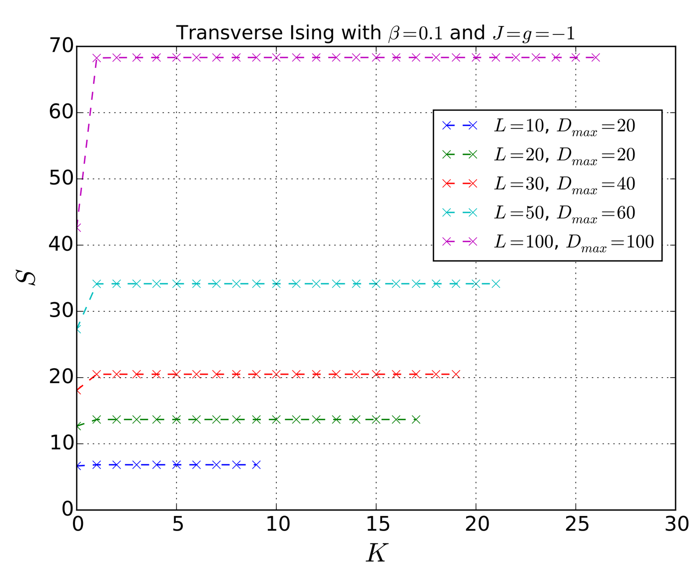
Background

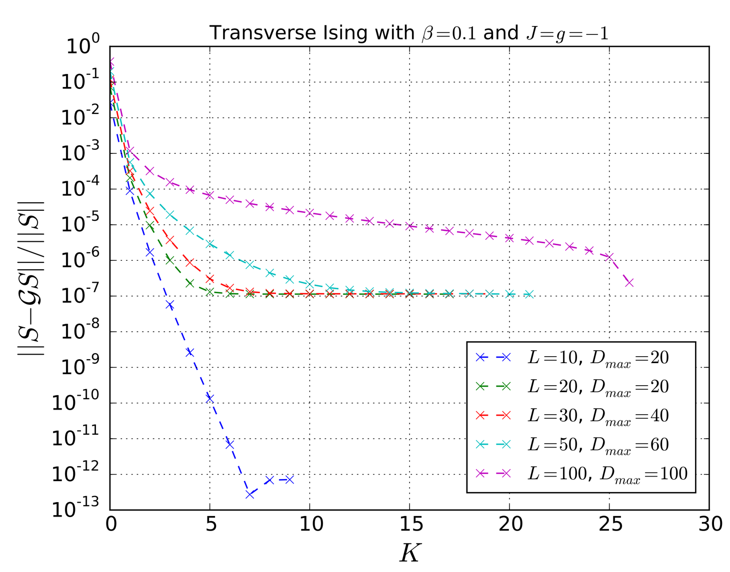
- Can we approximate functions Tr$f(A)$ like the entropy or the trace norm?
- Key idea: project MPO onto small, explicitly storable matrix
- Use Krylov basis of *MPOs* generated by $U, AU, A^2U, \cdots, A^{K-1}U$
- Starting MPO U is orthogonal and of the same dimensions as A
- Projection yields tridiagonal matrix $T_K \in \mathbb{R}^{K\times K}$ that is given by

$$T_K = \begin{bmatrix} \alpha_1 & \beta_2 & & \mathbf{0} \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_K \\ \mathbf{0} & & \beta_K & \alpha_K \end{bmatrix}$$

- Problem can be reformulated such that T_K is intimately related to Gauss quadrature
- We find that it holds Tr$f(A) =$ Tr$(U^* f(A) U) = \int_a^b f(\lambda)\mathrm{d}\mu(\lambda) \approx e_1^T f(T_K) e_1$

Results





Optimizing Dynamical Decoupling

Algorithm

Algorithm 2: Optimization Algorithm

Input : Number of initial models: n, Number of models to keep: k, Percentage of data: p, Set of possible topologies: $\mathcal{M}$, Size of data: d

$D \leftarrow$ `generateRandomData(`*d*`)` ;
$D, \langle\varsigma_s\rangle \leftarrow$ `keepBestData(`*D,p*`)` ;
$M \leftarrow$ `trainRandomModels(`*n, D, $\mathcal{M}$*`)` ;
$M \leftarrow$ `keepBestKModels(`*M,k*`)` ;
while $\langle\varsigma_s\rangle$ *not converged* **do**
  $M \leftarrow$ `trainBestModels(`*D*`)` ;
  $D \leftarrow$ `generateDataFromModels(`*M, d*`)` ;
  $D, \langle\varsigma_s\rangle \leftarrow$ `keepBestData(`*D,p*`)` ;
end

