

Deep Learning: A Tool for Computational Nuclear Physics

Gianina Alina Negoita^{*†}, Glenn R. Luecke[‡], James P. Vary[§], Pieter Maris[§], Andrey M. Shirokov^{¶||},
Ik Jae Shin^{**}, Youngman Kim^{**}, Esmond G. Ng^{††} and Chao Yang^{††}

^{*}Department of Computer Science, Iowa State University, Ames, Iowa, USA

Email: alina@iastate.edu

[†]Horia Hulubei National Institute for Physics and Nuclear Engineering, Bucharest-Magurele 76900, Romania

[‡]Department of Mathematics, Iowa State University, Ames, Iowa, USA

Email: grl@iastate.edu

[§]Department of Physics and Astronomy, Iowa State University, Ames, Iowa, USA

Email: jvary@iastate.edu, pmaris@iastate.edu

[¶]Skobeltsyn Institute of Nuclear Physics, Moscow State University, Moscow 119991, Russia

Email: shirokov@nucl-th.sinp.msu.ru

^{||}Department of Physics, Pacific National University, Khabarovsk 680035, Russia

^{**}Rare Isotope Science Project, Institute for Basic Science, Daejeon 34047, Korea

Email: geniean@ibs.re.kr, ykim@ibs.re.kr

^{††}Lawrence Berkeley National Laboratory, Berkeley, California, USA

Email: egng@lbl.gov, cyang@lbl.gov

Abstract—In recent years, several successful applications of the Artificial Neural Networks (ANNs) have emerged in nuclear physics and high-energy physics, as well as in biology, chemistry, meteorology, and other fields of science. A major goal of nuclear theory is to predict nuclear structure and nuclear reactions from the underlying theory of the strong interactions, Quantum Chromodynamics (QCD). With access to powerful High Performance Computing (HPC) systems, several *ab initio* approaches, such as the No-Core Shell Model (NCSM), have been developed to calculate the properties of atomic nuclei. However, to accurately solve for the properties of atomic nuclei, one faces immense theoretical and computational challenges. The present study proposes a *feed-forward* ANN method for predicting the properties of atomic nuclei like ground state energy and ground state point proton root-mean-square (rms) radius based on NCSM results in computationally accessible basis spaces. The designed ANNs are sufficient to produce results for these two very different observables in ${}^6\text{Li}$ from the *ab initio* NCSM results in small basis spaces that satisfy the theoretical physics condition: independence of basis space parameters in the limit of extremely large matrices. We also provide comparisons of the results from ANNs with established methods of estimating the results in the infinite matrix limit.

Keywords—Nuclear structure of ${}^6\text{Li}$; *ab initio* no-core shell model; ground state energy; point proton root-mean-square radius; artificial neural network.

I. INTRODUCTION

Nuclei are complicated quantum many-body systems, whose inter-nucleon interactions are not known precisely. The goal of *ab initio* nuclear theory is to accurately describe nuclei from the first principles as systems of nucleons that interact by fundamental interactions. With sufficiently precise many-body tools, we learn important features of these interactions, such as the fact that three-nucleon (NNN) interactions are critical for understanding the anomalous long lifetime of ${}^{14}\text{C}$ [1]. With access to powerful High Performance Computing (HPC) systems, several *ab initio* approaches have been developed to study nuclear structure and reactions, such as the No-Core

Shell Model (NCSM) [2], the Green's Function Monte Carlo (GFMC) [3], the Coupled-Cluster Theory (CC) [4], the Hyper-spherical expansion method [5], the Nuclear Lattice Effective Field Theory [6][7], the No-Core Shell Model with Continuum [2] and the NCSM-SS-HORSE approach [8]. These approaches have proven to be successful in reproducing the experimental nuclear spectra for a small fraction of the estimated 7000 nuclei produced in nature.

The *ab initio* theory may employ a high-quality realistic nucleon-nucleon (NN) interaction, which gives an accurate description of NN scattering data and predictions for binding energies, spectra and other observables in light nuclei. Daejeon16 is a NN interaction [9] based on Chiral Effective Field Theory (χEFT), a promising theoretical approach to obtain a quantitative description of the nuclear force from the first principles [10]. This interaction has been designed to describe light nuclei without explicit use of NNN interactions, which require a significant increase of computational resources. It has also been shown that this interaction provides good convergence of many-body *ab initio* NCSM calculations [9].

Properties of ${}^6\text{Li}$ and other nuclei, such as ${}^3\text{H}$, ${}^3\text{He}$, ${}^4\text{He}$, ${}^6\text{He}$, ${}^8\text{He}$, ${}^{10}\text{B}$, ${}^{12}\text{C}$ and ${}^{16}\text{O}$, were investigated using the *ab initio* NCSM approach with the Daejeon16 NN interaction and compared with JISP16 [11] results. The results showed that Daejeon16 provides both improved convergence and better agreement with data than JISP16. These calculations were performed with the code MFDn [12]–[14], a hybrid MPI/OpenMP code for *ab initio* nuclear structure calculations. However, one faces major challenges to approach convergence since, as the basis space increases, the demands on computational resources grow very rapidly.

The present work proposes a *feed-forward* Artificial Neural Network (ANN) method as a different approach for obtaining the properties of atomic nuclei such as the ground state (gs) energy and the ground state (gs) point proton root-mean-square (rms) radius based on results from readily-solved basis spaces. *Feed-forward* ANNs can be viewed as universal

non-linear function approximators [15]. Moreover, ANNs can find solution when algorithmic methods are computationally intensive or do not exist. For this reason, ANNs are considered a more powerful modeling method for mapping complex non-linear input-output problems. The output values of ANNs are obtained by simulating the human learning process from the set of learning examples of the input-output association provided to the network. Additional information about ANNs can be found in [16][17].

Although the gs energy and the gs point proton rms radius are ultimately determined by complicated many-body interactions between the nucleons, the variation of the NCSM calculation results appears to be smooth with respect to the two basis space parameters, $\hbar\Omega$ and $N_{\max}$, where $\hbar\Omega$ is the harmonic oscillator (HO) energy and $N_{\max}$ is the basis truncation parameter. In practice, these calculations are limited and one can not calculate the gs energy or the gs point proton rms radius for very large $N_{\max}$. To obtain the gs energy and the gs point proton rms radius as close as possible to the exact results, the results are extrapolated to the infinite model space. However, it is difficult to construct a simple function with a few parameters to model this type of variation and extrapolate the results to the infinite matrix limit. The advantage of ANN is that it does not need an explicit analytical expression to model the variation of the gs energy or the gs point proton rms radius with respect to $\hbar\Omega$ and $N_{\max}$. The *feed-forward* ANN method is very useful to find the converged result at very large $N_{\max}$.

