MSc in Photonics

PHOTONICS**BCN**

Universitat Politècnica de Catalunya (UPC)
Universitat Autònoma de Barcelona (UAB)
Universitat de Barcelona (UB)
Institut de Ciències Fotòniques (ICFO)

http://www.photonicsbcn.eu

Master in Photonics

MASTER THESIS WORK

Quantum Phase Transition detection via Machine Learning algorithms

Joel Pérez Díaz

Supervised by Prof. Dr. Maciej Lewenstein, (ICFO)
And Dr. Pietro Massignan, (UPC)

Presented on date 31st august 2019

Registered at

Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

Quantum phase transition detection via Machine Learning algorithms

Joel Pérez

ICFO - Institute of Photonic Sciences (Quantum Optics Theory Group), Mediterranean Technology Park, 08860 Castelldefels (Barcelona), Spain.

E-mail: joel.perez@icfo.eu

September 2019

Abstract. A Neural Network is trained to classify Mott Insulator and Superfluid phases in an optical lattice using data generated with Diffusion Monte Carlo algorithms (DMC). The trained model is used to predict the phase transition and its dependence with different training parameters is studied. The study of this dependence shows the existence of optimal training and simulation parameters, which cannot be used due to computational limitations. This prevents to calculate the phase transition diagram consistent with other theoretical and experimental results.

Keywords: Quantum Phase Transition, Machine Learning, Optical Lattices, Quantum Optics

1. Introduction

Machine Learning research began in the 1950s, it became a promising discipline quickly, specially in 1958 with the discovery of the perceptron by F. Rosenblatt [1]. After two decades of research in the field, the realization of many of its limitations led to a substantial decrease on its initial enthusiasm and financial support, which provoked what is known as the AI Winter: a period with almost no interest or financial support in Artificial Intelligence.

It wasn't until the decade of the 2010s, partially because of important advancements in computing power, that Machine Learning began delivering astonishing results and became a trending discipline not only in academia, but also in media and business. Machine Learning is present is most of our everyday life with applications in social media, medicine and gps navigation amongst others.

Physics can also benefit from the advancements in Machine Learning, its image recognition power has been proven to be useful in many physics related applications. The aim of this Master Thesis is to use Machine Learning algorithms to study quantum phase transitions in an optical lattice.

Optical lattices can confine atoms in a periodic structure using laser standing-waves. The properties of this periodic structure can be modified by properly tuning the laser

parameters, which offers the possibility of modelling systems that emulate the crystal lattice structure of some solids. Optical lattices are also useful tools to manipulate the internal states and positions of atoms, making them candidates for applications related with quantum information.

Depending on the relation between the lattice depth and the interaction between particles, atoms inside an optical lattice can be found in two well differentiated phases: Mott Insulator and Superfluid, whose transition is studied with Machine Learning in this Master Thesis.

2. Optical Lattices

In the presence of an electric field the energy of an atom is shifted, this energy shift can be treated as consequence of an external potential which deppends on the intensity of the electric field $\mathcal{E}$. The origin of this potential is referred as the ac Stark Effect and can be expressed as

$$V(\mathbf{r}) = -\frac{1}{2}\alpha'(\omega)\langle\mathcal{E}(\mathbf{r},t)\rangle_t \tag{1}$$

where α' is the real part of the polarizability of the atom.

When the intensity of the electric field is periodic in space, i.e: a standing-wave laser field, the potential (1) is also periodic, which produces an optical lattice in one, two or three dimensions.

Optical lattices in one dimension are the object of study of this Master Thesis. These kind of lattices are achieved by superimposing two counter-propagating laser beams, the most simple configuration are two lasers with identical amplitude, polarization and opposite wave-vector, which produces the following electric field

$$\mathcal{E}(x,t) = \mathcal{E}_0\cos(kx-\omega t) + \mathcal{E}_0\cos(-kx-\omega t) = 2\mathcal{E}_0\cos(kx)\cos(\omega t) \tag{2}$$

