UNIVERSITAT DE BARCELONA

Facultat de Física

Artificial neural network study of few boson systems

Name: Daniel Tamayo Mela

Degree: Physics

Name of the institution: University of Barcelona

Name of the tutor: Bruno Julià

Traineeship period: October 1, 2020 – June 1, 2021

Delivery date: July 9, 2021

Machine Learning Few-Body Quantum Systems

D. Tamayo[1*], P. Mujal[1,2], B. Juliá-Díaz[1,2] and S. Pilati[3]

1 Departament de Física Quàntica i Astrofísica, Universitat de Barcelona, Martí i Franquès 1, 08028 Barcelona, Spain
2 Institut de Ciències del Cosmos (ICCUB), Universitat de Barcelona, Martí i Franquès 1, 08028 Barcelona, Spain
3 School of Science and Technology, Physics Division, Università di Camerino, 62032 Camerino (MC), Italy
* danitamayo.adhoc@gmail.com

July 8, 2021

Abstract

We present a supervised machine-learning neural network capable of solving the ground energy problem of a few-body quantum system with a speckle potential, number of particles N and interaction strength g as a feature to the network.

Contents

1 Introduction & Objectives

From the *Departament de Física Quàntica i Astrofísica* of the University of Barcelona, we have collaborated with the University of Camerino to follow the study of a previous paper: "Supervised learning of few dirty bosons with variable particle number" [5].

The idea we are going to explore is if we can generalise an *Artificial Neural Network* (ANN) that computes the ground energy of a many-body system adding the strength parameter (g) as input, this parameter defines the interaction between two particles:

$$v(x_i, x_j) = g\delta(|x_i - x_j|) \tag{1}$$

we use $g > 0$, the repulsive case.

To compute the ground energies of many-body quantum systems it is usually used the following Hamiltonian:

$$\mathcal{H} = \sum_{i=1}^{N}(-\frac{\hbar}{2m}\frac{\partial^2}{\partial x^2} + V(x_i)) + \sum_{i<j}^{N} v(x_i, x_j) \tag{2}$$

which has proven its utility in experiments performed with ultracold atoms in optical speckle fields [1]; m is the particle mass, N the number of particles and x_i corresponds to the position of particle i, with i = 1,...,N.

Speckle potentials, $V(x)$, can be simulated by stochastic numerical algorithms [2][4], the generation of this data has been explained extensively in the other article, the important ideas we must know are:

- We can change how smooth is the potential by a parameter named *average intensity* V_0. This parameter will be of particular relevance as we will see in section 4, but until we get to that section, we will consider it as V_0=5.
- We have a characteristic energy scale, the correlation energy $E_c = \frac{\hbar^2}{m\ell}$ where ℓ is the correlation length.

With this model, it is been simulated a one-dimensional box of $L = 20\ell$ with a grid of 1024 points where it can be seen how $\delta x \approx 0.153\ell$, in other words, the discretization effects are negligible. This simulation allows us a numerical computation of the eigenvalues of the energy operator function with a threshold in which we only take the basis states with small kinetic energy. The larger the number of particles, the more complicated the diagonalization of the matrix will be, here is where transfer learning with artificial intelligence techniques allows us to obtain an interesting improvement.

In the previous article it is been shown how the error of the predicted energies by an ANN with the speckle and the number of particles as input decreases exponentially with the number of instances used in the training procedure, the objective is to obtain similar tendency with g as input.

2 First technique

2.1 Initial ANN structure

We have used a similar network structure of the last paper:

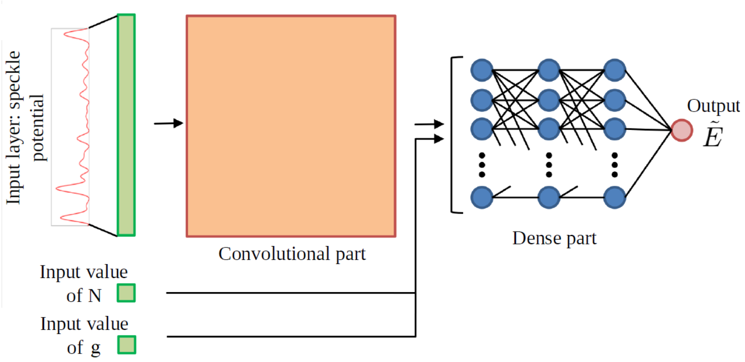

Figure 1: Schematic representation of the deep feed-forward neural network used to predict the ground-state energy, $\tilde{E}$, of few-boson systems (output). The input descriptors are the values of the speckle potential on a fine discrete grid. In the case of training with heterogeneous datasets, additional system descriptors are included, representing the particle number N and interaction strength g. This descriptors are connected directly to the dense part of the network.

The convolutional part is composed by 7 convolutional layers with:

- Kernel size = 5.
- Number of filters = 50.
- Strides = 1.
- Padding = "same". It means there is no change size between convolutional layers, only with the Local and Global MaxPooling layers.
- LocalMaxPooling layer between each convolutional layer. We are going to explore the pool size = 2 and 3
- Activation function = ReLu.
- Optimization method: Adam [3].
- Loss function: Mean Squared Error.
- Batch-size=32.

This part of the network ends with a Global Max Pooling (GMP) layer which is connected with the g interaction value an the number of particles N.

The dense part is composed by 3 layers of 30 neural networks with the same activation function ReLu and ends connecting our network with the energy ground state of the system.

Unless otherwise indicated, the percentages used in each training procedure will consist of a 60% of training data, 20% of validation data and 20% of test data.

2.2 Training procedure

Throughout this work we have verified how the parameters: number of hidden layers, number of filters and kernel-size exposed in the older network[1] were those that gave the most optimal training, even though, we have seen how the parameter pool-size seems to improve the results sometimes. To make a study of which is the optimal parameter we decide to analyse different networks:

- Network with pool size = 3.
- Network with pool size = 2.
- Network with pool size = 2, one more convolutional and LocalMaxPooolling layer giving us eight final neurons at the end of the convolutional part. We connect this end with the two other neurons (number of particles N and interaction strength) and then continue with the dense part.
- Network with pool size = 2, one more convolutional and LocalMaxPooolling layer giving us four final neurons at the end of the convolutional part. We connect this end with the two other neurons (number of particles N and interaction strength) and then continue with the dense part.

We perform the study of the accuracy in the training procedure with the parameters Mean Absolute Error (MAE) and coefficient of determination (R^2):

$$\begin{aligned} MAE &= \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} |E_{i,exact} - E_{i,pred}| \\ R^2 &= 1 - \frac{\sum_{i=1}^{N_{test}} (E_{i,exact} - E_{i,pred})^2}{\sum_{i=1}^{N_{test}} (E_{i,exact} - <E>)^2} \end{aligned} \tag{3}$$

and with the following distribution of data[2]:

- 34000 instances of g=0 separated in:
  - 20000 instances of N=1.
  - 10000 instances of N=2.
  - 2000 instances of N=3.
  - 2000 instances of N=4.
- 50000 instances of g ∈ [0,1] and N=2 distributed uniformly in g.
- 10505 instances of g ∈ [0,1] and N=3 distributed uniformly in g.

