Matrix Product States and Tensor Product States

Jan Müller

June 20, 2013

Even the fastest and most powerful computers of nowadays are not able to calculated bigger quantum mechanical systems. To get an impression of the information stored when computing quantum systems we have a look at a one-dimensional chain of N sites with a d-dimensional Hilbert space at each site. The total Hilbert space therefore has dimension $dim = d^N$, which already for 40 sites of a spin-$\frac{1}{2}$ system is too much to be calculated by a classical computer. The idea behind matrix product states (MPS) now is to avoid this huge Hilbert space if the entanglement entropie grows proportional to the size of the system's boundary (area law).

1 Constructing Matrix Product States

First we consider a general state of the quantum system:

$$|\Psi\rangle = \sum_{\{\sigma_i\}=1}^{d} \varphi_{(\{\sigma_i\})} |\sigma_1, ..., \sigma_N\rangle$$

Perform a singular value decomposition on $\varphi_{(\{\sigma_i\})}$, interpreting the $\varphi_{(\{\sigma_i\})}$ as the entries of a matrix φ:

$$\varphi_{(\{\sigma_i\})} = \varphi_{(\sigma_1),(\sigma_2,...,\sigma_N)} \overset{SVD}{=} \sum_{\alpha_1} U^{(1)}_{\sigma_1,\alpha_1} \lambda^{(1)}_{\alpha_1} V^{(1)}_{\alpha_1,(\sigma_2,...,\sigma_N)}$$

Continue iteratively decomposing the matrix V:

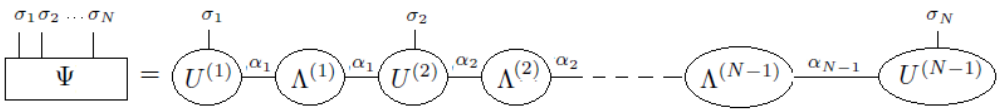
$$V^{(1)}_{(\alpha_1,\sigma_2),(\sigma_3,...,\sigma_N)} \overset{SVD}{=} \sum_{\alpha_2} U^{(2)}_{(\alpha_1,\sigma_2),\alpha_2} \lambda^{(2)}_{\alpha_2} V^{(2)}_{\alpha_2,(\sigma_3,...,\sigma_N)}$$

$$\varphi_{(\{\sigma_i\})} = \sum_{\{\alpha_i\}} U^{(1)}_{\sigma_1,\alpha_1} \lambda^{(1)}_{\alpha_1} U^{(2)}_{(\alpha_1,\sigma_2),\alpha_2} \lambda^{(2)}_{\alpha_2} ... \lambda^{(N-1)}_{\alpha_{N-1}} U^{(N)}_{\alpha_{N-1},\sigma_N} = U^{(1)}_{\sigma_1} \Lambda^{(1)} U^{(2)}_{\sigma_2} ... \Lambda^{(N-1)} U^{(N)}_{\sigma_N}$$

So the completely decomposed matrix product state becomes:

$$|\Psi\rangle = \sum_{\{\sigma_i\}} U^{(1)}_{\sigma_1} \Lambda^{(1)} U^{(2)}_{\sigma_2} ... \Lambda^{(N-1)} U^{(N)}_{\sigma_N} |\sigma_1, ..., \sigma_N\rangle$$

And in the pictorial notation:



2 Truncation of MPS

Use the Schmidt decomposition to cut the system after site l. The corresponding von Neumann entanglement entropie follows as:

$$S^{(l)} = -\sum_{\alpha=1}^{\kappa_l} \left(\lambda_\alpha^{(l)}\right)^2 log\left(\left(\lambda_\alpha^{(l)}\right)^2\right)$$

Use area law: The entanglement entropy grows proportional to the subsystems boundary and therefore is constant in one-dimensional systems. The $\lambda_\alpha^{(l)}$ therefore have to decay fast in κ_l which is related to the system size.

Idea: Introduce a limiting dimension D for the matrices and neglect all terms with $\alpha > D$.

Achievement: Now we only have about $N \cdot d \cdot D^2$ parameters left that describe the state instead of d^N compared to the previous notation. This is an outstanding result as a linear dependency on N is almost the best one can have for an alogithm compared to the exponential dependency being the worst.

Assuming maximal entanglement all states need to have the same probability. Due to the normalization follows:

$$S_{max}^{(D)} = log(D)$$

3 Tensor Product States

A generalization of MPS to tensor product states (TPS or also Tensor Network States TNS) can be achieved by simply adding more indices to the matrices and this way extending them to tensors of any order.

One possibility to do so is the PEPS (i.e. Projected Entangled Pair States) method which can most easily be understood by the pictorial representation. In this figure all indices which go with the sigmas are left out to avoid the whole construction getting too messy. Same holds for the U matrices.