Deep learning and the Schrödinger equation

KARAN SHAH, Technical University of Munich

In this paper, I report the work of [Mills et al. 2007] where they trained a certain Deep Learning Network to predict the ground-state energy of an electron in four different classes of electrostatic potentials confined in two dimensions. On randomly generated potentials, for which there is no analytical form for computing either the potential or the ground-state energy, the model was able to predict the ground-state energy to within chemical accuracy, with a median absolute error (MEA) of 1.49 mHa. I also briefly talk about their investigation of the performance of the model in predicting other fundamental quantities such as the kinetic energy and the first excited-state energy.

Additional Key Words and Phrases: Schrödinger Equation, Ground State Energy, Excited State Energy, Convolutional Neural Networks, Electrostatic Potential

1 ERWIN SCHRÖDINGER

Erwin Schrödinger was a Nobel Prize-winning Austrian physicist who developed a number of fundamental results in Quantum theory: His equation provides an analytical way to calculate the wave function of a quantum system and describes how it changes dynamically with time.

2 THE SCHRÖDINGER EQUATION

In Quantum Mechanics, wave function is defined as a variable quantity that describes the wave characteristics of a particle. The value of the wave function gives the likelihood of the particle to be at that point in space and time.

Conceptually, the Schrödinger equation is the quantum counterpart of what we know as Newton's Second Law of Motion in the realm of classical mechanics. Given a set of known initial conditions, Newton's second law makes a deterministic mathematical prediction as to what path any given physical system will follow over time. Equivalently, the Schrödinger equation gives the evolution of a wave function over time.

The most general form of the Schrödinger equation is the time-dependent variation, which gives a description of a quantum system evolving over time:

$$i\hbar\frac{d}{dt}|\Psi(t)\rangle = \hat{H}|\Psi(t)\rangle$$

In Quantum Mechanics, the Hamiltonian of a system is an operator corresponding to the total energy of that system, including both Kinetic Energy and Potential Energy.

Interestingly, the time-dependent Schrödinger equation described above predicts that wave functions can form standing waves, called stationary states. These stationary states can be described by a simpler form of the Schrödinger equation:

$$\hat{\mathrm{H}}\,|\Psi\rangle = E|\Psi\rangle$$

Analytical solutions of this equation are known for very few sample model Hamiltonians including the simple harmonic oscillator, the particle in a box, the dihydrogen cation, and the hydrogen atom. It is typically not possible to solve the Schrödinger equation exactly for situations of physical interest. Accordingly, approximate solutions are obtained using techniques like variational methods and

Fig. 1. Erwin Schrödinger, Source: Wikipedia Commons

This report is a part of the lecture, Master-Seminar - Deep Learning in Physics, Informatics, Technical University of Munich.

The original work was introduced by K. Mills, M. Spanner and I. Tamblyn in 2007 [Mills et al. 2007].

WKB approximation. It is also common to treat a problem of interest as a small modification to a problem that can be solved exactly, a method known as perturbation theory. [Press et al. 1990]

Even the helium atom – which contains just two electrons – has defied all attempts at a fully analytical treatment till date.

3 MACHINE LEARNING AND PHYSICS

The term Machine Learning is new; the approach is not. Even before the dawn of modern computing, we have been using various Machine Learning techniques (Pattern Matching, Grouping, Predicting, fitting parameters within a model etc.) to solve complex models and problems. In the field of Quantum mechanics, these approaches have proven to be quite powerful, yielding models trained for specific atomic species or based upon hand-selected geometric features. However, as is the case with most regular machine learning tasks, feature selection is a significant limitation of such approaches, as the outcomes depend upon the choice of input representation, and even more so, on the inclusion of all relevant features.

To overcome this limitation, here, the authors propose a fundamentally different approach inspired by the successful application of deep convolutional neural networks to problems in computer vision and computational games. Rather than trying to seek an appropriate input representation to capture all the relevant physical attributes of a quantum system, they train a highly flexible neural network on a considerable large training dataset. The intuition is that after sufficient training, the network will learn not only the features (in weight space) but also the mapping required to arrive at those results. This approach does not depend on the appropriate selection of input representations and features. This technique is commonly referred to as ''Featureless Learning".

