Machine-Learning in modern Many-Body Physics

Jonas Kell

University of Augsburg

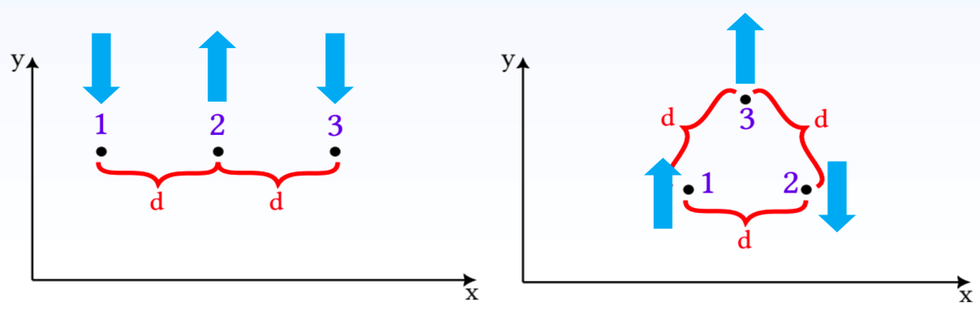
Introduction

While always a relevant scientific tool, machine-learning has hit mainstream adoption in the last few years, because of the introduction of *"AI"* agents like *Chat-GPT* to the masses. Originally the pattern-recognition capabilities of machine-learning models were used mainly in research and specialized industrial applications. Nowadays everyone thinks of *LLMs* when hearing of AI.

Yet there are many up-to-date scientific applications of machine-learning remaining. This poster is supposed to give a brief introduction to how machine learning is used in modern *many-body* physics research.

Physical Problem

Many-body physics attempts to describe the behavior of systems that are comprised of many individual particles. In this case specifically *condensed-matter* or *solid-state* physics describe the behavior of *crystalline* structures. This requires an underlying *lattice* with fixed sites. At the sites might be different particles. Easiest to visualize, they could be magnetic spins that either can point up or down.



Then different problems might be looked at, like finding the lowest energy state (ground-state-search) or trying to model how a system behaves dynamically (time-evolution). Both can be modeled by taking a matrix-exponential of the square *Hamiltonian* matrix.

$$
\begin{aligned}
|\Psi(t)\rangle &= e^{-i\mathcal{H}t/\hbar}\,|\Psi(0)\rangle \\
&= \sum_n e^{-i\mathcal{H}t/\hbar}\,\langle n|\Psi(0)\rangle\,|n\rangle
\end{aligned}
$$

$$
\begin{array}{c}
 \\
|\uparrow,\uparrow,\uparrow\rangle \\
|\uparrow,\uparrow,\downarrow\rangle \\
|\uparrow,\downarrow,\uparrow\rangle \\
|\uparrow,\downarrow,\downarrow\rangle \\
|\downarrow,\uparrow,\uparrow\rangle \\
|\downarrow,\uparrow,\downarrow\rangle \\
|\downarrow,\downarrow,\uparrow\rangle \\
|\downarrow,\downarrow,\downarrow\rangle
\end{array}
\begin{array}{c}
\begin{array}{cccccccc}
\langle\uparrow,\uparrow,\uparrow| & \langle\uparrow,\uparrow,\downarrow| & \langle\uparrow,\downarrow,\uparrow| & \langle\uparrow,\downarrow,\downarrow| & \langle\downarrow,\uparrow,\uparrow| & \langle\downarrow,\uparrow,\downarrow| & \langle\downarrow,\downarrow,\uparrow| & \langle\downarrow,\downarrow,\downarrow|
\end{array} \\
\begin{pmatrix}
3\cdot J & h & h & 0 & h & 0 & 0 & 0 \\
h & -1\cdot J & 0 & h & 0 & h & 0 & 0 \\
h & 0 & -1\cdot J & h & 0 & 0 & h & 0 \\
0 & h & h & -1\cdot J & 0 & 0 & 0 & h \\
h & 0 & 0 & 0 & -1\cdot J & h & h & 0 \\
0 & h & 0 & 0 & h & -1\cdot J & 0 & h \\
0 & 0 & h & 0 & h & 0 & -1\cdot J & h \\
0 & 0 & 0 & h & 0 & h & h & 3\cdot J
\end{pmatrix}
\end{array}
$$

While this is mathematically simple, the problem arises from the fact that the size of the base (possible up-down-combinations) grows exponentially with the number of sites. For specific shapes of the matrices (some problem-classes) one can find analytical solutions. But in general the problem has exponential complexity. And also: to model a real physical system, one needs a billion billion ($6{,}022\cdot10^{23}$) particles and numerical methods can solve exponential problems to maybe 100 particles.

Scaling to relevant sizes by brute-force is unfeasible even with all computational resources on earth.