When we train an ANN, there are a few concepts we should take into account. Make only one forward and back-propagation [6] doesn't give us the best weights, to obtain them we must repeat different times the training procedure with the same instances (60% of the total data defined as *training data*), we define each training as an *epoch*. The number of epochs that we must use should be large enough to capture the essence of the relations between variables but not too large to cause over-fitting [7].

The strategy we have followed to avoid over-fitting is use the 20% of our data as *validation data*, this data-set allows us to compute how the network improves for data not introduced in the training at each epoch, when the network does not improve its cost

[1] This study has been done in the network where we haven't g as a parameter.

[2] All the data-sets I have exposed has been computed by Pere Mujal

function over a number of epochs defined as *Early-Stopping* (we have used eight), the training stops and we have our final weights. The other 20% of data is defined as *test data* and is the one that we use to analyse the accuracy of the final network with the equations (3).

2.3 Results

Computing four different training with the ANNs of the previous section we have found the following accuracy parameters:

Parameters		pool=3 (GMP)		pool=2 (GMP)		pool=2 (Flatten 8)		pool=2 (Flatten 4)	
g	N	R^2	MAE	R^2	MAE	R^2	MAE	R^2	MAE
0 and (0,1]	1,2,3 and 4	0.9991	0.0286	0.9992	0.0252	0.9990	0.0313	0.9992	0.0246
0	1	0.9973	0.0150	0.9985	0.0123	0.9968	0.0176	0.9981	0.0131
	2	0.9981	0.0261	0.9990	0.0188	0.9979	0.0285	0.9986	0.0220
	3	0.9983	0.0378	0.9990	0.0325	0.9976	0.0446	0.9988	0.0323
	4	0.9978	0.0570	0.9975	0.0467	0.9972	0.0611	0.9967	0.0534
	1,2,3 and 4	0.9995	0.0225	0.9996	0.0175	0.9994	0.0250	0.9995	0.0192
(0,1]	2	0.9979	0.0283	0.9981	0.0253	0.9976	0.0305	0.9985	0.0236
	3	0.9969	0.0507	0.9971	0.0490	0.9966	0.0562	0.9969	0.0478
	2 and 3	0.9985	0.0321	0.9986	0.0295	0.9983	0.0349	0.9988	0.0278

Table 1: Different coefficients of determination and MAE for each structure in different intervals of the test-set (18901 instances). The whole data-set is composed by 94505 instances and has lasted three hours to each network to optimise the weights. The best network seems to be the one with pool=2.

We have obtained high accuracy parameters, to make an idea of how linear the predictions are we can represent the case poolsize=3:

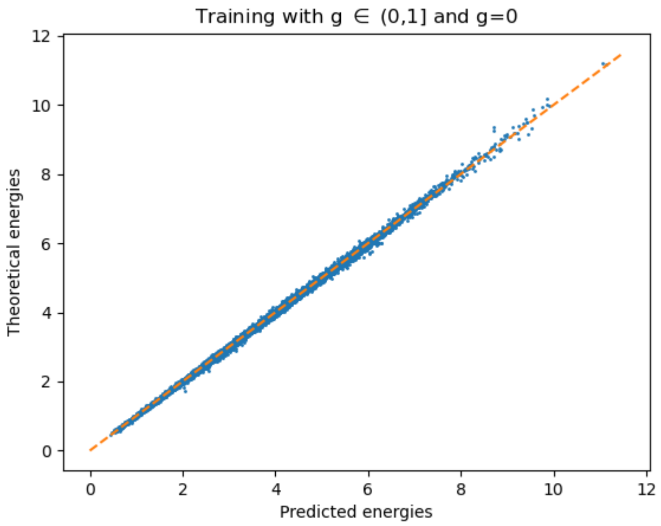

Figure 2: Theoretical energies vs the predicted ones representation in the 20% test set (instances with g=0, g ∈ [0,1] and N=1,2,3,4). We have a total of 18901 instances. The training procedure has lasted 3 hours.

Despite this great start, we must do a deep analysis; we can see how, when we explore for the same speckle (seed[3]=50235) and the same number of particles (N=2), the dependence of the energy with the strength parameter is sometimes not well captured:

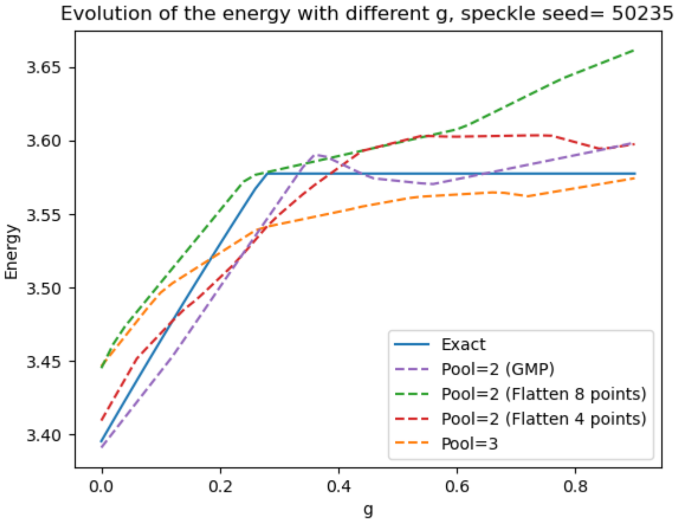

Figure 3: Theoretical and predicted energies dependence with g in networks with different pool sizes and a change of the GMP for a flatten output, we have chosen the seed=50235 to simulate the speckle and N=2.

Note how there is one value of g in which we have a clear change of tendency, we define this value as the *transition point* g_0. To have an idea of why is this produced we must imagine bosons as repulsive particles but that has a desire to stay in the lower parts of the speckle potential with a higher probability, we can see this behaviour in the figure 4 extracted from the article *"Scalable neural networks for the efficient learning of disordered quantum systems"* [8].

The larger the value of g, the more difficult it is to keep the bosons in positions close to this minimum energy position due to their strong repulsion. When g is higher enough (*transition point*) the second particle starts to stay (with a high probability) in far positions from the lower energy potential zone and the repulsive effect is no longer relevant in the evolution of energies.

The problem we have in figure 3 is how the constant part tendency is not well captured by our networks, e.g. for pool-size=2 and flatten=8, seems to predict a grow of energy for $g > g_0$.

The results of figure 2 are good but they could be better, for that reason, we have decided to try another method to compute the energy values, it is explained in the following section.

[3] Remind section 1, we compute potential speckle functions with pseudo-random numbers.