Developing a deep learning model involves both the design of the network architecture and the acquisition of training data. The latter is one of the most crucial aspects of training a machine learning model, as it determines the transferability of the resulting model.

In previous work, a NN was shown to interpolate the mapping of position to wavefunction for a specific electrostatic potential, but the fit was not transferable, a limitation also present in other applications of NNs to PDEs.

4 NUMERICAL SOLVER AND DATASET GENERATION

In the paper, the authors investigate four different classes of potentials in two dimensions: Simple Harmonic Oscillators (SHO), Infinite Wells (IW), Double-Well Inverted Gaussians (DIG), and Random Potentials (RND). Each of these are essentially a grayscale image: a 2D grid of floating-point numbers. The potentials are defined on a grid from x, y = 20 to 20 a.u. on a 256 × 256 grid. They used a standard finite-difference method to solve the eigenvalue problem:

$$\hat{H}\psi \equiv (\hat{T} + \hat{V})\psi = \varepsilon\psi$$

for each potential V. The potentials were generated with a dynamic range and length scale suitable to produce ground-state energies within a physically relevant range

4.1 Simple Harmonic Oscillator (SHO)

The harmonic oscillator that the authors refer to here is the quantum-mechanical analog of the classical harmonic oscillator. Because an arbitrary smooth potential can usually be approximated as a harmonic potential at the vicinity of a stable equilibrium point, it is one of the most important model systems in quantum mechanics. Furthermore, it is one of the few quantum-mechanical systems for which an exact, analytical solution is known.

The Simple Harmonic Oscillator (SHO) potentials were generated with the scalar function:

$$V(x, y) = \frac{1}{2}\left(k_x(x - c_x)^2 + k_y(y - c_y)^2\right)$$

where kx, ky, cx, and cy were randomly generated.

Theoretically, the SHO forms the simplest case for a purpose-built CNN as there is a closed form analytical solution to the problem dependent on two simple parameters (kx and ky). These two parameters uniquely define the ground-state energy of a single electron.

4.2 Infinite Wells (IW)

The Infinite Well (IW) potentials were generated with the scalar function:

$$V(x, y) = \begin{cases} 0 & \frac{1}{2}(2c_x - L_x) < x \leq \frac{1}{2}(2c_x + L_x) \text{ and} \\ & \frac{1}{2}(2c_y - L_y) < y \leq \frac{1}{2}(2c_y + L_y) \\ 20 & \text{otherwise} \end{cases}$$

where 20.0 was used as "numerical infinity". Just like the SHO, the ground state energy here depends analytically only on the width of the well in the two dimensions.

It is expected that both the SHO and the IW will serve as very trivial cases for even a modest network architecture to be able to accurately figure out the mapping between the potentials and the different energies.

4.3 Double-Well Inverted Gaussians (DIG)

The double-well inverted Gaussian (DIG) potentials were generated with the scalar function:

$$V(x, y) = -A_1 \exp\left[-\left(\frac{x - c_{x_1}}{k_{x_1}}\right)^2 - \left(\frac{y - c_{y_1}}{k_{y_1}}\right)^2\right] - A_2 \exp\left[-\left(\frac{x - c_{x_2}}{k_{x_2}}\right)^2 - \left(\frac{y - c_{y_2}}{k_{y_2}}\right)^2\right]$$

where the parameters are randomly sampled from a uniform distribution. The DIG dataset is reasonably more complex than the SHO and IW in two respects: First, the potential depends on significantly more parameters; Second, there is no known analytical solution for a single electron in a potential well of this nature.

4.4 Random Potentials (RND)

The random potentials were generated through a lengthy process motivated by three requirements: the potentials must (a) be random, (b) be smooth, and (c) go to a maximum of 20.0 at the boundary.

The random dataset formed the most generalized case as there was no closed-form equation to represent the potentials. The dataset

generation process is explained in some more detail in the original paper.