resulting in a lattice potential V_L according to (1) of the form

$$V_L(x) = V_0\cos^2(2kx) = V_0\cos^2(\pi x/a_0) \tag{3}$$

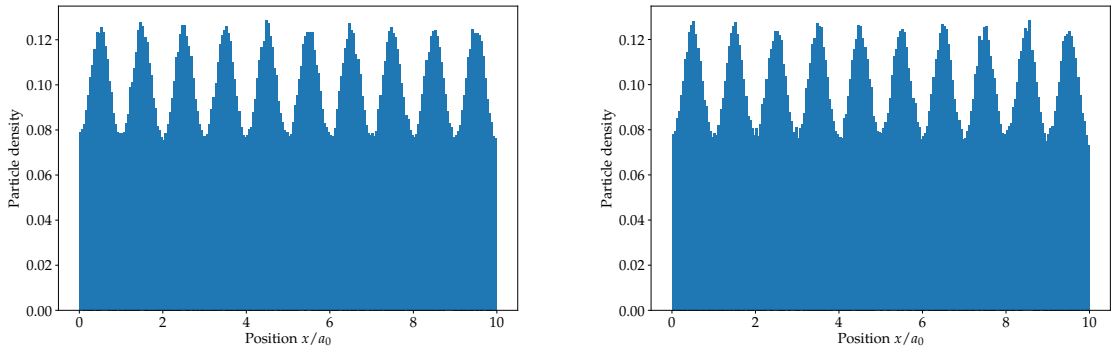
where $a_0 = \pi/k = \lambda/2$ is the lattice depth, i.e: the distance between the minima of the potential. The distance between those minima can be modified via the lasers wavelength λ. The lattice depth V_0 depends on the real part of the polarizability α' and the amplitude of the electric field $\mathcal{E}_0$, hence the potential V of an optical lattice is highly tunable, as well as the behaviour of the atoms it contains.

2.1. From Mott Insulator to Superfluid

The first quantization Hamiltonian of N bosons of mass m in an optical lattice with lattice constant a_0 and interacting with a contact potential of strength $g = -2\hbar/ma$ is

$$\hat{H} = \sum_{i=1}^{N}\left[-\frac{\hbar}{2m}\frac{\partial^2}{\partial x_i^2} + V_L(x_i)\right] + g\sum_{i<j}\delta(x_i - x_j) \tag{4}$$

The phase transition is obtained in [2] from the Luttinger parameter $K = v_F/c$, where $v_F = \hbar N\pi/La_0m$ is the Fermi velocity, fixed by the system setup, and c is the speed of



(a) (b)

Figure 1: Spatial distribution of 10 particles inside the one-dimensional optical lattice, calculated with a million configurations of 10 particles in (a) Mott Insulator and (b) Superfluid phase. Both phases show the same distribution.

sound which deppends on the strength of the interaction $g \propto 1/a$. The speed of sound is calculated via Diffusion Monte Carlo algorithms (DMC) and the transition is found for unit filling ($n = N/L = 1$) at $K = 2$ [3].

For deep optical lattices, where V_0 is high compared with the recoil energy $E_{rec} = \pi^2\hbar^2/2ma_0^2$, equation (4) reduces to the Bose-Hubbard model (BHM)

$$\hat{H}_{BH} = -J \sum_{\langle i,j \rangle}^{L} \hat{a}_i^\dagger \hat{a}_j + \frac{U}{2} \sum_{i=1}^{L} \hat{a}_i^\dagger \hat{a}_i^\dagger \hat{a}_i \hat{a}_i \tag{5}$$

where the first term accounts for tunnelling between lattice sites and the second for the interaction between particles at the same lattice site. The transition is found via exact diagonalization.

