Eur. Phys. J. A (2022) 58:186
https://doi.org/10.1140/epja/s10050-022-00839-y

THE EUROPEAN PHYSICAL JOURNAL A

Regular Article - Theoretical Physics

Estimating scattering potentials in inverse problems with Volterra series and neural networks

Gábor Balassa[a]

Institute for Particle and Nuclear Physics, Wigner Research Centre for Physics, 1525 Budapest, Hungary

Received: 8 June 2022 / Accepted: 20 September 2022

Communicated by Raphael-David Lasseri

Abstract Inverse problems often occur in nuclear physics, when an unknown potential has to be determined from the measured cross sections, phase shifts or other observables. In this paper, a data-driven numerical method is proposed to estimate the scattering potentials, using data, that can be measured in scattering experiments. The inversion method is based on the Volterra series representation, and is extended by a neural network structure to describe problems, which require a more robust estimation. The Volterra series method is first used to describe the one-dimensional scattering problem, where the transmission coefficients, and the phase shifts are used as inputs to determine the unknown potentials in the Fourier domain. In the second example the scattering process described by the radial Schrödinger equation is used to estimate the scattering potentials from the energy dependence of the phase shifts, where neural networks are used to describe the scattering problem. At the end, to show the capabilities of the proposed models, real-life data is used to estimate the $^3S_1$ NN potential with the neural network approach from measured phase shifts, where a few percent relative match is obtained between the measured values and the model calculations.

1 Introduction

Inverse problems in low-energy nuclear physics are related to the determination of (generally) local scattering potentials from observables such as scattering phase shifts, transmission coefficients, total- and differential cross sections, spin observables etc. [1–5]. These problems are generally ill-conditioned, and in many cases even ill-defined [6,7] and require a careful examination of the actual problem on which the inversion is done. In low-energy nuclear physics, usually the interaction potential is sought from elastic scattering data, which could be measured in colliding experiments [8]. At high enough energies, where inelastic channels could open, the potentials could also contain imaginary parts, that takes into account the absorption of the incoming flux, thus able to describe non-elasticity in the scattering experiments. Nuclear interactions are generally very complicated in nature, however at low energies it can be described fairly well, with optical potential models, which could take into account the elastic- and inelastic interactions as well [9,10].

There are several methods for estimating the interaction potentials from measured data that can be used in different scenarios. One such possibility is the original Gelfand–Levitan–Marchenko method, which uses the energy dependence of the phase shifts of a specific partial wave, to determine the scattering potential [11]. Another type of inversion is the Newton–Sabatier method, which solves the inversion problem at fixed energy from measured phase shifts of the different partial waves [12]. Apart from the fixed-angular momentum and fixed-energy cases, there also exist mixed case inversion techniques, where the phase shifts of several angular momenta, at different energies are used to obtain the potentials [13].

In this work, two different methods are proposed, which are both able to estimate the interaction potential from measurable data in quantum scattering problems. The first method uses a non-causal Volterra series representation to make a connection between the observables and the potentials in the Fourier domain, while the second method gives a sequential method to estimate the potentials at discrete space-points with neural networks, using only the energy dependent phase shifts at fixed angular momenta. The methods described here are purely data-driven and the identified systems are constrained to an operating range, in which the training has been done. If the differential equations, which govern the inverse systems are known, then it could be possible to analytically express e.g. the Volterra kernels [14],

[a] e-mail: balassa.gabor88@gmail.com (corresponding author)

Published online: 28 September 2022

thus giving a more robust estimation, in a much wider operating range. The inverse system, however is rarely known, and is usually highly nonlinear, in which case it is necessary to constrain ourselves into a well-defined range, where the nonlinearities could be locally modeled.

The paper is organized as follows. In Sect. 2 the necessary mathematical tools are summarized, which contains the Volterra series representation of nonlinear dynamical systems, and the applied neural network model as well. After the short summary, in Sect. 3 the one-dimensional scattering problem is addressed, where a non-causal Volterra representation is used to estimate the scattering potentials in the Fourier domain, using the phase shifts and the transmission coefficients as inputs. In Sect. 4 the low-energy elastic scattering is addressed, where a neural network model is used to describe the interaction potential in low-energy nuclear scattering, from the energy dependent phase shifts as inputs. The proposed neural network model is then used in Sect. 5 to describe the low-energy neutron + proton scattering data, giving an approximate 2% relative match between the measured-, and the obtained phase shifts calculated from the potential after inversion. At the end, Sect. 6 concludes the paper, giving some further possibilities on how to use and extend the proposed methods.

2 Mathematical models

In this section the necessary mathematical tools are summarized, including the causal and non-causal Volterra series, and the multilayer perceptron (MLP) type neural network structures, which will be both used in the later sections. Both mathematical models are able to identify nonlinear dynamical systems, and were used in many engineering applications, where the underlying mathematical models are too complex or not known exactly [15–17].

2.1 Volterra series

In linear system theory, where a physical system can be described by linear differential equations, the system can be characterized by a simple convolution integral, where an input $u(t)$ is convolved with a so-called transfer function $h(t)$, giving the $y(t)$ output of the system as follows [18]:

$$y(t) = \int_{-\infty}^{\infty} h(\tau)u(t-\tau)d\tau, \tag{1}$$

where $y(t)$ is the output of the system, $u(\tau)$ is the input, while $h(\tau)$ is the transfer function. Here, the value t is some parameter e.g. time, position, or any other variable on which the inputs, and outputs could depend. This simple representation can be extended even if the physical system is described by a system of linear differential equations, in which case a transfer matrix can be introduced [19]. In the linear case the behavior and the stability of the system is fully characterized by the transfer function, thus the main problem is to identify the $h(t)$ kernel. In real-life, however, many physical systems are nonlinear in nature, and the simple linear representation is simply not valid. In this case many simplifying assumptions can be made e.g. one could take the best linear approximation (BLA) of a weakly nonlinear system in the least-squares sense [20,21]. The best linear approximation could be enough in many practical applications, however if the nonlinearities are dominant, other methods are necessary to adequately model the nonlinear system. One of these methods is the Volterra series representation of nonlinear dynamical systems [22,23], which is the extension of the linear theory by introducing higher order polynomial terms in the convolutional integral, and can be written as:

$$\begin{aligned} y(t) = h_0 &+ \int_{-\infty}^{\infty} h_1(\tau)u(t-\tau)d\tau \\ &+ \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_2(\tau_1,\tau_2)u(t-\tau_1)u(t-\tau_2)d\tau_1 d\tau_2 + \cdots \\ &\ldots \\ &+ \int_{-\infty}^{\infty}\ldots\int_{-\infty}^{\infty} h_n(\tau_1,\ldots,\tau_n)u(t-\tau_1)\ldots u(t-\tau_n) \\ &\times d\tau_1\ldots d\tau_n, \end{aligned} \tag{2}$$

where $h_n(\tau_1,\ldots,\tau_n)$ is the nth order Volterra kernel of the system. In the most general case the integrals are going from $-\infty$ to ∞, in which case the model is called non-causal. In this case all the future and past values are used to describe the system, however, as it was mentioned before, the parameter t does not have to be time, but can be position or any other variable on which the system depends. If the system is causal, then the integral limits can be changed to $[0,\infty]$, which greatly simplifies the identification problem. As it will be shown later, the non-causal representation is a very useful tool in the addressed problems, which is also shown in [24], where a discrete, non-causal Volterra model is used to describe inverse quantum mechanical problems.

In practical applications, the system is usually discretized and the mathematical model could be cast into a system of linear equations, where the coefficient matrix contains the inputs with different lags, while the unknown parameter vector contains the identifiable kernel functions at the discrete points [25]. The identification of the kernel functions $h_1, h_2, \ldots, h_n$ are quite the cumbersome task, if the order, and the memory of the system is large, in which case alternative methods exist to identify the kernels [26,27]. One particularly interesting

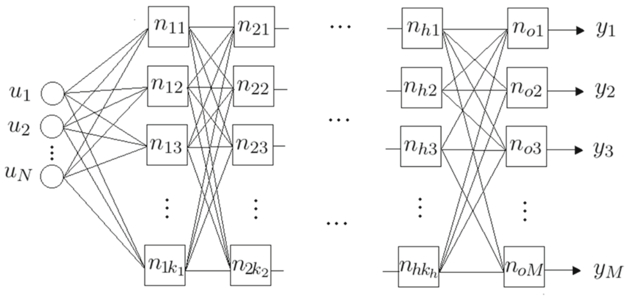

Fig. 1 The schematic view of a multiple input–multiple output, feed-forward, multilayer perceptron model, with h hidden layers

method is the neural network approach, where different neural network structures are used to obtain the nth order Volterra kernels by Taylor expanding the activation functions around the bias values [28].