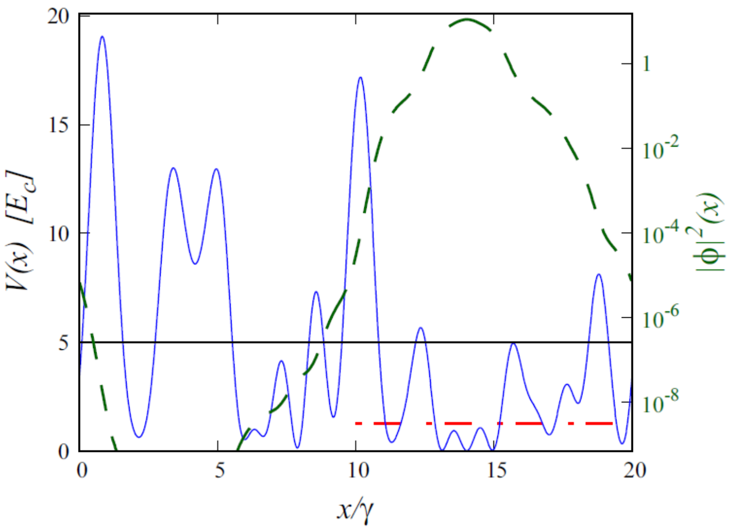

Figure 4: Profile of an instance of an optical speckle field $V(x)$ (blue curve) as a function of the spatial coordinate x/γ . The system size is $L = 20\gamma$ where γ is the disorder correlations length. The (black) horizontal line represents the average intensity $V_0 =< V(x) >$, and the (red) dot-dashed segment indicates the ground-state energy E. Energies are expressed in units of the disorder correlation energy E_c. The dashed (dark green) curve represents the squared modulus of the ground-state wave function (x) with ℓ_2 normalization. The corresponding scale is logarithmic and is represented on the right vertical axis.

3 Interactive potential as a perturbation

3.1 New strategy

In the previous section, we have seen how our network has problems predicting the constant part of the expected dependence between the energy and g (figure 3). Despite the bad news, we have notice something, as we can see in the same figure the increase in g has a relative impact of a 5% in the energy, our network is probably focusing the training capturing the strong dependence between the speckle and the number of particles with the energy and has problems finding other dependencies.

In order to force our network to focus on the calculation of the dependence on g, we can train a network to only compute the values $\Delta E \equiv E_g - E_{g=0}$. The network structure will be the same of the section 2 with pool-size=2 and GMP but we will change the output:

- INPUTS:
  - Speckle.
  - Number of particles.
  - Value of g.
- OUTPUT: $\Delta E \equiv E_g - E_{g=0}$

3.2 Computation of ΔE

In order to make a first try we use the exact $E_{g=0}$ to train the network exposed in section 3.1, we use our data-set of 50000 instances with N=2 and 10505 instances with N=3 distributed uniformly with $g \in [0, 1]$. We have found:

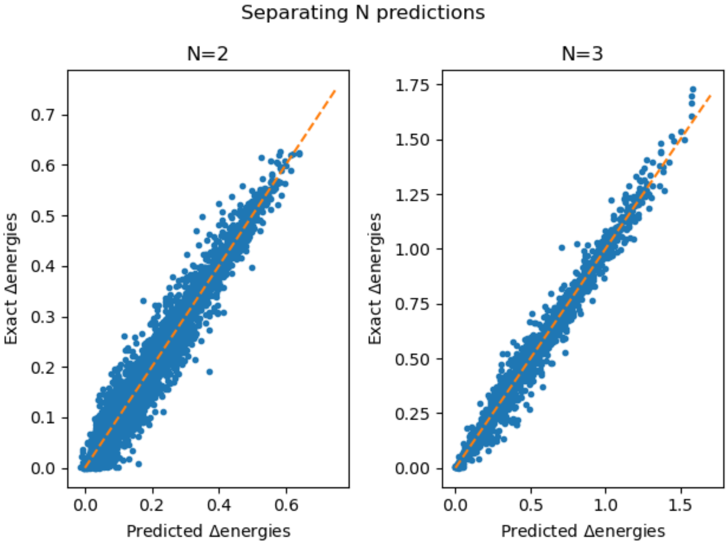

Figure 5: Representation of theoretical and predicted ΔE by the network of pool-size=2, the represented points are part of the test-data, 12101 instances with $g \in [0, 1]$. The training procedure has lasted 2 hours.

Accuracy parameters:

- N=2; $R^2 = 0.97$, MAE=0.014.
- N=3; $R^2 = 0.98$, MAE=0.030.

We can see how MAE has decreased compared to the first procedure (table 1), we can also see an improvement in the behaviour of this difference for different $g \in [0, 1]$ in the same speckle with seed=50235:

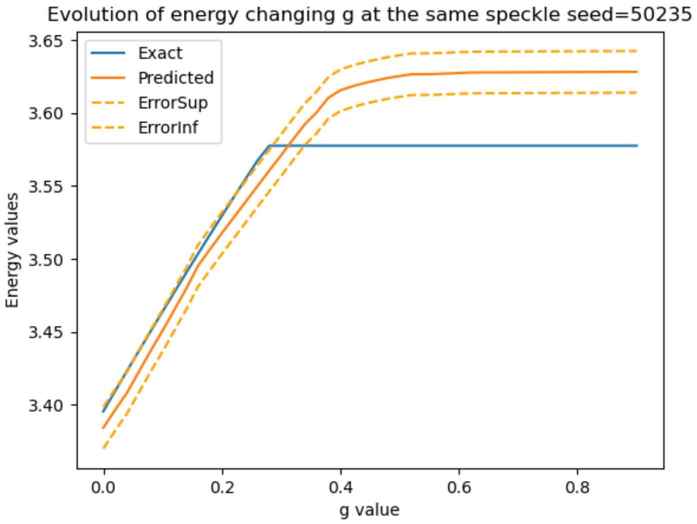

Figure 6: Evolution of the energies with the same speckle and N=2 predicted with the network explained before. The errorbar is the variance of the data-set.

Despite the fact we have not captured the exact *transition point*, we can see how good is our network predicting the stationary state.

3.2.1 Error exploration

In this section we are going to explore where are the trouble points:

- MAE representation in 19 different g-ranges:

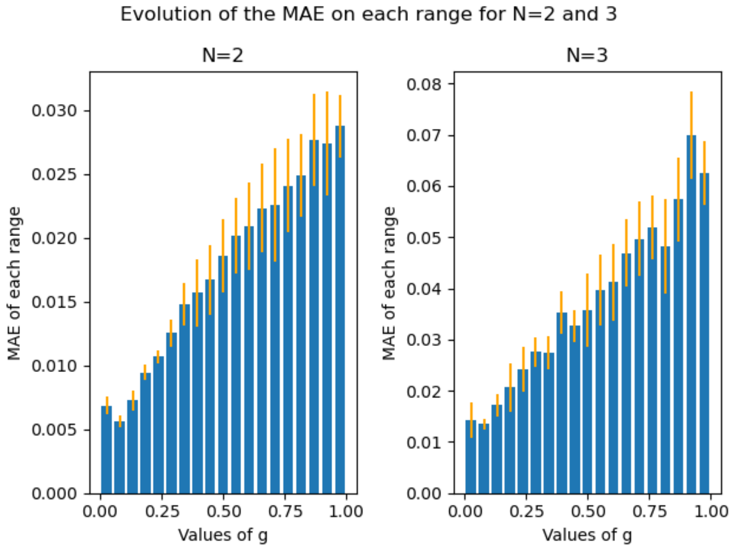

Figure 7: Evolution of the MAE for the N=2 (left) and N=3 (right) test data computed with the network of the previous section for 19 different g-ranges. The representation is the mean of five different training and the error-bar is its standard deviation of each range. We have different 30000 instances for N=2 and 6303 instances for N=3 in each training procedure.