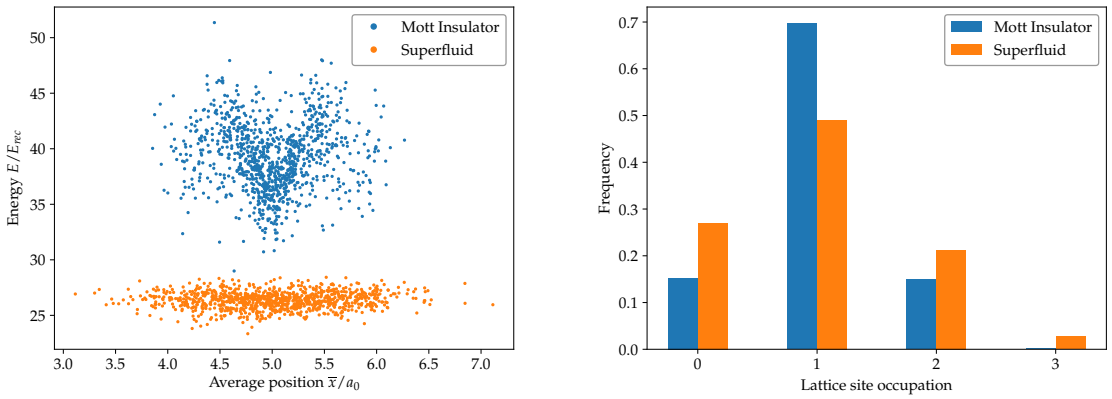
Experimental measurements of the transitions are made in [4], which mostly agree with BHM at $V_0/E_{rec} > 5$, while for lower values the results are closer to those predicted by sine-Gordon model.

The same DMC algorithms used in [2] are used in this Master Thesis to simulate the behaviour of a system of $N = 10$ particles with different scattering lengths in an optical lattice with occupation $n = 1$ and different depths (Fig. 1). DMC generates datasets containing snapshots of the position and energy of the particles in the lattice, and choosing a/a_0 and V_0/E_{rec} far enough from the transition provides examples of Mott Insulator and Superfluid configurations, whose differences are already visible in Fig. 2.

3. Machine Learning

A Machine Learning algorithm is such that can make a computer system perform a certain task without being explicitly programmed to perform that task in particular, but relying instead in inference and pattern detection.

One of the main applications of machine learning are classification problems, i.e: identifying to which of a set of categories a new observation belongs, on the basis of



(a) (b)

Figure 2: (a) Distribution of the average positions and energy of different configurations of 10 particles. (b) Probability distribution of the lattice site occupation.

a training set of data containing observations whose category is known. The object of study of this Master Thesis is the classification between two quantum phases without explicitly programming how to distinguish them.

Supervised methods require a large dataset where the actual category of each element is already known beforehand, which is used to do the training and then predict the category of new elements. Conversely, unsupervised methods don't require a known training dataset, but instead classify all the given dataset in unlabelled categories.

Unsupervised learning algorithms generally have a lower predictability power than supervised ones, however, the latter relies on the availability of a training set correctly classified, which isn't always easy to obtain or even possible, in those cases the former must be used.

Since the dataset is generated with tunable simulations, the phase of each generated element is known, which makes available a large training dataset with known categories and allow supervised methods to be used.

3.1. Neural Networks

A supervised algorithm that has become widely used are Artificial Neural Networks, inspired in biological neural networks. Neural Networks are made by artificial neurons grouped in layers, and artificial synapses that connect neurons between layers.

An artificial neuron is a computing object that has a certain number of inputs, each of those represented by a numerical value, which are summed together plus a characteristic term of the neuron, the bias. The resulting number is evaluated with an activation function and output by the neuron.

Artificial synapses are the connections between neurons, they are the abstract elements that connect the output of a neuron to the input of another in the next layer. Each synapse has a characteristic number by which its input is multiplied called weight, which represents the relevance of that particular connection between neurons.

A Neural Network is structured in layers, at least two, the input and the output layer. The number of neurons in the input layer is the dimension of the data, while the number of neurons in the output layer is the size of the set of categories. Between these two layers there can be an arbitrary number of internal layers with an arbitrary number of neurons. A higher number of layers and neurons allows the classification of more complex datasets, but increases the computing time required for its training and can result in overfitting if the complexity of the network is too high compared with the complexity of the data. Overfitting is a common issue in Machine Learning methods, an overfit model memorizes the training set instead of learning its subtle patterns, resulting in a high precision when predicting the actual training dataset but a significantly lower predicting an unknown one.

