QUANTUM MACHINE LEARNING

MIHA ROT

Fakulteta za matematiko in fiziko
Univerza v Ljubljani

Advances in processing power and algorithms have made machine learning a very potent tool. This article attempts to introduce machine learning and present two applications of machine learning in quantum physics. The first example deals with the ground state energy of an electron in 2D potential, while the second one touches upon the applicability of machine learning to the quantum many-body problems. The article concludes with a brief description of potential machine learning speed-ups promised by quantum computers.

KVANTNO STROJNO UČENJE

Zaradi napredkov v algoritmih in procesorski moči je postalo strojno učenje zelo močno orodje. Članek se začne s kratko predstavitvijo strojnega učenja, nato predstavi dva primera uporabe strojnega učenja v kvantni fiziki in zaključi z opisom potencialnih pohitritev, ki jih strojnemu učenju obljubljajo kvantni računalniki. Prvi primer obravnana metodo za določanje osnovnega stanje elektrona v 2D potencialu, drugi pa predstavi uporabo strojnega učenja pri opisu kvantnih mnogodelčnih sistemov.

1. Introduction

Pattern recognition and strategy optimisation is something that comes naturally to humans, but we quickly reach the limits of our processing power as the quantity and complexity of data increases. Machine learning algorithms are useful for discovering patterns in data that might not be apparent at first. They are designed to 'learn', that is to adjust their parameters, in one form or another, to the known training data set and construct a model that can subsequently be used on unknown data.

Algorithms capable of learning to execute tasks they were not explicitly programmed for are not a new idea. The first articles about neural nets appeared as early as 1958 [1], but they were limited by the technology of their day, as the implementation tends to be computationally complex. Improvements in processing power and data storage have led to the relatively recent explosion of machine learning applications ranging from image recognition to risk assessment and beyond.

Machine learning and quantum physics have a sort of symbiotic relationship. Quantum computers (and algorithms) can be used to speed up the computationally complex learning process [2], while machine learning allows for a different insight into quantum systems that tend to be large and unintuitive.

The first part of the article will provide a brief introduction to machine learning, followed by the main part presenting two examples of machine learning applied to quantum physics and a brief outline of potential speed-ups with quantum algorithms at the end.

2. Machine Learning

There are many different machine learning algorithms and it is important to choose the right one for the task at hand, as there are multiple trade-offs. Complex algorithms provide better accuracy, but it is much harder, if not impossible, to interpret the reasoning behind their results, which can be detrimental when conducting research or correcting errors. Over-fitting and under-fitting data should be avoided by using algorithms with the appropriate number of parameters for the training data that is available [3].

Algorithm selection is also influenced by the available type of data. Supervised learning (used in 3.1) works with labelled data[1] to infer a model that maps between the input and the desired output. Unsupervised learning groups and finds structure in unlabelled data. Reinforcement learning (used in 3.2) uses some sort of cost function to improve over time based on the previous results.

3. Simulating the quantum phenomenon with machine learning

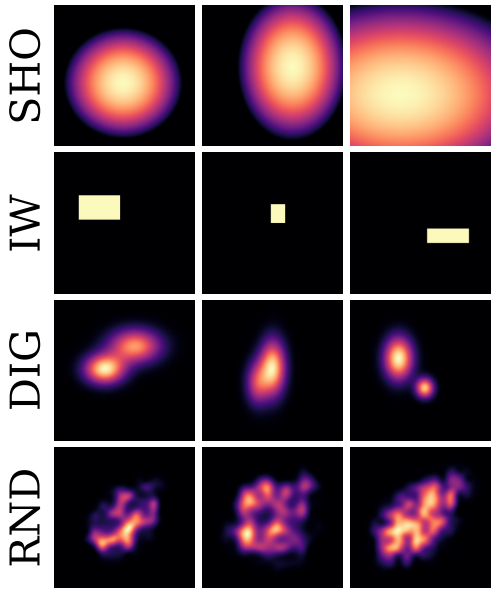
Applying machine learning algorithms to quantum physics is an interesting and relatively new field. Quantum problems are suitable for machine learning as there are large quantities of data with non-trivial correlations, especially in many-body problems where the dimension of the vector space increases exponentially with the number of particles. This section will deal with two examples from the literature that represent the various problems that can be tackled through machine learning.

