Machine learning and the Schrödinger equation

A V Pavlov¹, J A Serdyuk² and A B Ustinov¹

¹ Peter the Great St. Petersburg Polytechnic University, Polytechnicheskaya 29, 195251
St. Petersburg Russia
² Lomonosov Moscow State University, 2nd Education Building, Faculty CMC, GSP-1,
Leninskie Gory, 119991 Moscow Russia

E-mail: a.pavlov@physics.spbstu.ru

Abstract. In this research several methods of machine learning (ML), such as decision trees
and linear regression were used to predict ground-state energy (GSE) of an electron in a potential
well. Analysis has been done for various types of the potentials: ones with exact solution as
well as ones with only numerical one. It was shown that some methods can map the analytical
solution and predict GSE with chemical accuracy along with ability to find GSE for systems
which does not have analytical solutions. To increase the accuracy and performance of the ML
algorithms we proposed several methods of data preprocessing.

1. Introduction
For almost a hundred years solution of quantum-mechanical problems remains in focus of
challenging research [1,2]. With development of quantum mechanics importance of approximate
methods for calculating parameters of physical systems have increased. Many methods, based
either on approximations of underlying interactions (tight binding model, density functional
theory, k-p perturbation theory, etc) [3, 4] or on phenomenological fits of experimental results
(force fields) [5, 6], have emerged. These methods allow determining of electronic structure
of molecules, solids and heterostructures. Creating more efficient and/or robust methods for
solving quantum-mechanical problems is of fundamental importance for the further development
of physics, chemistry and material science.

Unlike physics, machine learning (ML) is a relatively new concept and a field of computer
science. However, despite the novelty of ML’s basic principles, it really have been used by
scientists for a long time. When they fit parameters of certain model to experimental data they
take advantage of linear regression, which is one of cornerstone techniques of machine learning.
But modern ML incorporates many methods that are nowadays used in various industrial and
scientific areas.

The main branches of machine learning are supervised, unsupervised, semi-supervised and
reinforcement learning. The supervised learning is used to predict a specific value based on
a given set of object’s measurable properties, called features. This branch of learning is used in
ranking systems of search engines [7], translation systems [8], speech recognition [9] and
many other areas. The unsupervised learning is used to extract patterns and find dependencies
in data. For example, collaborative filtering [10] is used in recommendation systems and topic
modelling [11,12] is an approach to create clusters of terms and documents. The semi-supervised
learning is similar to the supervised learning, but in this case only a relatively small part of the
training data has the target labels. It can be used to train an algorithm to predict a document class based on training set of documents written in another language [13]. The reinforcement learning consists of sparse target labels, rewards, affect algorithm’s predictions and decisions. Self-driving cars [14] are popular example of this approach.

In science ML algorithms find applications for example in biology and neural science [15–17]. In physics, machine learning has been used for materials discovery [18, 19] to target promising material candidates. Unsupervised machine learning was used to determine the phase diagram of the many-body atomic configuration [20]. Significant attention in scientific community is focused on application of neural networks (NN). One of the reasons for that is ability of significantly large artificial NN to map any continuous function with infinite accuracy [21]. Need for computational power in \textit{ab initio} methods can be bypassed by replacing exact algorithm with NN mapping [22,23].

In our work, we present results of supervised ML on prediction of electron ground state energy (GSE) in 2D potential well. Conventional methods of machine learning such as several types of decision trees and linear regression algorithms in combination with several preprocessing methods give outstanding performance - despite simplicity, in some cases they were able to outperform much more complicated convolutional NN [24].

2. Methodology
As already mentioned, supervised ML is based on training of prediction model on a set of data containing both inputs and corresponding desired outputs. In other words, to train the model to predict GSE of an electron in a potential well, one should create a training set consisting of potential profiles and GSE values for them. Classes of potential profiles and their parameters were chosen to meet the following requirements: (i) the generated potentials should produce GSEs within physically relevant energy range (in our case potentials were selected that have GSE in the interval 0-400 mHa, note: 1 Ha ≈ 27.2 eV), (ii) diversity inside one class of the potentials, in other words, potentials should have different parameters and GSE (see Ref. [24] for details). Additionally, the presence of an exact solution for some of potentials is desired. Following classes were chosen, that satisfy mentioned requirements: infinite well (IW), elliptic infinite well (EIW), honeycomb infinite well (HIW), harmonic oscillator (HO), double-well inverted Gaussian (DWIG) and random potential. Each potential $V(x, y)$ can be represented as a matrix of floating-point numbers, image representations for different classes of potentials are shown on figure 1.