3.2. Training a Neural Network

A Neural Network takes as input an array of real numbers which are output separately by each neuron of the input layer, each output is propagated to each neuron of the next layer being multiplied by the weight of its connection, in this layer each neuron sums all of its inputs plus its bias and outputs the result to the next layer, this process is repeated until arriving to the output layer. The output layer represents the result of the classification, each neuron represents a category and its output is a number from 0 to 1 representing the likelihood of the input belonging to that category.

The output of a Neural Network depends on its set of weights and biases, which begin as random values and through training algorithms are tuned in a way that can reproduce the classification of training dataset.

To evaluate how bad a Neural Network performs, a cost function is used. It measures the distance between the expected and obtained outputs. Slowly varying the weights and biases in a way that the cost decreases, the network arrives to a state where the cost function is minimal at least locally, and produce an output which is very close to the expected one.

Training a neural network consists therefore in finding the minimum of a function numerically, which can be achieved with a gradient descent. However, the cost function typically has a large number of variables and its evaluation for a given set of weights and biases requires the processing of large batches of data through the entire network. Directly performing a gradient descent would take an unfeasible amount of time. Backpropagation uses gradient descent individually on each layer, beginning from the output and going backwards, this reduces significantly the number of effective variables of the cost functions and the number of neurons that an input has to go through to compute it.

4. Binary Classification

4.1. Neural Network structure

The problem to solve is a classification between two categories: Mott Insulator and Superfluid. This is the particular case of a binary classifier, a classifier which only has two outputs. Since this configuration only needs an output neuron, which yields *zero* or

one depending on if the input is classified as Mott Insulator or Superfluid, the appropiate cost function to be used in the backpropagation is the binary cross-entropy

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log\left[p(y_i)\right] + (1-y_i)\log\left[1-p(y_i)\right] \tag{6}$$

An adequate optimizer is RMSprop [7], which automatically adjusts the learning rate in a convenient way at each learning step.

The layer structure of the neural network only consists of an input layer, an internal layer of 64 neurons using the rectified linear unit (ReLU) [5] as activation with dropout to reduce overfitting, and a single output neuron with the sigmoid activation function. The output of this last neuron varies from 0 to 1, and the precise meaning of this number is the probability that the input configuration is in Superfluid phase.

4.2. Training

To find the transition between phases, it is not required to detect the phase of a single configuration, but rather to detect the phase of a point in the $V_0/E_{rec}-\gamma^{-1}$ phase diagram. Many configurations in the same phase diagram coordinates can be used as a single input and then the network only has to guess one phase by seeing many examples. In practice, this translates to instead of using a vector with the positions of N particles as input, use an $R \times N$ matrix with R configurations of N particles.

Using more than one configuration at once produces more accurate results, but with a dataset consisting of M configurations, using R configurations a a single input reduces effectively the number of samples to M/R. Furthermore, the number of parameters of the network gets increased with R, which requires an even larger dataset for the training algorithm to work properly.

Finding an optimal value of R that delivers maximum accuracy with an stable training isn't trivial and depends on N and M. In this Master Thesis, where $2 \cdot 10^9$ configurations of $N = 10$ particles have been used for the training, the value of R which delivered good results is $R = 100$ (Fig. 3a).