In recent years, ANNs have been used in many areas of nuclear physics and high-energy physics. In nuclear physics, ANN models have been developed for constructing a model for the nuclear charge radii [18], determination of one and two proton separation energies [19], developing nuclear mass systematics [20], identification of impact parameter in heavy-ion collisions [21]–[23], estimating beta decay half-lives [24] and obtaining potential energy curves [25]. In high-energy physics, ANNs are used routinely in experiments for both online triggers and offline data analysis due to an increased complexity of the data and the physics processes investigated. Both the DIRAC [26] and the H1 [27] experiments used ANNs for triggers. For offline data analysis, ANNs were used or tested for a variety of tasks, such as track and vertex reconstruction (DELPHI experiment [28]), particle identification and discrimination (decay of the Z^0 boson [29]), calorimeter energy estimation and jet tagging. Tevatron experiments used ANNs for the direct measurement of the top quark mass [30] or leptoquark searches [31]. In terms of types of ANNs, the vast majority of applications in nuclear physics and high-energy physics were based on *feed-forward* ANNs, other types of ANNs remaining almost unexplored. An exception is the DELPHI experiment, which used a recurrent ANN for tracking reconstruction [28].

This research presents results for two very different physical observables for ${}^6\text{Li}$, gs energy and gs point proton rms radius, produced with the *feed-forward* ANN method. Theoretical data for ${}^6\text{Li}$ are available from the ab initio NCSM calculations with the MFDn code using the Daejeon16 NN interaction and HO basis spaces up through the cutoff $N_{\max} = 18$. This cutoff is defined for ${}^6\text{Li}$ as the maximum total HO quanta allowed in the Slater determinants forming the basis space less 2 quanta. The dimension of the resulting many-body Hamiltonian matrix is about 2.8 billion at this cutoff. We

return to discussing the many-body HO basis shortly. However, for the training stage of ANN, data up through $N_{\max} = 10$ was used, where the Hamiltonian matrix dimension for ${}^6\text{Li}$ is only about 9.7 million. Comparisons of the results from *feed-forward* ANNs with established methods of estimating the results in the infinite matrix limit are also provided. The paper is organized as follows: In Section II, short introductions to the ab initio NCSM method and ANN's formalism are given. In Section III, our ANN's architecture is presented. Section IV presents the results and discussions of this work. Section V contains our conclusion and future work.

II. THEORETICAL FRAMEWORK

The NCSM is an ab initio approach to the nuclear many-body problem for light nuclei, which solves for the properties of nuclei for an arbitrary NN interaction, preserving all the symmetries. Naturally, the results obtained with this method are limited to the largest computationally feasible basis space. We will show that the ANN method is useful to make predictions at ultra-large basis spaces using available data from NCSM calculations at smaller basis spaces. More discussions on these two methods are presented in each subsection.

A. Ab initio NCSM Method

In the NCSM method, the neutrons and protons (separate species of nucleons) interact independently with each other. The Hamiltonian of A nucleons contains kinetic energy (T_{rel}) and interaction (V) terms

$$H_A = T_{\text{rel}} + V = \frac{1}{A} \sum_{i < j} \frac{(\vec{p}_i - \vec{p}_j)^2}{2m} + \sum_{i < j} V_{ij} + \sum_{i < j < k} V_{ijk} + \dots, \quad (1)$$

where m is the nucleon mass, $\vec{p}_i$ is the momentum of the i -th nucleon, V_{ij} is the NN interaction including the Coulomb interaction between protons and V_{ijk} is the NNN interaction. Higher-body interactions are also allowed and signified by the three dots. The HO center-of-mass (CM) Hamiltonian with a Lagrange multiplier is added to the Hamiltonian above to force the many-body eigenstates to factorize into a CM component times an intrinsic component as in [32]. This way, the spurious CM excited states are pushed up above the physically relevant states, which have the lowest eigenstate of the HO for CM motion.

With the nuclear Hamiltonian specified above in (1), the NCSM solves the A -body Schrödinger equation using a matrix formulation

$$H_A \Psi_A(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_A) = E \Psi_A(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_A), \quad (2)$$

where the A -body wave function is given by a linear combination of Slater determinants ϕ_i

$$\Psi_A(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_A) = \sum_{i=0}^k c_i \phi_i(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_A), \quad (3)$$

and where k is the number of many-body basis states, configurations, in the system. To obtain the exact A -body wave function one has to consider infinite number of configurations, $k = \infty$. However, in practice, the sum is limited to a finite number of configurations determined by $N_{\max}$. The Slater determinant ϕ_i is the antisymmetrized product of single particle wave functions $\phi_\alpha(\vec{r})$, where α stands for the quantum

numbers of a single particle state. A common choice for the single particle wave functions is the HO basis functions. The matrix elements of the Hamiltonian in the many-body HO basis is given by $H_{ij} = \langle \phi_i | \hat{H} | \phi_j \rangle$. For these large and sparse Hamiltonian matrices, the Lanczos method is one possible choice to find the extreme eigenvalues [33].

To be more specific, our limited many-body HO basis is characterized by two basis space parameters: $\hbar\Omega$ and $N_{\max}$, where $\hbar\Omega$ is the HO energy and $N_{\max}$ is the basis truncation parameter. In this approach, all possible configurations with $N_{\max}$ excitations above the unperturbed gs (the HO configuration with the minimum HO energy defined to be the $N_{\max} = 0$ configuration) are considered. Even values of $N_{\max}$ correspond to states with the same parity as the unperturbed gs and are called the “natural” parity states, while odd values of $N_{\max}$ correspond to states with “unnatural” parity.

Due to the strong short-range correlations of nucleons in a nucleus, a large basis space, or model space, one that is often not feasible, is required to achieve convergence. To obtain the gs energy and other observables as close as possible to the exact results one has to choose the largest feasible basis spaces. Next, if numerical convergence is not achieved, which is often the case, the results are extrapolated to the infinite model space. To take the infinite matrix limit, several extrapolation methods have been developed (see, for example, [34]).

B. Artificial Neural Networks

ANNs are powerful tools that can be used for function approximation, classification and pattern recognition, such as finding clusters or regularities in the data. The goal of ANNs is to find a solution efficiently when algorithmic methods are computationally intensive or do not exist. An important advantage of ANNs is the ability to detect complex non-linear input-output relationships. For this reason, ANNs can be viewed as universal non-linear function approximators [15]. Employing ANNs for mapping complex non-linear input-output problems offers a significant advantage over conventional techniques, such as regression techniques, because ANNs do not require explicit mathematical functions.

ANNs are defined as computer algorithms that mimic the human brain, being inspired by biological neural systems. Similar to the human brain, ANNs can perform complex tasks, such as learning, memorization and generalization. They are capable of learning from experience, storing knowledge and then applying this knowledge to make predictions.

A biological neuron has a cell body, a nucleus, dendrites and an axon. Dendrites act as inputs, the axon propagates the signal and the interaction between neurons takes place at synapses. Each synapse has an associated weight. When a neuron ‘fires’, it sends an output through the axon and the synapse to another neuron. Each neuron then collects all the inputs coming from linked neurons and produces an output.

The artificial neuron (AN) is a model of the biological neuron. Figure 1 shows a representation of an AN. Similarly, the AN receives a set of input signals $(x_1, x_2, \dots, x_n)$ from an external source or from another AN. A weight w_i ($i = 1, \dots, n$) is associated with each input signal x_i ($i = 1, \dots, n$). Additionally, each AN that is not in the input layer has another input signal called the *bias* with value 1 and its associated weight b . The AN collects all the input signals and calculates a *net* signal as the weighted sum of all input signals as

$$net = \sum_{i=1}^{n+1} w_i x_i, \quad (4)$$

where $x_{n+1} = 1$ and $w_{n+1} = b$.