3.1 Neural networks and the single particle Schrödinger equation

This section will review the work in the article [4] related to the calculation of the ground-state energy of an electron in a 2D potential. Ground-state energy is a well-known problem with analytical solutions[2] which can serve as a convenient test case for machine learning algorithms.

Authors used four classes of potentials displayed in figure 1: simple harmonic oscillators (SHO), infinite wells (IW), double-well inverted Gaussians (DIG) and random potentials (RND). The first two are relatively simple, only dependent on 2 parameters, and analytically solvable, while the complexity of the latter two escalates and poses a real challenge.

Solving electronic structure problems is important in a large number of disciplines and it would prove beneficial to develop an algorithm that could provide efficient[3] solutions, especially as we increase the number of particles which causes the computational complexity of traditional algorithms to increase polynomially.



Slika 1. Heat-map visualization of the four different classes of potentials $V(x, y)$ with three representative examples for each one. Brighter colour represents lower values. Reproduced from [4]

The aim of machine learning considered here is constructing a model that provides an efficient mapping between the electrostatic potential and the corresponding ground-state energy, effectively bypassing the step of solving the Schrödinger equation $H\Psi = \epsilon\Psi$. Each potential is represented as a 256×256 matrix of floating point numbers that can be visualized as a heat-map.

3.1.1 Machine learning method selection

The authors have attempted a featureless[4] approach where a machine learning algorithm is provided the same data as a numerical method and is left to learn both the relevant attributes of the system and the mapping required to provide the relevant solution. Such an approach promises easier

[1] Prepared dataset that contains both the algorithm input and the corresponding desired output.
[2] For certain classes of potentials (eg. simple harmonic potential).
[3] Fixed initial training cost can be ignored in this context.
[4] An approach where the data/model is not specifically adapted to expose attributes specific to the problem.

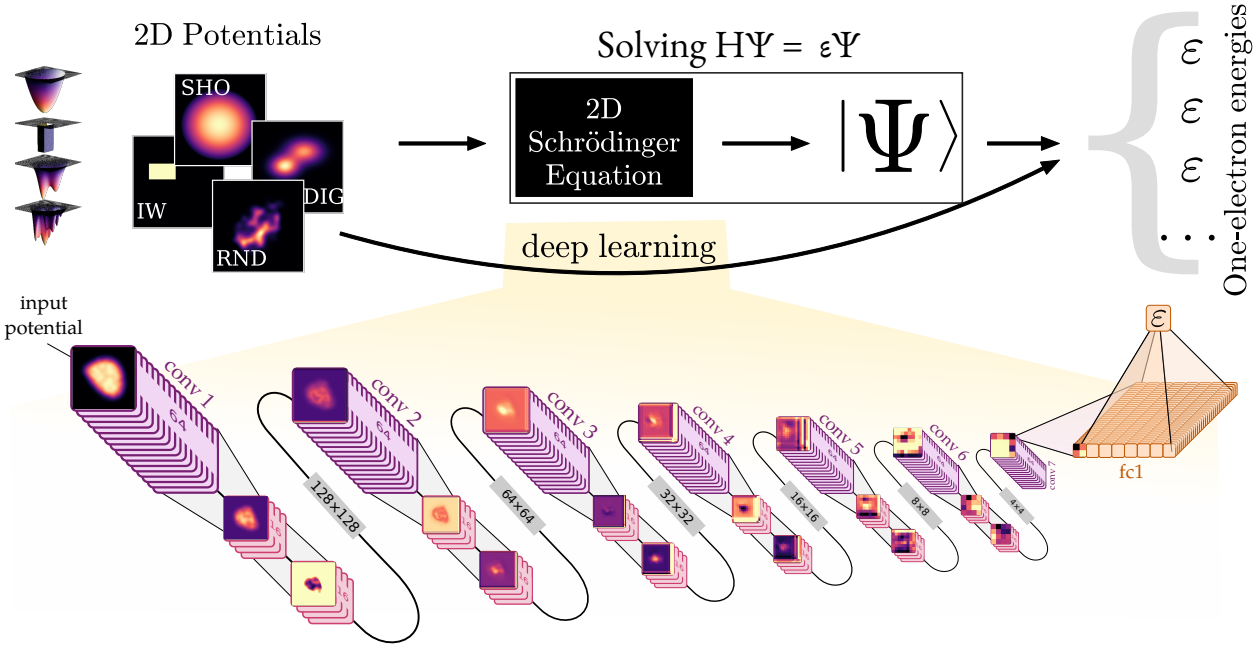
scalability for larger and more complex systems, that are the motivation for faster methods, but require deep learning methods.