In the applications in this paper, multiple input and single output (MISO) systems are in focus, in which case the Volterra representation changes to:

$$y(t) = h_0 + \sum_{i=1}^{N} H_1^i\{u_i(t)\} + \sum_{i_1,i_2=1}^{N} H_2^{i_1,i_2}\{u_{i_1}(t)u_{i_2}(t)\} + \dots + \sum_{i_1,\dots,i_N=1}^{N} H_N^{i_1,\dots,i_N}\{u_{i_1}(t),\dots,u_{i_N}(t)\}, \tag{3}$$

where $u_i(t)$ is the ith input variable, while the $H_N\{\cdot\}$ multi-input kernel functions are expressed as:

$$H_N^{i_1,\dots,i_N}\{u_{i_1}(t),\dots,u_{i_N}(t)\} = \int_{-\infty}^{\infty}\dots\int_{-\infty}^{\infty} h_N^{i_1,\dots,i_N}(\tau_1,\dots,\tau_N)u_{i_1}(t-\tau_1)\dots u_{i_N}(t-\tau_N)d\tau_1\dots d\tau_N. \tag{4}$$

As it can be seen, in the MISO case, the inputs can be mixed together giving higher order mixed terms e.g the second order $h_2(\tau_1,\tau_2)u_1(t-\tau_1,t-\tau_2)$, which could be included in the identification process.

2.2 Neural networks

The neural network model is a different approach, which can be used to describe nonlinear dynamical systems. In this case the system is described by many interconnected layers built up by so-called neurons, which could have nonlinear or linear activation functions as well [29,30]. In this paper only feed-forward multilayer perceptron structure is used, therefore in this section, only that will be described in more detail. The schematic view of a fully-connected, feed-forward MLP network can be seen in Fig. 1, where for the sake of generality the multiple input and multiple output (MIMO) case is shown. The number of inputs are set to N, while the number of outputs are set to M. The mapping from $u_1, u_2, \dots, u_N$ to $y_1, y_2, \dots, y_M$ are described by the block structure shown in Fig. 1, where each line corresponds to a weight parameter, and each rectangular block corresponds to a neuron, which is decribed by an activation function and a bias parameter. The first h columns are the so-called hidden layers, while the last layer labeled by o is the output layer. The input–output relationship can be expressed in a simple compact form if one introduces vector and matrix notations, in which case the output vector $\underline{y} = [y_1 y_2 \dots y_M]^T$ can be expressed as:

$$\underline{y} = \underline{N}^o\Big(\underline{b}^o + \underline{\underline{W}}^o\underline{N}^h\Big(\underline{b}^h + \underline{\underline{W}}^h\underline{N}^{h-1}\Big(\dots\dots\underline{N}^1\Big(\underline{b}^1 + \underline{\underline{W}}^1\underline{u}\Big)\Big)\Big)\Big), \tag{5}$$

where $\underline{N}^h = [n_{h1}n_{h2}\dots n_{hk_h}]^T$ is a vector of the activation functions at the hidden-layer h (or at the output layer o), while $\underline{b}^h = [b_1^h b_2^h \dots b_{k_h}^h]^T$ is the vector of bias paremeters, and $\underline{\underline{W}}^h$ is a weight matrix constructed by the elements w_{ij}^h.

This feed-forward structure is usually trained by the back propagation technique [31], which is a very efficient way in calculating the derivatives, that is needed for solving the optimization problem for the unknown weights and biases. A very good property of the neural network approach is its generalization capability, which is the ability to predict the output to previously unseen input data. With a careful construction of the network, very good generalization properties can be achieved [32] for different network structures as well.

3 One-dimensional scattering with a non-causal Volterra model

The first problem, which is addressed with the Volterra formalism will be the one-dimensional quantum scattering problem, where an incoming particle, described by the wave function $\phi(x) = e^{ikx}$ is scattered on a bounded potential $V(x)$, giving a scattered wave $\Psi(x)$. The scattering process can be described by the Lippmann–Schwinger equation [33], which in one-dimension has the following form:

$$\Psi(x) = \phi(x) + \int_{-\infty}^{\infty} dx' G(x,x')\Psi(x'), \tag{6}$$

where $G(x,x')$ is the Green function of the Schrödinger operator in one-dimension [34], k is the wave number, while $\phi(x)$ is the incoming-, and $\Psi(x)$ is the outgoing wave function. Assuming that the potential goes to zero at some finite distance, after discretization the integral equation can be rewritten into a linear system of equations, then can be solved by standard numerical methods. In the inverse problem, the unknown potential is sought from observables, which can be measured in scattering experiments. In [24] the above prob-

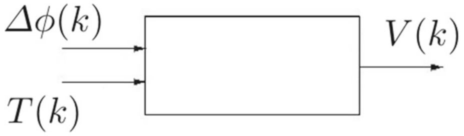

Fig. 2 The block diagram of the inverse system

lem was examined with a non-causal Volterra model using the real and imaginary parts of the scattered wave as inputs of the system. In an experimental setup however, the scattered wave is not accessible and the typical observables are the asymptotic phase shifts and/or transmission/reflection coefficients [35,36]. In this section, a non-causal Volterra system is used to describe the scattering potential in the Fourier domain, using the scattering phase shifts, and transmission coefficients as inputs. The block diagram of the system in question can be summarized in Fig. 2, where the input-output relationship is described by a dynamical system, which is essentially nonlinear. The nonlinear nature of the inverse scattering problem will be more apparent in Sect. 4, where the phase-shift → potential relation can be described by a first order nonlinear differential equation. In this section, a closed form of the dynamical system will not be derived, and a purely data-driven technique is proposed, which is able to describe the inverse problem sketched in Fig. 2. The two inputs of the system are the k-dependent phase-shifts, and transmission coefficients, which is defined as the ratio of the transmitted intensity to the incident intensity:

$$T(k) = \frac{|\Psi_{TR}(x)|^2}{|\Psi_{IN}(x)|^2}, \tag{7}$$

where $T(k)$ is the transmission coefficient, Ψ_{TR} is the transmitted wave function, Ψ_{IN} is the incoming wave function.

An example for the system inputs and outputs can be seen in Fig. 3, where the potential $V(x)$ and the corresponding $\Delta\phi(k)$ and $T(k)$ functions are also shown.

It is apparent from the block diagram in Fig. 2, that the output of the system is not $V(x)$, but instead its Fourier transform $V(k)$, which is a more natural choice if the inputs of the system are all k-dependent quantities. The non-causal nature of the applied Volterra system means, that after discretization, the output $V(k_i)$ depends on the previous $k_{i-1}, k_{i-2}, \dots k_{i-M}$, the current k_i, and the following values $k_{i+1}, \dots k_{i+M}$ as well, which can be summarized as follows:

$$\begin{aligned} V(k_i) = h_0 + H_1^{\Delta\phi}\Big[\Delta\phi[k]\Big] + H_1^T\Big[T[k]\Big] \\ + H_2^{\Delta\phi,T}\Big[\Delta\phi[k], T[k]\Big], \end{aligned} \tag{8}$$

where $\Delta\phi$ is the phase shift, T is the transmission coefficient, $H_1^{\Delta\phi}\Big[\Delta\phi[k]\Big]$ and $H_1^T\Big[T[k]\Big]$ are the Volterra kernels for the single inputs $\Delta\phi$ and T, while $H_2^{\Delta\phi,T}\Big[\Delta\phi[k], T[k]\Big]$ contains the cross-kernels for the two inputs. The Volterra kernels are practically truncated at order N, and with memory M, so that they can be identified with numerical techniques.

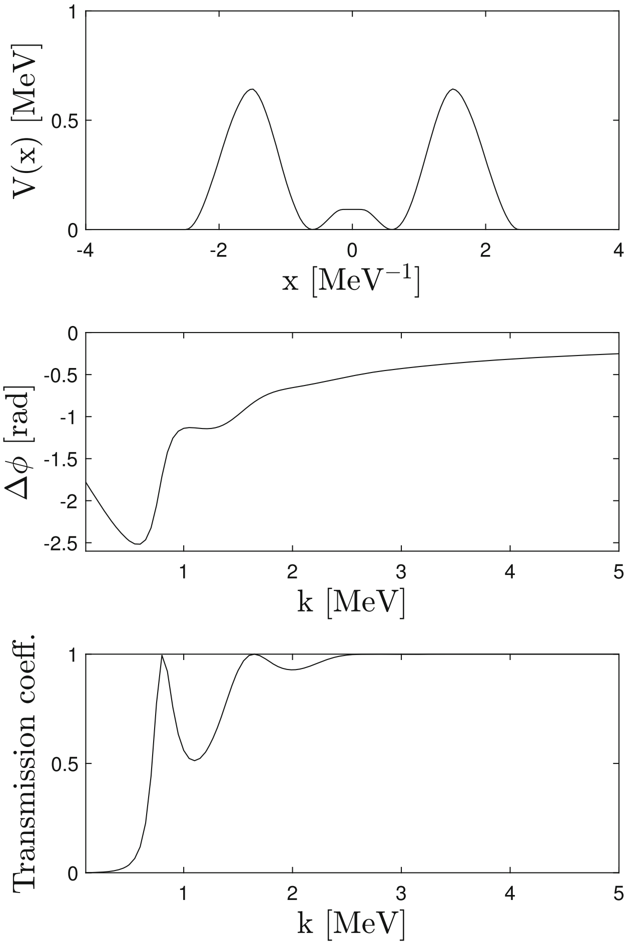

Fig. 3 Typical system inputs (phase shift, transmission coefficient) for a symmetric potential for $k \in [0.1, 5]$ MeV