The tendency seems to be monotonous increasing with g, we could try to change a little bit our data-set and see if giving more importance to higher ranges there's some improvement, we have tried to put the following distribution of instances:

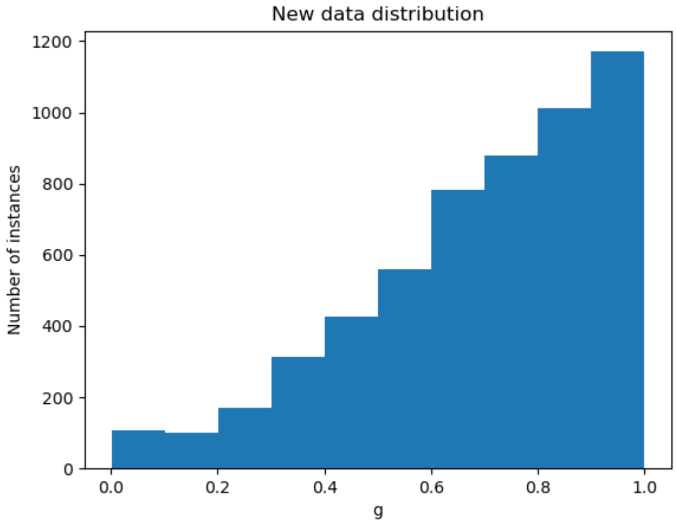

Figure 8: New data distribution of instances which we are going to use to train the network, we have reduced the 60505 original size of our data to 27629.

The results haven't improved, we have find:

- For N=2, MAE=0.03 and $R^2 = 0.90$.
- For N=3, MAE=0.06 and $R^2 = 0.92$.

But we can clearly see how the slope has decreased at least for N=2.

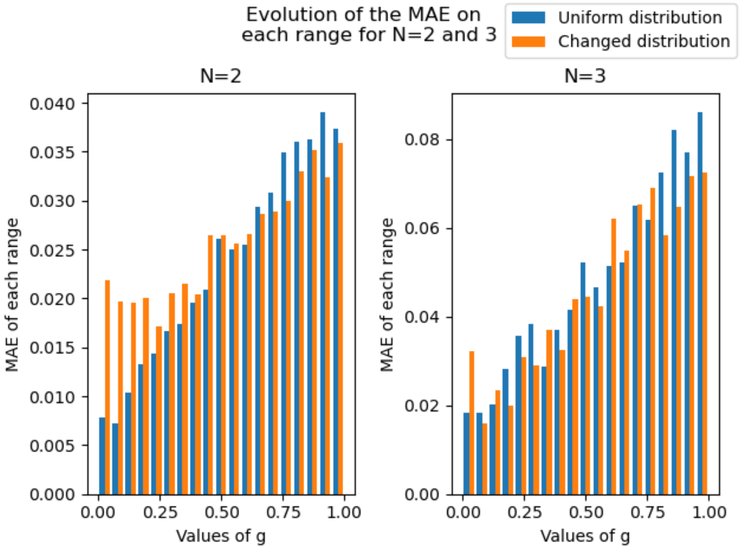

Figure 9: Comparison of the MAE in each range of g for the N=2 (left) and N=3 (right) of a smaller data-set. The training procedure have been done with g distributed uniformly (blue) or with the changed distribution (orange). We have 4566 instances for N=2 and 959 instances for N=3.

- Representation of the absolute error in each point for the higher r-range:

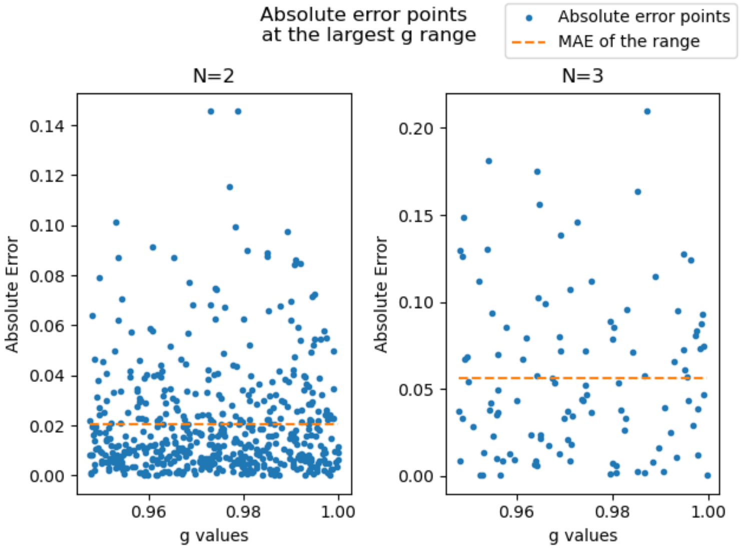

Figure 10: Evolution of the absolute error for the N=2 and N=3 test data computed with the network of the previous section (figure 65) for the largest g-range. We have 507 instances for N=2 and 110 instances for N=3.

We can clearly see how dense is the region where we don't have a higher absolute error than the MAE for N=2 but also how big can be the absolute error for some points, we will need to change something if we want more precision.

Is there a predominant speckle behavior for the highest AE points?

For $MAE > 0.01$ we can find the following speckles:

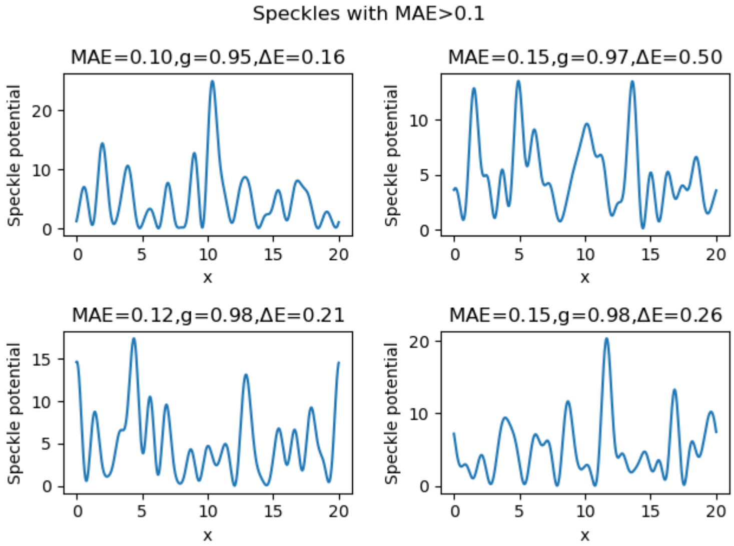

Figure 11: Different speckles obtained in which we have the worst predictions of the network ($MAE > 0.1$). We are only considering N=2 and the largest range of g.

With the same network in each speckle, we predict different g values to see where is the problem:

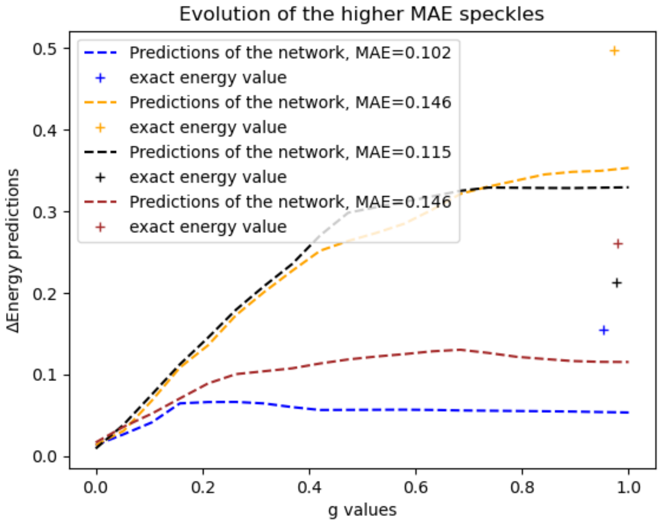

Figure 12: Predictions of the ΔE for two particles and different g values in the previous 4 speckles. We have also represented the exact energy value of our data set.