The first step towards a solution is choosing the correct method. Several machine learning algorithms[5] were compared based on the real time of training, execution, and errors at predicting ground-state energies for SHO and random potentials. While other algorithms provided comparable or better results at SHO, the deep neural network proved to be far superior at random potentials while also being the fastest (with a caveat of dissimilar code optimization).

The schematic version of the chosen deep neural network algorithm is displayed in figure 2. A convolutional neural network consists of subsequent 'reducing' and 'non-reducing' convolutional layers. In this case, there are 7 reducing layers, each reducing the image resolution by a factor of 2 and 2 non-reducing layers (12 in total) between each reducing layer, serving to add trainable parameters to the network. The final convolutional layer feeds into a fully connected layer that is subsequently used to calculate the result.

Convolutional layers work by sliding (convolving) a filter[6] over the input matrix to calculate the output, effectively combining neighbouring values based on the filter values. A layer consists of multiple filters with values that serve as trainable parameters. In this example, the reducing layers use 64 3×3 filters with a stride of 2×2, thus reducing the resolution by a factor of 2, while the non-reducing layers work with 16 4×4 filters and a unit stride that conserves the resolution. The output of a convolutional layer is a stack of D_l matrices where D_l is the number of filters in the layer. To avoid the exponential growth of resulting stacks in latter layers filters operate on the entire depth of the previous result as $n \times n \times D_{l-1}$ tensors.

A fully connected layer connects every value in the first layer with every value in the second layer. In this example, it is used to provide a set of learnable coefficients between the flattened result of the convolutional layers and the final ground-state energy.

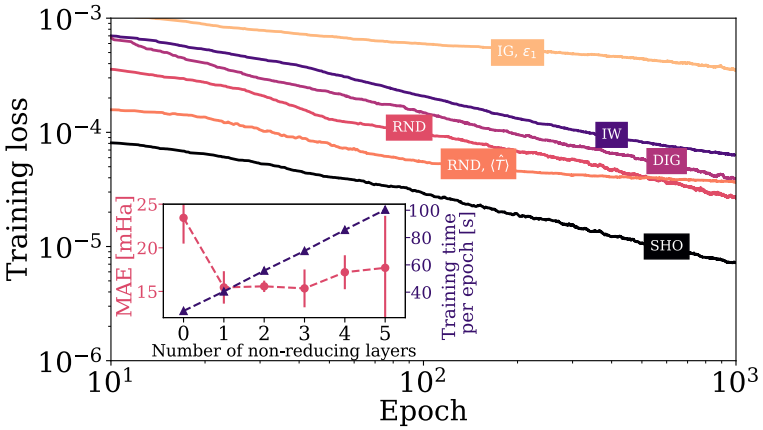


Slika 2. Visualization of the deep neural network used to bypass the numerical calculation of Schrödinger equation. The lower part of the image displays a schematic representation of 'reducing' and 'non-reducing' convolutional layers. Adapted from [4].

[5]Kernel ridge regression, random forests and deep neural networks.
[6]Usually a much smaller matrix than the input.

3.1.2 Machine learning training and results

A deep neural net was trained using supervised learning with potentials and corresponding labels (ground-state energies), that were calculated with a standard finite-difference method for the eigenvalue problem $H\psi = \epsilon\psi$. Neural nets were trained with 200000 randomly generated potentials of a specific class and corresponding ground-state energies.