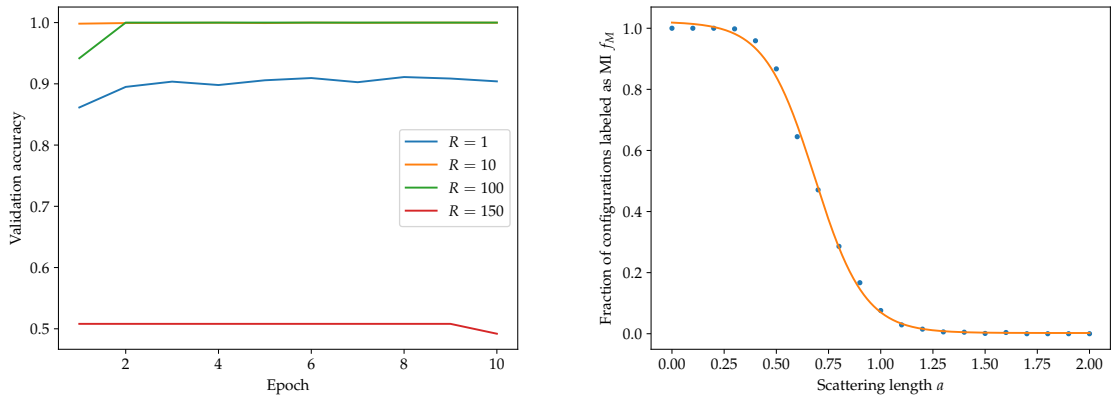
5. Phase Transition Detection

5.1. Sigmoid fitting

For a given lattice depth V_0/E_{rec}, there is a critical value of a, called a_t, for which those configurations with $a < a_t$ are in Mott Insulating phase, and those with $a > a_t$ in Superfluid phase.

And ideal model should be able to classify all $a < a_t$ configurations as Mott Insulator, and all those with $a > a_t$ as superfluid. Since the network has been training at extreme values of a which are far from the transition value, its behaviour at closer values is unexpected, and a more realistic result would be an increasing uncertainty in the phase of the configurations as the value of a gets closer to the transition.

The method to find a_t for a given lattice depth is to generate many configurations at different values of range of a wide enough to certainly contain the transition without the



(a) (b)

Figure 3: (a) Evolution of the validation accuracy for different number of matrix rows R. (b) The amount of configurations labelled as Mott Insulator with respect to the scattering length a is adjusted to a sigmoid, obtaining $a_t = 0.69$.

use of a priori information about its precise location. The generated configurations are grouped in matrices with the same number of rows as in the training, and each of this matrices is classified as either Mott Insulator or Superfluid.

For each a, a fraction of the configurations is labeled as Mott Insulator. This fraction is denoted by f_M and is a real number which should be equal to 0 at $a > a_t$ and 1 at $a < a_t$, with a sudden gap from 1 to 0 at $a = a_t$, i.e. the Heaviside step function (7). What is more realistic and what is observed is a sigmoid shape (8) where $f_M \simeq 1$ and $f_M \simeq 0$ at Mott Insulator and Superfluid configurations, respectively, which are far from the transition, with a smooth transition from 1 to 0 at $a \simeq a_t$.

$$f_M(a) = H(a_t - a) = \begin{cases} 1 & \text{for } a \leq a_t \\ 0 & \text{for } a > a_t \end{cases} \tag{7}$$

$$f_M(a) = f_0 + \frac{A}{1 - \mathrm{e}^{s(a-t)}} \tag{8}$$

For equation (8) to be valid, $f_0 \simeq 0$, $A \simeq 1$ and $s > 0$. The offset f_0 is the fraction of configurations labeled as Mott Insulator when $a \to \infty$, the amplitude A is the difference between the maximum and minimum value of f_M and the stepness s measures how sudden is the transition detected. In the limit and ideal case of $f_0 = 0$, $A = 1$, $s \to \infty$ and $t = a_t$, equation (8) is equivalent to equation (7).

A set of f_M–a points can be adjusted to equation (8), obtaining its parameters, and by inverting it, the value at which $f_M = 1/2$ is found (Fig. 3b). This value is proposed as the transition value a_t

$$a_t = \frac{1}{s} \log\left(\frac{A}{1/2 - f_0} - 1\right) - t \tag{9}$$

5.2. Using many trained models

Random processes are involved in the training. Two binary classifiers which classify the training dataset with perfect accuracy can have a different set of internal parameters, which can lead to slight differences at classifying configurations at non-extreme values of a and result in different values of a_t.

A solution to this problem would be to modify the neural network hyperparameters until the variance between different values of a_t is minimal. Nonetheless, the network already has a perfect accuracy classifying the values of a with which it is trained, and basing the tuning of the hyperparameters on a so posterior result, as it is the whole process of sigmoid fitting, is highly non-trivial.

The alternative is a more pragmatic numerical approach. The network is trained 300 times and for each training a transition a_t is calculated. Rather than obtaining a precise value of it, what it is obtained is statistical information about its expected value and uncertainty.