To identify the kernels, 12,000 training samples have been generated by solving the Lippmann–Schwinger equation, with $V(x) \in [0, 1]$ MeV, when $x \in [-4, 4]$ MeV^{-1}, and $V(x) = 0$, if $|x| > 4$ MeV^{-1}, which will be the operating range of the identified system. The potentials used for training were uniformly generated random samples, in the predefined range, which is a common technique in system identification to excite the system [37]. There are more refined methods for generating excitation signals e.g. random phase multisines [38], which could have been also used in this case as well. In addition, only symmetric potentials were generated so that the Fourier transform of $V(x)$ will be purely real. Here, the $\hbar = c = 1$ natural system of units is used, and the distance is measured in inverse energies. The Fourier transform for the potential has been calculated between $k \in [0, 20]$ MeV, with $\Delta k = 0.2$ MeV, so the Volterra system is discretized into 100 discrete points in momentum space. The choice of this discretization in momentum space corresponds to $\delta x \approx 0.163\ \mathrm{MeV}^{-1}$ in position space, which is approximately the maximum resolution, what we can get after inverse Fourier transforming the obtained $V(k)$ potential.[1] To each wave-number k_i, one Volterra system is identified. This does not take away from the generality of the system, as it is always possible to extend the model by introducing an extra input (in this case the wave number k), so that only one Volterra model is needed to describe the inverse system. It is, however, much simpler to identify separate Volterra systems for each wave number, due to the complicated cross-terms, which would arise otherwise. Identifying the kernels, it turned out that second order kernels, with $M = 35$ memories, and first order cross-terms were sufficient to describe every system, giving a few percent relative error at all of the wave numbers, for which the mean squared error (which is used in the optimization process, and is defined in Eq. 9) can be seen in Fig. 4.

$$MSE_j = \frac{1}{N}\sum_{i=1}^{N}\Big(V_{true}(k_j) - V_{est}(k_j)\Big)^2, \tag{9}$$

where N is the number of samples, V_{true} is the true potential, and V_{est} is the estimated potential given by the second order Volterra model, which has been identified at k_j. In Fig. 5 the results can be seen for a randomly generated test potential, discretized for 50 points between $x \in [-4, 4]\ \mathrm{MeV}^{-1}$ in position space, where a remarkable agreement is achieved between the estimated and the true potentials in Fourier-, and also in position space, after inverse Fourier transforming the obtained $V(k)$ potential. For better visibility, the true potential in Fig. 5 is linearly interpolated between the 100 inversion points in the Fourier space, and also between the 50 discrete points in position space, which are used in the inverse Fourier transform. The model works for more complicated test potentials as well, which can be seen in Fig. 6. Here, some larger discrepancies can be seen near $k \approx 6$ MeV, which corresponds to the largest MSE values in Fig. 4, and therefore is expected. With more memories and/or including higher order terms and cross-terms, the model can be further improved.

In this operating range, the nonlinearities are quite negligible, as only second order systems were enough to cover them, with a good accuracy. If the magnitude of the potential is larger, or the potential covers a wider range in space, the simple second order approximation might not be enough, and one has to include higher order terms as well. It was also important to include, at least the first order cross terms, in which case the combinations of the two inputs at different k_i-s e.g. $\Delta\phi(k_i)T(k_j)$, are also appearing in the Volterra representation.

In the next section, we go a step further, and only the phase shift information will be used at a wider operating range, where a different approach is more suitable to solve the inverse problem in low-energy elastic scattering problems, which arises in low-energy nuclear physics.

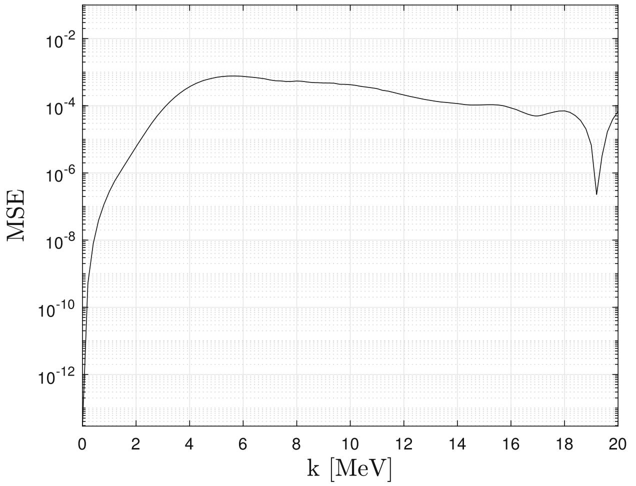

Fig. 4 Mean squared errors for the 100 Volterra models at different wave numbers

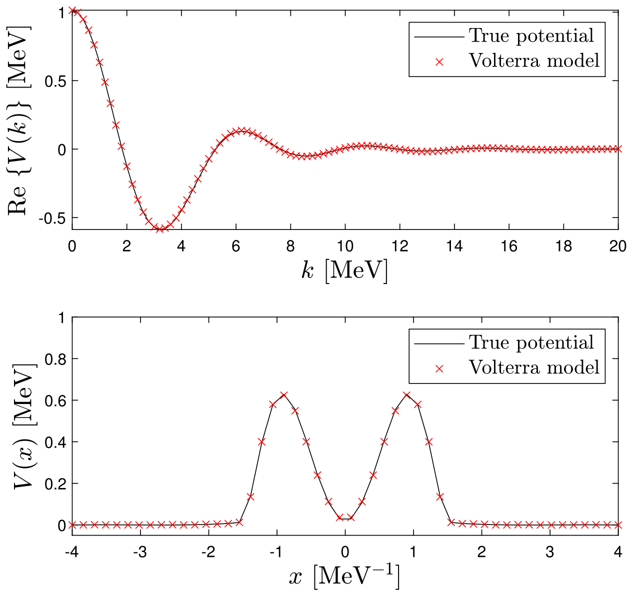

Fig. 5 Validation of the non-causal Volterra model for a discretized test potential. The upper figure show the Fourier transformed true-, and estimated potentials, while the lower figure shows the inverse Fourier transform of the true-, and estimated potentials

4 Low-energy elastic scattering with neural networks

In this section, a new data-driven method will be introduced, which is able to determine the scattering potentials in low-energy elastic scattering experiments, when the only observables are the energy dependent s-wave phase shifts. In this regard, the inversion is a fixed angular momenta method,

[1] The maximum resolution is the consequence of the Nyquist criteria.

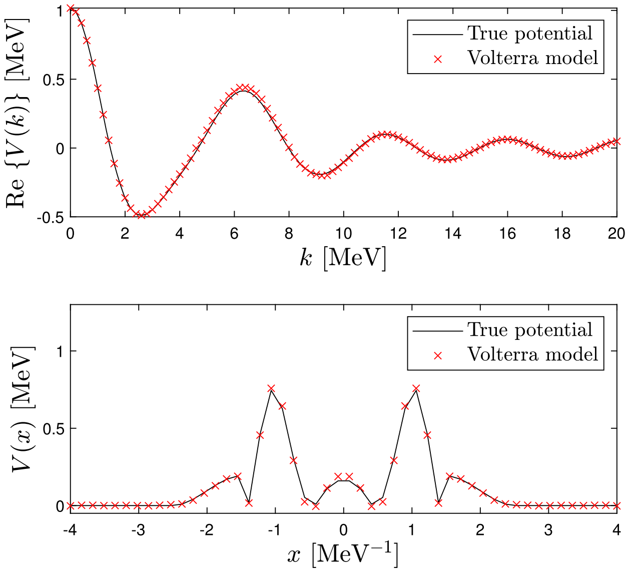

Fig. 6 Validation of the non-causal Volterra model for a discretized test potential. The upper figure show the Fourier transformed true-, and estimated potentials, while the lower figure shows the inverse Fourier transform of the true-, and estimated potentials

where the energy dependence of the phase shifts are used at fixed angular momentum to describe the inverse scattering problem. The method is universal, and it could be easily extended to different angular momenta as well, which is a future goal of the proposed method. In the previous section a Volterra model is used to describe the inverse problem, while here the method is extended with the use of a feed-forward neural network structure, which is able to grasp higher order nonlinearities. After describing the neural network model, and showing its working principles through simple examples, it will be used to describe real-life scattering data in Sect. 5.

