ARTIFICIAL NEURAL NETWORK METHODS IN FEW-BODY SYSTEMS

by

GAOTSIWE JOEL RAMPHO

submitted in part fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

in the subject

PHYSICS

at the

UNIVERSITY OF SOUTH AFRICA

SUPERVISOR : PROF. S. A. SOFIANOS.
JOINT SUPERVISOR : PROF. I. E. LAGARIS.

NOVEMBER 2002

****

Acknowledgments

I would like to thank the following persons :

Prof. S. A. Sofianos for suggesting this problem and for acting as supervisor. Without his guidance and inspiration this work would not have materialized.

Prof. I. E. Lagaris for acting as joint supervisor. I cannot thank him enough for having been able to obtain resources to visit South Africa to offer his expert guidance during enlightening discussions.

Members of the Physics Department (University of South Africa) for their encouragement and support.

My wife, Tebogo, and daughters, Tshepiso and Karabo, for their patience, understanding and motivation.

Summary

Artificial neural networks and optimization techniques are used to solve the relativistic and non-relativistic eigenvalue problem in two-body systems. The numerical results are in excellent agreement with known analytical or numerical ones obtained via other methods.

The artificial neural network method used, employs a multilayer perceptron with one hidden layer as the main approximating element. The trial solutions are constructed in such a way that they satisfy the required boundary conditions automatically. The dependence of the accuracy of the approximate solutions on the formulation of the trial solution and the size of the multilayer perceptron is investigated.

Keywords: Feedforward Neural Networks; Multilayer Perceptron; Optimization; Few-body Systems; Variational Methods; Coulombic systems; Ground States.

Contents

Chapter 1

Introduction

Different types of methods have been developed for solving a variety of differential equations. Some of the methods involve approximations of the differential operators in the equation by some difference equations. Such difference equations are derived from the Taylor expansion of the solution to the differential equation about a suitable point in the solution domain. Other methods obtain approximations of the solution to the differential equation by expanding the solution in terms of a set of known linearly independent basis functions. Artificial neural networks (ANNs) as numerical tools in solving differential equations belong to the latter class. In mathematical terms ANNs are non-linear functions having a set of variable parameters referred to as the *connection weights.* A typical ANN computational method generally involves approximating the solution to the problem at hand by some ANN. An appropriate error function, involving the solution approximation, is formulated and then minimized, by varying the network weights to extract the solution.

There is a number of advantages for employing ANNs in computational methods. One of these advantages is that the approximations obtained via neural networks are differentiable, compact, and can be used in further calculations [1]. Furthermore, it has been demonstrated that ANNs are universal approximators [2]. Many results supporting this fact have been reported [2, 3, 4, 5]. Moreover, there is a possibility for implementing the method on *neuroprocessors* [6].

ANNs are used in many areas of different disciplines. For example, ANN algorithms have been realized on specialized hardware and were also employed in data analysis techniques that have led to obtaining experimental results that would otherwise be extremely difficult to obtain in high energy physics (e.g. the mass of the top quark [7, 8]). As numerical tools ANN were previously employed by Takeda and Goodman [9] and by Barnard and Casasent [10] to solve systems of algebraic equations. Their approach involved mapping the equations to be solved on a Hopfield-type network. An energy function relating to the network was then minimized to obtain the required solution. Lee and Kang [11] used a similar approach to solve differential equations. They transformed the original problem, via finite difference methods, into a system of algebraic equations that were then solved using a Hopfield-type network. Yentis, Jr. and Zaghloul [6] also employed neural networks to solve finite difference equations resulting from partial differential equations. Wang and Mendel [12] used a feedforward ANN to solve systems of algebraic equations. To solve differential equations Meade Jr. and Fernandez [13] constructed splines by superposing linear transfer functions. The solution was then expressed as an expansion of these splines, the expansion coefficients of which are to be obtained by solving some predetermined equations. The splines are then mapped on to a feedforward ANN. These approaches, unfortunately, are very difficult to generalize and therefore have very limited application.

Recently, Lagaris *et al* [1] presented a method that employs a multilayer perceptron, with one hidden layer, as the main approximation element. The method requires the construction of the solution approximation to satisfy the required boundary conditions automatically. The solution approximation is mapped directly onto a chosen network. Collocation methods are then used to formulate a suitable error function which is to be minimized, under appropriate constraints, to extract the solution. The method was applied to a variety of differential equations [1, 14, 15]. In particular, the eigenvalue problems considered in Ref. [14] included the Dirac and the multi-dimensional Schrödinger equations. The method showed excellent performance in all the cases. However, the presentations in Refs. [1, 14, 15] do not explicitly establish aspects of the method related to the network structure stabilization or regularization.

In this dissertation we use the method presented in Ref. [1] to obtain solutions of the radial Schrödinger and Dirac equations for selected two-body bound states involving local potentials. We investigate the dependence of the accuracy of the numerical solutions on the form of the trial function and the number of neurons in the hidden layer of a one-hidden-layer multilayer perceptron. We also consider the influence of the form of the trial solution on the accuracy of the results. The question of the effect of the size of a multilayer perceptron with one hidden layer on the accuracy of solution approximations has been theoretically (or qualitatively) investigated (see for in stance [2, 3]). Most of the findings of such investigations can, in many cases, be generalized to solutions of differential equations [1, 14, 15]. We present results that suggest that approximations achieved by ANNs in certain two-body systems require special attention in order to achieve accurate solutions. In our studies we consider, for comparison, selected few-body potentials that have analytical solutions or have been solved in the literature, using other numerical methods. The results of this work are meant to complement the work presented in Ref. [1]. Since ANNs are beginning to receive increased attention in some areas of physics we believe that our results will serve as useful relevant reference for further application of ANNs in few-body physics.

This dissertation is organized as follows. In chapter 2 we recall the key equations to be solved. The main object of the chapter is the solution to the one-dimensional Schrödinger and radial Dirac equations. A brief outline of the working principle of variational techniques is also given. In chapter 3 we introduce the concept of artificial neural networks and how their architecture is related to function approximations. In chapter 4 we solve the eigenvalue problems for the following systems: a simple harmonic oscillator potential, a Wood-Saxon type potential, some Coulombic-plus-power law potentials, an attractive Coulombic potential, an exponential potential, and a Yukawa potential. The results obtained are presented and discussed. The relevant conclusions drawn are presented in chapter 5.

Chapter 2

Bound States

The main aim of this dissertation is to obtain numerical solutions of the eigenvalue problems of selected few-body bound systems. The equations we solve are the Schrödinger equation, for the non-relativistic treatment, and the Dirac equation, for relativistic considerations. In the present work we consider only the systems that interact via central potentials.

In this chapter we summarize relevant aspects of the time-independent Schrödinger equation in section 1.1, and highlight the properties of bound state solutions. In section 1.2 we look at the time-independent Dirac equation. We conclude this chapter with a short description, in section 1.3, of the application of variational methods to the radial Schrödinger and Dirac equations.

2.1 The Schrödinger Equation

The eigenvalue problem to be solved is

$$H\,\psi = E\,\psi \tag{2.1}$$

where E is the energy eigenvalue and ψ the corresponding eigenfunction of the Hamiltonian H. For the time-independent Schrödinger equation the Hamiltonian, in the center-of-mass system, reads

$$H = -\frac{\hbar^2}{2\,\mu}\,\nabla^2 + V(\vec{r})\,, \tag{2.2}$$

where μ is the reduced mass of and $V(\vec{r})$ the potential representing the interactions in a given system. If the prevailing forces in the system are central then the potential is spherically symmetric and Eq. (2.1) is best described in spherical polar coordinates.

2.1.1 The Radial Equation

For spherically symmetric potentials the Hamiltonian operator H in spherical polar coordinates, has the form

$$\begin{aligned} H \;=\; & -\frac{\hbar^2}{2\,\mu}\left[\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial}{\partial r}\right) + \frac{1}{r^2\,\sin\,\theta}\frac{\partial}{\partial\theta}\left(\sin\,\theta\frac{\partial}{\partial\theta}\right)\right. \\ & \left. + \frac{1}{r^2\,\sin^2\,\theta}\frac{\partial^2}{\partial\varphi^2}\right] + V(r)\,. \end{aligned} \tag{2.3}$$

The eigenfunctions $\psi(\vec{r})$ can be written as

$$\psi(\vec{r}) = R_l(r)\,Y_{lm}(\theta,\varphi) = \frac{1}{r}\,u_l(r)\,Y_{lm}(\theta,\varphi)\,. \tag{2.4}$$

Substituting expressions (2.3) and (2.4) into Eq. (2.1) yields two separate equations, the radial and angular eigenvalue equations. The angular equation can be readily solved for the functions $Y_{lm}(\theta,\varphi)$ called the *spherical harmonics.* The subscripts l and m are the *orbital angular momentum* and *magnetic* quantum numbers, respectively. These numbers are related to the angular eigenvalues corresponding to the functions $Y_{lm}(\theta,\varphi)$.

The radial equation for $u_l(r)$ reads

$$-\frac{\hbar^2}{2\mu}\frac{\mathrm{d}^2 u_l(r)}{\mathrm{d}r^2}+\left[\frac{\hbar^2}{2\mu}\frac{l(l+1)}{r^2}+V(r)\right]u_l(r)=E\,u_l(r)\,. \tag{2.5}$$

This equation is simplified by writing it in the form

$$\frac{\mathrm{d}^2 u_l(k,r)}{\mathrm{d}r^2}-\left[\frac{l(l+1)}{r^2}+U(r)\right]u_l(k,r)=-\,k^2\,u_l(k,r)\,, \tag{2.6}$$

where

$$U(r)=\frac{2\,\mu}{\hbar^2}\,V(r)\quad\text{and}\quad k^2=\frac{2\,\mu}{\hbar^2}\,E\,. \tag{2.7}$$

There are very few potentials $U(r)$ for which the exact solutions to Eq. (2.6) can be obtained and thus numerical methods are employed to obtain approximate solutions.

2.1.2 Boundary Conditions

Most numerical methods for solving equations like Eq. (2.6) require prior knowledge of the expected behavior of the solutions $u_l(k,r)$ in certain regions of space. Such behavior can be determined by switching off the potential $U(r)$. When $U(r)=0$ Eq. (2.6) has two sets of linearly independent solutions. One set consists of the Riccati-Bessel and the Riccati-Neumann functions, $j_l(k\,r)$ and $n_l(k\,r)$, while the other set is made up of the Riccati-Hankel functions of the first and the second kind, $h_l^{(1)}(k\,r)$ and $h_l^{(2)}(k\,r)$, respectively. These functions are related by the equation [16]

$$\begin{aligned} j_l(k\,r) &= \frac{1}{2}\left[h_l^{(1)}(k\,r)+h_l^{(2)}(k\,r)\right]\,, && (2.8)\\ n_l(k\,r) &= \frac{1}{2\,i}\left[h_l^{(1)}(k\,r)-h_l^{(2)}(k\,r)\right]\,, && (2.9)\\ h_l^{(1,2)}(k\,r) &= j_l(k\,r)\pm i\,n_l(k\,r)\,. && (2.10)\end{aligned}$$

The solution $j_l(k\,r)$ is referred to as *regular* (finite at the point $r=0$) and the others as *irregular*. The boundary conditions of the general solutions to Eq. (2.6) are derived from the behavior of the functions $j_l(k\,r)$ for small r and $h_l^{(1,2)}(k\,r)$ for large r, respectively.

The boundary behavior of these functions has the form [16]

$$j_l(z) \;\underset{z\to 0}{\longrightarrow}\; C_1\, z^{l+1} \tag{2.11}$$

$$h_l^{(1,2)}(z) \;\underset{z\to\infty}{\longrightarrow}\; \frac{C_2}{z} \exp\Big[\pm i \Big(z - (l+1)\pi/2\Big)\Big] \,, \tag{2.12}$$

where C_i are constants and $z = k\, r$.

If the potential $U(r)$ is different from zero and it fulfills the conditions

$$\lim_{r\to 0} r^2\, U(r) = 0 \tag{2.13}$$

and

$$\lim_{r\to\infty} r\, U(r) = 0\,, \tag{2.14}$$

then the regular solutions to Eq. (2.6) are defined through the boundary conditions

$$u_l(k,r) \;\underset{r\to 0}{\longrightarrow}\; j_l(k,r)\,, \tag{2.15}$$

and

$$u_l(k,r) \;\underset{r\to\infty}{\longrightarrow}\; f_1(k,l)\, h_l^{(1)}(k\, r) + f_2(k,l)\, h_l^{(2)}(k\, r)\,, \tag{2.16}$$

where $f_i(k,l)$ are the so-called *Jost* functions.

For bound systems the probability density, $|\,\psi(\vec{r})\,|^2$, is required to be finite everywhere. This implies that the bound state solutions $u_l(k,r)$ must be normalized so that

$$N^2 \int_0^\infty |\, u_l(k,r)\,|^2\, \mathrm{d}r = 1\,, \tag{2.17}$$

where N is a normalization constant. Also, this condition is fulfilled only when $u_l(k_b, r)$ obeys relations (2.15) and (2.16) with $f_2(i\, k_b) = 0$, where $k_b \,>\, 0$ and

$$k_b^2 = -\,\frac{\hbar^2}{2\,\mu}\,\mathcal{E}\,, \qquad \mathcal{E} \,>\, 0\,. \tag{2.18}$$

Therefore, the bound state eigenfunctions to Eq. (2.6) satisfy the boundary conditions

$$u_l(k_b, 0) = u_l(k_b, \infty) = 0\,. \tag{2.19}$$

2.2 The Dirac Equation

The material presented in this section is intended to serve only as a comprehensive summary of the structure of the Dirac equation for stationary states in a given system.