Figure 1. Graphical representation of the $V(x, y)$ profiles for different potential classes. Each picture has real world size 40 a.u. by 40 a.u., with sampling step 256 points along each side. False colors show the value of potential in given point (hotter the color, deeper the potential). The figure shows potentials for IW (a), EIW (b), HIW (c), HO (d), DWIG (e) and random potential (f).

To find GSE of the potentials, standard finite-difference method (see e.g. [25]) was used to find first eigenvalues of time-independent Schrödinger equation:

$$\hat{H}\psi = \left(-\frac{\hbar^2}{2m_e} \nabla^2 + \hat{V}(x, y)\right)\psi = \varepsilon\psi$$

(1)
for each potential $V(x, y)$. Atomic units were used during calculations, such that reduced plank constant $\hbar$ and mass of electron $m_e$ equal to 1.

Because comparison with Ref. [24] provided a good benchmark for our calculations, potentials were calculated on a square domain from -20 to 20 a.u. Since, for HO and IW potentials there are analytical solutions of energy eigenvalues problem, accuracy of finite-difference scheme can be controlled via comparison between analytically and numerically computed values. Using this approach and assuming that mean absolute error (MAE) $\approx 0.1$ mHa is sufficient for research purposes, $256 \times 256$ grid was chosen for applying to finite-difference method.

For machine learning computations Scikit-learn [26] and XGBoost [27] software packages were used. In our tests two algorithms that are based on decision trees showed the best performance, so Extra Trees (ET) [28] and XGBoosting (XGB) were chosen for the study. Decision trees predict the target label based on a sequence of conditions on $X$ values, e.g. if $x_1 < t_1$ and $x_2 > t_2$ then $y = y^*$. Ensemble of small decision trees (weak learners), combined into a single strong learner, provides good accuracy in many real-world problems [29–32]. One of the most popular example of such ensemble is random forest, where each tree is trained based on a random subset of training objects and a random subset of their features. Extra trees approach is similar to random forest, but values of the thresholds $t_i$ are picked at random also. Another way to build an ensemble of trees is used in XGB and is called gradient tree boosting. Each $m$-th tree is created to correct errors of the previous $m-1$ trees. This is done using the gradient, as the name implies.

3. Results and discussion

The calculations were carried out as follows: several datasets were generated in order to analyze behavior of the selected methods for different combinations of training potentials and preprocessing procedures. Data sets were split into two parts: training dataset, that was used to train the model, and test dataset, that have not been accessible to the model during the training process and was used to evaluate MAE, related to method and preprocessing procedure. Following datasets were generated: (i) "Random dataset" (RDS) which included only random potentials. The training part consisted of datasets including 500, 1000, 1500, 2000 and 4000 sample potentials. The test dataset included 2000 of potentials and corresponding GSEs. (ii) "Data + Random dataset" (DRDS) which included an equal number of IW, HO, DWIG and random potentials. Training and testing parts were organized in a similar to RDS way. (iii) "Harmonic dataset" (HDS) and "Infinite well dataset" (IWDS), to show peculiar behavior of some methods on tasks which have analytical solution (see Sec. 3.3). The first consisted of 1500 HO potentials in training and testing parts. The second was created from 1500 samples of IW, HIW and EIW potentials each.

3.1. Potential centering

Let us now discuss results of applying ET and XGB to RDS and DRDS datasets. Testing results for that are shown on figure 2 (c). As one can notice MAE of both methods are relatively close on both datasets. MAE is bigger for DRDS dataset, because it consists of four types of potentials and methods have less examples of each class to learn. However, with increase in size of the training dataset MAE reduces slowly for DRDS dataset and remains relatively stable for RDS. That means models reached their saturation of learning capabilities. However, the achieved accuracy is by approximately order of magnitude higher than the desired chemical accuracy (1 kcal/mol or 1.6 mHa). One way to drop MAE is to increase the size of the training set, but, as is obvious from dependencies for RDS, it slightly impacts the results.

Another way to increase the accuracy, which is widely used in ML, is to prepare data in order to help machine to learn faster. This procedure is called preprocessing. One of the possible procedures is centering. The idea of it is pretty simple: one can take the potential, find the
minima, maxima or any other specific point, and shift potential in a way to place feature in the center of the potential area. (see for example figures 2 (a) and (b) where results of centering procedure are shown for random and DWIG potentials respectively). Translations do not change the GSE, but, since the potentials become more uniform, such procedure simplify the process of parameter optimization, that algorithm performs in order to generate the solution.