The accuracy and precision of a_t is related with the width of the sampling interval. A narrow interval which contains the theoretical transition will be more accurate, since the non-horizontal part of the sigmoid will be more represented by the samples, which will result in a better sigmoid adjustment. Nonetheless, a too narrow sampling interval that involves a priori knowledge of the precise theoretical value of the transition would invalidate its predictability power.

A solution to the previous dilemma consists in finding a first value of a_t with a wide training interval with its extremes far from the transition. With the obtained values of a_t a second interval is determined based on the average and dispersion of a_t.

5.3. Training coordinates

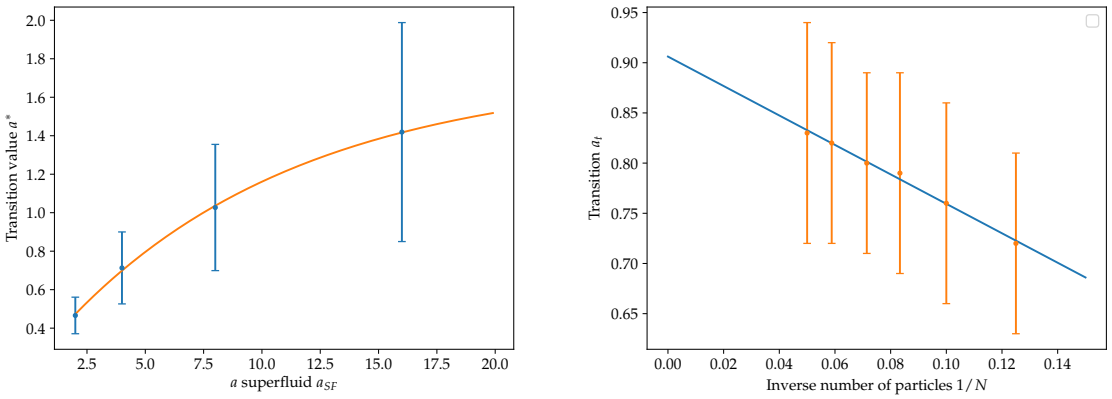
The sigmoidal shape show that the trained models do slightly distinguish between two a at the same theoretical phase, which may indicate that the choice of the two training coordinates from which the network learn the definition of Mott Insulator and Superfluid phase could be relevant to the transition value calculated.

This dependence is shown in Fig. 4a, where the training coordinates for Mott Insulator phase are $S_M = 1$ and $a_M = 0.0$ while for Superfluid phase the lattice height is fixed at $S_S = 1$ and a_S is varied from 2.0 to 16.0, showing an asymptotic dependence (10) with a value at infinity $a_t = c_0 = 1.73$.

$$a_t = c_0 + c_1 \mathrm{e}^{c_2 a_S} \tag{10}$$

This value is far from the value obtained with the continuous model at [2], where $a_t = 0.69$. This is justified by the fact that while a_S is infinitely far from the transition, a_M is still very close and the transition value obtained is shifted towards the Superfluid. Unfortunately, the DMC algorithm used doesn't work properly for $S > 5$, which prevents from using a S_M high enough to define a training point for Mott Insulator far enough from the transition.

With this limitation, what can be tested is whether a Neural Network that can properly detect the transition at $S = 1$ can find the transitions at different values of S. From Fig. 4a the training coordinates can be hand-picked in such a way that $a_t = 0.69$, this



(a) (b)

Figure 4: Transition a_t at $V_0/E_{rec} = 1$ with respect to (a) a superfluid a_S and (b) inverse number of particles $1/N$. Both dependencies show the tendency to a finite value at infinite N and a_S. With infinite particles and $a_S = 3.9$, $a_t = 0.91$; with infinite a_S and $N = 10$ particles, $a_t = 1.73$.

corresponds to $S_M = S_S = 1$, $a_M = 0.0$ and $a_S = 3.9$. As it is shown in Fig. 5, the different transitions found aren't significantly sensible to the lattice height, showing that at least with the chosen training coordinates, the networks aren't detecting the phase transition.

5.4. Finite size effects

The calculations for the phase transition of the different models mentioned in section 2.1 rely on approximations for a high number of particles $N \to \infty$, while the configurations studied have only $N = 10$ particles.