The time-independent Dirac equation for a particle of mass m in a central Lorentz vector $V_v(r)$ and scalar $V_s(r)$ potentials has the form (the units are such that $\hbar = c = 1$)

$$\Big\{ \vec{\alpha} \cdot \vec{p} + \beta \, [\, m + V_s(r) \,] + V_v(r) \Big\} \, \psi(\vec{r}) = E \, \psi(\vec{r}) \tag{2.20}$$

where E is the total energy and $\vec{p}$ the momentum operator of the system. The Dirac matrices, $\vec{\alpha}$ and β, are defined by

$$\vec{\alpha} = \begin{bmatrix} 0 & \vec{\sigma} \\ \vec{\sigma} & 0 \end{bmatrix} \quad \text{and} \quad \beta = \begin{bmatrix} \mathbf{1} & 0 \\ 0 & -\mathbf{1} \end{bmatrix} . \tag{2.21}$$

where $\vec{\sigma}$ are the Pauli matrices and $\mathbf{1}$ denotes the unit matrix. We parameterized $\psi(\vec{r})$ as follows

$$\psi(\vec{r}) = N \begin{bmatrix} R_+(r) \, \Omega_l^j(\frac{\vec{r}}{r}) \\ - \, i \, R_-(r) \, \Omega_{l'}^j(\frac{\vec{r}}{r}) \end{bmatrix} , \tag{2.22}$$

where Ω_l^j are two-component angular spinor functions for a state with orbital angular momentum l and total angular momentum j. N is an appropriate normalization factor defined by the condition

$$\int \psi^*(\vec{r}) \, \psi(\vec{r}) \, \mathrm{d}\vec{r} = 1 \, , \tag{2.23}$$

where ψ^* is the complex conjugate of ψ. Substituting Eq. (2.22) into Eq. (2.20) yields a set of coupled equations

$$i \, \vec{\sigma} \cdot \vec{p} R_+(r) \, \Omega_l^j \;\; = \;\; \Big[E + m + V_s(r) - V_v(r) \Big] \, R_-(r) \, \Omega_{l'}^j \, , \tag{2.24}$$

$$-i \, \vec{\sigma} \cdot \vec{p} R_-(r) \, \Omega_{l'}^j \;\; = \;\; \Big[E - m - V_s(r) - V_v(r) \Big] \, R_+(r) \, \Omega_l^j \, . \tag{2.25}$$

We are interested only in the radial component of these equations. To separate the radial component from the angular one we proceed as follows. Using the relations [17]

$$\left(\vec{\sigma}\cdot\frac{\vec{r}}{r}\right)\Omega_l\left(\frac{\vec{r}}{r}\right) = -\Omega_{l'}\left(\frac{\vec{r}}{r}\right), \qquad \vec{p}\cdot\frac{\vec{r}}{r} = -i\,\frac{2}{r}$$

and the vector identity

$$\vec{\sigma}\cdot\vec{A}\,\vec{\sigma}\cdot\vec{B} = \vec{A}\cdot\vec{B} + i\,\vec{\sigma}\cdot\vec{A}\times\vec{B}\,,$$

it can be shown that [17]

$$\begin{aligned}\vec{\sigma}\cdot\vec{p}\,\Omega_l &= \frac{i}{r}\left[2+\vec{L}\cdot\vec{\sigma}\right]\Omega_{l'} &(2.26)\\ &= \frac{i}{r}\,(1-\kappa)\,\Omega_{l'} &(2.27)\end{aligned}$$

and

$$\vec{\sigma}\cdot\vec{p}\,\Omega_{l'} = \frac{i}{r}\,(1+\kappa)\,\Omega_l\,, \tag{2.28}$$

where $\kappa = \mp(j+1/2)$. Therefore

$$\begin{aligned}i\,\vec{\sigma}\cdot\vec{p}\,R_+(r)\,\Omega^j_l &= i\,[\vec{\sigma}\cdot\vec{p}\,R_+(r)]\,\Omega^j_l + i\,R_+(r)\,\vec{\sigma}\cdot\vec{p}\,\Omega^j_l &(2.29)\\ &= -\left[\frac{\mathrm{d}R_+(r)}{\mathrm{d}r} + \frac{1-\kappa}{r}\,R_+(r)\right]\Omega^j_{l'} &(2.30)\end{aligned}$$

and

$$-\,i\,\vec{\sigma}\cdot\vec{p}\,R_-(r)\,\Omega^j_{l'} = \left[\frac{\mathrm{d}R_-(r)}{\mathrm{d}r} + \frac{1+\kappa}{r}\,R_-(r)\right]\Omega^j_l\,. \tag{2.31}$$

Substituting relations (2.30) and (2.31) into equations (2.24) and (2.25), respectively, reveals that the radial functions $R_+(r)$ and $R_-(r)$ satisfy the radial equations

$$\begin{aligned}-\left(\frac{\mathrm{d}}{\mathrm{d}r} + \frac{1-\kappa}{r}\right)R_+(r) &= \left[E+m+V_s(r)-V_v(r)\right]R_-(r)\\ \left(\frac{\mathrm{d}}{\mathrm{d}r} + \frac{1+\kappa}{r}\right)R_-(r) &= \left[E-m-V_s(r)-V_v(r)\right]R_+(r)\,, &(2.32)\end{aligned}$$

For the ground state, $\kappa = 1$. Using the transformation $f(r) = rR_+(r)$ and $g(r) = rR_-(r)$ it follows that

$$\frac{\mathrm{d}f}{\mathrm{d}r} = R_+ + r\,\frac{\mathrm{d}R_+}{\mathrm{d}r} \quad \text{and} \quad \frac{\mathrm{d}g}{\mathrm{d}r} = R_- + r\,\frac{\mathrm{d}R_-}{\mathrm{d}r}$$

hence the equations (2.32) take the form

$$\begin{aligned} -\left(\frac{\mathrm{d}}{\mathrm{d}r} - \frac{\kappa}{r}\right) f(r) &= \Big[E + m + V_s(r) - V_v(r)\Big]\, g(r) \\ \left(\frac{\mathrm{d}}{\mathrm{d}r} + \frac{\kappa}{r}\right) g(r) &= \Big[E - m - V_s(r) - V_v(r)\Big]\, f(r)\,, \end{aligned} \tag{2.33}$$

In this work we solve these equations numerically using artificial neural network methods.

2.3 Variational Methods

The Rayleigh-Ritz variational method has proved reliable in estimating ground state solutions to eigenvalue problems of many systems. The description of this method is now standard textbook material. In this section we, therefore, highlight the basic properties of this method.

Given an eigenvalue problem

$$H\,\psi = E\,\psi\,, \tag{2.34}$$

the ground state energy eigenvalue E_0 of the Hamiltonian H can be estimated via the functional

$$\langle\, H\,\rangle = \frac{\int \psi^*\, H\, \psi\, \mathrm{d}r}{\int \psi^*\, \psi\, \mathrm{d}r}\,, \tag{2.35}$$

where ψ^* denotes the complex conjugate of ψ. This can be trivially shown (for non-degenerate states) by expanding the orthogonal eigenfunctions of H, ψ_n, in terms of a complete set of basis functions ϕ_n in the form

$$\psi_n = \sum_n c_n\, \phi_n\,, \tag{2.36}$$

where n = 0,1,2,... and c_n are the variational parameters. Then Eq. (2.35) yields

$$\langle\, H\,\rangle = \frac{\sum_n \mid c_n \mid^2 E_n}{\sum_n \mid c_n \mid^2}\,. \tag{2.37}$$

Since $E_n \geq E_0$ then

$$\langle H \rangle - E_0 = \frac{\sum_n |\, c_n \,|^2 \, [\, E_n - E_0 \,]}{\sum_n |\, c_n \,|^2} \geq 0 \,. \tag{2.38}$$

Therefore, Eq. (2.35) is and upper bound to the ground state energy E_0.

The variational approach involves finding a function ψ such that the variation of Eq. (2.35) vanishes *i.e.*

$$\delta \langle H \rangle = 0 \,. \tag{2.39}$$

The technique requires an informed guess of the trial function, ψ_t, containing adjustable parameters, that can be varied to minimize $\langle H \rangle$. The procedure of adjusting these parameters varies from method to method [18, 19]. However, for most methods (e.g. basis expansion), it essentially begins with choosing the initial values for the least possible number of parameters. The chosen set of parameters is then appropriately varied to obtain the minimum of $\langle H \rangle$. With the first parameter set fixed, a new set is initialized and varied to further minimize $\langle H \rangle$. The lowest value of $\langle H \rangle$ thus obtained is the best estimate of the ground state energy and ψ_t the corresponding ground state wave function. If ψ differs greatly from the ground state of H then equation (2.35) yields misleading results [16]. This approach of strategic fixing and increasing the number of variational parameters, although effective, can be very demanding in terms of computer memory and time [18, 19].

For the variational technique to work, the eigenvalue spectrum of the problem considered must have an upper or lower bound. Fortunately, this requirement holds for the Schrödinger equation. For the Schrödinger equation, definition (2.35) translates into [16]

$$E = N_s \int \left[-\frac{\hbar^2}{2\mu} \, u_l(r) \, \frac{\mathrm{d}^2 \, u_l(r)}{\mathrm{d}r^2} + V(r) \, u_l^2(r) \right] \mathrm{d}r \tag{2.40}$$

$$= N_s \int \left[\frac{\hbar^2}{2\mu} \left(\frac{\mathrm{d} \, u_l(r)}{\mathrm{d}r} \right)^2 + V(r) \, u_l^2(r) \right] \mathrm{d}r \,, \tag{2.41}$$

where use has been made of the fact that $u_l(r)$ is real for bound states. The normalization constant N_s is given by

$$N_s = \left[\int_0^\infty u_l^2(r)\, \mathrm{d}r \right]^{-1} . \tag{2.42}$$

The dependence of $u_l(r)$ on variational parameters is implied.

The existence of both positive- and negative-energy solutions for the Dirac equation, on the other hand, indicates that the corresponding energy eigenvalue spectrum is not bounded. Therefore the straightforward application of the Rayleigh-Ritz variational technique in determining the ground state solutions of the Dirac equation constitute an ill-posed problem. However, with appropriate modification, the variational technique can be applied to the Dirac equation [20]. Most of such modifications, however, can be implemented by simple choices of the trial solutions. In this work we employ the form used in Ref. [14],

$$E = N_r \int \left\{ \left[m + V_s(r) \right] \left[f^2(r) + g^2(r) \right] \; + \; V_v(r) \left[f^2(r) - g^2(r) \right] \right\} \mathrm{d}r \,, \tag{2.43}$$

where

$$N_r = \left\{ \int_0^\infty \left[f^2(r) - g^2(r) \right] \mathrm{d}r \right\}^{-1} . \tag{2.44}$$

Again, the fact that the wave functions are real for bound states has been considered. This form of the variational energy is obtained by rearranging the sum of the coupled radial equations.

Chapter 3

Artificial Neural Network Methods

Artificial neural networks (ANNs) are devices constructed from weighted interconnections of processing units or *artificial neurons.* The terminology, structure, and processing mechanism of these devices are derived mainly from the biological nervous system. Even the explanation of their processing mechanism is simpler from the perspective of the biological nervous system. These devices have seen application in many disciplines. One of the many applications is the *function approximation*, which can be extended to solutions of differential equations.

The use of artificial neural networks in solving differential equations involves approximating the solution with a function that depends, directly or indirectly, on neural networks. An appropriate error function is then formulated in terms of the trial solution. The resulting error function, therefore, depends on the network parameters. Thus, to obtain the required approximate solution the error function is minimized with respect to these parameters. This procedure highlights general features of a typical artificial neural network numerical method. Neural network methods differ mainly by the technique of constructing trial solutions to a given problem and the way the trial solution relates to the chosen network. The effectiveness of the method relies essentially on these techniques.

In what follows we introduce the general features of neural networks and the accompanying terminology. In section 3.1 we summarize features and the mathematical realization of the basic operation of an artificial neuron. In section 3.2 we introduce the concept of neural networks where we highlight their basic attributes. In section 3.3 we summarize key aspects of the learning process. A summary of the properties of the optimization environment employed in this work, *Merlin* [21], is presented in this section. In section 3.4 we discuss the utilization of artificial neural networks as numerical tools. A reader interested in the details of the subject of artificial neural networks is referred to the relevant texts in the literature, see, for example, Refs. [22, 23] and references therein.

3.1 The Artificial Neuron

The basic operation of an artificial neuron can be mathematically described as mapping two different classes of vectors that are in the same space. This mapping is accomplished by the use of the so-called *activation* or *transfer function* embedded into the neuron. The input-output mechanism of an artificial neuron is generally described by the relation

$$y = v\, f(w\, x + u)\,, \tag{3.1}$$

and is illustrated by the block diagram in Fig. 3.1. Here x represents the input, v, w, and u are the connection weights, f the transfer function, and y the output of the neuron. The *input* weight w and the *bias* weight u transform x into a different variable, say z, with w scaling x and u setting the new origin $z = 0$ in the domain of x. The *output* weight v scales f. The scaling process (obtaining the desired values of the weights) is called *training* or *learning.* The internal processing mechanism, during training, of an artificial neuron can be conceptualized as in Fig. 3.2. In region 1 the inputs are summed and weighed. Then the weighed sum is transfered to region 2 where it is transformed according to the transfer function of the neuron. Depending on the threshold T the results of the transformation are weighed and reflected as the overall output of the neuron.

Any function with appropriate characteristics can be employed as a transfer function for a given neurons. The most frequently used functions for this purpose are :

- The step function

$$f(z) = \begin{cases} 1: & z \geq 0 \\ 0: & z < 0\,, \end{cases} \tag{3.2}$$

- The Piecewise-linear function

$$f(z) = \begin{cases} 1: & z > a \\ z: & -a \leq z \leq a \\ -1: & z < -a\,, \end{cases} \tag{3.3}$$

- The sigmoid function

$$f(z) = \frac{1}{1 + \mathrm{e}^{-z}}\,, \tag{3.4}$$

  This function is continuous and bounded in (0,1). The sigmoid function is also infinitely differentiable. Given the first derivative of the sigmoid function

$$\frac{\mathrm{d}\,\sigma}{\mathrm{d}\,z} = \sigma\,(1 - \sigma)\,, \tag{3.5}$$

  higher derivatives can be readily generated. The above mentioned scaling effects, when the transfer function is chosen to be the first derivative of the sigmoid function, are illustrated graphically in Fig. 3.3.

- The hyperbolic tangent function

$$f(z) = \frac{1 - \mathrm{e}^{-2z}}{1 + \mathrm{e}^{-2z}}\,. \tag{3.6}$$

  is continuous and bounded in (-1,1).