Here, first a short introduction is given of the theory of low-energy quantum scattering, where we solely focus on spherically symmetric potentials, but now in three-dimensions, so that the method can be readily used and compared with real experiments as well. In low-energy nuclear scattering the two-body scattering system with masses m_1 and m_2 can be described by the three-dimensional time-independent Schrödinger equation:

$$-\frac{\hbar^2}{2\mu}\Delta\Psi(\boldsymbol{r}) + V(\boldsymbol{r})\Psi(\boldsymbol{r}) = E\Psi(\boldsymbol{r}), \qquad (10)$$

where $\mu = m_1 m_2/(m_1 + m_2)$ is a reduced mass of the two colliding particles, $\Psi(\boldsymbol{r})$ is the normalized wave function, $\boldsymbol{r} = (x, y, z)$ space-vector, $\Delta = (\frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial y^2}, \frac{\partial^2}{\partial z^2})$ is the Laplace operator, while $V(\boldsymbol{r})$ is the potential, and E is the energy. If we only consider spherically symmetric potentials and separate the radial and the angular parts as $\Psi(\boldsymbol{r}) = R_l(r)Y_l^m(\theta, \phi)$ the problem can be simplified into solving the radial Schrödinger equation, which has the form of:

$$-\frac{\hbar^2}{2\mu}\frac{\partial^2 u_l(r)}{\partial r^2} + \left[V(r) + \frac{\hbar}{2\mu}\frac{l(l+1)}{r^2}\right] = E u_l(r), \qquad (11)$$

where $u_l(r) = rR_l(r)$ is the radial wave function, and l is the angular momentum of the system. In scattering experiments, we are interested in the asymptotic behavior of the wave functions, which can be expressed as the sum of an incoming plane wave $e^{i\boldsymbol{kr}}$ and an outgoing spherical wave, weighted by the scattering amplitude $f(k, \theta)$, which is related to the differential cross section as [39,40]:

$$\frac{d\sigma}{d\theta} = |f(k, \theta)|^2. \qquad (12)$$

The scattering amplitude can be expressed with the scattering phase shifts by doing a partial wave expansion, giving the following form:

$$f(k, \theta) = \frac{1}{2ik}\sum_{l=0}^{\infty}(2l+1)(e^{2i\delta_l - 1} - 1)P_l(cos\theta), \qquad (13)$$

where δ_l is the phase shift of the lth partial wave, and P_l is the lth order Legendre polynomial. In scattering experiments the differential-, and total cross sections are measured, and the phase shifts are determined by fitting the different partial waves [41,42].

The forward problem is therefore to solve the Schrödinger equation, and by fitting the expected asymptotic wave-function, to obtain the phase shifts. The Schrödinger equation, which describes the problem is linear, however the inverse problem, which appears in measurements are nonlinear, as it can be seen from the so-called Variable Phase Approximation (VPA) [43–46], which is a first order nonlinear differential equation, which relates a compact and local scattering potential to the asymptotic phase shifts as:

$$\frac{d\delta_l(r)}{dr} = -\frac{U(r)}{k}\Big(j_l(kr)\cdot\cos\delta_l(r) - n_l(kr)\cdot\sin\delta_l(r)\Big)^2, \qquad (14)$$

where $\delta_l(r)$ is the accumulated phase shift at r, while $U(r) = 2\mu V(r)/\hbar^2$, and $k = (2\mu E/\hbar^2)^{1/2}$. The j_l and n_l functions are the Ricatti–Bessel and Ricatti–Neumann functions. The initial condition for the VPA equation is $\delta_l(r = 0) = 0$, which corresponds to a zero phase shift, when the potential still not disturbs the incoming wave. The measured, accumulated phase shift will be the asymptotic $\delta_l(r \to \infty)$ value. This equation can be used in atomic and nuclear physics to obtain the scattering phase shift for a given potential [47,48], however for higher angular momenta the numerical solution is quite difficult, and a very precise solver e.g. fourth or fifth order Runge–Kutta method is necessary.

In this paper, only the s-wave ($l = 0$) scattering is considered, in which case the VPA equation becomes:

$$\frac{d\delta_0(r)}{dr} = -\frac{U(r)}{k}\sin^2\Big(kr + \delta_0(r)\Big). \tag{15}$$

In the inverse problem, the asymptotic $\delta(r \to \infty)$ values at different energies are used as inputs to describe the unknown potential $V(r)$.

Knowing that the phase shifts can be described directly by a first order nonlinear differential equation, one way for inversion would be to give a closed form to the direct problem (if its possible) at different energies, then solve the nonlinear system of equations with a suitable method. In ideal case, the closed form would be the analytical solution of Eq. (15), however due to the nonlinearities, it is not a trivial problem to solve. In the system identification side, it is possible to estimate the behavior of the nonlinear dynamical system in a well defined operating range, with either Volterra series or neural networks. This is, however, not an efficient way to do the inversion, because one still has to solve a system of nonlinear equations to obtain the potentials.

In the Volterra system theory there are methods on how to obtain the inverse kernels from the identified direct kernels [49], while in the neural network case, there are several methods exist for inversion [50–52]. Most of these methods require a numerical inversion scheme, where e.g. gradient methods are used to determine the inverse solution. One especially interesting method is the use of an invertible neural network structure [53,54], where the inverse of the system could be uniquely determined from the forward network. This could be a convenient way to solve these kind of problems, because the forward process is usually much simpler to train, however in this case a very careful training is needed if the inverse system is badly conditioned. Another interesting method, which could be tested in the future is the use of one-dimensional convolutional neural networks (1D CNN) [65], which could have nice properties e.g. insensitivity to small perturbations in the input data, and due to its convolutional form it could be well suited to describe dynamical systems as well. In the followings only multilayer perceptron type neural networks are used to describe the dynamical systems, and the usage of other types of networks is left to future works.

The method, which will be proposed, identifies the inverse system sequentially, using the phase shifts as inputs and the potentials as outputs. The schematic view of the inversion process can be seen in Fig. 7, where $\varDelta\phi$ refers to the energy dependent s-wave phase shifts, while $V(r)$ is the potential. In the method the position space (distance from the origin r) is discretized at M different space points, denoted by $r_1, r_2, r_2, \ldots, r_M$, where r_1 is the closest point to the origin, while r_M is the furthest. The inversion will be given in these so-called inversion points, which ideally spans the whole space, where the $V(r)$ potential differs significantly from zero. In this sense, we assumed that the potential goes to zero (or at least close to zero) at a finite distance, which is a reasonable assumption in nuclear physics. In practice, it is usually possible to give, at least a crude estimation to the range of the potential, thus this information could be easily put into the identification process. If nothing is known from the potential, then it also should not be a problem, however the identification process could take a longer time, because one have to span a larger region, with possibly more inversion points.

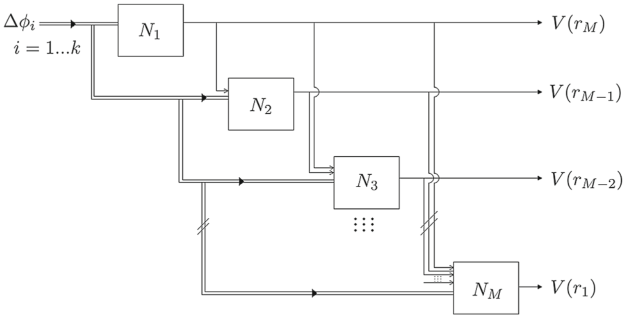

Fig. 7 The schematic view of the inversion process, where each N_i block corresponds to a fully connected MLP network

The method starts from the assumption, that a closed form can be given to the potential at the last inversion point $V(r_M)$, using only the phase shifts as inputs. To describe this relation, a multilayer perceptron model is identified, which is marked as N_1 in Fig. 7. For the next point, $V(r_{M-1})$, the phase shifts, and also the previously estimated $V(r_M)$ potential value is used as inputs for the second dynamical system N_2. The $V(r_{M-1})$ potential still only depends on the measured phase shifts, but now through N_1 as well. In this way the identification process turned out to be much simpler. The method can be generalized for the following points as well, where the last inversion point $V(r_1)$ will depend on the phase shifts, and all of the previously estimated potentials as well, which also depend on the input phase shifts, so ultimately we will arrive at a closed form between the phase shifts and the potentials in the inversion points. The method can be summarized as follows:

$$V(r_M) = N_1(\varDelta\phi_1, \varDelta\phi_2, \ldots, \varDelta\phi_k), \tag{16}$$

$$V(r_{M-1}) = N_2(\varDelta\phi_1, \varDelta\phi_2, \ldots, \varDelta\phi_k, V(r_M)),$$

$$\vdots \tag{17}$$

$$V(r_1) = N_M(\varDelta\phi_1 \ldots, \varDelta\phi_k, V(r_M), \ldots, V(r_2)), \tag{18}$$

where each $N_i(\cdot)$ expression represents a multilayer perceptron model. The number of free parameters can be expressed by counting the weight and bias parameters at every MLP used in the inversion points as:

$$N^{tot} = \sum_{i=1}^{M} \Bigg[(k+i-1)h_1^{(i)} + \sum_{j=1}^{n^{(i)}-1} h_j^{(i)} h_{j+1}^{(i)} + h_{n^{(i)}} + \sum_{j=1}^{n^{(i)}} h_j^{(i)} + 1 \Bigg], \tag{19}$$

where M is the number of inversion points, $h_j^{(i)}$ is the number of neurons in the j'th hidden layer of the i'th MLP, k is the number of phase shifts used as inputs, while $n^{(i)}$ is the number of hidden layers in the i'th MLP.