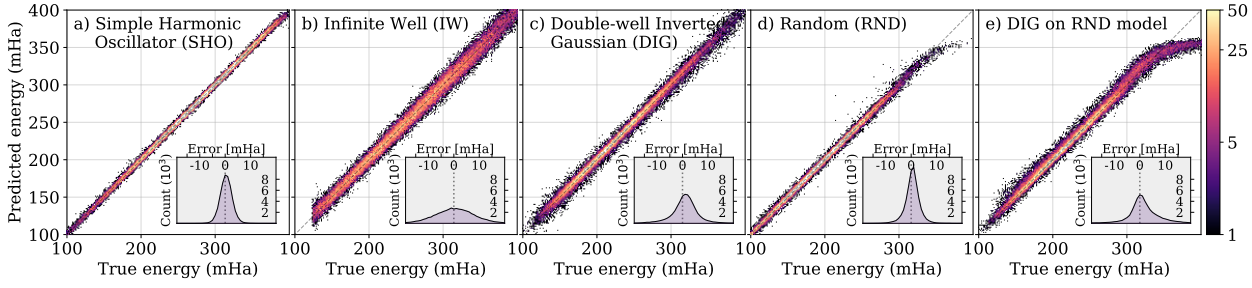
Slika 3. Model error compared to the training data in relation to the number of iterations through the data set. Inlay shows the impact of non-reducing layers on error and training time. Reproduced from [4].

Figure 3 displays the relation between the training length expressed in epochs and the training loss. The loss is defined as the absolute error of the neural net calculated result relative to the label value. Epoch is defined as one complete iteration[7] through the training dataset, in this case, that means 200000 potential-energy pairs.

Inlay in figure 3 displays the impact of non-reducing layers and consequently the number of trainable parameters. Choosing the appropriate number of layers is vital as it has a strong correlation with the training duration and prevents over/under-fitting.

The training was stopped after 1000 epochs as the loss no longer decreased significantly. Loss can be used to estimate the appropriate stopping point, but the model still needs to be cross-validated on independent data to determine the actual accuracy of prediction.

Trained models were evaluated on 50000 test potentials of the same class they were trained on, but not used during the training process. Results are displayed in figure 4 and table 1. Surprisingly, the infinite well potentials fared the worst, despite only being analytically dependent on two parameters, which is most likely due to the sharp discontinuity in their potential. The authors noted that the random potential model has not fully converged yet and provided another 200000 example 1000 epoch training, which significantly improved the median absolute error to 1.49 mHa[8]. Figure 4(d) is displaying the histogram for the further-trained model.



Slika 4. Histograms of true and predicted values for the different potentials. The models were evaluated on the same class of potential they were trained on, with the exception of panel e. Insets display the distribution of error. Reproduced from [4].

[7] One iteration step consists of the neural net evaluating a potential and modification of the neural net coefficients according to the loss.

[8] The Hartree [Ha] is a unit of energy commonly used in atomic physics and computational chemistry. 1 Ha $\sim$ 27.2 eV.

This well trained model for the RND potential shows promise for general use[9], as shown in figure 4(e), with MAE of only 2.94 mHa when solving DIG potentials with model trained on RND potentials, which is comparable with the MAE result of 2.70 mHa for the model that was trained on the same class of potentials.

Potential	MAE [mHa]
SHO	1.51
IW	5.04
DIG	2.70
RND	2.13
equal mixture	5.90
DIG with RND model	2.94

Tabela 1. Median absolute error (MAE) values for deep neural network evaluated on test data. Values from [4].

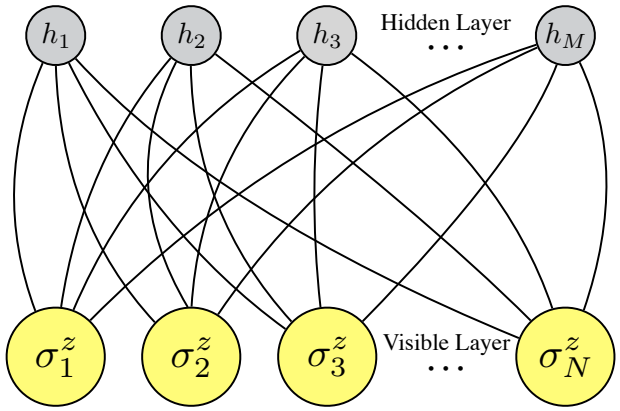
Both test evaluations of the model trained on random potential data show the model failing at high energies, which is a result of not being exposed to many high energy examples during training.