We can see how the constant tendency seems to be well captured in our network but there is some problem estimating where it should starts.

3.3 Computation of $E_{g=0}$

From the paper on which this work is based, we already know that ANNs can capture the behaviour of energy for constant; furthermore, we are in the simplest case in which there is no interaction between particles.

We use the same parameters of the network exposed in previous sections but without N and g as input this time:

- INPUT: Speckle potential.
- OUTPUT: $E_{N=1}$

We haven't worried about the number of particles because we will always be able to compute the energy of the system multiplying $E_{N=1}N$.

We have done four different training with different number of data and hyper-parameters[4]:

- Training 100000 instances:
  - $MAE_{pool=2}$=0.0041, $R^2 = 0.9998$.
  - $MAE_{pool=3}$=0.0046, $R^2 = 0.9998$.
- Training 200000 instances:
  - $MAE_{pool=2}$=0.0043, $R^2 = 0.9998$.
  - $MAE_{pool=3}$=0.0051, $R^2 = 0.9997$.

We can see how the effect of adding 100000 instances in the training procedure does not improve the results, there is a limitation in the accuracy of this network but we consider this error is good enough to make predictions.

As we have four different networks with similar accuracy parameters, we can take the mean of the predicted values by each ANN an represent its linearity with the exact values:

[4] It is hard to say the computational time of this section due to problems in my computer reading so many instances; despite that we can affirm that the four networks have required approximately 24 hours to be trained.

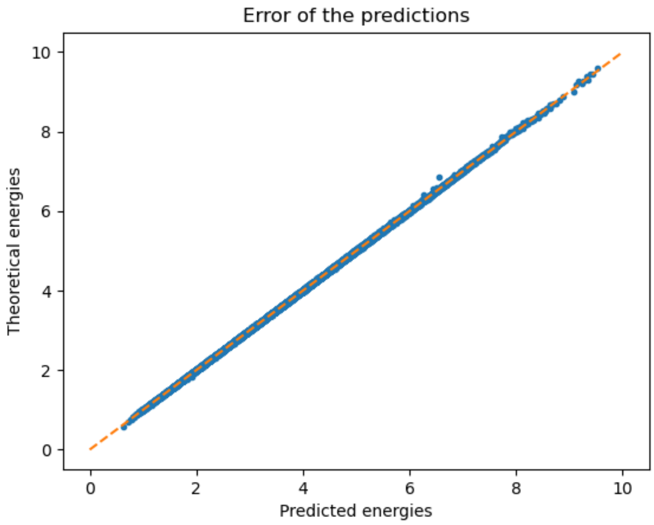

Figure 13: Representation of the theoretical ground energies of the potential speckle and the mean of the predicted ones by the explained 4 networks, the represented data is the one we will use to feed the second network (60505 instances). The total number of instances used in the training 200000 and the percentages are 20% of test data, 20% of validation data and 60% of training data.

Accuracy parameters:

- MAE=0.005.
- R^2=0.99995.

We can also compare the relation between $E_{g,exact} - E_{g=0,pred}$ and $E_{g,exact} - E_{g=0,exact}$:

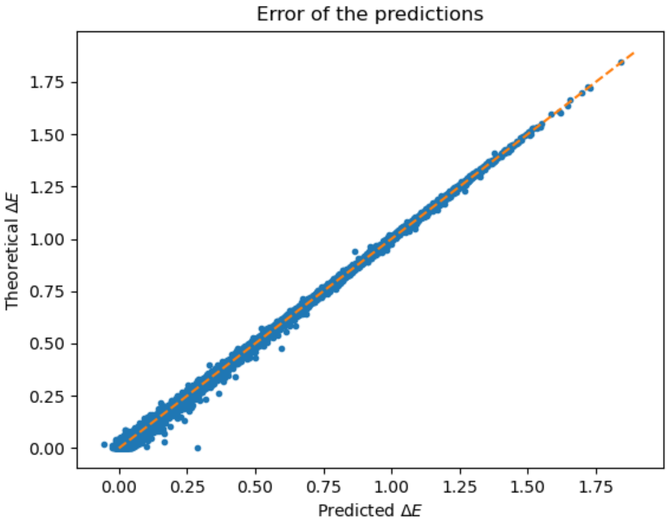

Figure 14: Representation of the theoretical ΔE of the potential speckle and the mean of the predicted ones by the explained 4 networks, the represented data is composed by 60505 instances.

We can see how it seems to be a higher error in the closest g=0 energies but we still have the same MAE and a high coefficient of correlation:

- MAE=0.005.
- R^2=0.9990.

3.4 Using predicted $E_{g=0}$ (-)

Is reasonable to think that as the MAE=0.005 obtained in the previous section is one order of magnitude lower than the error in predictions of ΔE, we could try to train another network with this data:

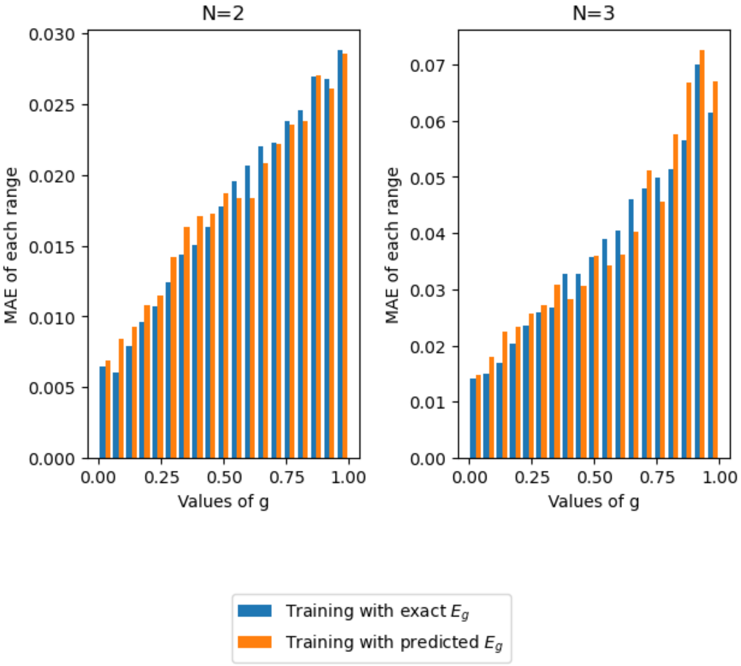

Figure 15: Evolution of the MAE for the N=2 (left) and N=3 (right) test data computed with our second network for 19 different g-ranges. We have represented the test set and it is composed by 10000 instances of N=2 and 2101 instances of N=3.

As we expected there is no significant change between the two training procedures. We have found the following parameters:

- N=2: $MAE = 0.016, R^2 = 0.96$.
- N=3: $MAE = 0.04, R^2 = 0.97$.