Output: $\langle\varsigma_s\rangle, D, M$

Background

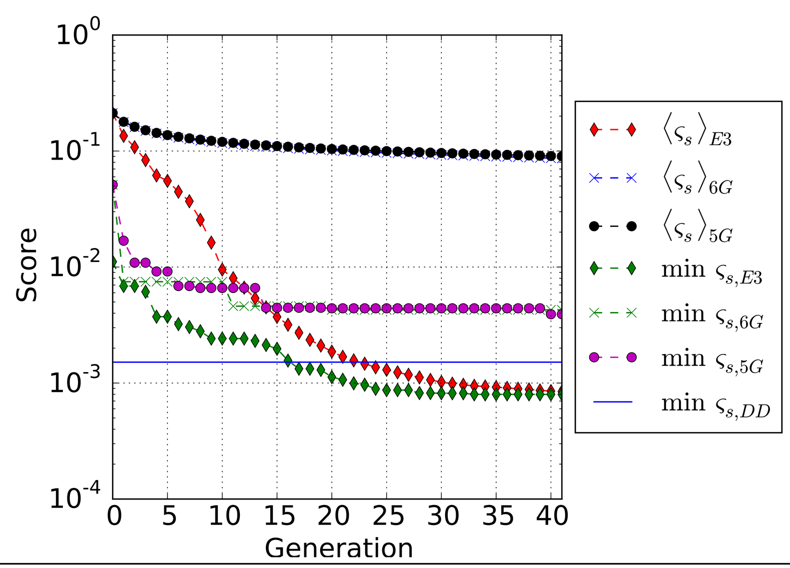
- Can we learn to generate good dynamical decoupling sequences?
- Assuming no access to fidelity measure except for effcient evaluation
- Conjecture: RNNs can learn structure of and generate good sequences
- The fidelity measure we would like to optimize is defined as

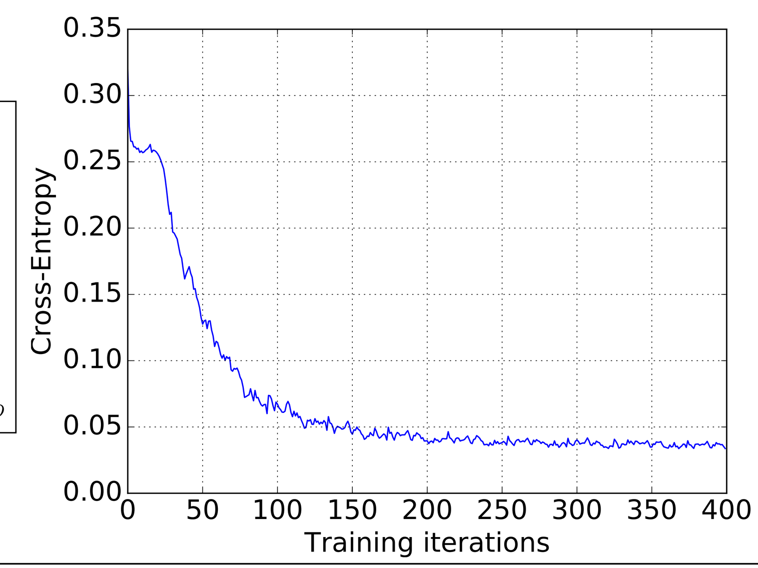
$$D(U, I) = \sqrt{1 - \frac{1}{d_S d_B} \|\mathrm{Tr}_S(U)\|_{\mathrm{Tr}}}$$

- Piecewise constant, finite strength control Hamiltonian with Pauli gates
- Ansatz: use sequence data to train RNNs, use RNNs to generate better data
- We use the *cross entropy* as error function for RNN-training

$$CE(m, \{i\}) = -\sum_t \delta_{s_t,i} \log p_{m,i}(s_{t-1}, \ldots, s_1)$$

Results





[1] M. August, M. C. Bañuls, and T. Huckle. On the approximation of functionals of very large hermitian matrices represented as matrix product operators. *arXiv preprint arXiv:1610.06086*, 2016.
[2] M. August and X. Ni. Using recurrent neural networks to optimize dynamical decoupling for quantum memory. *arXiv preprint arXiv:1604.00279*, 2016.