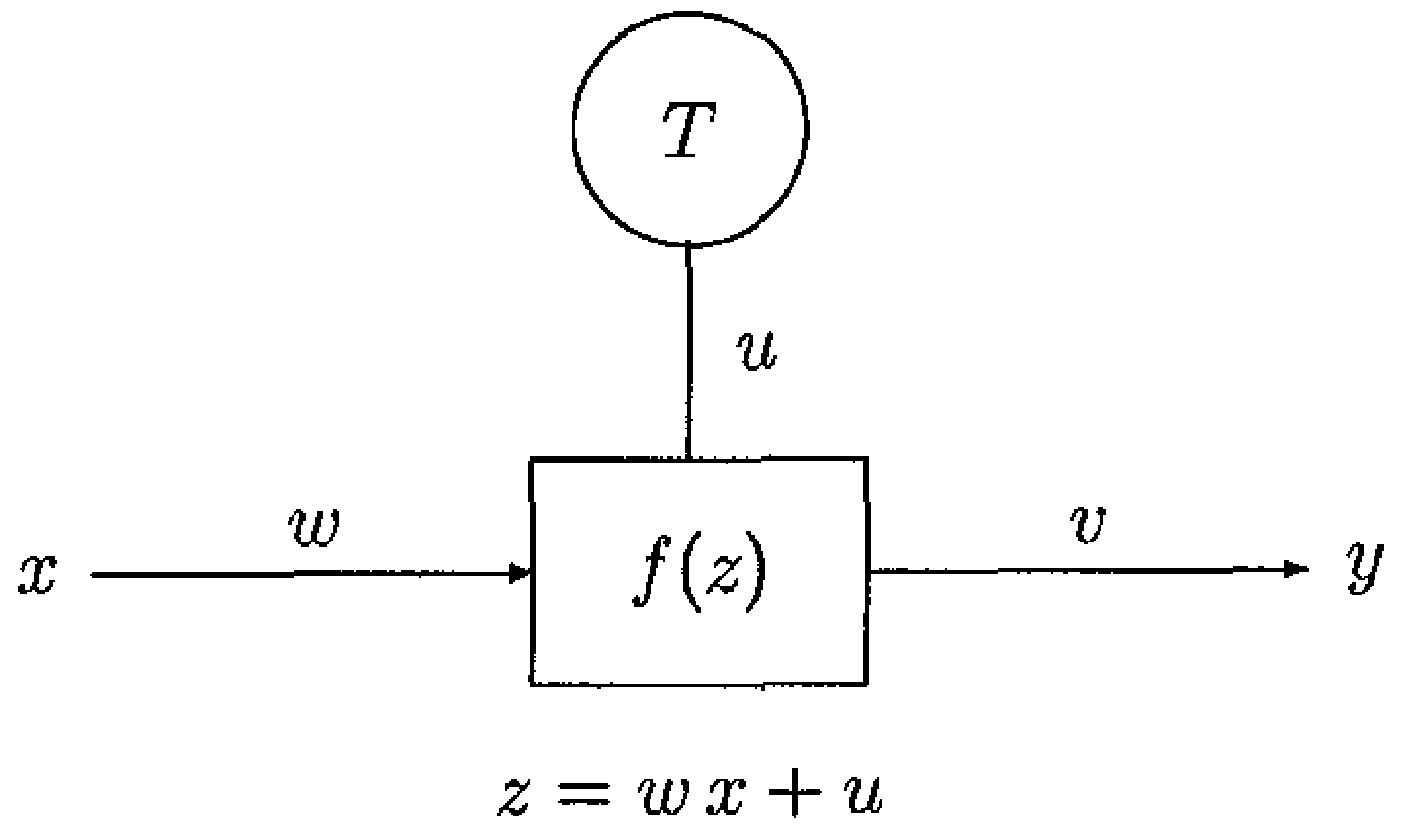

Figure 3.1: An schematic representation of an artificial neuron.

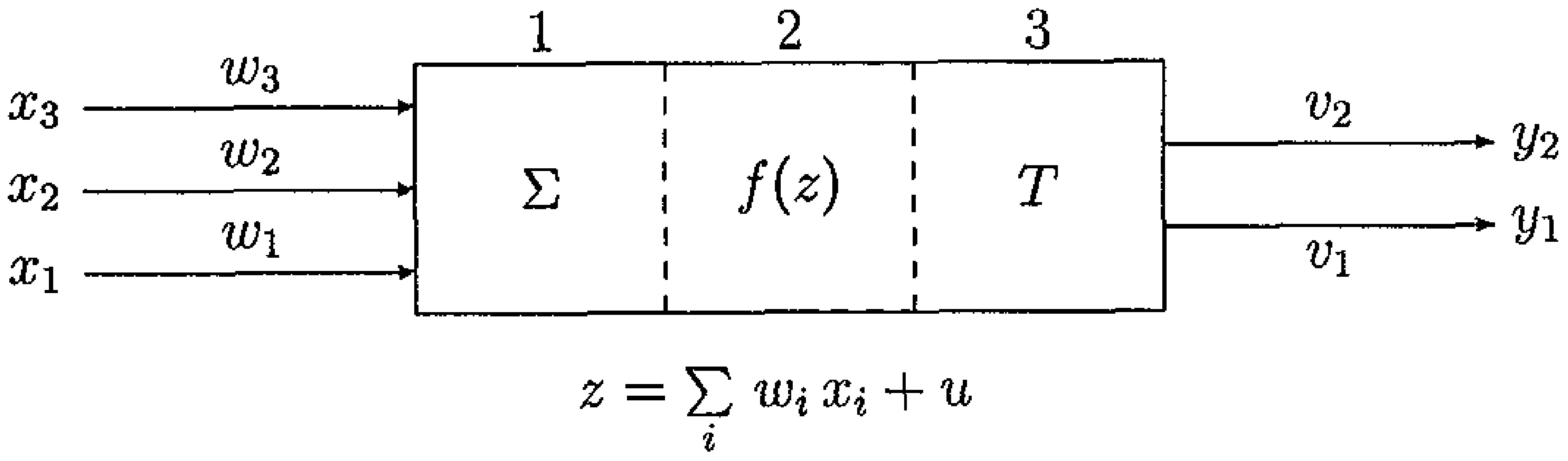

Figure 3.2: A conceived internal processing mechanism of an artificial neuron.

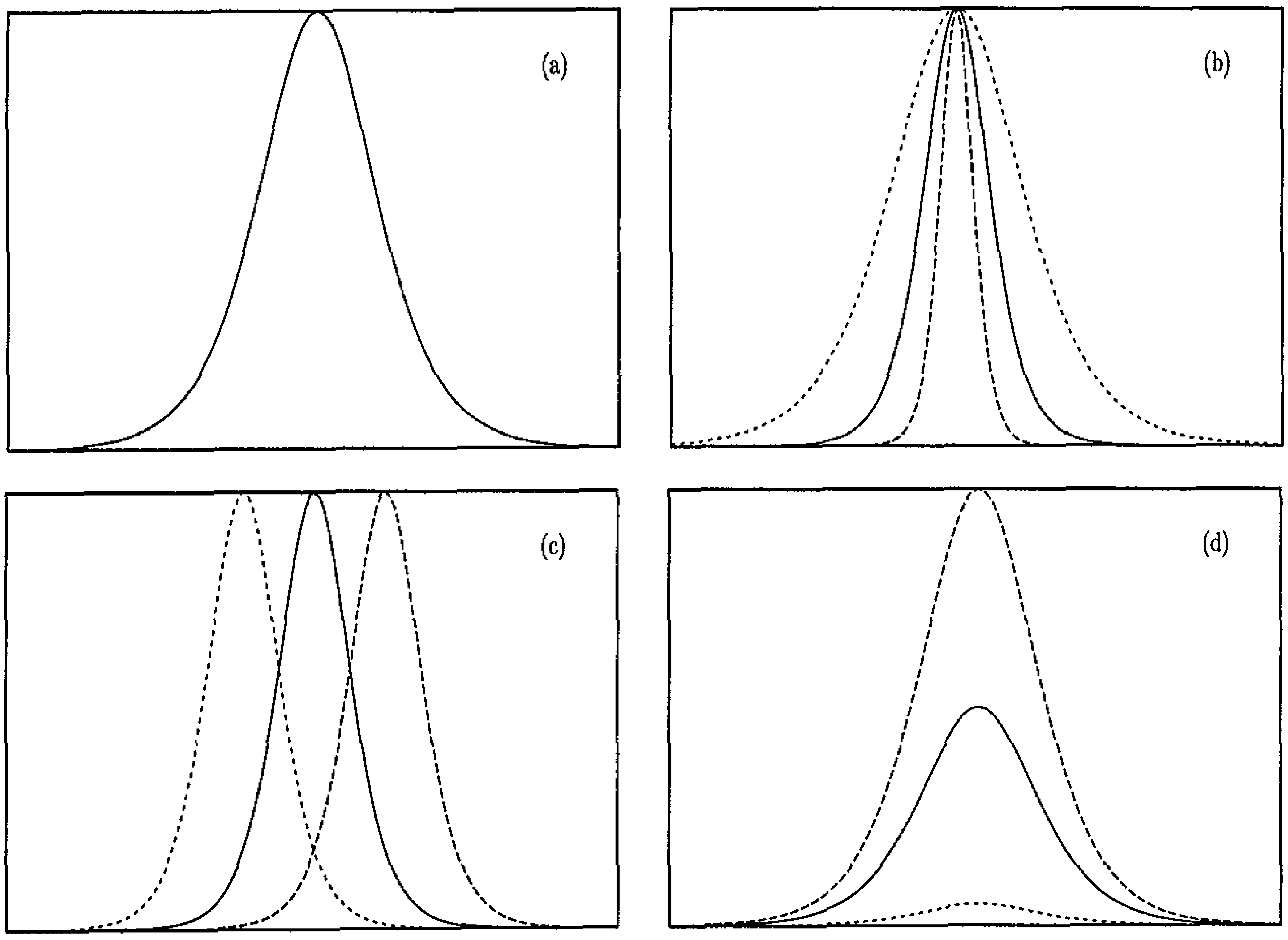

Figure 3.3: (a) The first derivative of the sigmoid. The effect of varying (b) w, (c) u and (d) v on the function shown in (a) when the function is used in neuron activation.

The choice of a transfer function is generally informed by the role of the neuron and the task at hand.

3.2 A Multilayer Perceptron

For increased efficiency, artificial neurons are usually connected to form clusters, *artificial neural networks*. The processing mechanisms of such networks are generally similar to that of a single neuron. These networks, in general, possess structure (architecture) that is determined by the arrangement of and the role assigned to the constituent neurons. The classification of artificial neural networks is generally based on the architecture and the input processing mechanism. In this work we employ a layered network in the approximation of the solutions to differential equations. As a result we summarize general aspects of this type of networks.

A layered network is constructed by arranging all the neurons performing the same function in layers. This architecture is illustrated in Fig. 3.4. These networks are further classified according to their input-output processing mechanisms. For instance, a *feedforward* network is one in which processing is progressive, one-directional only. That is, the output of neurons in a given layer serves as input only to neurons in the proceeding layer. This type of network consists of an input layer, an output layer and one or more hidden layers, with no interconnections among neurons in the same layer. A *recurrent* network, on the other hand, allows for forward and backward processing, as illustrated in Fig. 3.5. When the output of neurons in a given layer does not satisfy certain predetermined conditions, then such an output is propagated backward and fed back into the input layer, in this type of network. *Radial basis function* networks employ radial basis functions as transfer functions of the hidden layer. The most general network architecture is the Williams-Zipser architecture [24], which does not assign fixed roles to the network neurons.

A *feedforward* artificial neural network (or *multilayer perceptron*)[1] is popular in the field of function approximation. In function approximation, it is customary to use linear transfer functions, (3.3), in the input and output layers, respectively. The hidden layer(s), on the other hand, require(s) nonlinear transfer functions to alter the representation of the inputs. The output of a three-layer feedforward network, $N(\vec{r},\vec{p})$, for a given input vector,

$$\vec{r} = (r_1, r_2, \ldots, r_n)\,, \tag{3.7}$$

[1] The meaning of the terms *input-,output-*, and *hidden-layer* are illustrated in Fig. 3.4.

is described by the weighted sum

$$N(\vec{r}, \vec{u}, \vec{v}, w) = \sum_{i=1}^{m} v_i \, f(z_i) \,, \tag{3.8}$$

where $f(z)$ is the transfer function of the hidden layer and

$$z_i = \sum_{j=1}^{n} w_{ij} \, r_j + u_i \,, \tag{3.9}$$

is the transformation variable. The output of the four-layer network, illustrated in Fig. 3.4, is given by

$$N(\vec{r}, \vec{u}, \vec{v}, \hat{w}) = \sum_{i=1}^{m_1} v_i \, f_2 \left[\sum_{j=1}^{m_2} w_{ij}^{(2)} \, f_1(z_j) - u_i \right] \,, \tag{3.10}$$

with

$$z_j = \sum_{k=1}^{n} w_{jk}^{(1)} \, r_k + u_j \,, \tag{3.11}$$

and $\hat{w}$ is an array of matrices. For every additional hidden layer the corresponding output of the network is obtained by replacing z with $f(z)$ in Eq. (3.10).

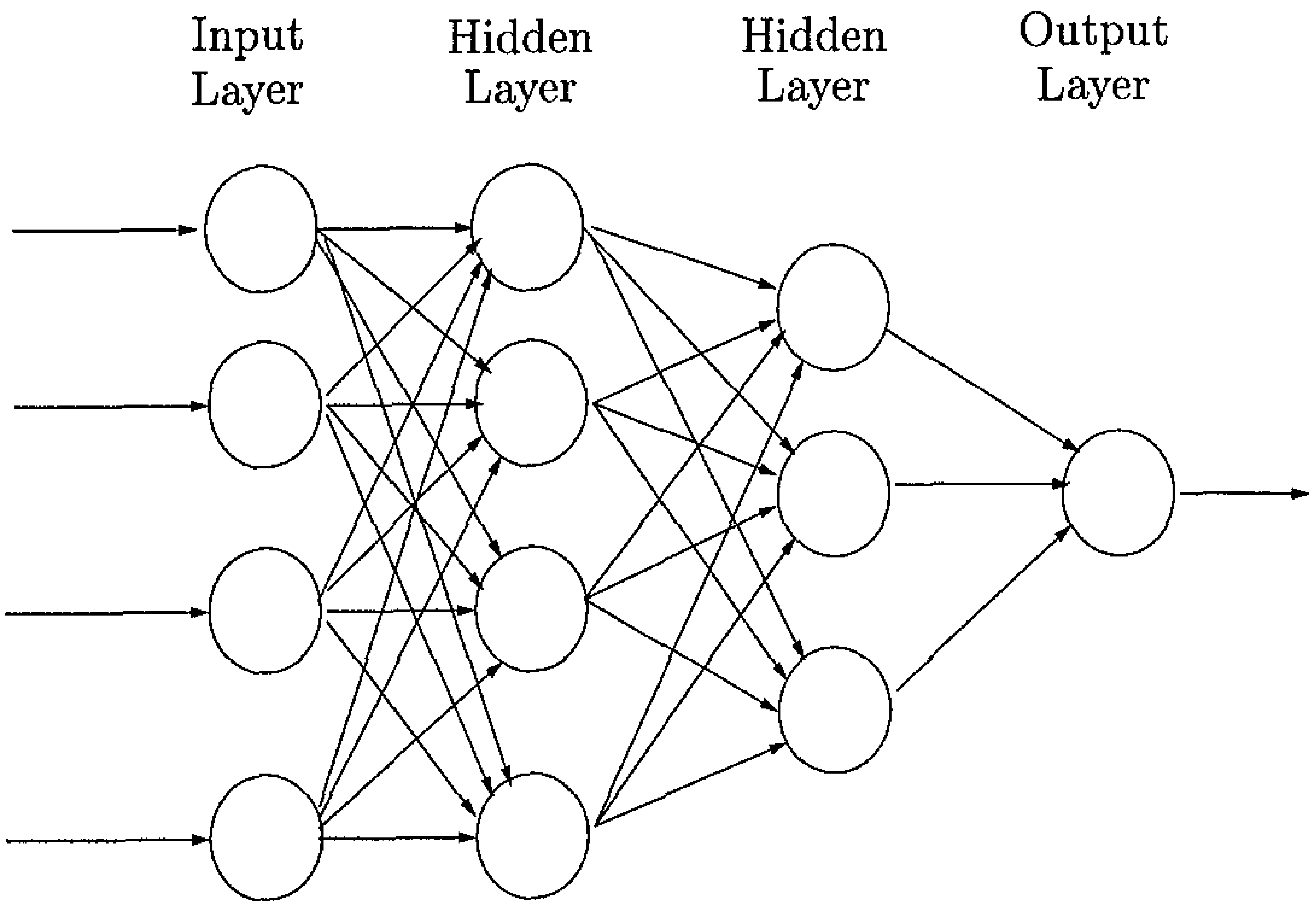

Figure 3.4: A typical architecture of a multilayer feedforward artificial neural network with two hidden layers.