Ansatz

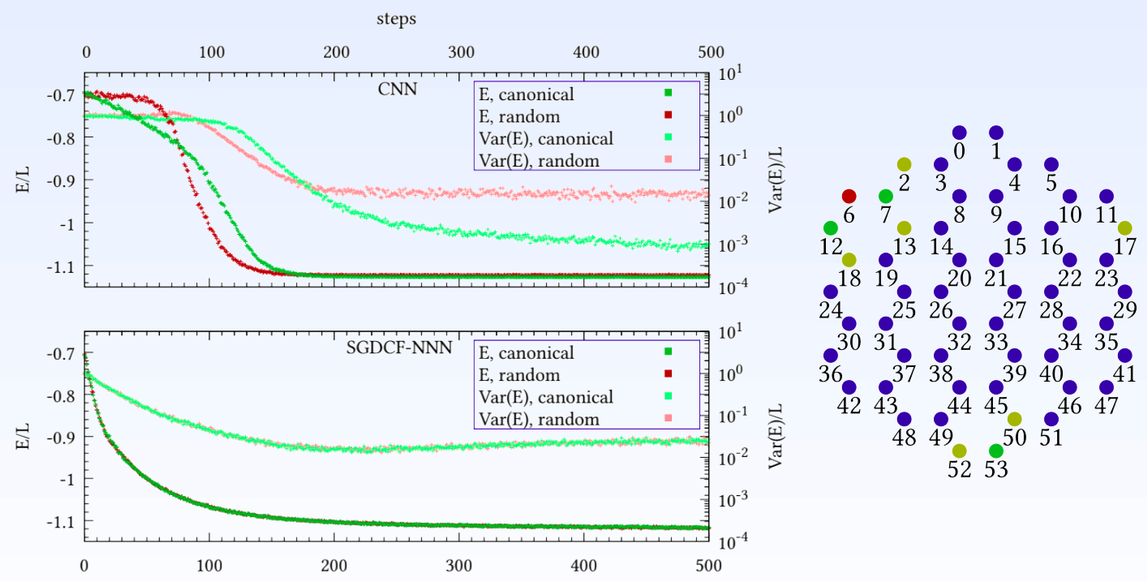
A possible solution to this dilemma could be the employment of machine-learning inspired methods to get a "close-enough" solution approximating the wave-function without requiring the exponential input size. This is exactly the same reason why image-recognition works (similar features have similar "effects").

Parametrization

This is achieved by replacing a full wave-function with a parametrization, that has parameters that can be trained like in a machine-learning model.

$$
\mathcal{H}_N(t) = \sum_i C_i(t)\cdot\Phi_i(N) \quad\leftrightarrow\quad \mathcal{H}_{\mathrm{VCN}}(N,\vec{\eta},t) = \sum_i \eta_i(t)\cdot\Phi_i(N)
$$

The choice of parametrization here-by makes or breaks the success of the procedure. The parametrization might have locational biases, akin to image-recognition methods. Here the graph-transformers were investigated to bring the transformer architecture of LLMs to physics and bake locational crystal information into the model.



Other strategies include the so-called *PINs*, the *physically-inspired-networks*, where a result is taken from a physical reasoning process (like *perturbation-theory*) and then the structure of the formulas is kept and the analytic coefficients are replaced with learnable network-parameters.

From known starting values, a time-evolution can be obtained with repeated small time steps. This is the *TDVP* (*time-dependend variational principle*).

$$
\left|\Psi_{\vec{\eta}+\delta\cdot\dot{\vec{\eta}}}(t+\delta)\right\rangle \quad\leftrightarrow\quad e^{-i\mathcal{H}\delta}\left|\Psi_{\vec{\eta}}(t)\right\rangle
$$

$$
\sum_{k'} \overleftrightarrow{S}_{k,k'}\dot{\vec{\eta}}_{k'} = -i\vec{F}_k
$$

$$
\dot{\vec{\eta}} = -i\overleftrightarrow{S}^{-1}\vec{F}
$$

Learning / Sampling

The *training* of the parametrized models has many similarities to other machine-learning optimizations. However here there is too much available training-data and not too little. Yet by the same principle, one might hope that a subset of the states – obtained via *Monte-Carlo sampling* – is enough to reveal patterns in the problem and extract a "higher truth", a few-parameter solution.

$$
\mathrm{P}(N,t) = \frac{\left|e^{\mathcal{H}_{\mathrm{eff}}(N,t)}\right|^2|\Psi_N|^2}{\sum_K \left|e^{\mathcal{H}_{\mathrm{eff}}(K,t)}\right|^2|\Psi_K|^2} \quad \sum_N \mathrm{P}(N,t)\hat{\mathcal{O}}_{\mathrm{loc}}(N,t)\cdot \approx \frac{1}{|\{N\}_{\mathrm{MC}}|}\sum_{\{N\}_{\mathrm{MC}}} \hat{\mathcal{O}}_{\mathrm{loc}}(N,t)
$$

Bibliography

1. **https://github.com/jonas-kell/bachelor-thesis-documents**
2. **https://github.com/jonas-kell/master-thesis-documents**