Overall the convolutional neural networks show promise in transferability and efficient solutions for electronic structure problems, with further work required to achieve scaling to variable input sizes and rotational invariance.

3.2 Restricted Boltzmann machine and the quantum many-body problem

This section will review the work in the article [5] related to the application of machine learning on a quantum many-body problem. Machine learning is used in the effort to condense the information from the full wave function to its most essential features.

Authors chose a relatively simple neural net called a restricted Boltzmann machine (RBM), that is schematically presented in figure 5. RBM is an energy-based model with one layer of visible and one layer of hidden variables. This model is considered restricted because there is no intra-layer variable interaction which greatly simplifies the training process.



Slika 5. Restricted Boltzmann machine, a type of artificial neural network with N visible and M hidden variables. Reproduced from [5].

Basic idea is to interpret the wave function $\sum_S \Psi(S)\,|S\rangle$ with $\Psi(\mathcal{S})$ treated as a black box that returns a phase and an amplitude (a complex number) based on the input $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_N)$. Neural-network quantum states (NQS) method aims to replicate this black box with an artificial neural network specified by a set of internal variables $\mathcal{W}$.

The visible layer od RBMs dealing with spin 1/2 quantum systems is represented by N nodes corresponding to the physical spin variables (eg. $\mathcal{S} = (\sigma_1^z, ..., \sigma_N^z)$ where $\sigma_i^z \in \{-1, 1\}$) and a hidden layer of M auxiliary spin variables $(h_1, ..., h_m)$ where $h_i \in \{-1, 1\}$. This description corresponds[6] to a variational expression for coefficients of quantum states

$$\Psi_M(\mathcal{S}; \mathcal{W}) = \sum_{\{h_i\}} e^{\sum_j a_j \sigma_j^z + \sum_1 b_b h_i + \sum_{ij} W_{ij} h_i \sigma_j^z},$$

where the weights $\mathcal{W} = \{a_i, b_j, W_{ij}\}$ fully specify the neural network's response to an input state $\mathcal{S}$.

Hidden variables can be explicitly traced out, as there are no intra-layer interactions, and the

[9] Use outside of the exact potential class it was trained on.

wave function can be stated as

$$\Psi_M(\mathcal{S};\mathcal{W}) = e^{\sum_i a_i \sigma_i^z} \times \prod_{i=1}^{M} F_i(\mathcal{S}), \quad \text{where} \quad F_i(\mathcal{S}) = 2\cosh\left(b_i + \sum_j W_{ij}\sigma_j^z\right).$$

The parameter M plays a role similar to the bond dimension in matrix product states (MPS) method, with an increased number of hidden variables leading to better accuracy at the expense of longer training time. It is useful to express the number of hidden and visible variables as a density $\alpha = M/N$.