To test the method in a simple scenario, let us assume, that the sought potential is bounded between $V(r) \in [-10, 0]$ MeV in $r \in [0, 6]$ fm, and dies out at $r > 6$ fm, while for the mass, we fix $m = m_1 = m_2 = 1$ GeV. These values give a good benchmark to examine the method, as the nonlinearities in this case are becoming non-negligible, thus a nonlinear dynamical model is necessary. Here, instead of the wave number k, we introduce the laboratory kinetic energy T_{lab} as:

$$k = \frac{m_2^2(T_{lab}^2 + 2m_1 T_{lab})}{(m_1+m_2)^2 + 2m_2 T_{lab}}, \tag{20}$$

where T_{lab} is the laboratory kinetic energy. In the following examples T_{lab} is going to be used as the input parameter for the phase shifts, so $\Delta\phi(k) \rightarrow \Delta\phi(T_{lab})$ exchange is used, to be able to compare the model results directly with the measurements in Sect. 5. To do the identification, 10,000 training-, and 2000 validation samples has been generated in the predefined range, where the phase shifts has been calculated by the VPA equation shown in Eq. (15).

The identification has been done for different systems containing different number of hidden layers and/or neurons to be able to determine an optimal system complexity for the actual problem. The inputs for the system were the laboratory kinetic energy dependent phase shifts, $\Delta\phi(T_{lab})$, from 1 to 99 MeV, with $\Delta T_{lab} = 5$ MeV steps, so in overall 20 points at $T_{lab} = [1, 6, 11, 16, \ldots, 99]$ MeV. The number of inversion points is set to $M = 20$ between 0 and 5.713 fm, with $\Delta r = 0.3007$ fm. To train the system the Levenberg-Marquardt back propagation algorithm is used [63] sequentially for every inversion point one after another, starting from the last point at the largest distance. The inputs and outputs were normalized between $[-1, 1]$ in every neural block, and to avoid overfitting, the training is stopped after the validation error starts to increase, or the gradient reaches a minimal value set to be 10^{-7}.

To make a first guess for the complexity, four different systems were trained and compared by calculating the relative errors in the inversion points for 2000 test samples, which can be seen in Fig. 8, where the relative errors are shown in each of the inversion points. In each of these systems all of the N_i neural blocks are assume to have a feed-forward MLP structure, with the same number of hidden layers, and the same number of neurons.

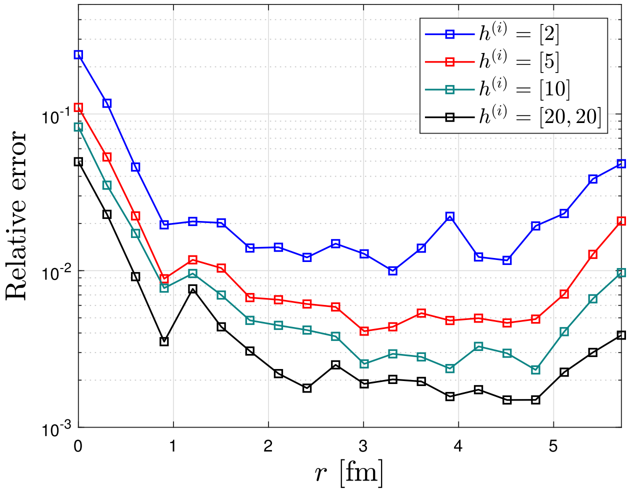

Fig. 8 Relative errors in the inversion points for four different MLP's

Three of the four systems consists only one hidden layer $h^{(i)}$, with 2, 5, 20 neurons in each of the neural blocks N_i, while in the last system two hidden layers were used, with 20 neurons in each. The activation functions were $\tanh(\cdot)$ nonlinearities in the hidden layers, and a linear relationship in the output layer. For the first inversion point in $r = 5.713$ fm only the input phase shifts could be used for teaching, therefore one could expect a larger error, and a slower learning rate, which can be seen in the calculated relative errors as well. It can be seen that with increasing complexity, the relative errors are decreasing in each of the inversion points. With using 10 neurons and one hidden layer, the relative error is just a few percent in almost all of the points. In all cases, the largest errors can be seen in the last few points at small distances below 0.5 fm, which is the consequence of the large sensitivity of the inversion in this range. With using two hidden layers (or more neurons), the errors could be further reduced, however to achieve a very good inversion it is not necessary to use two hidden layers in each of the inversion points. The final model, which will be used in the followings consists one hidden layer, with [10] neurons in the first 16 inversion points $r_{20}, r_{19}, \ldots, r_5$, and [20, 20] neurons in two hidden layers in the last 4 inversion points $r_4, \ldots, r_1$, where the errors are the largest. Following Fig. 7 this means, that N_1,N_2,. . .,N_{16} are MLP's with one hidden layer and 10 neurons, while N_{17},N_{18},N_{19}, and N_{20} are MLP's with two hidden layers, each having 20 neurons. The relative errors in the inversion points for this system can be seen in Fig. 9, where again, a few percent relative error has been obtained in all of the points.

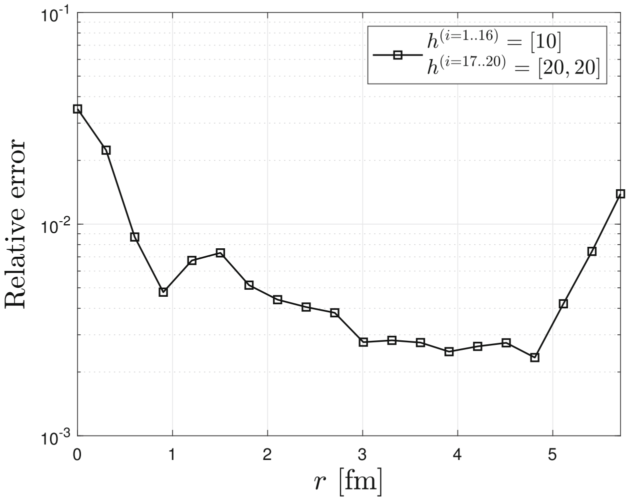

Fig. 9 Relative errors in the inversion points for the identified MLP. The $h^{(i)}$ notation refers to the number of neurons used in the hidden layers in the i'th inversion point, where i corresponds to the distance r_{20-i+1}

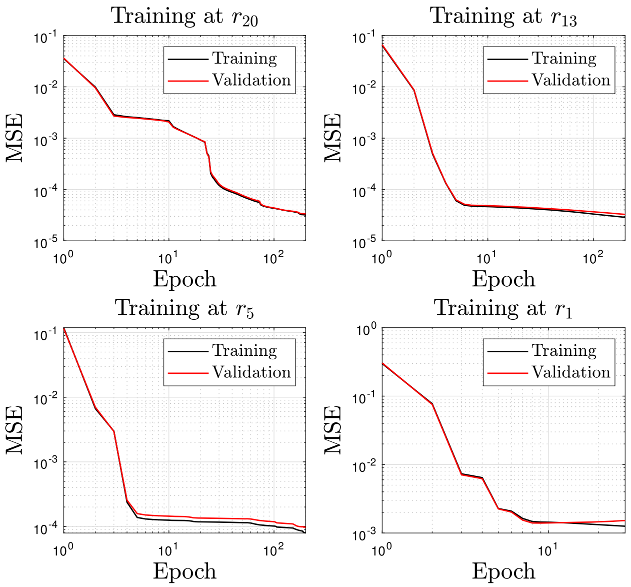

Fig. 10 Training and validation losses during the training process for four inversion points

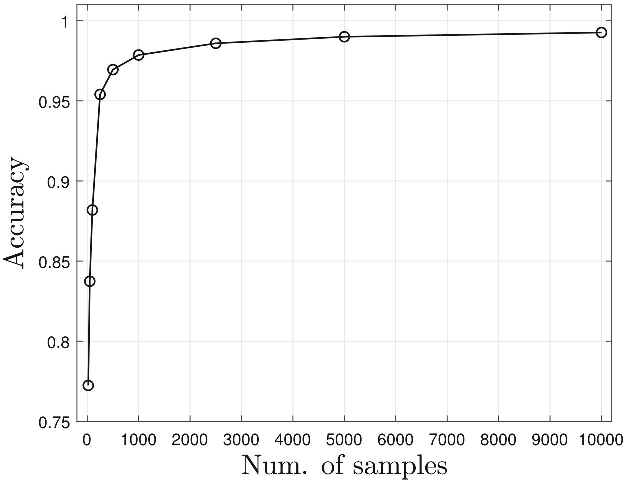

Fig. 11 Model accuracy dependence on the number of training samples

In Fig. 10 the training process can be followed, where the dependence of the mean squared error on the number of epochs determined from the normalized inputs/outputs are shown in four different inversion points r_{20}, r_{13}, r_5, r_1, where the training and validation loss is shown as well. The figures clearly show the previously mentioned behavior regarding the time to reach an optimal solution. The first inversion point (r_{20}) needs more than 10^2 epochs to reach an MSE, which corresponds to a few percent relative error in that point, while in the subsequent points a convergence can be achieved relatively fast, after 10–20 epochs. Near the last inversion point (r_1) the MSE of the converged solution is a little larger than in the previous points, which is also expected from the previously determined relative errors in Fig. 9.