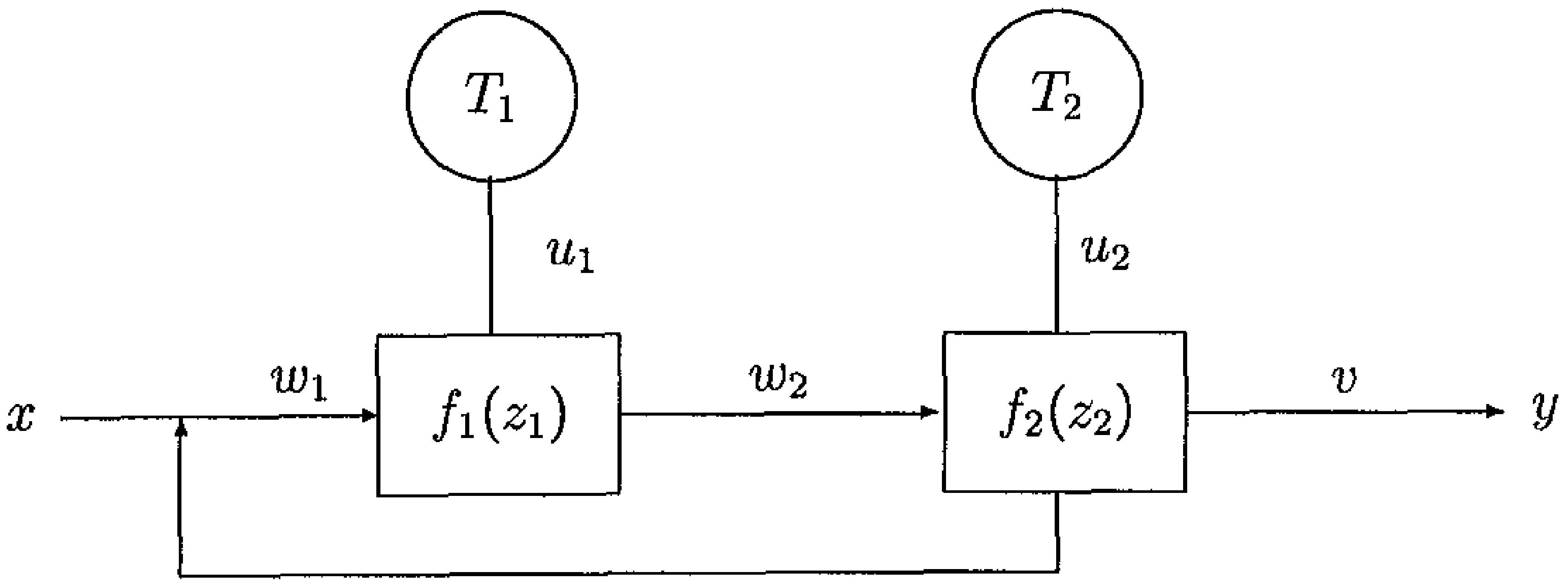

Figure 3.5: An illustration of structural connections of neurons allowing for reprocessing of the unsatisfactory output.

3.3 Minimization

The use of artificial neural networks in *function approximation* can be derived from the generalization of concepts in curve-fitting. Suppose that we wish to approximate a given function $f(x)$ with a neural network $g(x,\vec{p})$, where $\vec{p}$ are the network weights. This task can be achieved by minimizing the *error distance*

$$e(\vec{p}) = f(x) - g(x,\vec{p})\,. \tag{3.12}$$

It is found that minimizing the function

$$L(\vec{p}) = \int e^2(\vec{p})\,\mathrm{d}x\,, \tag{3.13}$$

instead, is more effective. However, in practice, the simpler form

$$L(\vec{p}) = \sum_i e^2(x_i,\vec{p})\,, \tag{3.14}$$

is the one minimized. Minimization in this case becomes the result of training the network.

Learning can be described by the general relation

$$\vec{p}(n+1) = \vec{p}(n) + \Delta\,\vec{p}\,, \tag{3.15}$$

between the given value of the parameters, $\vec{p}(n)$, and the changed (*updated*) one $\vec{p}(n+1)$. The techniques of determining the desired values of the *updates* $\Delta\vec{p}$ are referred to as *learning algorithms.* Many such algorithms exist and are available in the literature [23]. However, for convenience, most of the algorithms have been compiled into computer software packages that are also available to the research community. One such software is *Merlin,* [21]. This particular software package offers a portable, open and integrated optimization environment with numerous desirable features. Its main objective is to locate a local minimum of a real function that depends on some variable parameters, within suitable constraints. The error functions (3.12) and (3.14) belong to those functions that can be minimized via *Merlin.*

Some of the key features of Merlin include an *operating system* and a *control language* to help in devising suitable optimization strategies, and a variety of optimization algorithms. The optimization algorithms included are direct, gradient, and conjugate gradient methods. The direct methods are the *Roll* method and *Simplex* (or Polytope) method. The gradient methods are the *Quasi-Newton* methods that use the so-called BFGS and DFP updates [21]. The different implementations of these methods are

- BFGS - uses a line search approach with the Choleski factorization of the Hessian matrix.
- TRUST - uses a trust region search approach and maintains the Choleski factorization.
- TOLMIN - uses a line search approach with Goldfarb-Idnani factorization of the Hessian matrix.

The conjugate gradient methods are those of Fletcher-Reeves and Polak-Ribiere (including the generalized Polak-Ribiere method). The Levenberg-Marquardt method is included for sum of squares.

This software is maintained and supported. It is also freely available to the scientific community. It can be downloaded from :

http://nrt.cs.uoi.gr/merlin/

All the minimization in this dissertation is done using Merlin.

3.4 Numerical Applications

Some methods of solving differential equations involve approximating the differential operators in the equation by some finite difference equations. Such difference equations are derived from the Taylor expansion of the solution about a suitable point. The resulting system of equations can then be transformed into matrix equations, which can then be solved by using relevant standard techniques. Some of the previous applications of ANNs as computational tools involved solving such systems of equations. The system of equations was mapped onto an ANN. The energy error function of the network was then minimized to extract the desired solution [9, 11].

3.4.1 Neural Splines

The solution, $\phi(r)$, to a differential equation can be expanded in terms of a known linearly independent set of (basis) functions $s_i(r)$

$$\phi(r) = \sum_i^n v_i \, s_i(r) \, , \tag{3.16}$$

where v_i are the expansion coefficients. There are different types of basis functions available for use in this expansion. One of them are the *spline functions*, which are said to offer a lot of advantages [25]. A *spline* function is a piecewise polynomial which is non-zero only inside a small interval. The optimal degree of the polynomial chosen is determined by the requirement that the polynomial should have a continuous kth derivative for it to approximate efficiently the solution to a $(k+1)$th order differential equation.

The piecewise-linear functions, Eq. (3.3), can be combined to construct Chapeau functions [13],

$$\begin{aligned}
\chi_i(x) &= \frac{x - x_{i-1}}{x_i - x_{i-1}}\,, \qquad x_{i-1} \le x \le x_i\,,\\
\chi_i(x) &= \frac{x - x_i}{x_{i+1} - x_i}\,, \qquad x_i \le x \le x_{i+1}\,,\\
\chi_i(x) &= 0\,, \qquad \text{otherwise}\,.
\end{aligned} \tag{3.17}$$

This is accomplished by adding two piecewise-linear functions with certain constraints imposed on the weights. As a *neural spline* such a construction involves adding the output of two neurons to obtain

$$s_i(x) = v_i^A\, f_A(z_i^A) + v_i^B\, f_B(z_i^B)\,. \tag{3.18}$$

and then require the output weights to relate in the form

$$v_i^A = -\,v_i^B = \frac{1}{2}\,v_i\,.$$

The labels A and B denote two adjacent intervals in which the piecewise-linear functions are non-zero. The sum of the linear transfer functions leads to

$$z_i^A(x) - z_i^B(x) = 2\,\chi_i(x) - 1\,, \tag{3.19}$$

whence the input and bias weights are given by

$$w_i^A = \frac{2}{x_i - x_{i-1}}\,, \qquad u_i^A = \frac{2\,x_{i-1}}{x_i - x_{i-1}} - 1\,, \tag{3.20}$$

$$w_i^B = \frac{2}{x_{i+1} - x_i}\,, \qquad u_i^B = \frac{2\,x_i}{x_{i+1} - x_i} - 1\,. \tag{3.21}$$

To obtain the required solution only the expansion coefficients v_i need to be determined. In Ref. [13] an inner product of the differential equation with some weight function was used to construct a system of algebraic equations which were then solved by standard techniques of solving matrix equations.

3.4.2 The ANN Method

In the approaches summarized above the ANNs were employed not as the main approximating elements of the solutions to differential equations. Therefore, the accuracy of the resulting approximate solutions cannot be entirely attributed to the performance of ANNs. A method that employs ANNs as the main approximating elements of the solutions to differential equations was presented in Ref. [1]. In this work we investigate the performance of this method in the solution of eigenvalue problems involving differential equations.

Eigenvalue problems are described by the general equation

$$H\,\psi(r) = E\,\psi(r)\,. \tag{3.22}$$

Bound state solutions to this problem are characterized by discrete eigenvalues E and the specific behavior of the eigenstates $\psi(r)$. Eigenstates of a bound system vanish at the origin of the coordinate system, and at large distances. Solving this equation for bound states means determining the quantities E and $\psi(r)$.

The method presented in Ref. [1] relies on, firstly, discretizing the coordinate space. By doing so Eq. (3.22) is transformed into a system of equations

$$H\,\psi(r_i) = E\,\psi(r_i), \qquad r_i \in D\,. \tag{3.23}$$

where D defines the domain of the variable r. This set of equations is to be solved subject to the appropriate boundary conditions. This method uses the artificial neural network as the basic approximating element. In this case the solution to the eigenvalue problem is approximated by the output of the network. The construction of the approximate (trial) solution is done in such a way that the desired boundary conditions are satisfied automatically. To extract this solution the error function [1]

$$L(\vec{p}) = N_t \sum_i \left[H\,\psi_t(r_i,\vec{p}) - E\,\psi_t(r_i,\vec{p}) \right]^2, \qquad r_i \in D\,, \tag{3.24}$$

where

$$N_t = \left[\int |\,\psi_t(r_i,\vec{p})\,|^2\,\mathrm{d}r \right]^{-1},$$

is minimized by varying the network parameters $\vec{p}$, weights, subject to suitable constraints. The eigenvalue E is also expressed in terms of the parameters $\vec{p}$ via the Rayleigh-Ritz variational technique :

$$E(\vec{p}) = N_t \int \psi_t(r_i,\vec{p})\,H\,\psi_t(r_i,\vec{p})\,\mathrm{d}r \tag{3.25}$$

The lowest value of the function $L(\vec{p})$ thus attained yields the approximate solution to the problem. The ideal minimum of $L(\vec{p})$ in this case is zero. However, not any combination of parameters that yield the lowest minimum for the error function solve the problem. A very reliable minimization technique is thus required.

The explicit error function for the radial Schrödinger equation is

$$L(\vec{p}) = N_s \sum_i \left\{ -\frac{\hbar^2}{2\,\mu}\,\frac{\mathrm{d}^2\,\psi_t(r_i)}{\mathrm{d}r^2} + \left[V(r_i) - E \right] \psi_t(r_i) \right\}^2, \tag{3.26}$$

while for the Dirac equation we use

$$\begin{aligned} L(\vec{p}) \;=\; & N_d \sum_i \left\{ \left\{ \frac{\mathrm{d}\,f_t(r_i)}{\mathrm{d}r} - \frac{f_t(r_i)}{r_i} + \left[m + E + V_s(r_i) - V_v(r_i) \right] g_t(r_i) \right\}^2 \right. \\ & \left. + \left\{ \frac{\mathrm{d}\,g_t(r_i)}{\mathrm{d}r} + \frac{g_t(r_i)}{r_i} + \left[m - E + V_s(r_i) + V_v(r_i) \right] f_t(r_i) \right\}^2 \right\} \\ & . \end{aligned} \tag{3.27}$$

N_s (N_d) is the normalization coefficient of the Schrödinger (Dirac) trial solutions

$$\begin{aligned} N_s &= \left[\int_0^\infty \psi_t^2(r)\,\mathrm{d}r \right]^{-1} && (3.28) \\ N_d &= \left\{ \int_0^\infty \left[f_t^2(r) + g_t^2(r) \right] \mathrm{d}r \right\}^{-1} . && (3.29) \end{aligned}$$

These factors appear in the error function to ensure that the trial solutions are normalized.

In the method summarized above the type of the network used and the functional form of the trial solution remain arbitrary. In Ref. [14] the trial solutions were formulated in such a way that the behavior of these solutions near the boundaries,

$$\lim_{r\to 0} r^{-1} f(r) = 1 \quad \text{and} \quad \lim_{r\to\infty} \mathrm{e}^{a\,r} f(r) = 1\,, \tag{3.30}$$

where a is a constant, is explicitly included. The trial function has the general form

$$\psi_t(r,\vec{p}) = B(r,\,a)\,N(r,\vec{p})\,. \tag{3.31}$$

$N(r,\vec{p})$ is a multi-layer perceptron with one hidden sigmoid layer. The function $B(r,a)$ serves to enforce the correct behavior of the numerical solution at the boundaries. The constant a is treated as an additional adjustable parameter. The trial solutions can be expressed in the form

$$\psi_t(r,\vec{p}) = r\,N(r,\vec{p})\,, \tag{3.32}$$

where the asymptotic behavior of the trial solution is implied by the transfer function embedded in the hidden neurons. This parameterization, (3.32), is similar to expression (3.31) but with the factor B free from adjustable parameters. This increases the dependence of the trial solution on the network. We consider the first derivative of the sigmoid as the transfer function. Note that the first derivative of the sigmoid displays the asymptotic behavior,

$$\lim_{r\to\infty} N(r,\vec{p}) = 0\,, \tag{3.33}$$

as desired. The third derivative of the sigmoid, on the other hand, satisfies both the conditions (3.30). However, we choose the first derivative to establish the general effect of the form of the solution approximation. The input and output layers are linear.