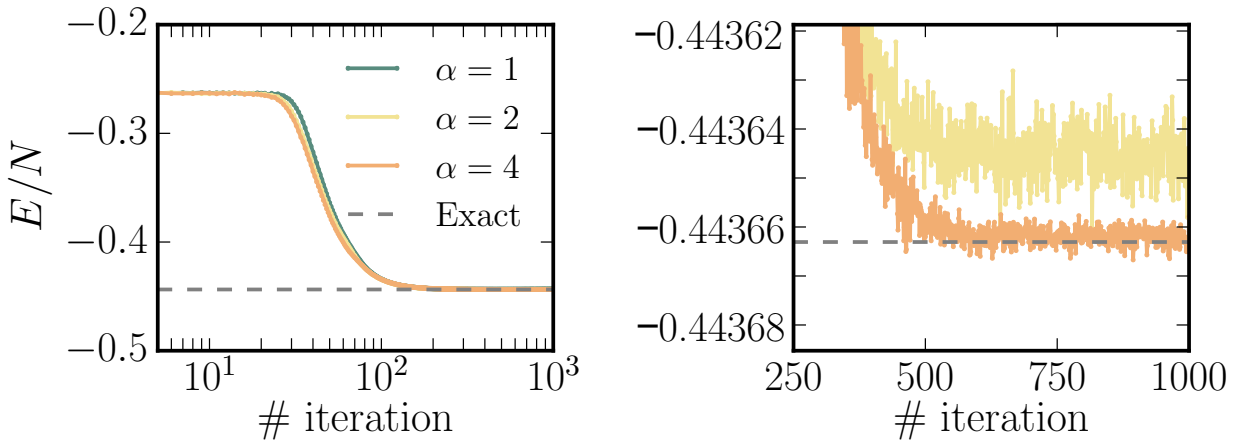
3.2.1 Ground state

Neural network representation of the ground state for a given Hamiltonian $\mathcal{H}$ is optimised with reinforcement learning realised through the minimization of expected energy

$$E(\mathcal{W}) = \frac{\langle \Psi_M | \mathcal{H} | \Psi_M \rangle}{\langle \Psi_M | \Psi_M \rangle}$$

with respect to the weights $\mathcal{W}$. Minimization is achieved through iterative Monte Carlo sampling with a stochastic estimate of energy gradient and $|\Psi_M(\mathcal{H}, \mathcal{W}_k)|^2$ calculated at each iteration k. Improved weight estimation $\mathcal{W}_{k+1}$ is then calculated with a gradient-descent method.

Results of iteration for one of the examples are displayed in figure 6, where it appears that there is no correlation between α and convergence point.



Slika 6. Convergence of energy through iterations of learning algorithm for 1D Heisenberg model with $N = 40$ and periodic boundary conditions. Multiple hidden variable densities α are tested with the right panel displaying a zoom view of convergence. Reproduced from [5].

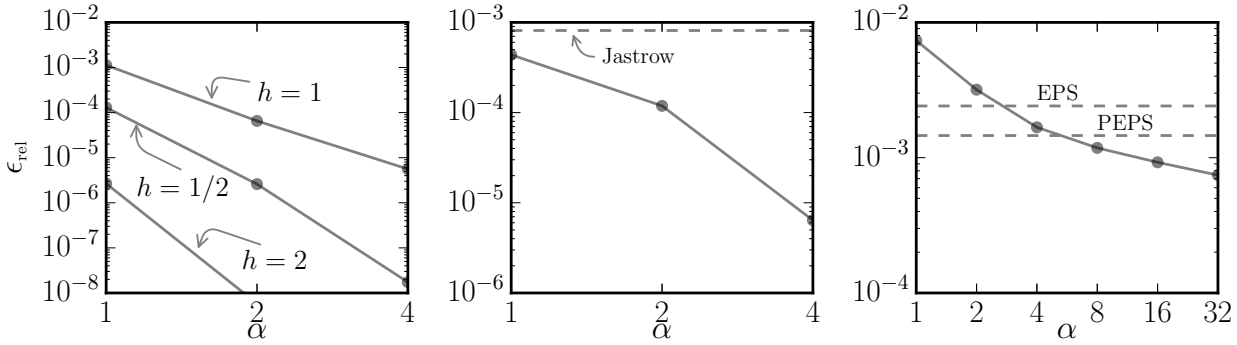
The method is validated on the transverse-field Ising (TFI) and the antiferromagnetic Heisenberg (AFH) models with periodic boundary conditions in 1D and AFH in 2D on a 10×10 lattice.

$$\mathcal{H}_{TFI} = -h\sum_i \sigma_i^z - \sum_{\langle i,j\rangle} \sigma_i^z\sigma_j^z \qquad \mathcal{H}_{AFH} = \sum_{\langle i,j\rangle} \sigma_i^x\sigma_j^x + \sigma_i^y\sigma_j^y + \sigma_i^z\sigma_j^z$$

where i and j sum over the spins with $\langle i,j\rangle$ representing all valid neighbour combinations.

Accuracy of NQS is displayed in figure 7. The left panel displays accuracy for the 1D TFI, where even a modest density of hidden variables achieves a remarkable accuracy (even for the critical point $h = 1$) and seems to show power law behaviour in α. The middle panel shows that a similar accuracy

is also replicated on the more complex 1D AFH model[10]. The right panel shows accuracy for the 2D AFH model where the increased dimension noticeably affected the rate of convergence. The NQS method shows improvement as the hidden variable density α increases, eventually beating other state-of-the-art methods.