To further examine the model, the accuracy has been calculated using different number of training samples for the previously determined model, which can be seen in Fig. 11. The accuracy shows a sharp rise at a few hundred samples, then a slowly saturating region after a few thousand samples, which is the usual shape one will get with multilayer perceptron models [64]. This shows us, that in this system, with only a few thousand training samples a good generalization can be obtained with the proposed model.

To show the model capabilities through some examples, the results for two randomly generated potential compared with the original potentials can be seen in Fig. 12. The inversion technique is capable to give satisfying results within a few percentage of relative error, even with this simple neural network structure. It is worth mentioning, that the discrepancies, especially at small r-s, could be improved, by setting up more inversion points, and/or using more phase shifts as inputs, however in that case the time of training and the complexity of the model could also increase.

5 Estimating the $^3S_1$ NN scattering potential with neural networks

After introducing the neural network model through a simple example, let us proceed to the main application of the inversion method, which is the determination of the scattering potential in low-energy elastic S-wave NN scattering from the experimental phase shifts. In particular, we are interested in the $^3S_1$ channel, in which case one bound state, the deuteron, appears in the measured phase shifts [55,56]. In the

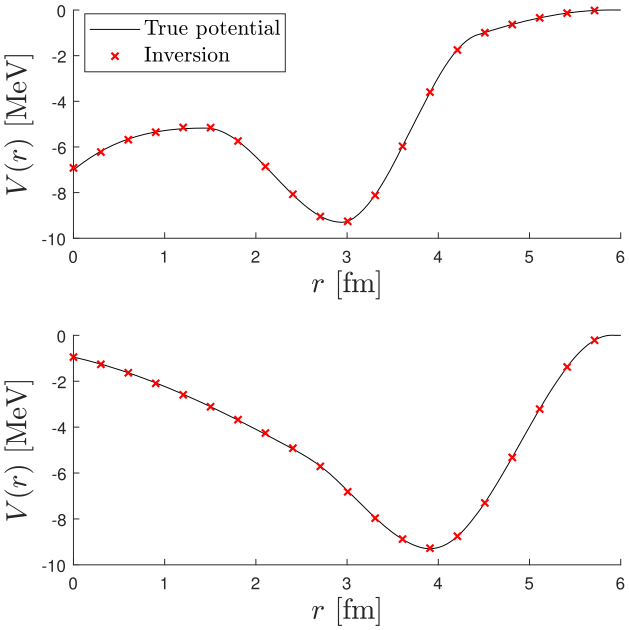

Fig. 12 Validation of the model for two randomly generated potentials. The red markers represent the values at the inversion points, while the black lines are the true potentials

identification process, there are no restrictions to the number of possible bound states, however, knowing apriori the number of bound states, it could be possible to restrict the training samples to only consist potentials, which can give a specific number of bound states. In this case the training and the identification of the system could be greatly simplified, however the obtained model will be more restricted.

Before generating the training samples, it is necessary to gather some information about the identifiable system, as it could greatly simplifies the identification process. In this case, it is known, that the effective range of the scattering process should be a few Fermi in position space, however it is also known, that strong repulsive, and Coulomb-like terms could also appear in the potentials. As it was described in the previous section, the first inversion point (furthest from the origin) does not have to be zero, but it is desirable, to be at least close to zero, as in that case, the identification is much simpler, and a simpler model could be used to describe that point. In this case, we can assume, that at approximately $r \approx 3$ fm, the potential is negligible, so for first approximation, we set the range of the inversion to $r \in [0, 3]$ fm, with $M = 30$ inversion points, starting from $r_1 = 0$ fm, and ending at $r_{30} = 2.9986$ fm, which corresponds to $\Delta r = 0.1034$ fm.

The operating range for the inversion has been set, so the model could describe a bounded attractive and repulsive potential in the whole range, and also a highly repulsive term e.g. a Coulomb-, or a nonperturbative strong repulsive core, at low r-s near $r = 0$. The potentials are also assumed to go to zero at $r \approx 3$–4 fm. Furthermore, it would be possible to include infinities at $r = 0$ as well, in which case the inversion point at $r = 0$ fm has to be excluded. In this case, we assumed that the potential at $r < 0.5$ fm is a finite value between -500 and $20,000$ MeV, while at $r > 0.5$ fm the values are constrained in the range of $V(r) \in [-500, 500]$ MeV, and go to zero at 3–4 fm. In the generation of the excitation functions, or in other words the $V(r)$ training potentials, it is necessary to include the previous assumptions, while keeping the randomness of the generated functions. This is realized with the help of Piecewise Hermite Interpolating Polynomials (PCHIP) [57], which is a polynomial interpolation method, that is able to give continuous functions in a predefined range, so it is possible to generate $V(r)$ functions, where $|V(r)| < V_{max}$ constraint can be easily satisfied. To do the interpolation a number of control points has to be given, which are used as fixed points, in which between the interpolation is done. To keep the randomness of the potentials, the control points are chosen randomly between $r_c \in [0, r_{max}]$ for each $V_i(r)$ excitation, by first generating a random number N, which gives the number of control points, then generating N number of r_i points, which will be the place of the control points. At the final step a random number is generated for each control point in the operating range of $V(r)$, then we interpolate between the control points. Two randomly generated potential, which respects our previous assumptions are shown in Fig. 13, where it can be seen, that at small distances the potential could be very large, while in the region $r > 3$ fm it goes to zero. It can also be seen, that if the number of control points are large, then the potential keeps its randomness, that we desire in an identification process, where one optimally wants to excite all of the frequencies until the Nyquist limit. To train the neural network, 10,000 samples have been generated with different number of control points in each sample from $N = 2$ up to $N = 18$, while for validation another 2000 samples have been generated, which are also used in the training process to avoid overfitting. Following the reasoning of the previous example, the complexity of the network has been chosen, so that it could describe the system with a few percent relative errors in all of the inversion points. The final structure for all of the inversion points are MLP's with two hidden layers, having 30 neurons in each, with $\tanh(\cdot)$ activation functions, and a linear output layer. In this model, every inversion point is described by this structure, so every $N_i(\cdot)$ $(i = 1, \ldots, 30)$ has the same MLP structure, which has to be trained sequentially, starting from the first inversion point r_{30}.

The training process for two representative inversion points (r_{30}, r_{23}) can be followed in Fig. 14, where the expected behavior is seen again. The first inversion point r_{30}, which only depends on the input phase shifts, the training process is slower and need more epochs to reach convergence, than what is observed in the following inversion

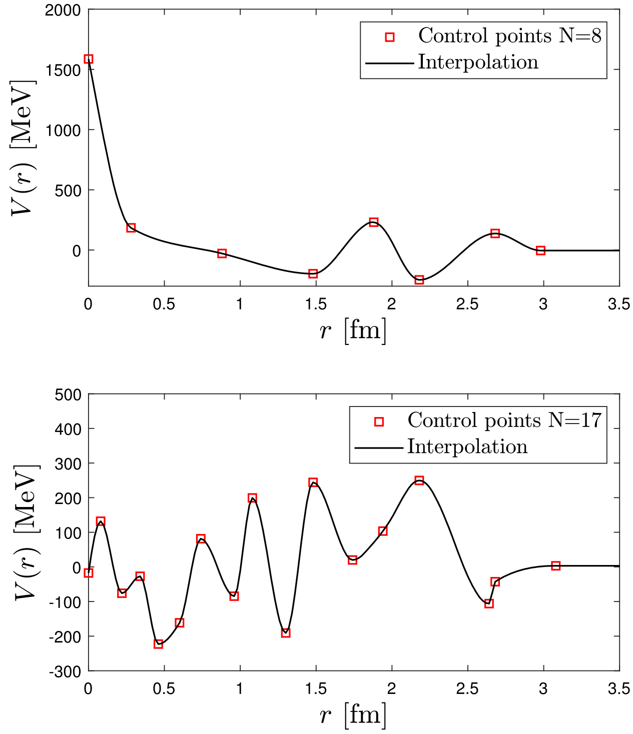

Fig. 13 Two randomly generated potential, used for training the neural network, with different number of control points in the PCHIP representation

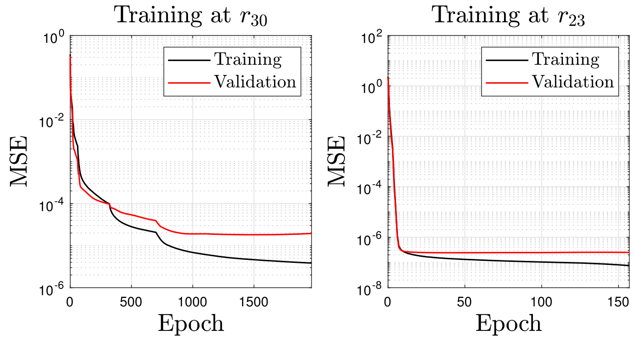

Fig. 14 Loss function (MSE) dependence on the number of epochs of the training and validation samples during the training process

points, which depend on the previous inversion points and the input phase shifts as well.