3.5 Trying another approaches

3.5.1 Range training (-)

We have seen how the main problem is in the point in which we have a change of tendency so, we could try to make four different training selecting $g \in [0 + 0.25i, 0.25(1 + i)]$ where i=0,1,2 & 3. If we represent the evolution of the error we will see a different tendency now:

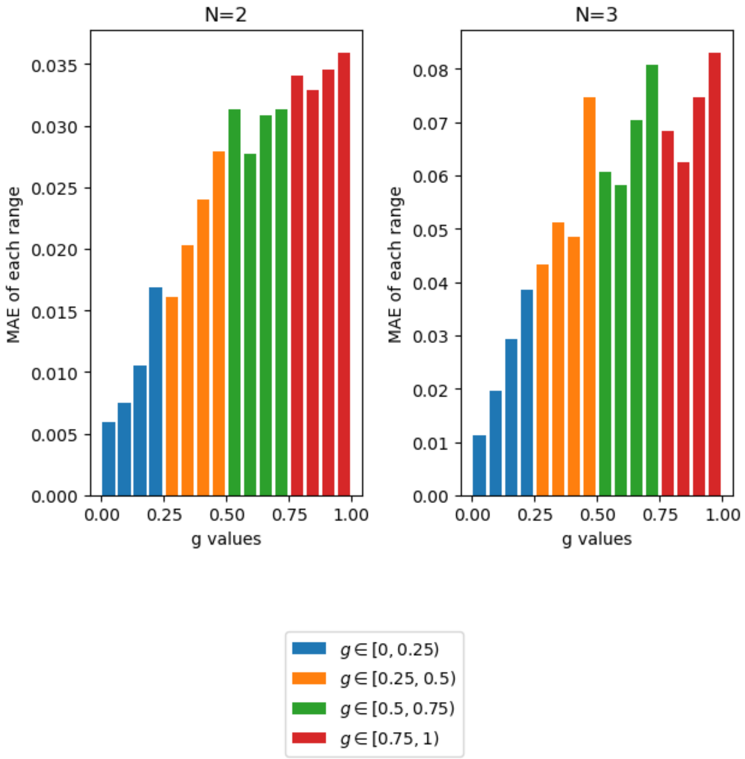

Figure 16: Evolution of the MAE for the N=2 (left) and N=3 (right) test data computed with our second network for 16 different g-ranges (4 ranges for each different training). We have represented the test set and it is composed by 13333 instances of N=2 and 2768 instances of N=3. The whole training procedure has lasted 4 hours.

We can also see the following accuracy parameters:

	N=2		N=3	
Ranges	MAE	R^2	MAE	R^2
$g \in [0, 0.25)$	0.01	0.82	0.02	0.88
$g \in [0.25, 0.5)$	0.02	0.73	0.06	0.79
$g \in [0.5, 0.75)$	0.03	0.81	0.07	0.86
$g \in [0.75, 1)$	0.03	0.87	0.07	0.90

Table 2: Different coefficients of determination and MAE for each different g intervals used in training.

The error is higher, we can also see how bad are our different networks extrapolating values:

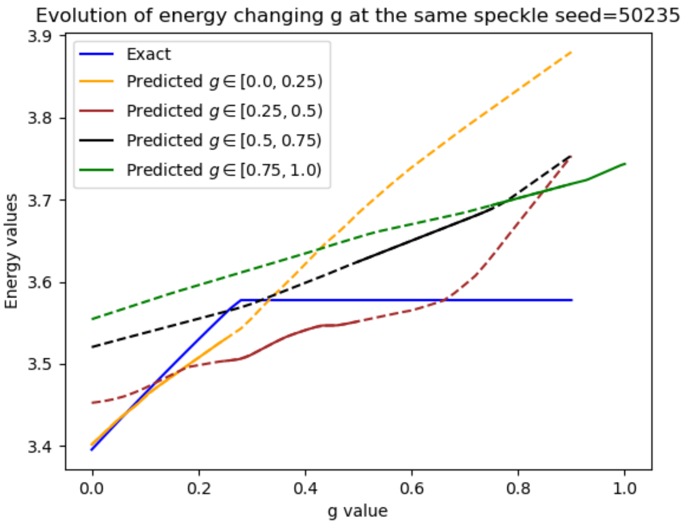

Figure 17: Evolution of the predicted energies by the different networks trained in each different range, the dashed lines are the extrapolations of the network. We have used an speckle with iseed=50235.

3.5.2 Change of patience parameter (-)

We can see how the error evolves in each epoch for 4 different ranges of the test set:

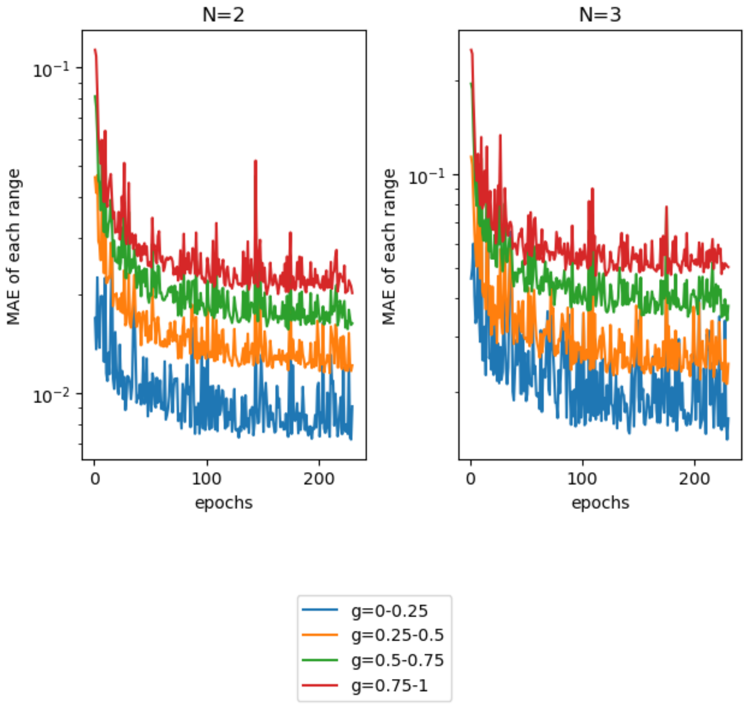

Figure 18: Evolution of the MAE for the N=2 (left) and N=3 (right) test data in 4 different g-ranges along different training epochs. Test set is composed by 10000 instances of N=2 and 2101 instances of N=3. The training procedure has lasted 43 hours.

We can see how it seems to be a constant MAE value for each g-range.