Slika 7. Results for the ground state problem displaying relative error compared to the exact result in relation to the density of hidden variables. The left panel displays the 1D TFI, the middle panel the 1D AFH and the right panel the 2D AFH. Reproduced from [5].

3.2.2 Unitary dynamics

A similar approach is also possible for the time dependent problem $i\frac{d}{dt}\Psi = \mathcal{H}\Psi$. The time dynamic is introduced through the time dependent weights $\mathcal{W}(t)$. Weights are determined by minimizing the residual

$$R(t, \dot{\mathcal{W}}(t)) = \text{dist}(\delta_t \Psi(\mathcal{W}(t)), -i\mathcal{H}\Psi(W(t)))$$

with respect to the time derivative of weights $\dot{\mathcal{W}}(t)$ using the time variational Monte Carlo method. $\mathcal{W}(t)$ constructed through the numerical integration of optimised $\dot{\mathcal{W}}(t)$ is then used to express the time dependant coefficients $\Psi_M(\mathcal{S}; \mathcal{W}(t))$.

Dynamics are induced by the instantaneous change of transverse field h for the TFI model and the change of longitudinal coupling J_z for the AFH model as indicated by the labels in the figure. Results are displayed in figure 8 with the TFI on the left and the AFH on the right. High accuracy obtained in these time-dependent solutions shows promise for the NQS method not only as a ground-state method but also for the evolution of complex states.

4. Quantum speed-ups for classical algorithms

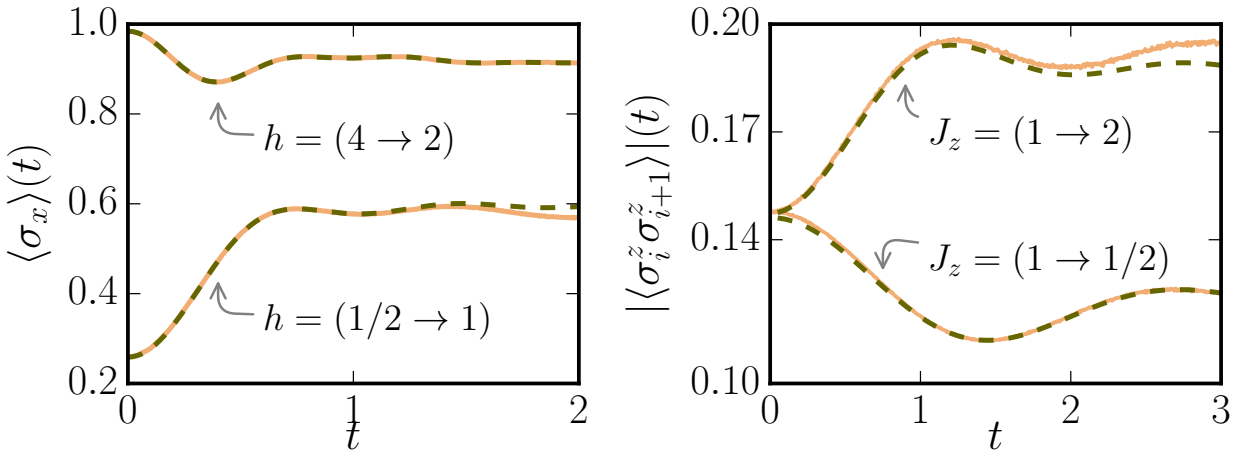
Quantum computers offer substantial improvements to operations that are common in machine learning algorithms. Databases can be searched with time complexity of $\mathcal{O}(\sqrt{N})$ which is quadratically faster compared to classical computers, that do it in $\mathcal{O}(N)$. Fourier transforms over N points, inverting sparse[11] $N \times N$ matrices and finding their eigenvalues is possible in $\mathcal{O}(\text{poly}(\log N))$, which is an exponential speed-up compared to $\mathcal{O}(N \log N)$[12] achieved by the best classical algorithms [7].