Next, the AN calculates and transmits an output signal, y . The output signal is calculated using a function called an *activation* or *transfer* function, which depends on the value of the *net* signal, $y = f(net)$.

input signals

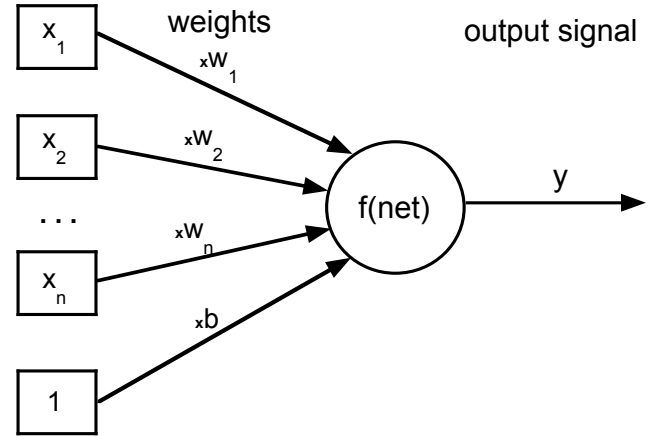


Figure 1. An artificial neuron.

ANNs consist of a number of highly interconnected ANs which are processing units. One simple way to organize ANs is in layers, which gives a class of ANN called multi-layer ANN. ANNs are composed of an input layer, one or more hidden layers and an output layer. The neurons in the input layer receive the data from outside and transmit the data via weighted connections to the neurons in the hidden layer, which, in turn, transmit the data to the next layer. Each layer transmits the data to the next layer. Finally, the neurons in the output layer give the results. The type of ANN, which propagates the input through all the layers and has no *feed-back* loops is called a *feed-forward* multi-layer ANN. For simplicity, throughout this paper we adopt and work with a *feed-forward* ANN. For other types of ANN, see [16][17].

Figure 2 shows an example of a *feed-forward* three-layer ANN. It contains one input layer, one hidden layer and one output layer. The input layer has n ANs, the hidden layer has m ANs and the output layer has p ANs. The connections between the neurons are weighted as follows: v_{ji} are the weights between the input layer and the hidden layer, and w_{kj} are the weights between the hidden layer and the output layer, where $(i = 1, \dots, n)$, $(j = 1, \dots, m)$ and $(k = 1, \dots, p)$. In this example, the input layer has no *activation* function, the hidden layer has *activation* function f and the output layer has *activation* function g . It is also possible to have a different *activation* function for each individual neuron.

The *activation* function in the hidden layer, f , is different from the *activation* function in the output layer, g . For function approximation, a common choice for the *activation* function for the neurons in the hidden layer is a *sigmoid* or *sigmoid-like* function, while the neurons in the output layer have a *linear*

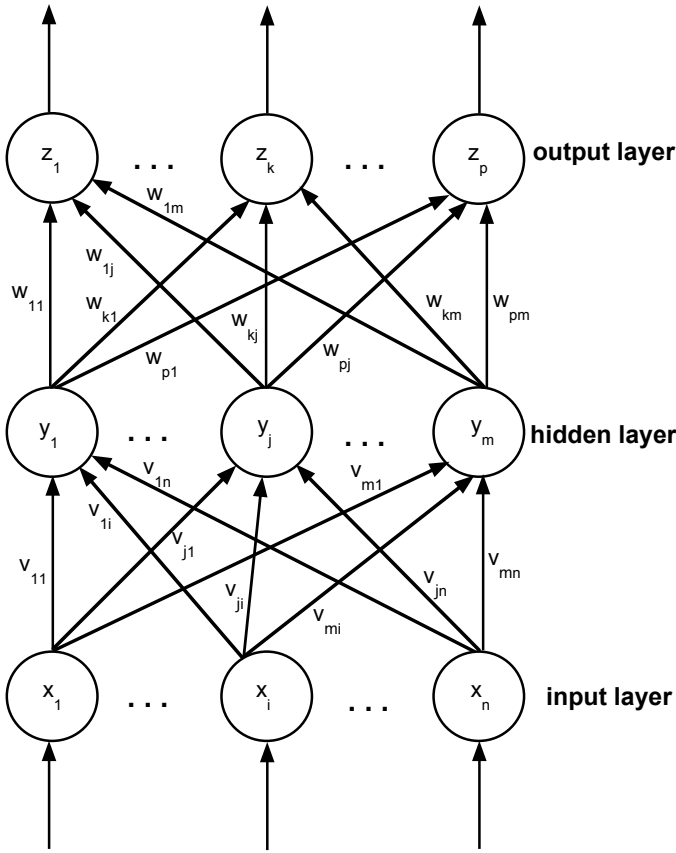


Figure 2. A three-layer ANN.

function:

$$f(x) = \frac{1}{1 + e^{-ax}}, \quad (5)$$

where a is the slope parameter of the *sigmoid* function and

$$g(x) = x. \quad (6)$$

The neurons with *non-linear activation* functions allow the ANN to learn *non-linear* and *linear* relationships between input and output vectors. Therefore, sufficient neurons should be used in the hidden layer in order to get a good function approximation.

In the example shown in Figure 2 and with the notations mentioned above, the network propagates the external signal through the layers producing the output signal z_k at neuron k in the output layer

$$\begin{aligned} z_k &= g(\text{net}_{z_k}) = g\left(\sum_{j=1}^{m+1} w_{kj} f(\text{net}_{y_j})\right) \\ &= g\left(\sum_{j=1}^{m+1} w_{kj} f\left(\sum_{i=1}^{n+1} v_{ji} x_i\right)\right). \end{aligned} \quad (7)$$

The use of an ANN is a two-step process, training and testing stages. In the training stage, the ANN adjusts its weights until an acceptable error level between desired and predicted outputs is obtained. The difference between desired and predicted outputs is measured by the error function, also

called the performance function. A common choice for the error function is *mean square error* (MSE).

There are multiple training algorithms based on various implementations of the *back-propagation* algorithm [35], an efficient method for computing the gradient of error functions. These algorithms compute the *net* signals and outputs of each neuron in the network every time the weights are adjusted as in (7), the operation being called the *forward pass* operation. Next, in the *backward pass* operation, the errors for each neuron in the network are computed and the weights of the network are updated as a function of the errors until the stopping criterion is satisfied. In the testing stage, the trained ANN is tested over new data that was not used in the training process. The predicted output is calculated using (7).

One of the known problems for ANN is overfitting: the error on the training set is within the acceptable limits, but when new data is presented to the network the error is large. In this case, ANN has memorized the training examples, but it has not learned to generalize to new data. This problem can be prevented using several techniques, such as early stopping, regularization, weight decay, *hold-out* method, *m-fold cross-validation* and others.

Early stopping is widely used. In this technique the available data is divided into three subsets: the training set, the validation set and the test set. The training set is used for computing the gradient and updating the network weights and biases. The error on the validation set is monitored during the training process. When the validation error increases for a specified number of iterations, the training is stopped, and the weights and biases at the minimum of the validation error are returned. The test set error is not used during training, but it is used as a further check that the network generalizes well and to compare different ANN models.