After preliminary tests, it was found that centering of the minima of potentials significantly improved accuracy. Results for both methods when using centered potentials are shown on figure 2 (d). As one can notice MAE drops significantly compared to the results of work with non-centered potentials. Accuracy, in that case, slowly increases with the size of training set for both ET and XGB. Additionally, with increased accuracy it becomes clear that XGB is better on RDS, while ET is better with DRDS (MAEs on 4000 potentials training dataset equal to 4.42 mHa and 3.61 mHa respectively). The lowest MAE that was achieved by ET on DRDS with 8000 samples training dataset is equal to 2.45 mHa. Extrapolation of the results shows that chemical accuracy should be achieved at 15000 – 20000 training dataset sizes. Convolutional NN achieves comparable accuracy on similar problems without preprocessing on 10 times larger datasets [24].

3.2. Heterogeneity of the dataset
Another factor that can impact the accuracy of predictions made by ML algorithms is significant difference of dataset parts in one or several parameters. To demonstrate this phenomenon, additional data for DWIG was generated. By modulating depth and width of each Gaussian, GSE distribution in a range from 200 to 1200 mHa was achieved. Then we replaced DWIG data of already existing DRDS dataset by new Gaussians with "out of the scope" GSE and conducted the same tests.

The results of computation are presented on figure 3. All calculations have been done with centering preprocessing procedure. It may be noted that the dependencies of MAE on size of the training set for modified DRDS dataset have the same pattern as for the unmodified one. MAE for ET in normal dataset equals to 3.61 mHa compared to 8.35 mHa in modified one. For
XGB, MAE are 5.11 mHa and 10.19 mHa respectively.

These observations are connected to the fact that prediction accuracy of ML methods strongly depend on homogeneity of input data. If at least some of them are "out of the scope" for general trend, application of method leads to reduction in the overall prediction accuracy. Trying to nullify influence of heterogeneity requires either a deep analysis of the problem or use of brute force (increase the size of the training dataset etc). Using the latter, however, does not guarantee the desired accuracy and, in some cases, is difficult or even impossible to implement.

3.3. Dimension reduction

Finally, let’s discuss another type of preprocessing known in ML as feature extraction. Previously discussed methods had the goal of increasing the prediction accuracy. Higher prediction accuracy is always desired, but it comes at expense of time that is needed to train an algorithm. On one hand training time is rather one-time investment, if the parameters of the system remain constant, one can use the same apparatus multiple times, but, on the other hand, training of the complicated model on large enough dataset can take from several hours to several days, depending on algorithm and available computing power. Proper preprocessing can significantly reduce this time minimal or no loss in accuracy. This becomes especially important when properties of a system change in time and online learning is needed.

One of the ways to overcome the problem with training time is to reduce the number of features of input data. For example, standard potential for this work consists of 65536 features.
The value of a potential in every point of a 256×256 grid) which ML algorithms should map to one GSE. If we reduce the potential matrix size to 32×32 the number of features will drop to 1024. Examples of potentials before and after scaling are depicted on figures 4 (a) and (b).

In random potential, which doesn’t have an exact solution, features of an object are not correlated with each other, so scaling can significantly affect the accuracy, but HO and IW have exact solutions, i.e. there is a dependence between values in each node of the grid and resulting GSE. The results of tests for the latter case are shown on figures 4 (c) and (d). They show that MAE generally doesn’t grow with decreasing number of features and even diminishes for XGB. This behavior is very different from the behavior of finite difference methods, error in which depends strongly on the grid size. However, ML methods do not solve differential equations, but find dependencies between some potential properties and GSE value. For the HO, there is an exact solution for energy spectrum, which can be represented as follows: energy only depends on the curvature of the potential in the two primary axes. For IW, the energy is determined by the width of the potential in two axes, in the case of EIW and HIW dependencies are more complex, but, in general, the energy is determined by the ”area” of the potential. And if a ML method guesses these patterns, it will be able to predict the GSE independently from grid spacing. Thus, dimension reduction works great on decision trees, which are able to fit the pattern.

But let’s see how it works on much simpler ML methods – linear regressions. Linear regressions work in a following way, let $X$ be a matrix of independent variables (features), and $Y$ be the vector of the target labels. The goal of the linear regression approach is to find such a vector $w$ that $Xw = Y$. In many cases there is no exact solution to this equation, and it makes us to choose the solution $w$ based on a set of restrictions (regularization) and a measure of prediction quality (e.g. mean absolute error, mean squared error, etc). This leads to various linear approaches, such as least squares method (linear regression in Scikit-learn) and linear Kernel Ridge regression (Kernel Ridge with the linear kernel in Scikit-learn).

![Figure 5. Bar plot for MAE of Kernel Ridge and linear squares method for different problem dimensions. Horizontal axis shows the potential grid. Note: that the log-scale of the vertical axis is used.](image)

Prediction results made by linear regression and Kernel Ridge regression for IW potentials are shown on the figure 5. As one can notice scaling drastically drops the prediction accuracy of both methods. Linear regressions have limited predictive potential, and a small amount of features limits the ability to fine-tune all parameters. However, for original matrices with the size of 256 × 256 both methods gives MAE equal to 0.5 mHa, which makes them suitable for dealing with IW potentials.