In this work we compare the influence of the two forms of the trial solutions (3.31) and (3.32) on the accuracy of the numerical results. The method can be used also to compute excited states. These may be constructed in the form

$$\psi_{n+1}(r,\vec{p}) = \phi_t(r,\vec{p}) - \sum_{n=0} c_n\, \psi_n(r)\,, \tag{3.34}$$

where

$$c_n = \int \phi_t(x)\, \psi_n(x)\, \mathrm{d}x \tag{3.35}$$

and ψ_n are known normalized numerical states. In this case the functions ϕ_t are the ones formulated as in (3.31) or (3.32). With this construction the excited states are, in addition, automatically orthogonal to the lower states, as required.

Chapter 4

Results and Discussions

We applied the neural network method using a three-layer feedforward ANN with two different formulations of the trial solutions, as described in the previous chapter. Integrals were calculated using Gauss-Legendre quadratures and the networks were trained only at the integration points. To simplify discussions of our results we employed the following notations: We used N to denote the number of neurons in the hidden layer, $\Delta E = E_a - E_n$ the deviation of the numerical energy E_n from the analytical one E_a, and $\Delta u = u_a - u_n$ the deviation of the numerical wave function u_n from the analytical one u_a.

4.1 The Schrödinger Equation

In what follows we present results for the lowest, at most, three excited states of selected potentials. We generated the excited states from the most accurate of the lower numerical ones.

4.1.1 The Simple Harmonic Oscillator Potential

This potential is given by the simple form

$$V(x) = x^2 , \tag{4.1}$$

and is known to approximate the vibrational modes of most diatomic molecules quite well [26]. The variable x is dimensionless in this case. The Schrödinger equation for this potential has analytical solutions

$$u_n(x) = N_n \, \mathrm{e}^{-x^2/2} \, H_n(x) , \quad n = 0, 1, 2, \ldots , \tag{4.2}$$

where N_n is the normalization constant, $E_n = 2\,n+1$ are the corresponding energies and $H_n(x)$ the Hermite polynomials. We considered the domain [-10,10] with two hundred integration points. In the case of the *ansatz* (3.31) we approximated the boundary behavior of the trial solution by the form

$$B(x, a) = \mathrm{e}^{-a\,x^2} , \tag{4.3}$$

and used the sigmoid function for the transfer function of the hidden neurons. With this representation the *ansatz* (3.31) boils down to approximating just the Hermite polynomials. For the first excited state we exploited the knowledge that $u_1(0) = 0$ and formulated the problem independently from the ground state. We display the energies obtained for the lowest three states of this problem as a function of N in table 4.1 for the *ansatz* (3.32) while in table 4.2 we show the deviations of the numerical energies from the analytical values, as a function of N, for the *ansatz* (3.31). In Fig. 4.1 we show the graphs of the numerical wave functions of the *ansatz* (3.32) corresponding to the most accurate eigenvalue in table 4.1.

The results obtained with *ansatz* (3.32) display, in general, some familiar features. As seen in table 4.1 the accuracy in the energies generally converged to the analytical value as N is increased. This observation is in agreement with the findings of Ref. [2]. The accuracy in the energies of the excited states is generally lower than that of the ground state. Also, the rate of convergence appears to be slower in the case of the excited states. This is due to cumulative errors as the excited states are obtained using lower eigenstates.

It is customary to compare the deviations Δu and ΔE [28]. In Fig. 4.2 we present the deviations Δu. As can be observed from this figure the maximum Δu are generally less than $\sqrt{\Delta E}$. The deviations are very small around the minimum of the potential. This accuracy shows that this *ansatz* represents the correct solution quite well in this region. However, the deviations become significant toward the tail of the wave functions. The *ansatz* overestimates the asymptotic exponential decay of the exact solution. This can be attributed to the strong convergence of the derivative of the sigmoid. The *ansatz* (3.31), on the other hand, produces exceptionally accurate results, especially for the ground and first excited states. The deviations ΔE are shown in table 4.2. It is seen in this table that the calculated energies, to all practical purposes, for the ground and first excited states are exact. The second excited state was also computed within nine digits of accuracy. It is clear from these results that the way the trial solution is constructed is crucial.

Table 4.1: The energies for the lowest three states of the simple harmonic oscillator potential as functions of the number N of neurons in the hidden layer. These energies were obtained using the *ansatz* (3.32).

N	$E_0 = 1.$	$E_1 = 3.$	$E_2 = 5.$
1	1.0158384	3.0755280	6.0369568
2	1.0033040	3.0193306	5.0856387
3	1.0009653	3.0064398	5.0310690
4	1.0003345	3.0024667	5.0099337
5	1.0001285	3.0010273	5.0025493
6	1.0000531	3.0004557	5.0008563
7	1.0000232	3.0002114	5.0003645
8	1.0000106	3.0001018	5.0001464
9	1.0000050	3.0000012	5.0000714
10	1.0000024	3.0000258	5.0000344

Table 4.2: The deviations of the numerical energies, from the analytical values, for the lowest three states of the simple harmonic oscillator potential as functions of the number N of neurons in the hidden layer. The energies were obtained using *ansatz* (3.31). $a(-b) \Longrightarrow a \times 10^{-b}$.

N	ΔE_0	ΔE_1	ΔE_2
1	3.75(-16)	5.01(-16)	-1.98(-10)
2	6.26(-16)	0.	-3.25(-11)
3	6.26(-16)	-8.77(-16)	1.17(-10)
4	5.01(-16)	6.26(-16)	3.81(-11)
5	7.52(-16)	5.01(-16)	1.70(-10)
6	7.52(-16)	-5.01(-16)	4.39(-11)
7	2.51(-16)	0.	1.52(-11)
8	7.52(-16)	-5.01(-16)	2.78(-10)
9	5.01(-16)	1.25(-15)	-8.04(-12)
10	3.75(-16)	2.50(-16)	-1.37(-11)

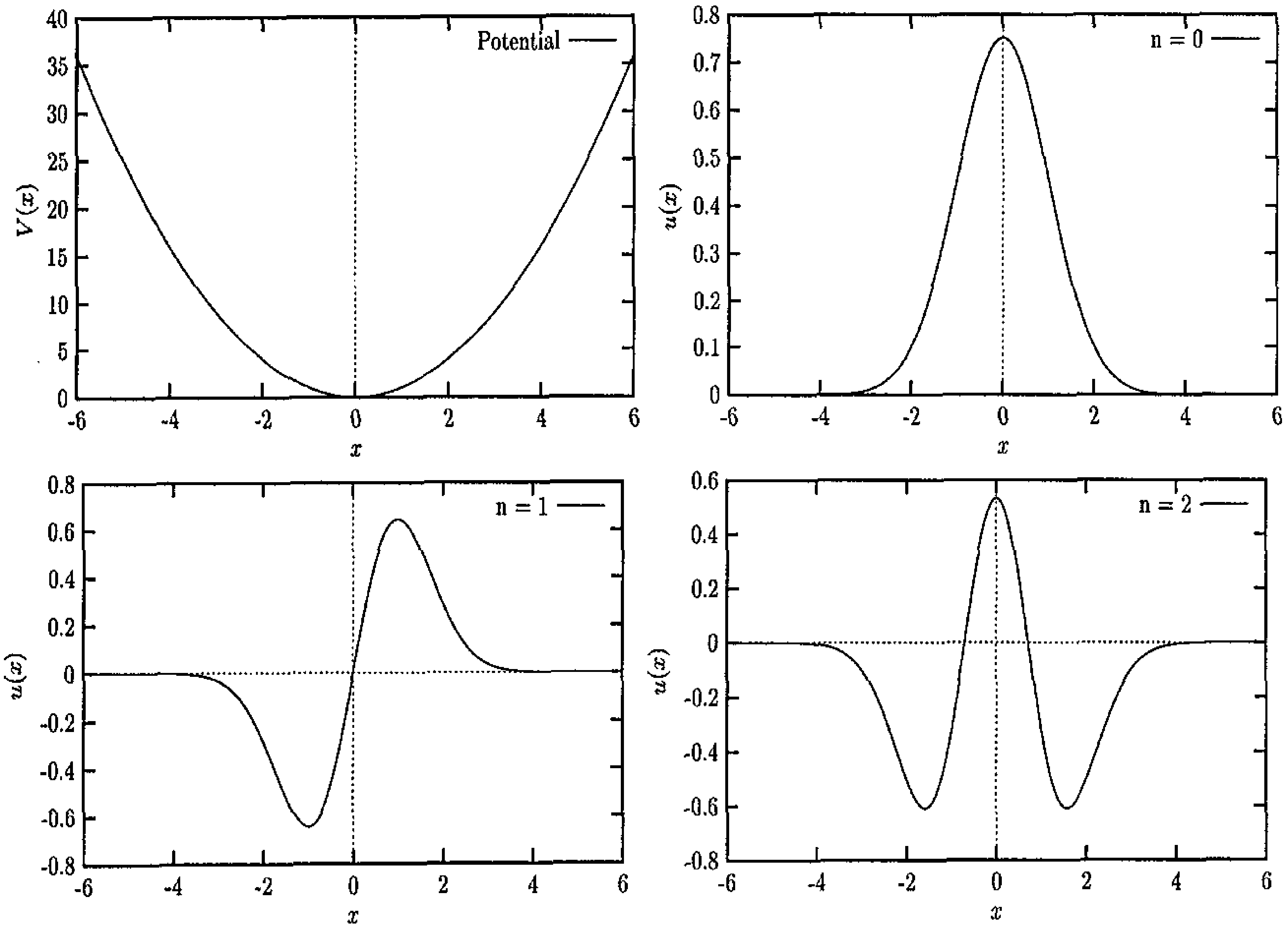

Figure 4.1: The simple harmonic oscillator potential and the numerical wave functions of its lowest three states for the *ansatz* (3.32).

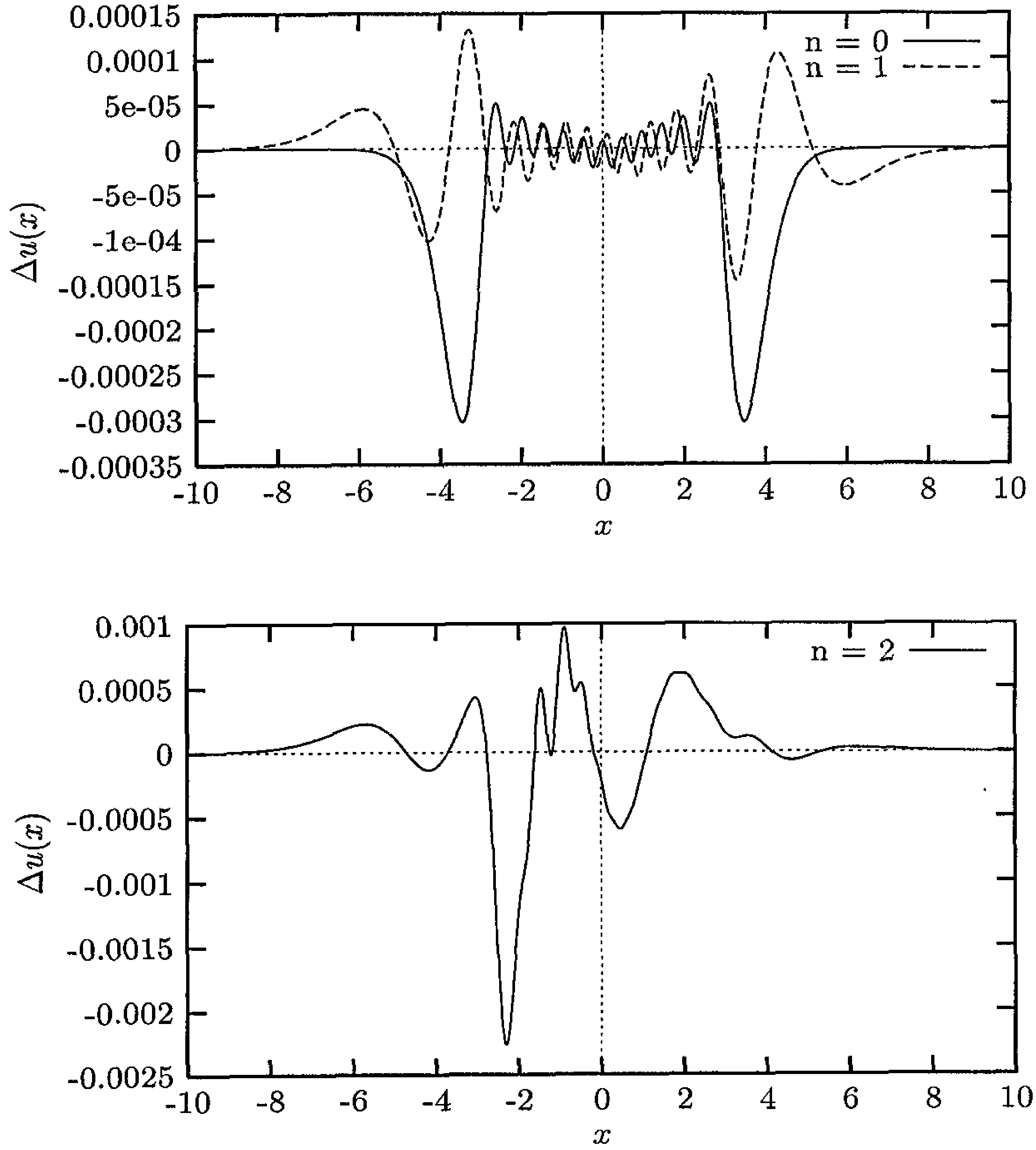

Figure 4.2: The deviations of the numerical wave functions, shown in Fig. 4.1, from the corresponding analytical ones.

4.1.2 The Optical Potential

This potential has the form

$$V(x) = \frac{u_o}{1+t}\left[1 - \frac{t}{a_0\,(1+t)}\right] \tag{4.4}$$

where $t = \exp[(x - x_o)/a_o]$, The values of the constants are given by [27]

$$u_o = -50\,\mathrm{fm}^{-2}, \quad a_o = 0.6\,\mathrm{fm}, \quad \text{and} \quad x_o = 7\,\mathrm{fm}\,.$$

The bound state energies, for the s-states, of this potential have been calculated in Ref. [27]. The lowest three states are relatively close to one another: E_1 = -49.457788728 fm^{-2}, E_2 = -48.148430420 fm^{-2}, and E_3 = -46.290753954 fm^{-2} [27]. Thus, this potential poses an interesting test for the present approach. We solved the problem in the interval [0,15] with two hundred integration points. We approximated the boundary behavior of the trial solution by the form

$$B(r,a) = r\,\mathrm{e}^{-a\,r}\,, \tag{4.5}$$

for the *ansatz* (3.31) and used the sigmoid function for the transfer function of the hidden neurons. We show the variation of the energies of the three lowest states as a function of N in tables 4.3 and 4.4 for the *ansatz* (3.32) and *ansatz* (3.31), respectively. We compare these energies with the corresponding ones quoted above. Fig. 4.3 shows the graphs of the numerical wave functions obtained with the *ansatz* (3.32) corresponding to the most accurate eigenvalue in table 4.3 .