Regularization modifies the performance function by adding a term that consists of the mean of the sum of squares of the network weights and biases. However, the problem with regularization is that it is difficult to determine the optimum value for the performance ratio parameter. It is desirable to determine the optimal regularization parameters automatically. One approach to this process is the Bayesian regularization of David MacKay [36]. The Bayesian regularization algorithm updates the weight and bias values according to *Levenberg-Marquardt* [35][37] optimization. It minimizes a linear combination of squared errors and weights and it also modifies the regularization parameters of the linear combination to generate a network that generalizes well. See [36][38] for more detailed discussions of Bayesian regularization.

For further and general background on the ANN and how to prevent overfitting and improve generalization refer to [16][17].

III. ANN DESIGN

The topological structure of ANNs used in this study is presented in Figure 3. The designed ANNs contain one input layer with two neurons, one hidden layer with eight neurons and one output layer with one neuron. The inputs were the basis space parameters: the HO energy, $\hbar\Omega$, and the basis truncation parameter, $N_{\max}$, described in Section II. The desired outputs were the gs energy and the gs point proton rms radius of ${}^6\text{Li}$. An ANN was designed for each desired output: one ANN for gs energy and another ANN for gs point proton rms radius. The optimum number of neurons in the hidden layer was obtained according to a trial and error process.

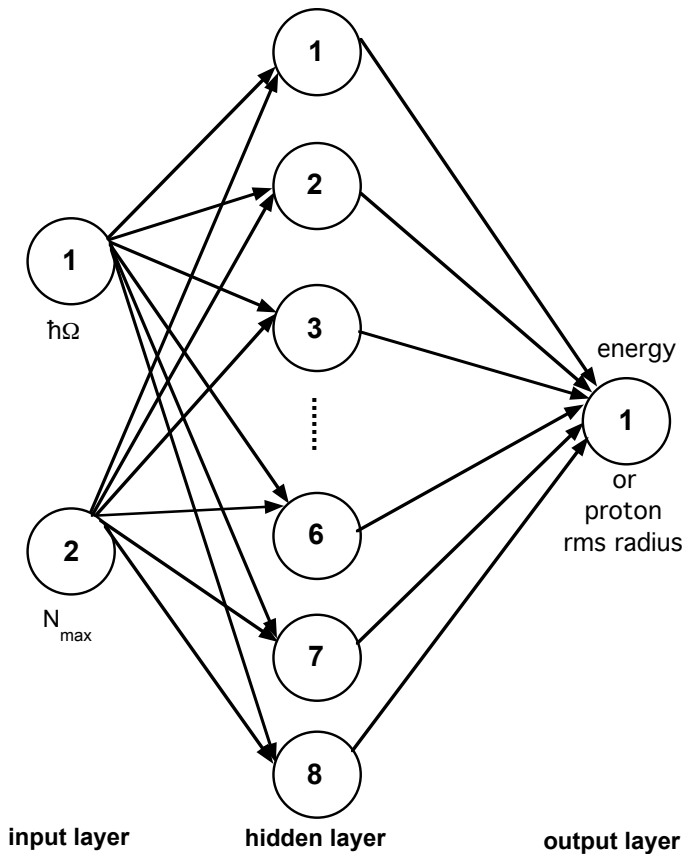


Figure 3. Topological structure of the designed ANN.

The *activation* function employed for the hidden layer was a widely-used form, the *hyperbolic tangent sigmoid* function

$$f(x) = \text{tansig}(x) = \frac{2}{(1 + e^{-2x})} - 1, \quad (8)$$

where x is the input value of the hidden neuron and $f(x)$ is the output of the hidden neuron. *tansig* is mathematically equivalent to the *hyperbolic tangent function*, *tanh*, but it improves network functionality because it runs faster than *tanh*. It has been proven that one hidden layer and *sigmoid*-like *activation* function in this layer are sufficient to approximate any continuous real function, given sufficient number of neurons in the hidden layer [39].

MATLAB software v9.2.0 (R2017a) with *Neural Network Toolbox* was used for the implementation of this work. As mentioned before in Section I, the data set for ${}^6\text{Li}$ was taken from the *ab initio* NCSM calculations with the MFDn code using the Daejeon16 NN interaction [9] and basis spaces up through $N_{\max} = 18$. However, only the data with even $N_{\max}$ values corresponding to “natural” parity states and up through $N_{\max} = 10$ was used for the training stage of the ANN. The training data was limited to $N_{\max} = 10$ and below since future applications to heavier nuclei will likely not have data at higher $N_{\max}$ values due to exponential increase in the matrix dimension. This $N_{\max} \leq 10$ data set was randomly divided into two separate sets using the *dividerand* function in MATLAB: 85% for the training set and 15% for the testing set. A *back-propagation* algorithm with Bayesian regularization

with MSE performance function was used for ANN training. Bayesian regularization does not require a validation data set.

For function approximation, Bayesian regularization provides better generalization performance than early stopping in most cases, but it takes longer to converge. The performance improvement is more noticeable when the data set is small because Bayesian regularization does not require a validation data set, leaving more data for training. In MATLAB, Bayesian regularization has been implemented in the function *trainbr*. When using *trainbr*, it is important to train the network until it reaches convergence. In this study, the training process is stopped if: (1) it reaches the maximum number of iterations, 1000; (2) the performance has an acceptable level; (3) the estimation error is below the target; or (4) the Levenberg-Marquardt adjustment parameter μ becomes larger than 10^{10} . A good typical indication for convergence is when the maximum value of μ has been reached. During training, one can choose to show the Neural Network Training tool (nntraintool) GUI in MATLAB to monitor the training progress. Figure 4 illustrates a training example as it appears in nntraintool.

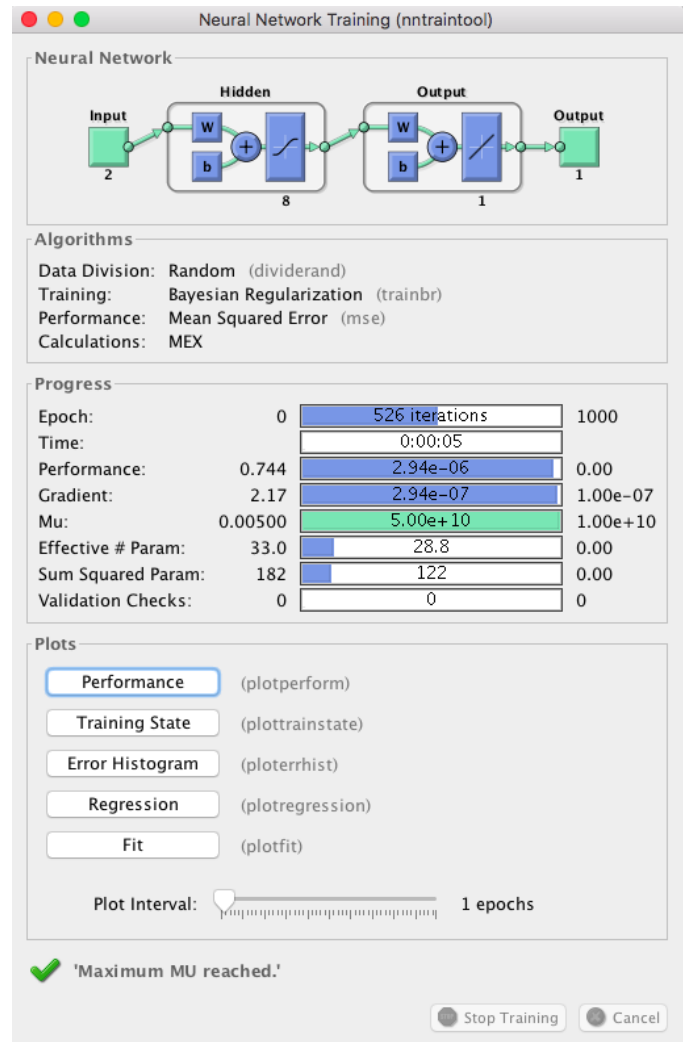


Figure 4. Neural Network Training tool (nntraintool) in MATLAB.