4. Conclusions
To summarize, in this work we present results of application of ML to solution of 2D stationary Schrödinger equation. We were able to achieve MAE close to chemical accuracy. Several methods of ML were used to find mapping between potential profile and it’s GSE. We show that MAE for each method strongly depends on preprocessing method. Latter is important, because preprocessing and analysis of data can help to choose the right method and training procedure. Eventually, ML is not black box that can solve any problem, but another way to map (fit) model to the data. Improper use of these methods can lead to mistakes and erroneous theories that are just results of optimization of mathematical models.
References

[1] Sanin A L and Smirnovsky A A 2018 Materials Physics and Mechanics 39 61–67
[2] Sanin A L and Smirnovsky A A 2007 Phys. Lett. A 372 21–27
[3] Vrubel I I, Polozkov R G and Ivanov V K 2016 Eur. Phys. J. D 70 167
[4] Ipatov A N, Ivanov V K and Polozkov R G 2014 Eur. Phys. J. D 68 249
[5] Verkhovtsev A, Polozkov R, Ivanov V, Korol A and Solov'yov A 2012 J. Phys. B 45 215101
[6] Polozkov R, Ivanov V, Verkhovtsev A, Korol A and Solov'yov A 2013 J. Phys.: Conf. Series 438 012009
[7] Dorogush A V, Ershov V and Gulin A 2018 arXiv:1810.11363
[8] Wu Y et al. 2016 arXiv:1609.08144
[9] Rabiner L R, Juang B H and Rudge J C 1993 Fundamentals of speech recognition vol 14 (PTR Prentice Hall Englewood Cliffs)
[10] Linden G, Smith B and York J 2003 IEEE Internet computing 76–80
[11] Deerwester S, Dumais S T, Furnas G W, Landauer T K and Harshman R 1990 Journal of the American society for information science 41 391–407
[12] Wang Y, Zhao X, Sun Z, Yan H, Wang L, Jin Z, Wang L, Gao Y, Law C and Zeng J 2015 ACM Transactions on Intelligent Systems and Technology (TIST) 6 47
[13] Shi L, Mihalcea R and Tian M 2010 Proc. of the 2010 Conf. on Empirical Methods in Natural Language Processing (Association for Computational Linguistics) pp 1057–1067
[14] Bojarski M, Del Testa D, Dworakowski D, Firner B, Flepp B, Goyal P, Jackel L D, Monfort M, Muller U, Zhang J et al. 2016 arXiv:1604.07316
[15] Kovalev M S, Ivolkina A A, Samsonova M G and Nuzhdin S V 2018 Frontiers in plant science 9
[16] Plekhanova E, Nuzhdin S V, Utkin L V and Samsonova M G 2019 Evolutionary applications 12 18–28
[17] Utkin L V 2019 Neurocomputing 331 18–32
[18] Hautier G, Fischer C C, Jain A, Mueller T and Ceder G 2010 Chem. Mater. 22 3762–3767
[19] Saad Y, Gao D, Ngo T, Bobbitt S, Chelikowsky J R and Andreoni W 2012 Phys. Rev. B 85 104104
[20] Wang L 2016 Phys. Rev. B 94 195105
[21] Cybenko G 1989 Mathematics of control, signals and systems 2 303–314
[22] Snyder J C, Rupp M, Hansen K, Müller K R and Burke K 2012 Phys. Rev. Lett. 108 253002
[23] Yao K and Parkhill J 2016 J. Chem. Theory Comput. 12 1139–1147
[24] Mills K, Spanner M and Tamllyn I 2017 Phys. Rev. A 96 042113
[25] Smith G D 1985 Numerical solution of partial differential equations: finite difference methods (Oxford university press)
[26] Pedregosa F et al. 2011 Journal of Machine Learning Research 12 2825–2830
[27] Chen T and Guestrin C 2016 Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (New York, NY, USA: ACM) pp 785–794
[28] Geurts P, Ernst D and Wehenkel L 2006 Machine learning 63 3–42
[29] Shenoy G G, Wagle M A and Shaikh A 2017 arXiv:1703.02915
[30] Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q and Liu T Y 2017 Advances in Neural Information Processing Systems pp 3146–3154
[31] Bahnsen A C, Aouada D and Ottersten B 2015 Expert Systems with Applications 42 6609–6619
[32] Gupta A, Nagarajan V and Ravi R 2017 Math. Oper. Res. 42 876–896