The relative accuracy of the energies obtained with the *ansatz* (3.32) appear to converge more rapidly and smoothly to the values reported in Ref. [27] as N is increased. We achieved an accuracy of ten digits with N = 10 for the ground state. The accuracy in the energies of the excited states is, once again, lower than that of the ground state. Since the states are close to one another the range of the wave functions is the same. This can be clearly seen from Fig. 4.3

Table 4.3: The energies for the lowest three states for the optical potential, as functions of the number N of neurons in the hidden layer, obtained with the *ansatz* (3.32).

N	$E_1[\mathrm{fm}^{-2}]$	$E_2[\mathrm{fm}^{-2}]$	$E_3[\mathrm{fm}^{-2}]$
1	-49.273309476	-47.942383505	-45.953188341
2	-49.319483697	-48.094244993	-46.192208217
3	-49.443115265	-48.131857890	-46.260181905
4	-49.457702399	-48.127867361	-46.280937589
5	-49.457774686	-48.146200212	-46.290170033
6	-49.457787225	-48.147453435	-46.289555696
7	-49.457788181	-48.148382580	-46.290448035
8	-49.457788664	-48.148415021	-46.290094972
9	-49.457788622	-48.148137226	-46.290708008
10	-49.457788724	-48.148425068	-46.290620574
Ref. [27]	-49.457788728	-48.148430420	-46.290753954

Table 4.4: The energies for the lowest three states of the optical potential, as functions of the number N of neurons in the hidden layer, obtained with the *ansatz* (3.31).

N	$E_1[\mathrm{fm}^{-2}]$	$E_2[\mathrm{fm}^{-2}]$	$E_3[\mathrm{fm}^{-2}]$
1	-49.445441445	-48.033460663	-46.147913397
2	-49.454784635	-48.111376154	-46.238465097
3	-49.456828832	-48.146419706	-46.290484170
4	-49.457731332	-48.143231053	-46.289516582
5	-49.457733292	-48.148323225	-46.290647515
6	-49.457781998	-48.148299499	-46.290667077
7	-49.457784639	-48.148390012	-46.290661592
8	-49.457788692	-48.148395420	-46.290656837
9	-49.457787283	-48.148381978	-46.290655162
10	-49.457787569	-48.148410132	-46.290656265
Ref. [27]	-49.457788728	-48.148430420	-46.290753954

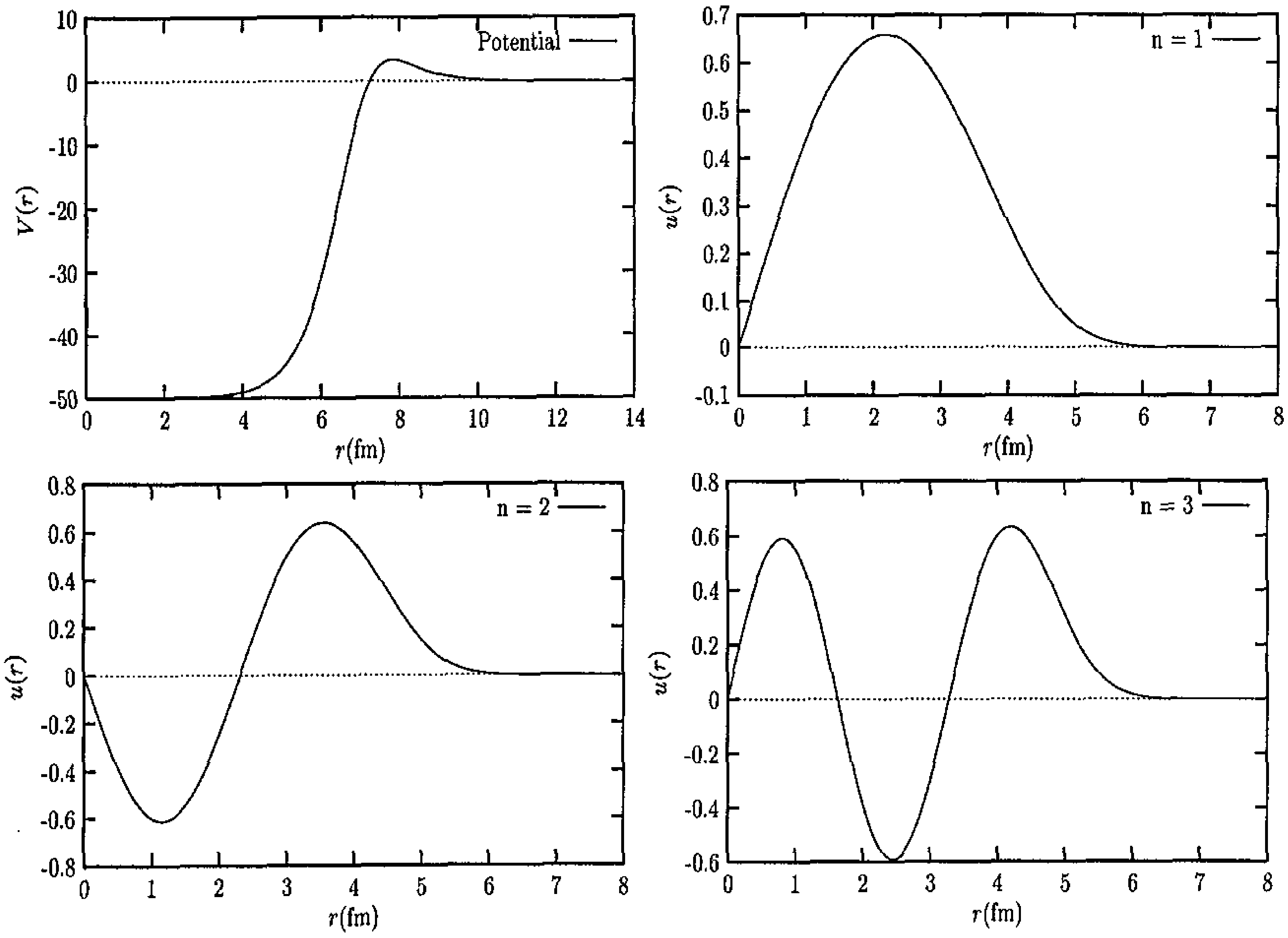

Figure 4.3: The optical potential and the numerical wave functions of its lowest three s-states for the *ansatz* (3.32).

4.1.3 Quarkonium

The interactions of the constituent 'particles' in a mesonic system, quark-antiquark bound state, can be estimated by potentials of the form

$$V(r) = -\frac{a}{r} + b\,r^c, \tag{4.6}$$

where a, b and c are constants. In this work we chose the combinations

$$\begin{aligned} V_1(r) &= -\frac{1}{r} + 2\,r && (4.7)\\ V_2(r) &= -\frac{1}{r} + 2\,r^2 && (4.8)\\ V_3(r) &= -\frac{1}{r} + 2\,(\,r + r^2\,)\,, && (4.9) \end{aligned}$$

where the unit of length is the *Bohr radius.* The ground state of the latter potential $V_3(r)$ corresponds to the exact eigenfunction [28]

$$u(r) = N_3\,r\exp(\,-r - r^2\,) \tag{4.10}$$

with eigenvalue 2.5. N_3 is the normalization coefficient. We treated $V_1(r)$ and $V_3(r)$ with fifty integration points in the interval [0,10], and $V_2(r)$ with forty integration points in the interval [0,4]. For these potentials we computed only the ground states. Also, we treated only $V_3(r)$ with both *ansatz* 3.31 and *ansatz* 3.32 and compared results obtained with the exact ones (see Fig. 4.4). We again approximated the boundary behavior of the trial solution by

$$B(r,a) = r\,\mathrm{e}^{-a\,r}\,, \tag{4.11}$$

for the *ansatz* (3.31) and used the sigmoid function for the transfer function of the hidden neurons. For $V_1(r)$ and $V_2(r)$ we only used the *ansatz* (3.32). The numerical energies as a function of N are given in table 4.5.

The energies of these potentials also showed convergence as N was increased. This convergence is illustrated in Fig. 4.4(a) and Fig. 4.5. Again the convergence is very fast.

The highest accuracy of the energy for $V_3(r)$ we could obtained with *ansatz* (3.32) is nine digits (N = 9). The *ansatz* (3.31) on the other hand produced the lowest deviation $\Delta E \sim 10^{-12}$ with the same number of neurons. This *ansatz*, (3.31), could fit the ground state energy to about nine digits with only three neurons in the hidden layer. Using the results of $V_3(r)$ and Fig. 4.5 we estimate the accuracy of the energies for $V_1(r)$ and $V_2(r)$ for N = 9 to at least eight digits. We compared the numerical wave functions obtained with the analytical one. The wave functions used were the ones for which the value of the corresponding numerical energies differed from the analytical one by $\sim 10^{-7}$. The deviations Δu are shown in Fig. 4.4(b) and (c). These figures show that the maximum Δu, in both cases, is approximated by $\sqrt{\Delta E}$. These deviations occur mainly in the interaction region. However, the *ansatz* (3.32) produced relatively higher deviations, Δu.

In the studies of the properties of few-body systems the quantities $\langle\, r\,\rangle$, $\langle\, r^2\,\rangle$, and $\psi(0)$ are usually required. One way of approximating the quantity $\psi(0)$ numerically is via the relation [28]

$$\psi^2(0) = 2\left\langle \frac{\partial V(r)}{\partial r} \right\rangle . \tag{4.12}$$

In table 4.6 we compare our numerical results of these quantities with results from other numerical methods in the literature. Except for the value of $\psi^2(0)$ the ANN method produces, comparatively, accurate results. The energy is reproduced to nine digits, and the corresponding wave function to four digits, of accuracy. The expectation values, $\langle\, r\,\rangle$ and $\langle\, r^2\,\rangle$, on the other hand, are reproduced to at least seven digits of accuracy. As can be observed from the results of Ref. [28] it is not unusual for a numerical wave function to produce accurate energy values but poor estimates of the quantity $\psi(0)$.

Table 4.5: Ground state energies, in a.u., for the quarkonium potentials as functions of the number N of neurons in the hidden layer.

N	$V_1(r)$	$V_2(r)$	$V_3(r)$	
			(3.32)	(3.31)
1	1.421317794	1.287659018	2.567574234	2.505377798
2	1.404523739	1.230396959	2.506794595	2.500006103
3	1.403314774	1.223708670	2.500423735	2.500000006
4	1.403161714	1.223705266	2.500027409	2.500000065
5	1.403130685	1.223705136	2.500000628	2.500000000
6	1.403130512	1.223705128	2.500000370	2.500000102
7	1.403130213	1.223705126	2.500000001	2.500000019
8	1.403130350	1.223705117	2.500000002	2.500000000
9	1.403130220	1.223705121	2.500000000	2.500000000
10	1.403130211	1.223705117	2.499999956	2.500000001
Exact			2.500000000	2.500000000

Table 4.6: Results for the potential $V_3(r) = -\frac{1}{r} + 2\,(\,r + r^2\,)$.

	E	$\langle\, r\,\rangle$	$\langle\, r^2\,\rangle$	$2\left\langle \frac{\partial V(r)}{\partial r} \right\rangle$
Ref. [28]	2.500 000 000	0.605 867	0.447 072 56	24.960 17
		0.605 863	0.447 069	25.693 80
This work	2.500 000 002	0.605 862 91	0.447 068 58	24.423 451
Exact	5/2	0.605 862 89	0.447 068 56	25.693 806

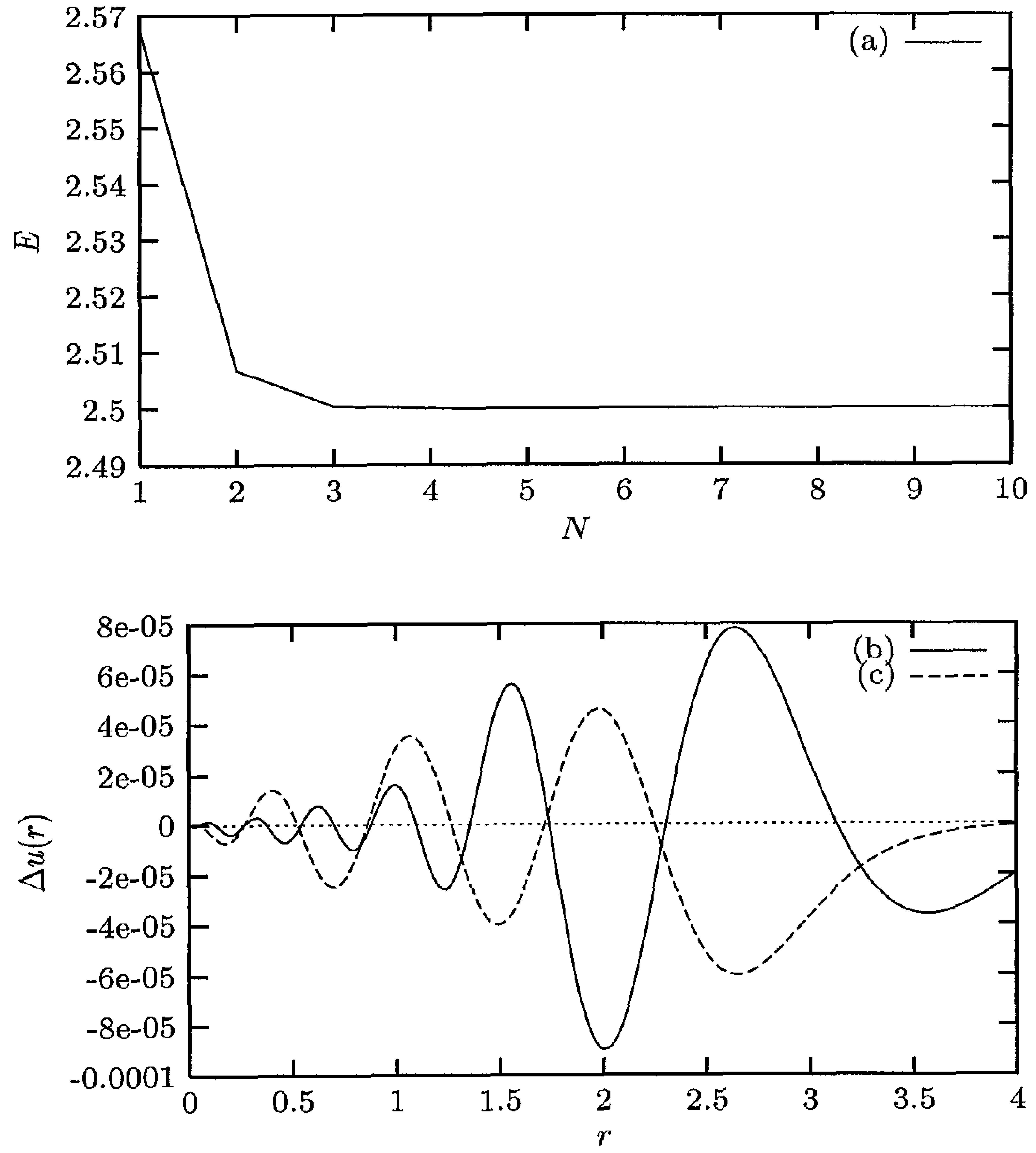

Figure 4.4: (a) The variation of the numerical ground state energy of $V_3(r)$ with the number of neurons in the hidden layer. The deviations of the neural ground state wave functions of $V_3(r)$ from the analytical one. (b) was computed with *ansatz* (3.32) (N = 6), while (c) was obtained with *ansatz* (3.31) (N = 3). $\Delta E \sim 10^{-7}$ in both cases. The unit of length is the *Bohr radius.*