The obtained relative error in the 30 inversion points for 2000 test samples after training are shown in Fig. 15. The largest errors again occur in the first few inversion points at the largest distances, and at the last few inversion points at small distances, which was the observed behavior in the last example as well. In overall a few percent error could be achieved with this model. An example of the model capabilities to a randomly generated potential can be seen in Fig. 16.

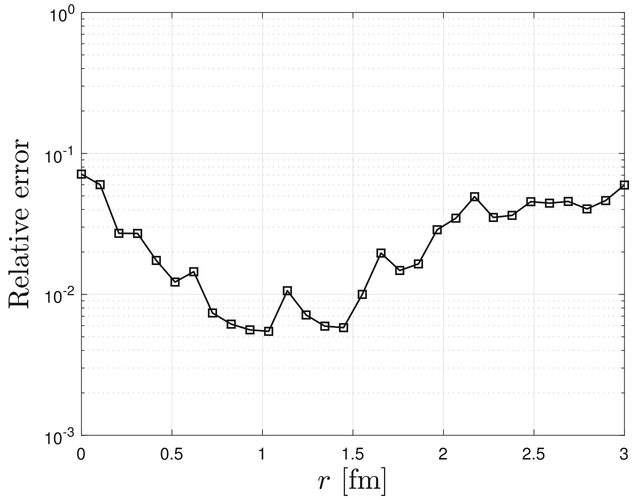

Fig. 15 Relative errors of the trained neural network in the inversion points

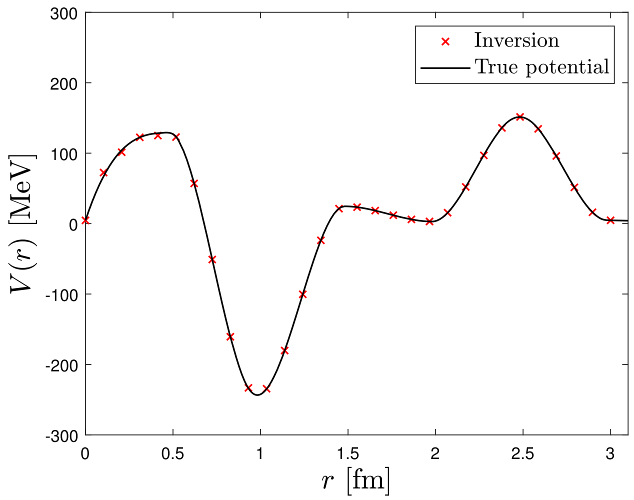

Fig. 16 Validation of the model for a randomly generated test potential

After the inverse system is identified in the desired operating range, it is possible to estimate the low-energy NN scattering potentials from the measured phase shifts. In this section the neutron+proton scattering is investigated between $T_{lab} \in [1, 300]$ MeV laboratory kinetic energies, for which the measured phase shifts are taken from [58]. After inversion, the obtained potential and the laboratory kinetic energy dependent, re-calculated phase shifts can be seen in Fig. 17, where a very good match has been achieved between the original-, and the re-calculated phase shifts. The averaged relative error, through the measured points, between $T_{lab} \in [1, 300]$ MeV is approximately 2%. The obtained potential has a strong repulsive core at small distances, while at larger distances a moderate attractive force emerges. This picture is comparable with the usual nuclear potential models for s-wave scattering [59,60], where the strong repulsive core is the consequence of the strong force at small distances, while the medium and large distance attractive terms are related

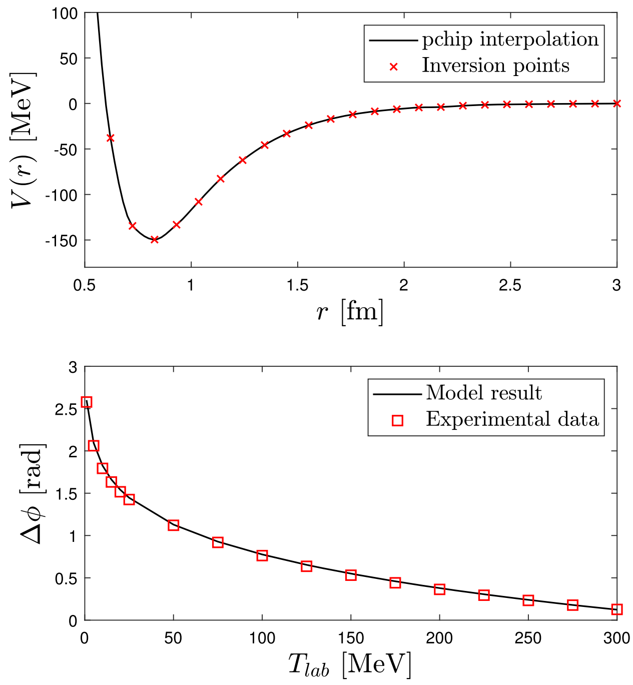

Fig. 17 Estimated ${}^3\mathrm{S}_1$ NN potential, obtained from the neural network inversion model. The upper panel shows the calculated values at the inversion points, and also the PCHIP interpolation between the inversion points, while the lower panel shows the calculated and measured phase shifts

to the various meson exchange forces [61,62]. It is worth mentioning, that we did not have to make any distinction for scattering with-, and without bound states, and could include them both in the identified model. The identified system in this form can only be used to describe $l = 0$ scatterings, however by adding the angular momenta to the inputs, it can be easily extended to describe potentials, with different angular momenta as well.

6 Conclusions

In this paper, two data-driven inversion methods are proposed, which are both able to describe inverse quantum mechanical scattering systems, in a well-specified operating range, using nonlinear system identification techniques. In these problems, the scattering potential is sough from the known observables e.g. phase-shifts, and transmission coefficients, using finite dimensional, truncated Volterra systems, and feed-forward neural networks. Two main structure is proposed, which are tested in two scattering scenarios. Firstly, a non-causal Volterra system is used to describe the one-dimensional quantum scattering problem, where the inputs were the energy dependent phase shifts, and the transmission coefficients, while the output was the Fourier transformed scattering potential. The system was designed, so that in every k_i (in the range, constrained by the Nyquist limit), one non-causal Volterra system is identified, which used the input functions at $k_{i-M}, k_{i-M+1}, \ldots, k_i, \ldots, k_{i+M}$, defined by the memory of the system. To test the method, the operating range has been set to $V(x) \in [0, 1]$ MeV, with a test mass of $m = 1$ MeV. With these parameters the nonlinearity of the system could be modeled by second order Volterra series, with $M = 35$ memories, by including only the first order cross-terms of the input phase shifts, and transmission coefficients.

For the second problem, a more realistic model is proposed, where the low-energy elastic scattering on spherically symmetric potential is addressed, with neural networks. In this model, only the energy dependent phase shifts were used as inputs to the inverse system, while the potential in position space is used as the output. To solve the inversion, a sequential neural network architecture is proposed, which consists one MLP network at each inversion point, which are connected by their outputs. To test the model, the operating range has been set to $T_{lab} \in [1, 99]$ MeV, and $V(r) \in [-10, 0]$ MeV, with test masses of $m_1 = m_2 = 1$ GeV. The network has been trained by generating a sufficient number of training samples, with the help of the Variable Phase Equation, then tested to randomly generated continuous potentials, giving good results in each case, with a few percent relative error.

The neural network construction then used to describe the low-energy elastic neutron+proton scattering from measured phase shifts between $T_{lab} \in [1, 300]$ MeV. To train the network, the operating range has been set to be able to describe the short distance repulsive-, and the larger distance attractive terms as well, where the training samples were constructed with the help of Piecewise Cubic Hermitian Interpolating Polynomials, which were able to easily constraint the potential functions between the prespecified range. The results of the inversion shows a physically reasonable potential, with a short range repulsive, and a long range attractive part. The phase shifts were calculated back from the obtained potential, and compared to the measured phase shifts, giving good results, with around 2% relative error, averaged through the number of measurement points.

The methods proposed here, were both able to solve the inversion problem in quantum scattering, at a predefined operating range. As the inverse system is essentially nonlinear, by extending the operating range, the nonlinearities will have greater roles, and a more complex system has to be identified, which in the case of the Volterra model, means more memories, and higher order terms, with more cross-terms as well, while in the neural network structure, it means more hidden-layers, with more neurons. This method is advantageous, if the full nonlinear dynamical equations, which governs the system, are not known, or one only wants to operate the system in a well-defined range. In this case one option would be to identify a model (Volterra/Neural Network) for the forward system, than by generating a sufficient number of samples, identify the inverse system as well, with one of the described methods. Another possibility would be to directly identify the inverse system by using only the measured data, which could be more problematic if there are no sufficient number of observables to work with. In low-energy quantum scattering experiments the dynamical equations, which governs the system are well-known, therefore there is no need to identify a black box system, and one can utilize the apriori knowledge in quantum mechanics to build a sufficient model even for more complex systems as well.