Since the simulation and training time grows exponentially with N, the transition for large values of N cannot be explicitly calculated, but exploring a range of different small values N can give an approximate idea of its behaviour as $1/N \to 0$, as it is shown in Fig. 4.

6. Conclusions

A Neural Network can be successfully trained to distinguish between two coordinates in the Mott Insulator – Superfluid phase diagram. The sigmoidal behaviour of f_M show that there is a small region compared to the training interval where the classification isn't either as Mott Insulator or Superfluid, which is proposed to be the physical phase transition. The location of this transition has a strong dependence with the number of particles and the training coordinates, which show that the optimal values are those computationally more exigent.

With the available training and simulation parameters, the phase transition can't be found with the implemented method.

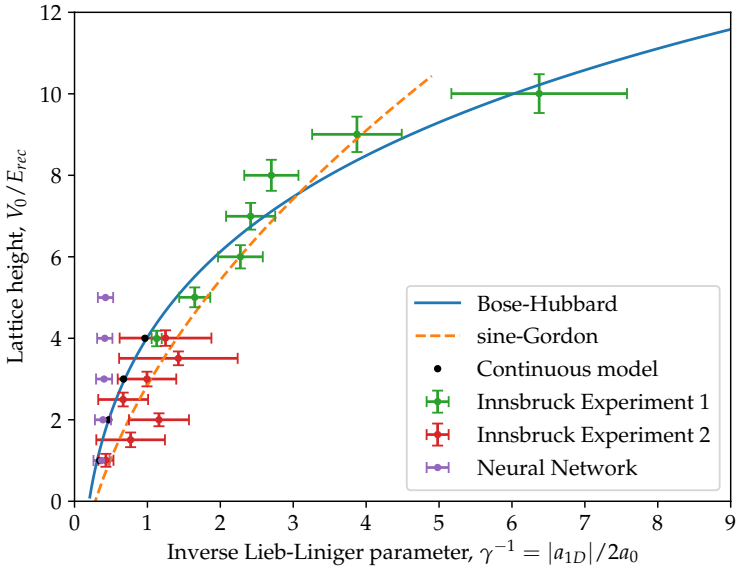


Figure 5: Phase diagram comparing various theories [2] and experiments [4]. The training coordinates of the Neural Network are chosen a way such that the transition value coincides with that from the continuous model at $V_0/E_{rec} = 1$, which isn't capable of finding the transition at different lattice depths.

The data used to train the models used single–coordinate definitions of the phases, i.e: one training coordinate for Mott Insulator and another of Superfluid. It is possible that the information about the phase doesn't lie on a single point at the phase diagram and a good approach would be to use many of them for both phases. Although this approach is very likely to improve the results, using an effective surface on the phase diagram as definition for the phases requires more a priori knowledge of the actual value of the transition, which could make irrelevant any finding if the training surfaces are too close to the transition.

7. Bibliography

[1] Rosenblatt F 1958 *Psychological Review* 65–386
[2] Astrakharchik G, V Krutitsky K, Lewenstein M and Mazzanti F 2015 *Physical Review A* **93**
[3] Giamarchi T 2004 *Quantum Physics in One Dimension*
[4] Haller E, Hart R, Mark M, G Danzl J, Reichsöllner L, Gustavsson M, Dalmonte M, Pupillo G and Nägerl H C 2010 *Nature* **466** 597–600
[5] Hahnloser R H R, Sarpeshkar R, Mahowald M A, Douglas R J and Seung H S 2000 *Nature* **405** 947–951 ISSN 1476-4687 URL `https://doi.org/10.1038/35016072`
[6] Greiner M and Fölling S 2008 *Nature* **453** 736 EP – URL `https://doi.org/10.1038/453736a`
[7] Hinton G 2015 Neural Networks for Machine Learning notes URL `http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture\_slides\_lec6.pdf`
[8] Gersch H A and Knollman G C 1963 *Phys. Rev.* **129**(2) 959–967 URL `https://link.aps.org/doi/10.1103/PhysRev.129.959`