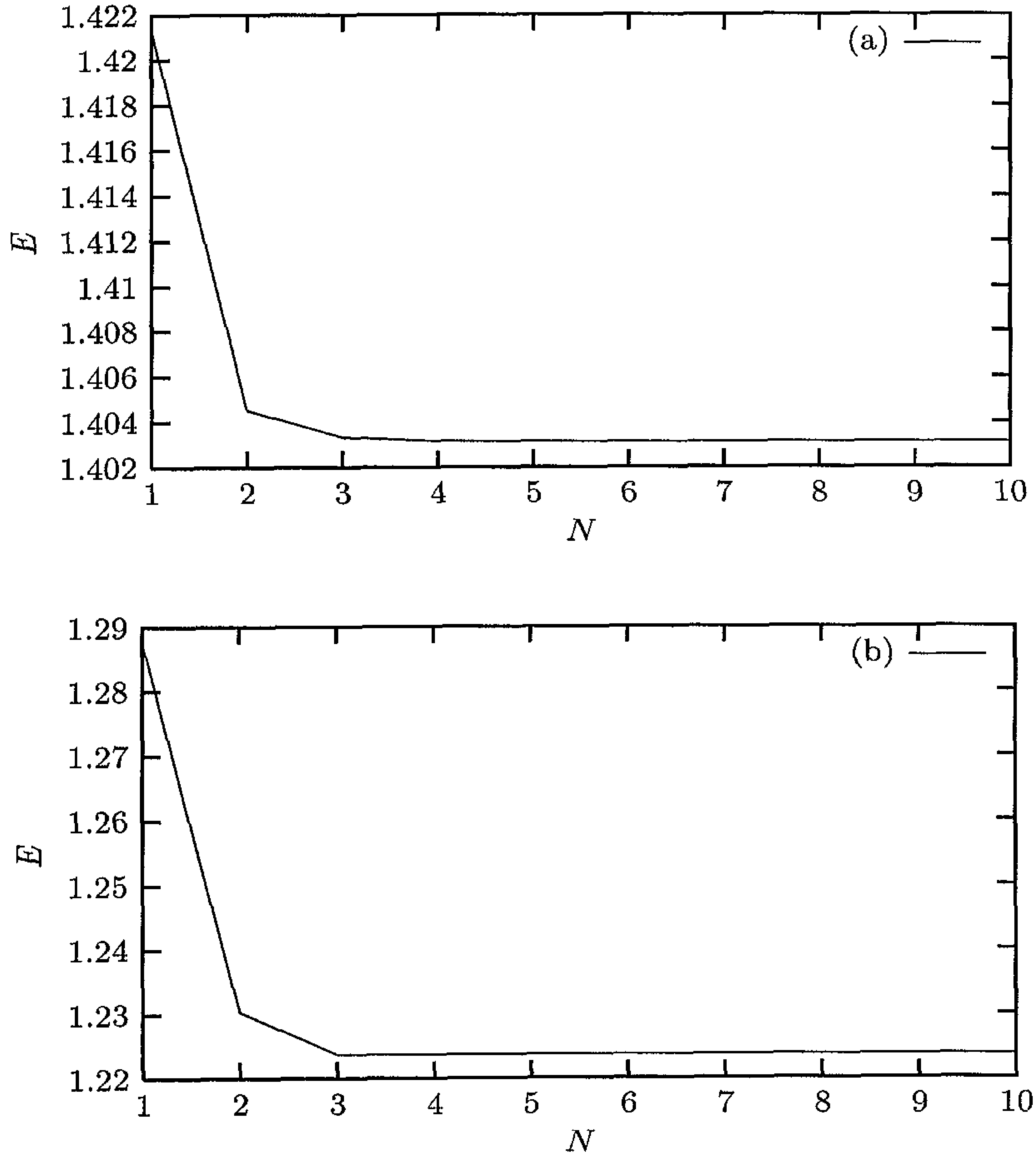

Figure 4.5: The variation of the numerical ground state energies of $V_1(r)$, (a), and $V_2(r)$, (b), with the number of neurons in the hidden layer.

4.1.4 The Coulomb Potential

The ground state of the potential

$$V_c(r) = -\frac{1}{r}, \tag{4.13}$$

in Hartree atomic units (a.u), admits the analytical solution

$$u(r) = 2\,r\,\mathrm{e}^{-r} \tag{4.14}$$

with the corresponding bound state energy being E_1 = -0.5 a.u. We solved this problem with twenty-one integration points in the interval [0,19]. For the *ansatz* (3.31) we approximated the boundary behavior of the trial solution by the form

$$B(r,a) = r\,\mathrm{e}^{-a\,r}, \tag{4.15}$$

and used the sigmoid function for the transfer function of the hidden neurons. In table 4.7 we give the deviations ΔE of the computed energies as a function of N.

As seen in table 4.7 the accuracy in the ground state energies of this system shows little dependence on N for *ansatz* 3.31 and *ansatz* 3.32. The accuracy of these energies seemed to depend mainly on the range of integration. The average deviation of the numerical energies from the analytical value is $\sim 10^{-13}$ a.u. But the lowest deviation, $\sim 10^{-14}$ a.u., was obtained with the *ansatz* (3.32) and N = 2. We compared the numerical ground state wave functions, obtained with N = 1 for both *ansatz* 3.31 and *ansatz* 3.32, with the analytical one. And we obtained the results shown in Fig. 4.6. As seen from this figure the maximum Δu are considerably less than $\sqrt{\Delta E}$. Also, the deviations in the case of the *ansatz* (3.31) are comparatively higher than those of the *ansatz* (3.32).

Table 4.7: The deviations, from the analytical value, of the ground state energies, in a.u., of the Coulombic potential as functions of the number N of neurons in the hidden layer. $a(-b) \Longrightarrow a \times 10^{-b}$.

N	(3.32)	(3.31)
1	-1.24(-13)	-1.27(-13)
2	1.60(-14)	-1.27(-13)
3	-8.93(-14)	-1.27(-13)
4	-1.20(-13)	-1.27(-13)
5	-1.10(-13)	-1.27(-13)
6	-1.27(-13)	-1.27(-13)
7	-1.27(-13)	-1.27(-13)
8	-1.27(-13)	-1.27(-13)
9	-1.27(-13)	-1.27(-13)
10	-1.27(-13)	-1.27(-13)

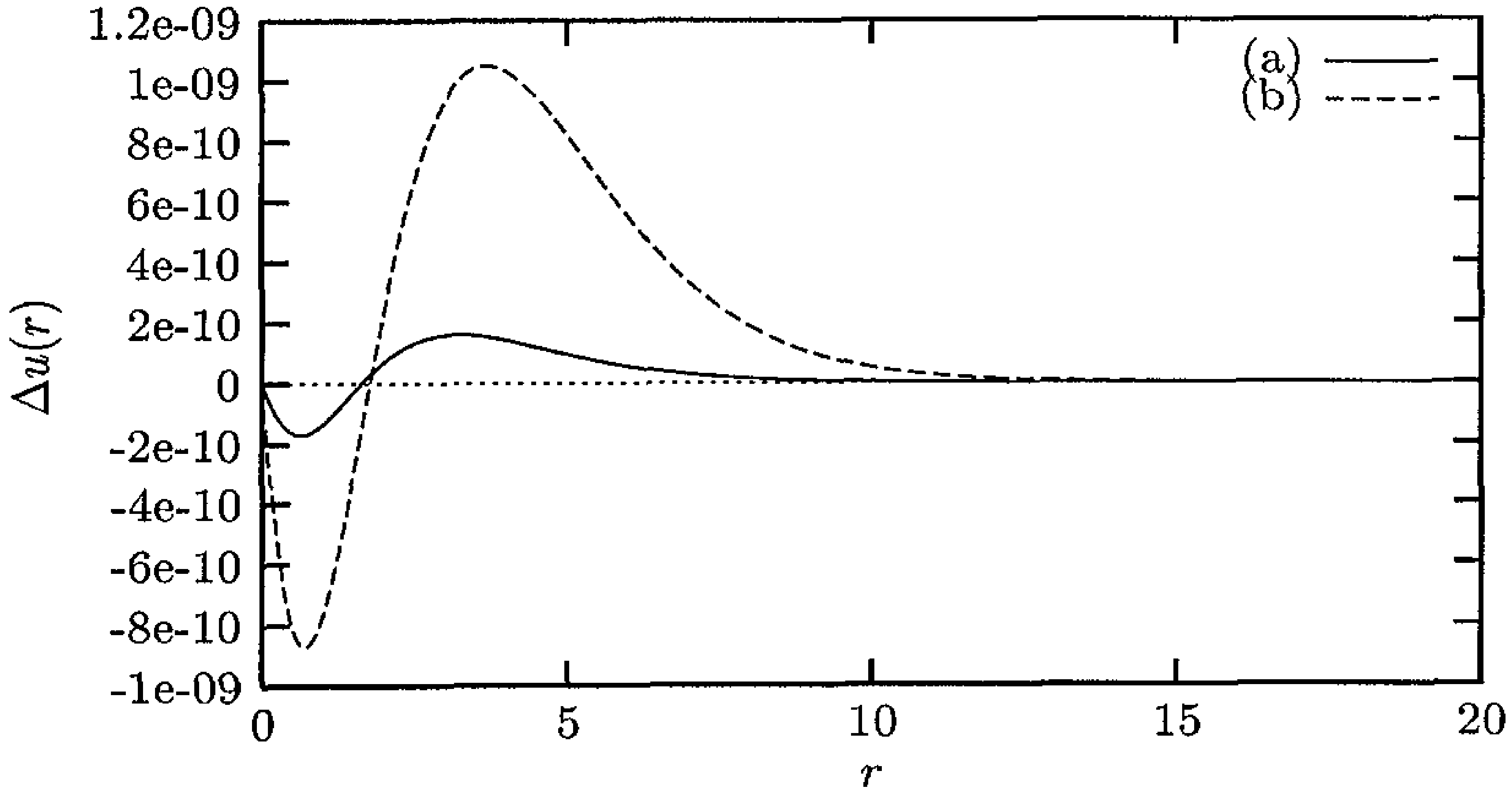

Figure 4.6: The deviations of the neural ground state wave functions of the Coulomb potential computed with (a) *ansatz* (3.32) and (b) *ansatz* (3.31) for N = 1. The unit of length is the *Bohr radius.*

4.1.5 Deuteron

We assume that the interactions between the nucleons in the deuteron can be represented by the exponential potential

$$V_e(r) = -A\,\mathrm{e}^{-\alpha r}\,, \tag{4.16}$$

or, alternatively, by the Yukawa potential

$$V_y(r) = -B\,\frac{\mathrm{e}^{-r/\beta}}{r/\beta} \tag{4.17}$$

where A, α, B, and β are constants. We computed the ground state energies of the deuteron ($\hbar^2/2\,\mu = 41.47$ MeV) for each of the two potentials. We considered the intervals [0,3] with fifty integration points in each case. Again, we represented the boundary behavior of the trial solutions by the form

$$B(r,a) = r\,\mathrm{e}^{-a r} \tag{4.18}$$

and used the sigmoid function for the transfer function of the hidden neurons, for the *ansatz* (3.31). The results are given in table 4.8. In Fig. 4.7 we show the wave functions corresponding to the singlet s-state, for $V_e(r)$, and the triplet s-state, for $V_y(r)$.

For the exponential potential, $V_e(r)$, we used the parameter set A = 104.2 MeV and α = 0.73 fm [29]. These parameters give rise to a virtual state with energy 0.039913 fm^{-1} [29] . In this work we obtained the value 0.039049 fm^{-1}. The convergence of the energy value, as N is increased, is relatively fast and smooth with the *ansatz* (3.32). For the Yukawa potential, $V_y(r)$, we used the parameter set suggested in [30], B = 14.3 MeV and β = 2.5 fm. These values of the parameters yield an energy value of 0.2354038389 fm^{-1}. This value is close the one obtained in Ref. [29], 0.23640 fm^{-1}, with the exponential potential using the parameter set A = 155.17 MeV and α = 0.76 fm. For this potential, the convergence of the energy value, as N is increased, is faster with the *ansatz* (3.31).

Table 4.8: The ground state energies of the deuteron, in fm^{-1}, for the exponential and Yukawa potential as functions of the number N of neurons in the hidden layer.