Note the ANN architecture view and the training stopping parameters with their ranges.

IV. RESULTS AND DISCUSSIONS

Every ANN creation and initialization function starts with different initial conditions, such as initial weights and biases, and different division of the training, validation, and test data sets. These different initial conditions can lead to very different solutions for the same problem. Moreover, it is also possible to fail in obtaining realistic solutions with ANNs for certain initial conditions. For this reason, it is a good idea to train several networks to ensure that a network with good generalization is found. Furthermore, by retraining each network, one can verify a robust network performance.

Figure 5 shows the training procedure of 100 ANNs with architecture mentioned in Section III using the *trainbr* function for Bayesian regularization. Each ANN is trained starting from different initial weights and biases, and with different division for the training and test data sets. To ensure good generalization, each ANN is retrained 5 times.

```

1 net = fitnet(8, 'trainbr');
2 net.performFcn = 'mse';
3 numNN = 100;
4 numNNr = 5;
5 NN = cell(numNNr, numNN);
6 trace = cell(numNNr, numNN);
7 perfs = zeros(numNNr, numNN);
8 % train numNN ANNs
9 for i = 1:numNN
10     % retrain each ANN numNNr times
11     for j = 1:numNNr
12         [NN{j}{i}, trace{j}{i}] = train(net, x, t);
13         y2 = NN{j}{i}(x2);
14         perfs(j, i) = perform(NN{j}{i}, t2, y2);
15         net = NN{j}{i};
16     end
17     % reinitialize initial weights and biases
18     net = init(net);
19 end
20 minPerf = min(perfs(:))
21 [rowMin, colMin] = find(perfs == minPerf)
22 net = NN{rowMin}{colMin};
23 tr = trace{rowMin}{colMin};

```

Figure 5. Training 100 ANNs and retraining each ANN 5 times to find the best generalization.

The performance function, such as MSE, measures how well ANN can predict data, i.e., how well ANN can be generalized to new data. The test data sets are a good measure of generalization for ANNs since they are not used in training. A small performance function on the test data set indicates an ANN with good performance was found. In this work, the ANN with the lowest performance on the test data set is chosen to make future predictions.

Using the methodology described above, two ANNs are chosen to predict the gs energy and the gs point proton rms radius. The ANN prediction results for the gs energies and gs proton rms radii of ${}^6\text{Li}$ are presented in detail in this section. Comparison with the ab initio NCSM calculation results is also provided for the available data at $N_{\max} = 12 - 18$.

Figure 6 presents the gs energy of ${}^6\text{Li}$ as a function of the HO energy, $\hbar\Omega$, at selected values of the basis truncation parameter, $N_{\max}$. The dashed curves connect the NCSM calculation results using the Daejeon16 NN interaction for $N_{\max} = 2 - 10$, in increments of 2 units, used for ANN training and testing. The solid curves link the ANN prediction results for $N_{\max} = 12 - 70$. The sequence from $N_{\max} = 12 - 30$ is in increments of 2 units, while the sequence from

$N_{\max} = 30 - 70$ is in increments of 10 units. The lowest horizontal line corresponds to $N_{\max} = 70$ and represents the nearly converged result predicted by ANN. Convergence is defined as independence of both basis space parameters, $\hbar\Omega$ and $N_{\max}$. The convergence pattern shows a reduction in the spacing between successive curves and flattening of the curves as $N_{\max}$ increases. The gs energy provided by the ANN decreases monotonically with increasing $N_{\max}$ at all values of $\hbar\Omega$. This demonstrates that the ANN is successfully simulating what is expected from theoretical physics. That is, in theoretical physics the energy variational principle requires that the gs energy behaves as a non-increasing function of increasing matrix dimensionality at fixed $\hbar\Omega$ and, furthermore, matrix dimension increases with increasing $N_{\max}$.

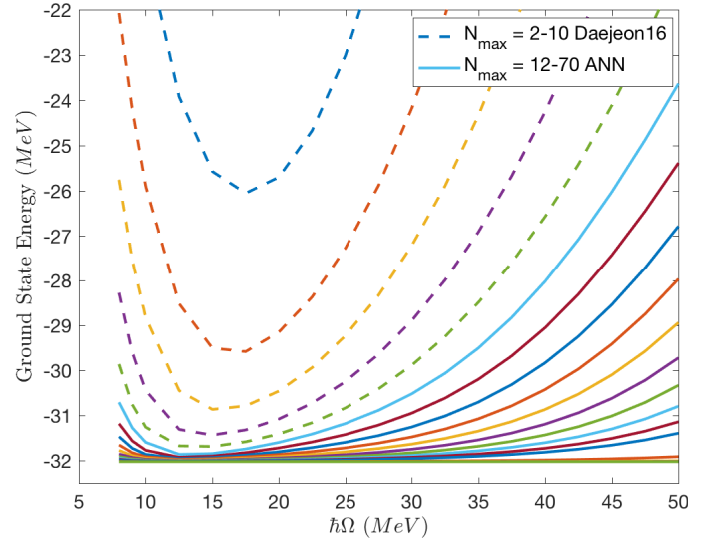


Figure 6. Calculated and predicted gs energy of ${}^6\text{Li}$ as a function of $\hbar\Omega$ at selected $N_{\max}$ values.

To illustrate the ANN prediction accuracy, the NCSM calculation results and the corresponding ANN prediction results of the gs energy of ${}^6\text{Li}$ are presented in Figure 7 as a function of $\hbar\Omega$ at $N_{\max} = 12, 14, 16$, and 18. The dashed curves connect the NCSM calculation results using the Daejeon16 NN interaction and the solid curves link the ANN prediction results. The nearly converged result predicted by ANN is also shown above the horizontal axis at $N_{\max} = 70$. Figure 7 shows good agreement between the calculated NCSM results and the ANN predictions up through $N_{\max} = 18$. Actual NCSM results always converged from above towards the exact result and become increasingly independent of the basis space parameters, $\hbar\Omega$ and $N_{\max}$. That the ANN result is essentially a flat line at $N_{\max} = 70$ and that the curves preceding it form an increasingly dense pattern approaching $N_{\max} = 70$ both provide indications that the ANN is producing a valid estimate of the converged gs energy.