3.5.3 Fixing $\Delta E(g = 0) = 0$ (-)

We have seen how our network seems to understand that $\Delta E(g = 0) = 0$ but we can try to improve the results adding 20000 different instances with this values. We have decided to keep the same proportions of data from N=2 and N=3 so, we have added 16666 instances of N=2 and 3334 instances of N=3. If we represent the evolution of the MAE we can see:

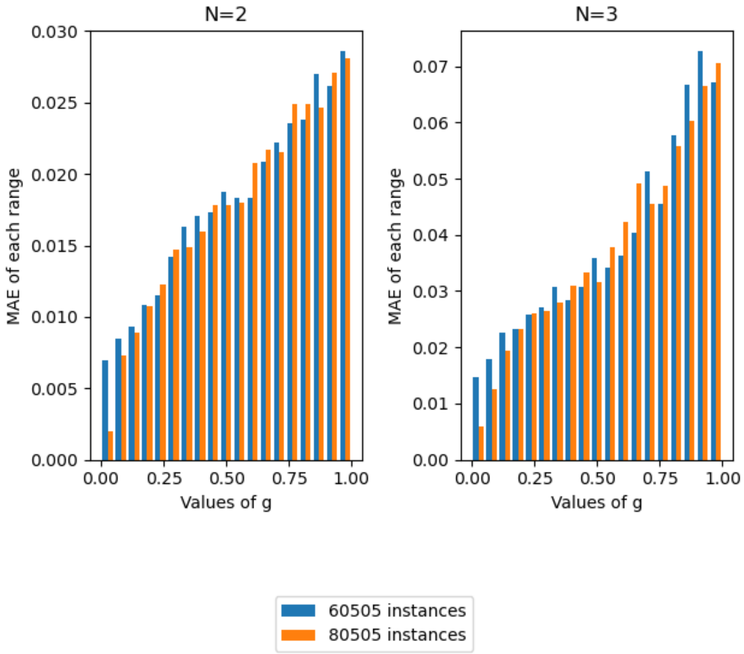

Figure 19: Evolution of the MAE for the N=2 (left) and N=3 (right) test data computed with our second network for 19 different g-ranges in the training procedure with and without $\Delta E(g = 0) = 0$. We have represented the test set and it is composed by 13333 instances of N=2 and 2768 instances of N=3. The training procedure has lasted 4 hours.

We have found the following accuracy parameters:

- N=2: $MAE = 0.013, R^2 = 0.97$.
- N=3: $MAE = 0.03, R^2 = 0.98$.

3.5.4 Addition of training examples g = 0.05,0.26 and 1

As we already have a large data-set of energies for g=0.05,0.26 and 1, we could try to see if adding non-distributed g values we find a better *transition point*, we use:

- 50000 instances of g ∈ [0,1] and N=2 distributed uniformly in g.
- 10505 instances of g ∈ [0,1] and N=3 distributed uniformly in g.
- 50000 instances of g=0.05 and N=2.
- 50000 instances of g=0.26 and N=2.
- 50000 instances of g=1 and N=2.
- 2000 instances of g=0.05 and N=3.

- 2000 instances of g=0.26 and N=3.
- 2000 instances of g=1 and N=3.

TOTAL INSTANCES = 216505. It's relevant to see how we have reduced the relative number of examples to N = 3

Training this amount of data we have found:

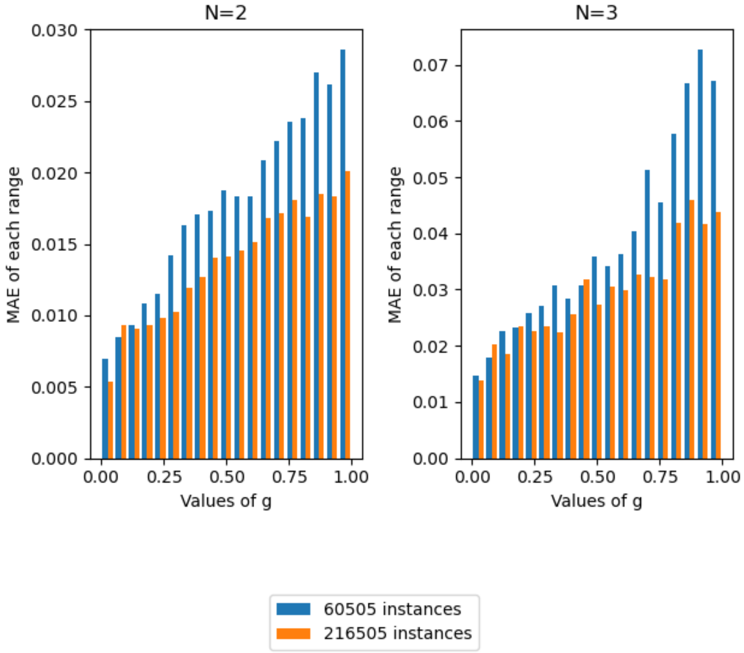

Figure 20: Evolution of the MAE for the N=2 (left) and N=3 (right) test data computed with our second network for 19 different g-ranges i the training procedure. We have represented the test set and it is composed by 40000 instances of N=2 and 3301 instances of N=3. The training procedure has lasted 19 hours.

We have found the following accuracy parameters:

- N=2: $MAE = 0.012, R^2 = 0.98$.
- N=3: $MAE = 0.03, R^2 = 0.98$.

3.5.5 Adding $E_{g=0}$ as a parameter of the network(-)

As we know, second order value energy in quantum mechanics depend on the zero order, we can try to see if adding in the same position of N and g the $E_{g=0}$ value there is some improvement:

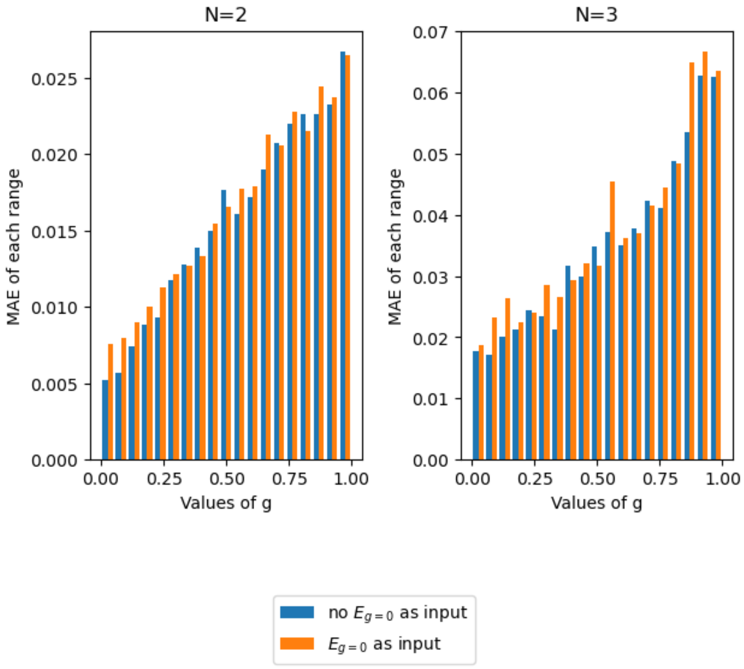

Figure 21: Evolution of the MAE for the N=2 (left) and N=3 (right) test data computed with our second network for 19 different g-ranges in the training procedure with and without $E(g=0)$ as input. We have represented the test set and it is composed by 12101 instances of N=2 and 2000 instances of N=3. The training procedure has lasted 4 hours.

We have found the following parameters:

- N=2: $MAE = 0.017, R^2 = 0.96$.
- N=3: $MAE = 0.04, R^2 = 0.97$.

We conclude it does not improves the predictions.