5 CNN ARCHITECTURE

With the ultimate aim of better generalization and reproducibility, the architecture of the neural network was designed to be as generic as possible. They used a simple deep neural network architecture, as shown in the figure above, composed of a number of repeating units of convolutional layers. The sizes of all the layers were chosen for a balance of execution speed and result accuracy.

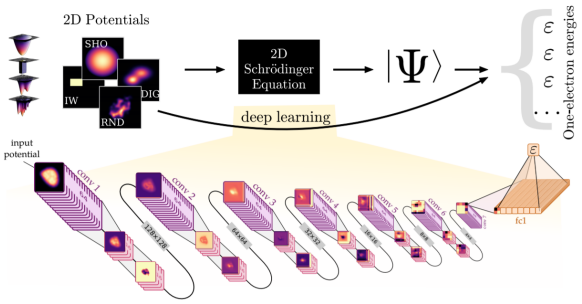

Fig. CNN Architecture

As shown in the figure, the convolutional layers are of two different types: "Reducing" and "Non-Reducing". The 7 Reducing layers operate with filter (kernel) sizes of 3 × 3 pixels. Each reducing layer operates with 64 filters and a stride of 2×2, effectively reducing the image resolution by half at each step. Between each pair of these reducing convolutional layers, there are "Non-Reducing" layers which operate with 16 filters of size 4 × 4. These filters have unit stride, and therefore preserve the resolution of the image. The purpose of these layers is to add additional trainable parameters to the network.

All convolutional layers have ReLU activation. The final convolutional layer is fed into a fully connected layer of width 1024, also with ReLU activation. This layer feeds into a final fully-connected layer with a single output. This output is then used to compute the loss.

It is important to note here that as the simple harmonic oscillator potentials have an analytic solution, they were used as reference for validating the accuracy of the numerical solver.

6 IMPLEMENTATION AND RESULTS

For training, they used the AdaDelta optimization scheme with a learning rate of 0.001 to minimize the loss function, continuously monitoring its value as training proceeded. Interestingly, it was observed that after 1000 epochs, the loss no longer decreased significantly. They devised a custom TensorFlow implementation in order to make use of 4 graphical processing units (GPUs) in parallel. All training datasets consisted of 200,000 training examples and training was run for 1000 epochs. All reported errors are based on evaluating the trained model on validation datasets consisting of 50,000 potentials that were not accessible to the network during the training process.

Table 1. Key Observations

Potential	MAE
SHO	1.51mHa
IW	5.04mHa
DIG	2.70mHa
RND	1.49mHa
DIG on RND	2.94mHa

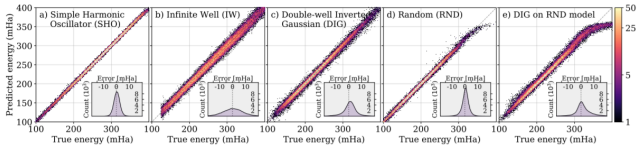

Fig. Histograms of the true vs. predicted energies for each example in the test set indicate the performance of the various models

The above figure shows the Histograms of the true vs. predicted energies for each example in the test set indicating the performance of each model. The insets show the distribution of error away from the diagonal line representing perfect predictions.

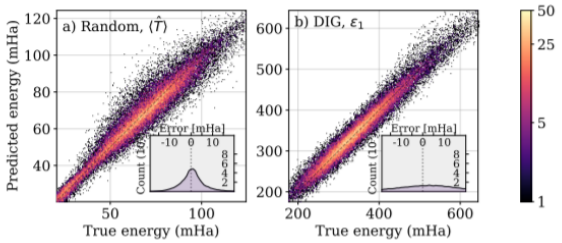

Fig. Histograms of the true vs. predicted energies for the model trained on the (a) kinetic energy, and (b) excited-state energy of the double-well inverted Gaussian.

The above figure shows the Histograms of the true vs. predicted energies for the model trained on the (a) kinetic energy, and (b) excited-state energy of the double-well inverted Gaussian.