In the case, that the dynamical equations are known precisely, the Volterra kernels can be calculated analytically and the higher order terms can be obtained more easily. As the Variable Phase Equation describes the forward problem, it could be possible to analytically calculate the Volterra kernels for the forward process, then invert the obtained Volterra system, to be able to estimate the potentials from the phase shifts, which could be a future application of the proposed methods.

Acknowledgements The author was supported by the Hungarian OTKA fund K138277. This article is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement STRONG-2020-No 824093.

Funding Information Open access funding provided by ELKH Wigner Research Centre for Physics.

Data Availability Statement This manuscript has no associated data or the data will not be deposited. [Authors' comment: All the data shown in the paper are available from the corresponding author, upon request.]

References

1. K. Chardan, P.C. Sabatier, *Inverse Problems in Quantum Scattering Theory*, 2nd edn. (Springer, New York, Berlin, 1989)
2. R.G. Newton, *Scattering Theory of Waves and Particles* (McGraw-Hill, New York, 1966)
3. V.I. Kukulin, R.S. Mackintosh, J. Phys. G **30**, R1 (2004)
4. R. Lipperheide, H. Fiedeldey, Z. Phys. A **286**, 45 (1978)
5. R. Lipperheide, H. Fiedeldey, Z. Phys. A **301**, 81 (1981)
6. H.W. Engl, C.W. Groetsch, *Inverse and Ill-Posed Problems* (Academic Press, Cambridge, MA, 2014)
7. S. Kabanikhin, N. Tikhonov, V. Ivanov, V.M. Lavrentiev, J. Inverse Ill-Posed Probl. **16**, 317 (2008)
8. H. Isozaki, J. Math. Phys. **45**, 2613 (2004)
9. L.J. Allen et al., Phys. Lett. B **298**, 36 (1993)
10. C.A. Coulter, G.R. Satchler, Nucl. Phys. A **293**, 269 (1977)
11. T.H. Kirst, K. Amos, L. Berge, M. Coz, H.V. von Geramb, Phys. Rev. C **40**, 912 (1989)
12. R.G. Newton, J. Math. Phys. **3**, 75 (1962)
13. R.S. Mackintosh, S.G. Cooper, Phys. Rev. C **43**, 1001 (1991)
14. E.G.F. Thomas, J.L. van Hemmen, W.M. Kistler, SIAM J. Appl. Math. **61**, 1 (2000)
15. C.M. Cheng, Z.K. Peng, W.M. Zhang, G. Meng, Mech. Syst. Signal Process. **87**, 340 (2017)
16. M.J. Korenberg, I.W. Hunter, Ann. Biomed. Eng. **24**, 250 (2007)
17. M. Balajewicz, F. Nitzsche, D. Feszty, AIAA J. **48**, 56 (2010)
18. C.L. Phillips, J.M. Parr, E.A. Riskin, *Signals, Systems, and Transforms*, 4th edn. (Pearson College Div, London, England, 2008)
19. H.J. Bessai, *MIMO Signals and Systems* (Springer, Berlin/Heidelberg, Germany, 2005)
20. A.V. Kamyad, H.H. Mehne, A.H. Borzabadi, Appl. Math. Comp. **167**, 1041 (2005)
21. M. Schoukens, R. Pintelon, T.P. Dobrowiecki, J. Schoukens, IEEE Trans. Autom. Control **65**, 1514 (2020)
22. G. Palm, T. Poggio, J. Appl. Math. **33**, 195 (1977)
23. G. Palm, B. Pöpel, Quart. Rev. Biophys. **18**, 135 (1985)
24. G. Balassa, Mathematics **10**(8), 1257 (2022)
25. G. Stepniak, M. Kowalczyk, J. Siuzdak, Sensors **18**, 1024 (2018)
26. K. Zhong, L. Chen, Math. Prob. Eng. **2020**, 1 (2020)
27. B. Zhang, S. Billings, Mech. Syst. Sign. Proc. **84**, 39 (2017)
28. J. Wray, G. Green, Biol. Cybern. **71**, 187 (1994)
29. K. Gurney, *An Introduction to Neural Networks* (UCL Press, London, England, 1997)
30. V. Marmarelis, X. Zhao, IEEE Trans. Neural Netw. **8**, 1421 (1997)
31. J. Schmidhuber, Neural Netw. **61**, 85 (2015)
32. S. Urolagin, K.V. Prema, N.V.S. Reddy, Generalization Capability of Artificial Neural Network Incorporated with Pruning Method: Advanced Computing, Networking and Security. ADCONS 2011. Lecture Notes in Computer Science, vol. 7135 (Springer, Berlin, Heidelberg, 2012)
33. G. Baym, *Lectures on Quantum Mechanics* (W. A. Benjamin Inc., Reading, MA, USA, 1969)
34. V.E. Barlette, M.M. Leite, S.K. Adhikari, Am. J. Phys. **69**, 1010 (2001)
35. J.E. Bowcock, H. Burkhardt, Rept. Prog. Phys. **38**, 1099 (1975)
36. D.L. Price, F.F. Alonso, *Experimental Methods in the Physical Sciences*, vol. 44 (Academic Press, Cambridge, MA, 2013)
37. J. Schoukens, R. Pintelon, Y. Rolain, *Mastering System Identification in 100 Exercises* (John Wiley & Sons, Hoboken, NJ, USA, 2012)
38. K. Tiels, M. Schoukens, J. Schoukens, Automatica **60**, 201 (2015)
39. R.C. Fernow, *Introduction to Experimental Particle Physics* (Cambridge University Press, Cambridge, England, 1989)
40. R.G. Newton, *Scattering Theory of Waves and Particles* (McGraw Hill, New York, USA, 1966)
41. S. Dubovichenko et al., Chin. Phys. C **41**, 014001 (2017)
42. V.G.J. Stoks, R.A.M. Klomp, M.C.M. Rentmeester, J.J. de Swart, Phys. Rev. C **48**, 792 (1993)
43. V. Viterbo, N. Lemes, J. Braga, Rev. Bras. Ensino Fís. **36**, 1 (2014)
44. J.M. Clifton, R.A. Leacock, J. Comp. Phys. **38**, 327 (1980)
45. F. Calogero, Am. J. Phys. **36**, 566 (1968)
46. A. Romualdi, G. Marchetti, Eur. Phys. J. B **94**, 249 (2021)
47. A.P. Palov, G.G. Balint-Kurti, Comp. Phys. Commun. **263**, 107895 (2021)
48. P.M. Morse, W.P. Allis, Phys. Rev. **44**, 269 (1933)
49. M. Morháč, Nonlinear Anal. Theory Methods Appl. **15**, 269 (1990)

50. B.-L. Lu, H. Kita, Y. Nishikawa, IEEE Trans. Neural Netw. **10**, 1271 (1999)
51. G.A. Padmanabha, N. Zabaras, J. Comput. Phys. **433**, 110194 (2021)
52. C.A. Jensen et al., Inversion of feedforward neural networks: algorithms and applications. Proc. IEEE **87**(9), 1536–1549 (1999)
53. I. Ishikawa et al. Universal approximation property of invertible neural networks, arXiv preprint arXiv:2204.07415 (2022)
54. L. Ardizzone et al. Analyzing inverse problems with invertible neural networks, arXiv preprint arXiv:1808.04730 (2018)
55. M. Elbistan, P. Zhang, J. Balog, J. Phys. G Nucl. Part. Phys. **45**, 105103 (2018)
56. A. Khachi, L. Kumar, O.S.K.S. Sastri, J. Nucl. Phys. Mater. Sci. Rad. A. **9**, 87 (2021)
57. C.A. Rabbath, D. Corriveau, Defence Technol. **15**, 741 (2019)
58. R.A. Arndt et al., Phys. Rev. D **28**, 97 (1983)
59. R.B. Wiringa, V.G.J. Stoks, R. Schiavilla, Phys. Rev. C **51**, 38 (1995)
60. E. Shuryak, J.M. Torres-Rincon, Phys. Rev. C **100**, 024903 (2019)
61. M. Chemtob, J.W. Durso, D.O. Riska, Nucl. Phys. B **38**, 141 (1972)
62. M. Naghdi, Phys. Part. Nucl. **45**, 924 (2014)
63. C. Lv et al., IEEE Trans. Ind. Inf. **14**, 3436 (2018)
64. W. Ng, B. Minasny, W.D.S. Mendes, J.A.M. Demattê, SOIL **6**, 565 (2020)
65. S. Kiranyaz et al., Mech. Syst. Signal Process. **151**, 107398 (2021)