4 Change of the intensity speckle & different g values for each speckle

Until now we have taken $V_0 = 5$ but, as we have seen in section 3.2.1, there is a problem detecting the transition point. Since we know that the speckle intensity is directly related to the *transition point* (section 2.3), we have decided to change the value to $V_0 = 2$ to reduce this effect. We have generated 10000 speckles in which we have computed the energy for five different g random values between zero and one with the idea of giving to the network the ability of localise the *transition point*.

We cannot use the network of section 3.3 to predict $E_{g=0}$ values due to the fact that the change of the intensity affects directly to the ground state energy. Throughout this section we will be using the exact $E_{g=0}$ values to evaluate the accuracy of ΔE. We segment the data-set into a training data-set of 6000 speckles with five g values (30000 total instances), a validation data-set 2000 speckles and a test data-set of 2000 speckles. We obtain a significant improvement:

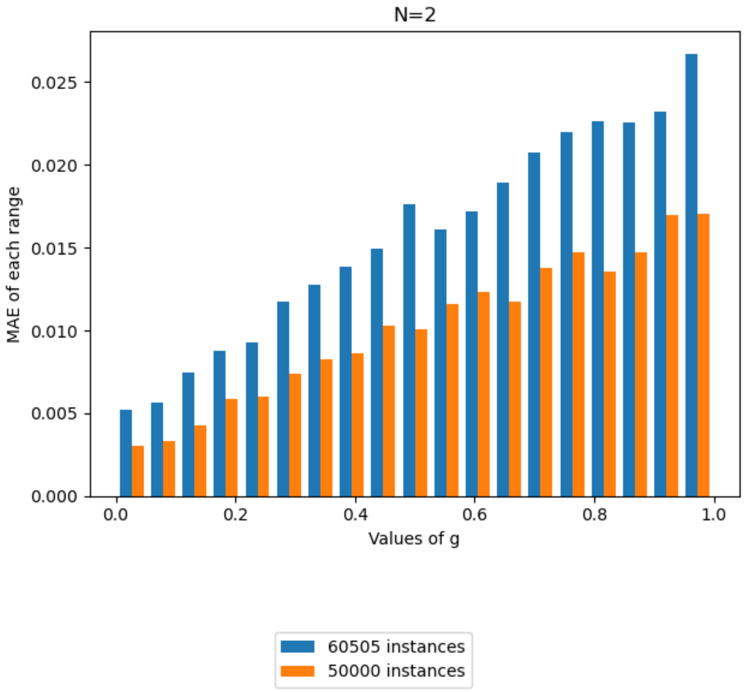

Figure 22: Evolution of the MAE for the N=2 test data computed with network for 19 different g-ranges. We have used 60505 and 50000 instances in each procedure and the 20% is represented (test data). The training procedure has lasted 2 hours.

We have found the following parameters:

- MAE=0.01.
- $R^2 = 0.95$.

Although we know that the previous network also predicts values of N=3, we can see that the tendency with the new data-set with fewer examples improves substantially on the previous one.

4.1 Transfer learning to compute $E_{g=0}$

We have exposed how we cannot use the networks of the section 3.3 to compute $E_{g=0}$, despite this, we can expect that the weights of those networks to be more close to the optimal ones than a random initialization. We can transfer learning from the four models that we use to compute $E_{g=0}$ and make a training procedure with the total 10000 speckle instances.We obtain:

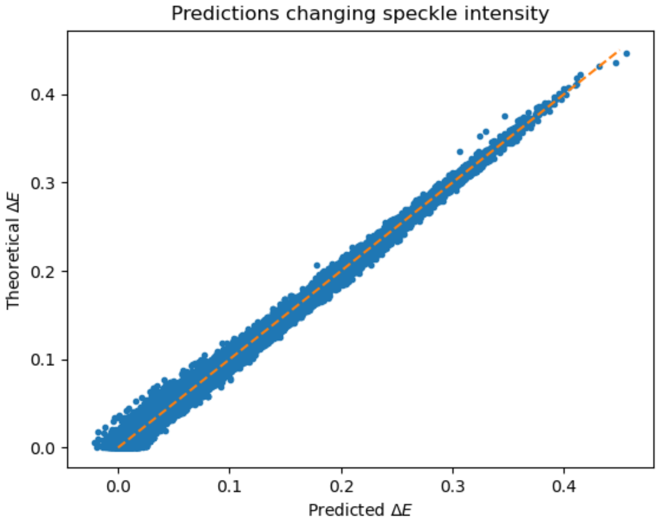

Figure 23: Representation of theoretical vs predicted ΔE with the data set, in which we have changed the intensity speckle parameter. In this figure, we have represented 10000 instances, repeating the same speckle for five different g values.

We have found the following parameters:

- $MAE = 0.004$.
- $R^2 = 0.995$.

4.2 Future analysis - Normalization approach

We have no more time in this internship to continue improving the results but this work is going to follow the idea of separate the computation of E_g in three different networks:

- **Network used to predict $E_{g=0,N=1}$.** Here we will only have the speckle as input and we will only use this network for one particle. When we want to compute the ground state for many-body systems we will only have to multiply $E_{g=0,N=1}N$. It has already been computed.
- **Network used to predict $E_{g=\infty}$.** Here we will have the speckle and the value N as input.
- **Network used to predict $\frac{E_g - E_{g=0}}{E_{g=\infty} - E_{g=0}}$.** Here we will put g, N and the speckle as input.

Some interesting advances have already been seen with this method, but they will not belong to this work.

5 Conclusions

5.1 Work

We have reaffirmed the usefulness of ANNs to solve non-linear problems. We have reached predictions of ΔE with an error of approximately 0.01 and we have seen how the error of

$E_{g=0}$ predictions is negligible. We still have some points that are not well captured at the higher g range values but, as we have exposed in section 4.2, we have the confidence to obtain the last improvement with other network structures.

5.2 Personal

This work has been an enriching experience that has allowed me to grow professionally, I have learned how to use Keras, Pandas and improve my skills with matplotlib and object-oriented programming; but the most important is how it also allowed me to grow as a person, in my career I had never imagined having the opportunity to meet with a research team so soon, I have learned a lot how teamwork works with some great professionals to whom I can only thank for all their time and dedication.

References

[1] JM Huntley. "Speckle photography fringe analysis: assessment of current algorithms". In: *Applied optics* 28.20 (1989), pp. 4316–4322.

[2] JM Huntley. "Speckle photography fringe analysis: assessment of current algorithms". In: *Applied optics* 28.20 (1989), pp. 4316–4322.

[3] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[4] Michele Modugno. "Collective dynamics and expansion of a Bose-Einstein condensate in a random potential". In: *Physical Review A* 73.1 (2006), p. 013606.

[5] Pere Mujal et al. "Supervised learning of few dirty bosons with variable particle number". In: *arXiv preprint arXiv:2010.03875* (2020).

[6] Michael A Nielsen. *Neural networks and deep learning*. Vol. 25. Determination press San Francisco, CA, 2015. Chap. How the backpropagation algorithm works.

[7] Michael A Nielsen. *Neural networks and deep learning*. Vol. 25. Determination press San Francisco, CA, 2015. Chap. Improving the way neural networks learn.

[8] N Saraceni, S Cantori, and S Pilati. "Scalable neural networks for the efficient learning of disordered quantum systems". In: *Physical Review E* 102.3 (2020), p. 033301.