7 KEY OBSERVATIONS

The SHO, being one of the simplest potentials, performed extremely well. The IW potentials performed notably poorer than the SHO potentials, despite their similarity in being analytically dependent upon two simple parameters. This is likely due to the sharp discontinuity associated with the infinite well potentials, combined with the sparsity of information present in the binary valued potentials. The DIG and RND potentials performed moderately well. In case of RND, since the loss did not completely converge after 1000 epochs, additional 200,000 training samples were provided to the network and it was allowed to train for 1000 more epochs. 1.49 mHa was the

MEA after additional training To investigate the transferability of the learnt model to another class of potentials, the model trained on RND was able to predict the ground-state energy of the DIG potentials within a reasonable MEA.

As indicated by the graphs, the model fails at very high energies. However, this is an expected result given that the training set did not have many examples in this particular energy band. The reasonably good performance on DIG on RND dataset is not entirely surprising; One possible explanation is that during the generation of the random potentials, there's an element of Gaussian blurring. However, this moderate performance is still a testament to the generalizability of CNNs Since the ground-state and first excited-state are both eigenvalues of the Hamiltonian, the authors were also able to demonstrate the training of a seperate model on the expected value of the kinetic energy. The trained model predicted the kinetic energy value with a MAE of 2.98 mHa. While the spread of testing examples in figure 4 above suggests the model performed more poorly, the absolute error was still reasonably small. The model was versatile enough to be able to predict the ground-state, first excited-state and the kinetic energies separately. This, again, serves as a testament to the power of CNNs.

8 CONCLUSION

The authors were aware of there being other supervised/unsupervised Machine Learning approaches to solve the same problem, and were glad to report that their CNN based approach performed reasonably better than other existing known approaches. A detailed comparison with Kernel Ridge Regression and Random Forests can be found in the original paper. As the number of electrons in a system increases, the computational complexity of both the analytical methods as well as other ML models grow polynomially. CNNs exhibit no such scaling. Although the focus here was on a particular type of problem, namely an electron in a confining 2D potential well, the concepts here are directly applicable to many problems in physics and engineering. This experiment can also be considered as a successful demonstration of the ability of NNs to learn how to approximate the solutions to PDEs. A generalizable, scalable and transferable approach to solving PDEs would naturally impact many other verticals of theoretical physics, chemistry and mathematics.

Thus, in conclusion, CNNs are one of the more promising candidates for application to electronic structure calculations as they are inherently designed for data which has a spatial encoding of information.

9 FUTURE SCOPE

One notable limitation of the author's approach is that the efficient training and evaluation of the CNN requires uniformity in the input size. Future work can focus on an approach that would allow transferability to variable input sizes. This approach to training the CNN is not absolutely rotationally invariant. Recent proposals to modify the network architecture itself to make it rotationally invariant, without using an additional augmentation, are promising [Dieleman et al. 2016].[Worrall et al. 2016] This exact approach to solving PDEs rapidly and accurately can be used in other areas of physics and engineering. To scale this implementation to support multi-particle systems. As the number of electrons in a system increases, the computational complexity of the analytical methods as well as other ML models grow polynomially. CNNs exhibit no such scaling. (Eg. Helium Atom)

ACKNOWLEDGMENTS

The authors would like to acknowledge fruitful discussions with P. Bunker, P. Darancet, D. Klug, and D. Prendergast. K.M. and I.T. acknowledge funding from NSERC and SOSCIP. Compute resources were provided by SOSCIP, Compute Canada, NRC, and an NVIDIA Faculty Hardware Grant.

REFERENCES

S. Dieleman, J. De Fauw, and K. Kavukcuoglu. 2016. Exploiting Cyclic Symmetry in Convolutional Neural Networks. (2016). https://arxiv.org/abs/1602.02660

K. Mills, M. Spanner, and I. Tamblyn. 2007. Deep learning and the Schrodinger equation. (2007). https://arxiv.org/pdf/1702.01361.pdf

William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. 1990. Numerical recipes: The art of scientific computing. (1990).

D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow. 2016. Harmonic Networks: Deep Translation and Rotation Equivariance. (2016). https://arxiv.org/abs/1612.04642