The gs rms radii provide a very different quantity from NCSM results as they are found to be more slowly convergent than the gs energies and they are not monotonic. Figure 8 presents the calculated gs point proton rms radius of ${}^6\text{Li}$ as a function of $\hbar\Omega$ at selected values of $N_{\max}$. The dashed curves connect the NCSM calculation results using the Daejeon16 NN interaction up through $N_{\max} = 10$, while the solid curves link the ANN prediction results above $N_{\max} = 10$. The highest

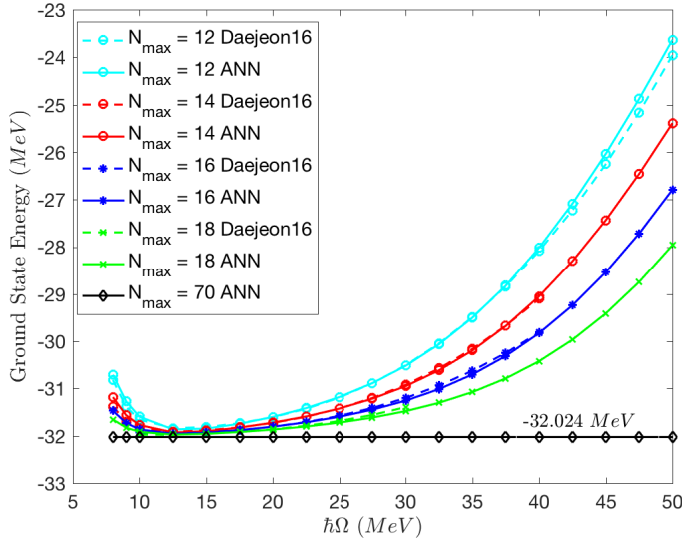


Figure 7. Comparison of the NCSM calculated and the corresponding ANN predicted gs energy values of ${}^6\text{Li}$ as a function of $\hbar\Omega$ at $N_{\max} = 12, 14, 16$, and 18 . The lowest horizontal line corresponds to the ANN nearly converged result at $N_{\max} = 70$.

curve corresponds to $N_{\max} = 90$ and successively lower curves are obtained with $N_{\max}$ decreased by 10 units until the $N_{\max} = 30$ curve and then by 2 units for each lower $N_{\max}$ curve. The rms radius converges monotonically from below for most of the $\hbar\Omega$ range shown. More importantly, the rms radius shows the anticipated convergence to a flat line accompanied by an increasing density of lines with increasing $N_{\max}$. These are the signals of convergence that we anticipate based on experience in limited basis spaces and on general theoretical physics grounds.

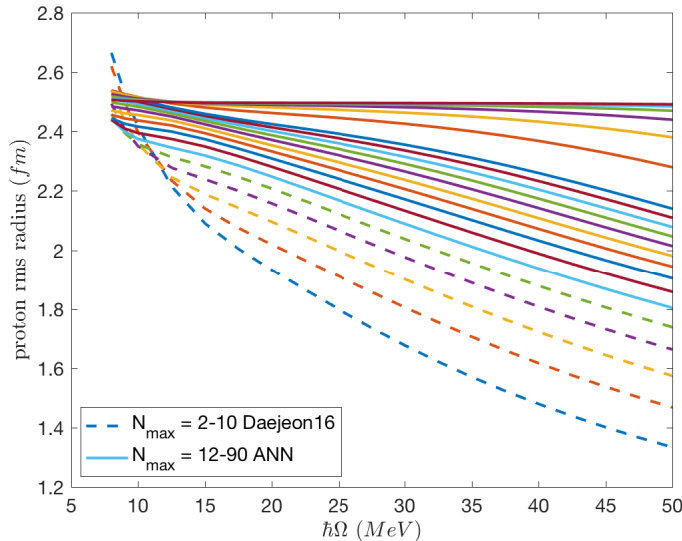


Figure 8. Calculated and predicted gs point proton rms radius of ${}^6\text{Li}$ as a function of $\hbar\Omega$ at selected $N_{\max}$ values.

The NCSM calculated values and the corresponding prediction values of the gs point proton rms radius of ${}^6\text{Li}$ are presented in Figure 9 for $N_{\max} = 12, 14, 16$, and 18 . The dashed curves link the NCSM calculation results using the

Daejeon16 NN interaction and the solid curves connect the ANN prediction results. As seen in this figure, the ANN predictions are in good agreement with the NCSM calculations, showing the efficacy of the ANN method.

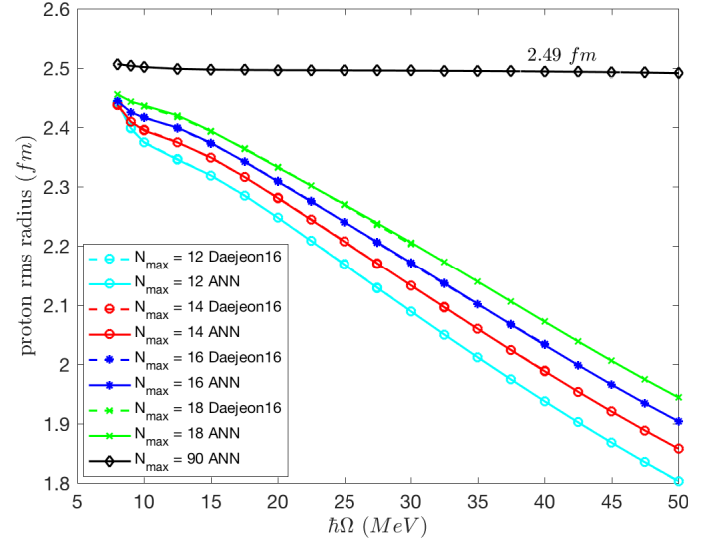


Figure 9. Comparison of the NCSM calculated and the corresponding ANN predicted gs point proton rms radius values of ${}^6\text{Li}$ as a function of $\hbar\Omega$ for $N_{\max} = 12, 14, 16$, and 18 . The highest curve corresponds to the ANN nearly converged result at $N_{\max} = 90$.

Table I presents the nearly converged ANN predicted results for the gs energy and the gs point proton rms radius of ${}^6\text{Li}$. As a comparison, the gs energy results from the current best theoretical upper bounds at $N_{\max} = 10$ and $N_{\max} = 18$ and from the Extrapolation B (Extrap B) method [34] at $N_{\max} \leq 10$ are provided. Similar to the ANN prediction, the Extrap B result arises when using all available results through $N_{\max} = 10$. The ANN prediction for the gs energy is below the best upper bound, found at $N_{\max} = 18$, which is about 85 KeV lower than the Extrap B result.

There is no extrapolation available for the rms radius, but we quote in Table I the estimated result by the *crossover-point* method [40] to be ~ 2.40 fm. The *crossover-point* method takes the value at $\hbar\Omega$ in the table of rms radii results through $N_{\max} = 10$, which produces an rms radius result that is roughly independent of $N_{\max}$.

TABLE I. COMPARISON OF THE ANN PREDICTED RESULTS WITH RESULTS FROM THE CURRENT BEST UPPER BOUNDS AND FROM OTHER ESTIMATION METHODS.