Many quantum algorithms rely on the existence of a quantum random access memory[8] (QRAM) as a black box algorithm capable of accessing N pieces of data in $\mathcal{O}(\log_2 N)$ and encoded in $\log_2 N$ qubits.

[10] As an interesting note this NQS representation with $\alpha = 4$ provides a similar accuracy as the DMRG algorithm with the bond dimension of 140.

[11] Some algorithms only require low-rank matrices.

[12] Finding eigenvalues of a tridiagonal matrix.



Slika 8. Results for the time dependant NQS (full line) compared to the numerically accurate methods (dashed line). Left panel shows results for the TFI model after a quench in the transverse field h. The right panel shows results for the AFH model after a quench in the coupling constant J_z. Reproduced from [5].

Improvements for some common algorithms are summarised in table 2 but it is important to note some caveats about their applicability. Quantum algorithms themselves are fast but there are issues with input and output operations remaining relatively slow and dominating the cost of quantum algorithms when dealing with classical data. Time complexity analysis is useful for the asymptotic behaviour of large systems but it is unclear what the actual computation speed on quantum computers might be and what the crossover point[13] is. Neither quantum computers nor QRAM have truly advanced beyond proof of concept stage and still need a lot of work before such benchmarks could be available.

Method	Speedup	QRAM dependency
Online perceptron [9]	$\mathcal{O}(\sqrt{N})$	Optional
Least squares fitting [10]	$\mathcal{O}(\log N)$	Yes
Classical Boltzmann machine [11]	$\mathcal{O}(\sqrt{N})$	Optional
Quantum principal component analysis [12]	$\mathcal{O}(\log N)$	Optional
Quantum support vector machine [13]	$\mathcal{O}(\log N)$	Yes

Tabela 2. Time complexity improvements of some quantum algorithms compared to their classical versions.

5. Conclusion

Quantum mechanics provides ample opportunity to either improve on and speed up machine learning algorithms or use them to ease calculations and discover new concepts. As shown in section 3.1, there is ample opportunity for machine learning to speed up computation for repetitive or complex multi-particle electronic structure problems in physics, chemistry and material science. Machine learning can also improve on stochastic methods like Monte Carlo and helps in finding more efficient wave function representation allowing for larger systems and better accuracy, as shown in section 3.2

Machine learning is a very active field and it is important for physicists to keep up, find applications and help improve algorithms which will certainly further our understanding of nature.

[13]Size of data where the algorithmic efficiency overcomes the difference in operation speed.

6. Acknowledgement

I would like to thank my mentor doc. dr. Martin Horvat for his help and support.

LITERATURA

[1] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.

[2] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015.

[3] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre G.R. Day, Clint Richardson, Charles K. Fisher, and David J. Schwab. A high-bias, low-variance introduction to machine learning for physicists. *Physics Reports*, 2019.

[4] Kyle Mills, Michael Spanner, and Isaac Tamblyn. Deep learning and the schrödinger equation. *Phys. Rev. A*, 96:042113, Oct 2017.

[5] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

[6] Zhih-Ahn Jia, Biao Yi, Rui Zhai, Yu-Chun Wu, Guang-Can Guo, and Guo-Ping Guo. Quantum neural network states: A brief review of methods and applications. *Advanced Quantum Technologies*, 0(0):1800077, 03 2019.

[7] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549:195 EP –, Sep 2017.

[8] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Phys. Rev. Lett.*, 100:160501, Apr 2008.

[9] Nathan Wiebe, Ashish Kapoor, and Krysta M Svore. Quantum perceptron models. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 4006–4014, USA, 2016. Curran Associates Inc.

[10] Nathan Wiebe, Daniel Braun, and Seth Lloyd. Quantum algorithm for data fitting. *Phys. Rev. Lett.*, 109:050505, Aug 2012.

[11] Nathan Wiebe, Ashish Kapoor, and Krysta M. Svore. Quantum deep learning. *Quantum Info. Comput.*, 16(7-8):541–587, May 2016.

[12] Seth Lloyd, M Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10, 07 2013.

[13] Seth Lloyd, Silvano Garnerone, and Paolo Zanardi. Quantum algorithms for topological and geometric analysis of big data. *Nature Communications*, 08 2014.