Observable	Upper Bound $N_{\max} = 10$	Upper Bound $N_{\max} = 18$	Estimation ^a $N_{\max} \leq 10$	ANN $N_{\max} \leq 10$
gs energy (MeV)	-31.688	-31.977	-31.892	-32.024
gs rms radius (fm)	—	—	2.40	2.49

^a The Extrap B method [34] for the gs energy and the crossover-point method [40] for the gs point proton rms radius

It is clearly seen from Figures 7 and 9 above that the ANN method results are consistent with the NCSM calculation results using the Daejeon16 NN interaction at $N_{\max} = 12, 14, 16$, and 18 . Table I also shows that ANN's results are consistent with the best available upper bound in the case of the gs energy. The ANN's prediction for the converged rms radius is slightly larger than the result from the *crossover-point*

method and more consistent with the trends visible in Figure 9 at the higher $N_{\max}$ values. To measure the performance of ANNs, MSE for the training subsets up through $N_{\max} = 10$, as well as on the second test set for data at $N_{\max} = 12, 14, 16$, and 18, are provided in Table II.

TABLE II. THE MSE PERFORMANCE FUNCTION VALUES ON THE TRAINING AND TESTING DATA SETS AND ON THE $N_{\max} = 12, 14, 16$, AND 18 DATA SET.

Data Set	Whole Set $N_{\max} \leq 10$	Training Set $N_{\max} \leq 10$	Testing Set ₁ $N_{\max} \leq 10$	Testing Set ₂ $N_{\max} = 12 - 18$
gs energy (MeV)	4.86×10^{-4}	5.04×10^{-4}	3.80×10^{-4}	0.0072
gs rms radius (fm)	7.88×10^{-7}	4.49×10^{-7}	2.74×10^{-6}	9.24×10^{-7}

The small values of the performance function in Table II above indicate that ANNs with good generalizations were found to predict the results.

V. CONCLUSION AND FUTURE WORK

Feed-forward ANNs were used to predict the properties of the ${}^6\text{Li}$ nucleus such as the gs energy and the gs point proton rms radius. The advantage of the ANN method is that it does not need any mathematical relationship between input and output data. The architecture of ANNs consisted of three layers: two neurons in the input layer, eight neurons in the hidden layer and one neuron in the output layer. An ANN was designed for each output.

The data set from the ab initio NCSM calculations using the Daejeon16 NN interaction and basis spaces up through $N_{\max} = 10$ was divided into two subsets: 85% for the training set and 15% for the testing set. Bayesian regularization was used for training and doesn't require a validation set.

The designed ANNs were sufficient to produce results for these two very different observables in ${}^6\text{Li}$ from the ab initio NCSM. The gs energy and the gs point proton rms radius showed good convergence patterns and satisfy the theoretical physics condition, independence of basis space parameters in the limit of extremely large matrices. Comparisons of the results from ANNs with established methods of estimating the results in the infinite matrix limit are also provided. By these measures, ANNs are seen to be successful for predicting the results of ultra-large basis spaces, spaces too large for direct many-body calculations.

As future work, more Li isotopes such as ${}^7\text{Li}$, ${}^8\text{Li}$ and ${}^9\text{Li}$ will be investigated using the ANN method and the results will be compared with results from improved extrapolation methods currently under development.

ACKNOWLEDGMENT

This work was supported by the Department of Energy under Grant Nos. DE-FG02-87ER40371 and DESC000018223 (SciDAC-4/NUCLEI). The work of A.M.S. was supported by the Russian Science Foundation under Project No. 16-12-10048. Computational resources were provided by the National Energy Research Scientific Computing Center (NERSC), which is supported by the Office of Science of the U.S. DOE under Contract No. DE-AC02-05CH11231. Personnel time for this project was also supported by Iowa State University.

REFERENCES

[1] P. Maris et al., "Origin of the Anomalous Long Lifetime of ${}^{14}\text{C}$," Physical Review Letters, vol. 106, no. 20, May 2011, pp. 202502–202505, DOI: 10.1103/PhysRevLett.106.202502.

[2] B. R. Barrett, P. Navrátil, and J. P. Vary, "Ab Initio No Core Shell Model," Progress in Particle and Nuclear Physics, vol. 69, Mar 2013, pp. 131–181, DOI: 10.1016/j.ppnp.2012.10.003, ISSN: 0146-6410.

[3] S. C. Pieper and R. B. Wiringa, "Quantum Monte Carlo Calculations of Light Nuclei," Annual Review of Nuclear and Particle Science, vol. 51, no. 1, Dec 2001, pp. 53–90, DOI: 10.1146/annurev.nucl.51.101701.132506.

[4] K. Kowalski, D. J. Dean, M. Hjorth-Jensen, T. Papenbrock, and P. Piecuch, "Coupled Cluster Calculations of Ground and Excited States of Nuclei," Physical Review Letters, vol. 92, no. 13, Apr 2004, pp. 132501–132504, DOI: 10.1103/PhysRevLett.92.132501.

[5] W. Leidemann and G. Orlandini, "Modern Ab Initio Approaches and Applications in Few-Nucleon Physics with $A \geq 4$," Progress in Particle and Nuclear Physics, vol. 68, Jan 2013, pp. 158–214, DOI: 10.1016/j.ppnp.2012.09.001, ISSN: 0146-6410.

[6] D. Lee, "Lattice Simulations for Few- and Many-Body Systems," Progress in Particle and Nuclear Physics, vol. 63, no. 1, July 2009, pp. 117–154, DOI: 10.1016/j.ppnp.2008.12.001, ISSN: 0146-6410.

[7] E. Epelbaum, H. Krebs, D. Lee, and U. G. Meißner, "Ab Initio Calculation of the Hoyle State," Physical Review Letters, vol. 106, no. 19, May 2011, pp. 192501–192504, DOI: 10.1103/PhysRevLett.106.192501.

[8] A. M. Shirokov, A. I. Mazur, I. A. Mazur, and J. P. Vary, "Shell Model States in the Continuum," Physical Review C, vol. 94, no. 6, Dec 2016, pp. 064320–064323, DOI: 10.1103/PhysRevC.94.064320.

[9] A. Shirokov et al., "N3LO NN Interaction Adjusted to Light Nuclei in ab Exitu Approach," Physics Letters B, vol. 761, Oct 2016, pp. 87–91, DOI: 10.1016/j.physletb.2016.08.006, ISSN: 0370-2693.

[10] R. Machleidt and D. Entem, "Chiral Effective Field Theory and Nuclear Forces," Physics Reports, vol. 503, no. 1, June 2011, pp. 1–75, DOI: 10.1016/j.physrep.2011.02.001, ISSN: 0370-1573.

[11] A. Shirokov, J. Vary, A. Mazur, and T. Weber, "Realistic Nuclear Hamiltonian: Ab Exitu Approach," Physics Letters B, vol. 644, no. 1, Jan 2007, pp. 33–37, DOI: 10.1016/j.physletb.2006.10.066, ISSN: 0370-2693.

[12] P. Sternberg et al., "Accelerating Configuration Interaction Calculations for Nuclear Structure," in Proceedings of the 2008 ACM/IEEE Conference on Supercomputing – International Conference for High Performance Computing, Networking, Storage and Analysis (SC 2008) Nov. 15–21, 2008, Austin, TX, USA. IEEE, Nov 2008, pp. 1–12, DOI: 10.1109/SC.2008.5220090, ISSN: 2167-4329, ISBN: 978-1-4244-2834-2.

[13] P. Maris, M. Sosonkina, J. P. Vary, E. Ng, and C. Yang, "Scaling of Ab-initio Nuclear Physics Calculations on Multicore Computer Architectures," Procedia Computer Science, vol. 1, no. 1, May 2010, pp. 97–106, ICCS 2010, DOI: 10.1016/j.procs.2010.04.012, ISSN: 1877-0509.

[14] H. M. Aktulga, C. Yang, E. G. Ng, P. Maris, and J. P. Vary, "Improving the Scalability of a Symmetric Iterative Eigensolver for Multi-core Platforms," Concurrency and Computation: Practice and Experience, vol. 26, no. 16, Nov 2014, pp. 2631–2651, DOI: 10.1002/cpe.3129, ISSN: 1532-0634.

[15] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks are Universal Approximators," Neural Networks, vol. 2, no. 5, Mar 1989, pp. 359–366, DOI: 10.1016/0893-6080(89)90020-8, ISSN: 0893-6080.

[16] C. M. Bishop, Neural Networks for Pattern Recognition. Oxford University Press, 1995, ISBN: 978-0198538646.

[17] S. Haykin, Neural Networks: A Comprehensive Foundation. Prentice-Hall Inc., 1999, Englewood Cliffs, NJ, USA, ISBN: 978-0132733502.

[18] S. Akkoyun, T. Bayram, S. O. Kara, and A. Sinan, "An Artificial Neural Network Application on Nuclear Charge Radii," Journal of Physics G: Nuclear and Particle Physics, vol. 40, no. 5, Mar 2013, pp. 055106–055112, DOI: 10.1088/0954-3889/40/5/055106.

[19] S. Athanassopoulos, E. Mavrommatis, K. A. Gernoth, and J. W. Clark, "One and two Proton Separation Energies from Nuclear Mass Systematics Using Neural Networks," Sep 2005, arXiv:0509075 [nucl-th].

[20] S. Athanassopoulos, E. Mavrommatis, K. Gernoth, and J. Clark, "Nuclear Mass Systematics Using Neural Networks," Nuclear

- Physics A, vol. 743, no. 4, Nov 2004, pp. 222–235, DOI: 10.1016/j.nuclphysa.2004.08.006, ISSN: 0375-9474.
- [21] C. David, M. Freslier, and J. Aichelin, “Impact Parameter Determination for Heavy-ion Collisions by use of a Neural Network,” *Physical Review C*, vol. 51, no. 3, Mar 1995, pp. 1453–1459, DOI: 10.1103/PhysRevC.51.1453.
 - [22] S. A. Bass, A. Bischoff, J. A. Maruhn, H. Stöcker, and W. Greiner, “Neural Networks for Impact Parameter Determination,” *Physical Review C*, vol. 53, no. 5, May 1996, pp. 2358–2363, DOI: 10.1103/PhysRevC.53.2358.
 - [23] F. Haddad et al., “Impact Parameter Determination in Experimental Analysis Using a Neural Network,” *Physical Review C*, vol. 55, no. 3, Mar 1997, pp. 1371–1375, DOI: 10.1103/PhysRevC.55.1371.
 - [24] N. Costiris, E. Mavrommatis, K. A. Gernoth, and J. W. Clark, “A Global Model of β^- -Decay Half-Lives Using Neural Networks,” Jan 2007, arXiv:0701096 [nucl-th].
 - [25] S. Akkoyun, T. Bayram, S. , and N. Yildiz, “Consistent Empirical Physical Formula for Potential Energy Curves of 38–66Ti Isotopes by Using Neural Networks,” *Physics of Particles and Nuclei Letters*, vol. 10, no. 6, Nov 2013, pp. 528–534, DOI: 10.1134/S1547477113060022, ISSN: 1531-8567.
 - [26] “DIRAC Experiment,” URL: <http://www.cern.ch/DIRAC> [accessed: 2018-01-17].
 - [27] “H1 Experiment,” URL: <http://www-h1.desy.de> [accessed: 2018-01-17].
 - [28] R. Frühwirth, “Selection of Optimal Subsets of Tracks with a Feed-back Neural Network,” *Computer Physics Communications*, vol. 78, no. 1–2, Dec 1993, pp. 23–28, DOI: 10.1016/0010-4655(93)90140-8, ISSN: 0010-4655.
 - [29] P. Abreu et al., “Classification of the Hadronic Decays of the Z^0 Into b and c Quark Pairs Using a Neural Network,” *Physics Letters B*, vol. 295, no. 3–4, Dec 1992, pp. 383–395, DOI: 10.1016/0370-2693(92)91580-3, ISSN: 0370-2693.
 - [30] S. Abachi et al., “Direct Measurement of the top Quark Mass,” *Physical Review Letters*, vol. 79, no. 7, Aug 1997, pp. 1197–1202, DOI: 10.1103/PhysRevLett.79.1197.
 - [31] B. Abbott et al., “Search for Scalar Leptoquark Pairs Decaying to Electrons and Jets in $\bar{p}p$ Collisions,” *Physical Review Letters*, vol. 79, no. 22, Dec 1997, pp. 4321–4326, DOI: 10.1103/PhysRevLett.79.4321.
 - [32] D. H. Gloeckner and R. D. Lawson, “Spurious Center-of-Mass Motion,” *Physics Letters B*, vol. 53, no. 4, Dec 1974, pp. 313–318, DOI: 10.1016/0370-2693(74)90390-6.
 - [33] B. N. Parlett, *The Symmetric Eigenvalue Problem*. Classics in Applied Mathematics, 1998, DOI: 10.1137/1.9781611971163, ISBN: 978-0-89871-402-9.
 - [34] P. Maris, J. P. Vary, and A. M. Shirokov, “Ab Initio No-Core Full Configuration Calculations of Light Nuclei,” *Physical Review C*, vol. 79, no. 1, Jan 2009, pp. 014 308–014 322, DOI: 10.1103/PhysRevC.79.014308.
 - [35] M. T. Hagan and M. B. Menhaj, “Training Feedforward Networks with the Marquardt Algorithm,” *IEEE Transactions on Neural Networks*, vol. 5, no. 6, Nov 1994, pp. 989–993, DOI: 10.1109/72.329697, ISSN: 1045-9227.
 - [36] D. J. MacKay, “Bayesian Interpolation,” *Neural Computation*, vol. 4, no. 3, May 1992, pp. 415–447, DOI: 10.1162/neco.1992.4.3.415, ISSN: 0899-7667.
 - [37] D. W. Marquardt, “An Algorithm for Least-Squares Estimation of Nonlinear Parameters,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, June 1963, pp. 431–441, SIAM, DOI: 10.1137/0111030, ISSN: 2168-3484.
 - [38] F. D. Foresee and M. T. Hagan, “Gauss-Newton Approximation to Bayesian Learning,” in *Proceedings of the International Joint Conference on Neural Networks*, vol. 3. IEEE, Jun 1997, pp. 1930–1935, DOI: 10.1109/ICNN.1997.614194.
 - [39] G. Cybenko, “Approximation by Superpositions of a Sigmoidal Function,” *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, Dec 1989, pp. 303–314, DOI: 10.1007/BF02551274, ISSN: 1435-568X.
 - [40] S. K. Bogner et al., “Convergence in the No-Core Shell Model with Low-Momentum Two-Nucleon Interactions,” *Nuclear Physics A*, vol. 801, no. 1, Mar 2008, pp. 21–42, DOI: 10.1016/j.nuclphysa.2007.12.008, ISSN: 0375-9474.