N	$V_e(r)$		$V_y(r)$	
	(3.32)	(3.31)	(3.32)	(3.31)
1	0.0612097533	0.0612097519	0.2357756418	0.2354104880
2	0.0391158653	0.0391114703	0.2354071620	0.2354038462
3	0.0390502394	0.0390493855	0.2354038491	0.2354038389
4	0.0390493825	0.0390499544	0.2354038389	0.2354038389
5	0.0390493730	0.0390493837	0.2354038389	0.2354038389
6	0.0390493729	0.0390493742	0.2354038389	0.2354038389
7	0.0390493729	0.0390493732	0.2354038389	0.2354038389
8	0.0390493729	0.0390493770	0.2354038389	0.2354038389
9	0.0390493729	0.0390493734	0.2354038389	0.2354038389
10	0.0390493729	0.0390493742	0.2354038389	0.2354038389
Ref. [29]	0.039913	0.039913	0.23640	0.23640

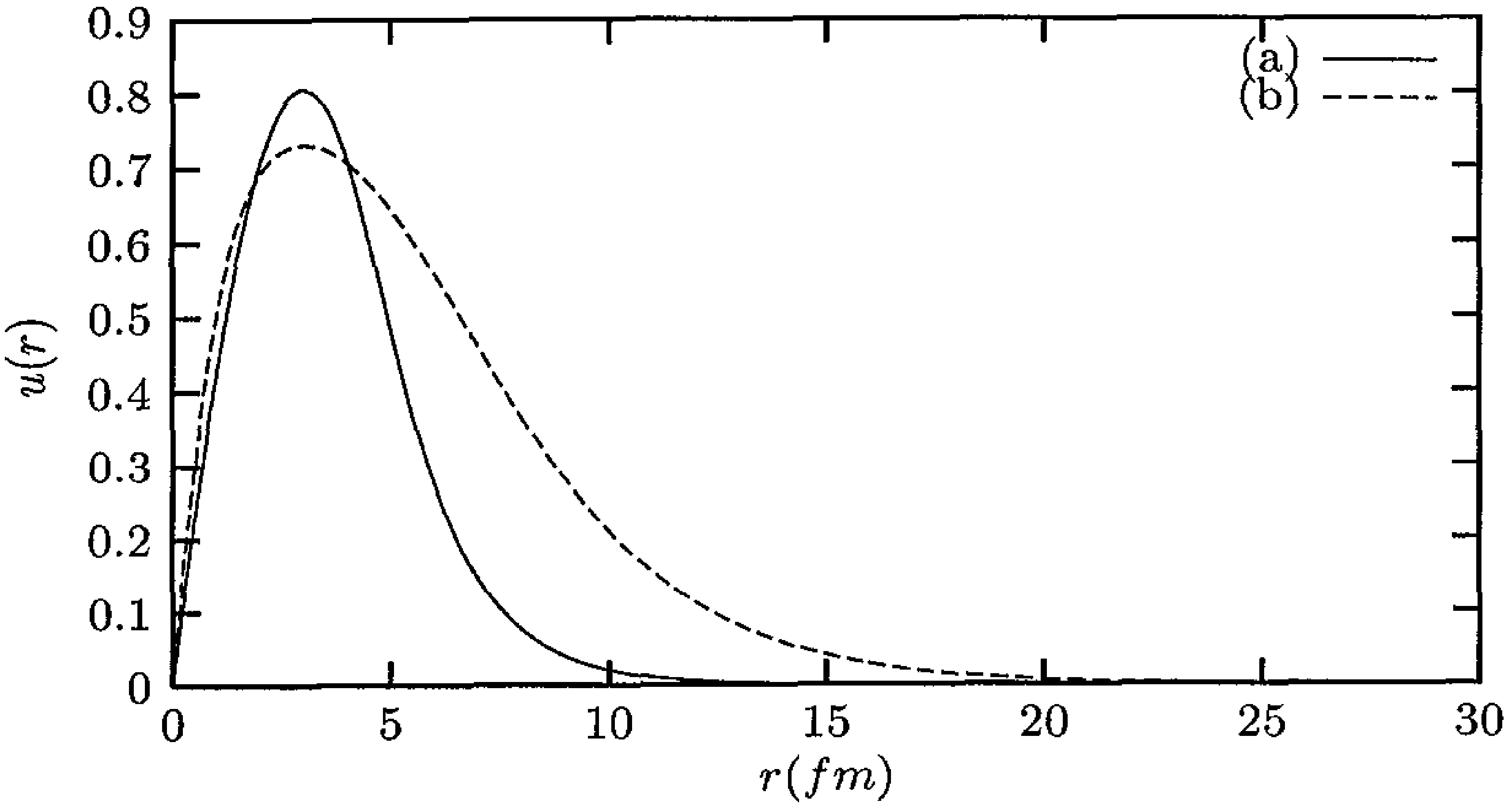

Figure 4.7: The singlet, (a), and triplet, (b), s-state wave functions of the deuteron computed with the *ansatz* (3.32), N = 5.

4.2 The Dirac Equation

There are very few potentials for which the Dirac eigenvalue equation can be solved analytically. The Coulomb potential is one such potential [31].

4.2.1 The Dirac-Coulomb Problem

The non-relativistic treatment of the Coulomb potential suggested that the accuracy of the results could be independent of the size of the hidden layer of the perceptron. We therefore considered the Dirac-Coulomb eigenvalue problem for further investigation. In this case we solved the problem with the same *ansatz* but for different transfer functions.

The Exact Solution

The Dirac eigenvalue problem for the Coulomb potential constitutes the Dirac-Coulomb problem [26]. The analytical ground state solution of this problem can be found in the literature [16],

$$f(x) = C\sqrt{1+\epsilon}\, x^{\gamma}\, \mathrm{e}^{-\alpha x} \qquad (4.19)$$

$$g(x) = \sqrt{\frac{1-\epsilon}{1+\epsilon}}\, f(x)\,, \qquad (4.20)$$

where C is a constant, and

$$x = \frac{r}{\hbar/\mu\, c^2} \quad \text{and} \quad \epsilon = \gamma = \sqrt{1-\alpha^2}\,. \qquad (4.21)$$

For the present work we chose the Coulomb potentials

$$V_{c1}(x) = -\frac{\alpha_1}{x}\,, \qquad (4.22)$$

$$V_{c2}(x) = -\frac{\alpha_2}{x} \qquad (4.23)$$

where $\alpha_1 = 1/137.0359895$ and $\alpha_2 = 0.49$. In the case of α_2 we used the mass 2.385 GeV, which is the reduced mass of the *bottonium.*

We treated the Coulomb potentials as Lorentz vectors. We began by computing the normalization coefficients parameterizing the trial solutions in the exact solution form,

$$f_t(x, \vec{p}_f) = p_1\, x^{p_2} \exp(-p_3\, x) \qquad (4.24)$$

$$g_t(x, \vec{p}_g) = p_4\, x^{p_5} \exp(-p_6\, x)\,, \qquad (4.25)$$

where the p_i were treated as minimization parameters. With this parameterization we determined the numerical values of the p_i that solved the relevant eigenvalue equation for the ground state, for each of the two potentials, and compared them with the analytical ones. The results are shown in table 4.9.

As can be seen in table 4.9 the reproduction of the values of all the parameters is excellent, especially for the $V_{c1}(x)$ case. With α_2 ($> \alpha_1$) the range of the Coulomb potential becomes shorter. The amplitude of the small part of the wave function for a short range potential is comparatively significant. As a result the probability of convergence of the variational energy to the ground state energy is small. That is one of the reasons for the lower accuracy attained in the case of α_2.

Table 4.9: Comparison of the numerical and analytical values of the parameters of the exact solutions to the ground state of the Dirac equation for the Coulombic potentials $V_{c1}(x)$ and $V_{c2}(x)$.

	$V_{c1}(x)$	$V_{c2}(x)$
p_1	1.0000061750086550	0.73940238035
p_4	0.0036487476462711	0.19356885132
p_4/p_1	0.0036487251153620	0.26179095215
γ	0.0036487251153640	0.26179095329
p_2	0.9999733739645460	0.87172243251
p_5	0.9999733739645862	0.87172243047
b	0.9999733739645427	0.87172243289
p_3	0.0072973530796413	1.16864996100
p_6	0.0072973530796413	1.16864996144
α	0.0072973530796448	1.16865000000
E	0.9999733739645437	2.07905793126
E_a	0.9999733739645427	2.07905800243

The Numerical Treatment

We then proceeded to solve the problems by parameterizing the trial solutions in terms of neural networks. Each radial function was represented by a separate network,

$$f_t(x,\vec{p}_f) \;=\; x\,N(x,\vec{p}_f) \qquad (4.26)$$

$$g_t(x,\vec{p}_g) \;=\; x\,N(x,\vec{p}_g)\,. \qquad (4.27)$$

We solved the Dirac-Coulomb problem for the coupling constant α_1 in the interval [0,2500] with two hundred integration points. The 'variations' of the energies with N for both transfer functions are displayed in table 4.10.

As seen in the table the energies appear to be almost independent of N for both transfer functions. However, the exponential transfer function produced higher accuracy than the derivative of the sigmoid. We managed to reproduce the ground state energy up to ten digits of accuracy with the exponential transfer function (N = 2). The reason for this high accuracy could be the relationship between the transfer function and the form of the exact solutions. In contrast, the best fit attained with the sigmoidal derivative was only six digits of accuracy (N = 4). We plotted the graphs of the large (small) component of the ground state wave function in Fig. 4.8(a) (Fig. 4.9(a)) for the exponential and Fig. 4.10(a) (Fig. 4.11(a)) for the sigmoidal derivative transfer functions. The average deviation from the analytical wave functions of the numerical large (small) component, illustrated in Fig. 4.8(b) (Fig. 4.9(b)) for the exponential and Fig. 4.10(c) (Fig. 4.11(c)) for the sigmoidal derivative, does not exceed the square root of the absolute error in the numerical energies in both cases. In general, the numerical wave functions tend to approach zero a little slower as $x \to 0$ with the sigmoidal derivative vanishing a little faster as $x \to \infty$ than the analytical ones.

In the case of the coupling constant α_2 we solved the problem in the domain [0,24] with three hundred integration points. As observed from table 4.11 and Fig. 4.12 to Fig. 4.15 the results have the same features as those obtained with α_1. However, the accuracy is unsatisfactory in this case. The average deviation of the numerical energies is $\sim$ 0.001 GeV while the maximum deviation of the numerical wave functions is $\sim$ 0.001 fm. The numerical energies do not indicate any pattern in the dependence on N, as was the case in the non-relativistic treatment of a similar potential.

Table 4.10: Ground state energies for the Coulomb potential with the coupling constant α_1 as functions of the number N of neurons in the hidden layer. These values were obtained with the sigmoidal and the exponential as transfer functions. The exact value is 0.9999733739645427.

N	*Sigmoid*	*Exponential*
1	0.99997855	0.9999733738579
2	0.99997647	0.9999733739217
3	0.99997424	0.9999733739215
4	0.99997380	0.9999733739214
5	0.99997484	0.9999733739212
6	0.99997466	0.9999733739212
7	0.99997461	0.9999733739213
8	0.99997452	0.9999733739213
9	0.99997448	0.9999733739211
10	0.99997439	0.9999733739211

Table 4.11: Ground state energies, in GeV, for the Coulomb potential with the coupling constant α_2 as functions of the number N of neurons in the hidden layer. These values were obtained with the sigmoidal and the exponential as transfer functions. The exact value is 2.07905800243 GeV.

N	*Sigmoid*	*Exponential*
1	2.08105	2.07888
2	2.08049	2.08496
3	2.08145	2.08571
4	2.08477	2.08676
5	2.08557	2.08620
6	2.08031	2.08610
7	2.07666	2.08483
8	2.07640	2.08560
9	2.08076	2.08227
10	2.09139	2.08399

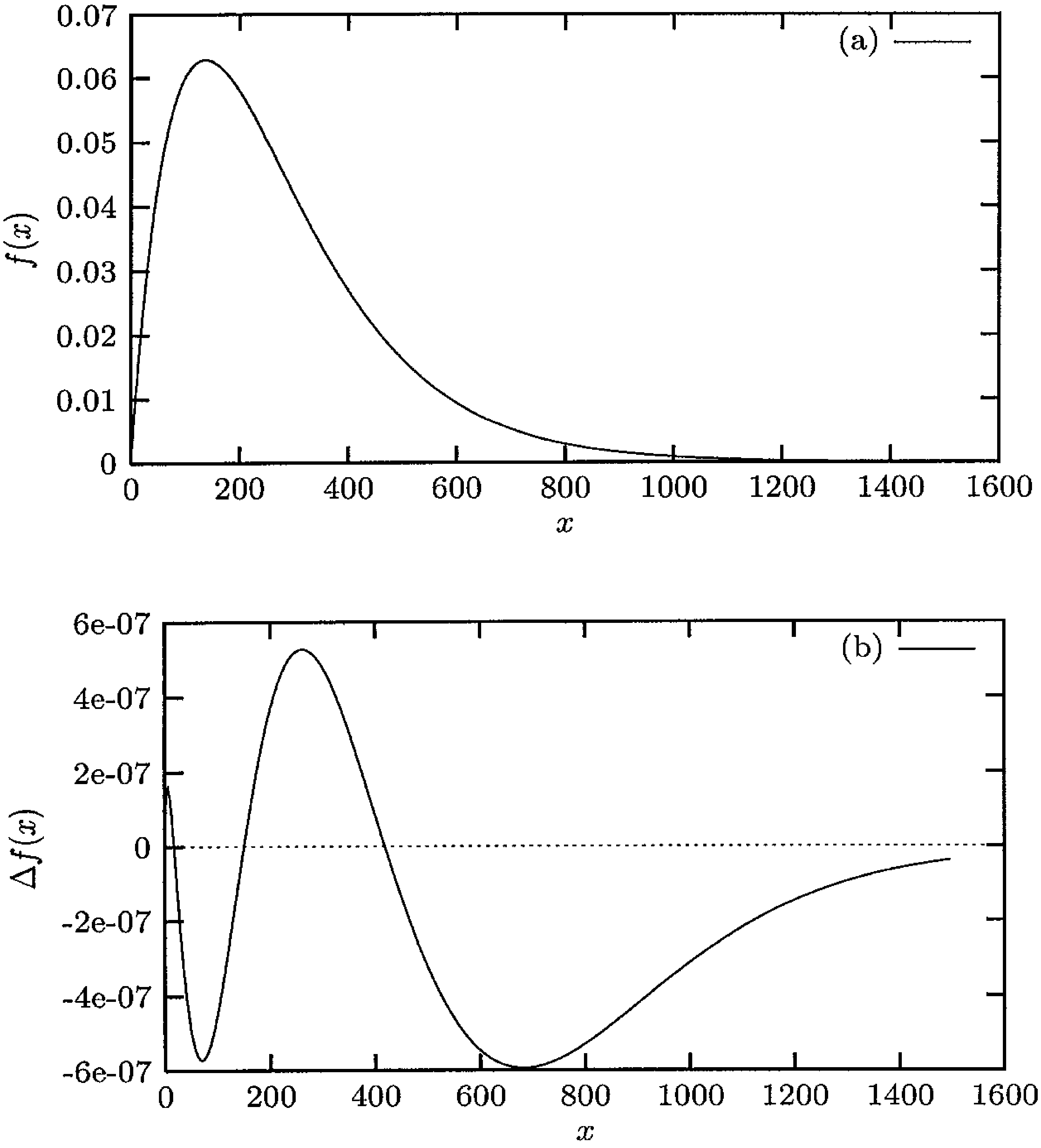

Figure 4.8: The normalized large part of the exponential neural ground state wave function, upper figure (a), of the Dirac-Coulomb problem (α_1), computed with N = 2 and its deviation, lower figure (b), from the analytical one.

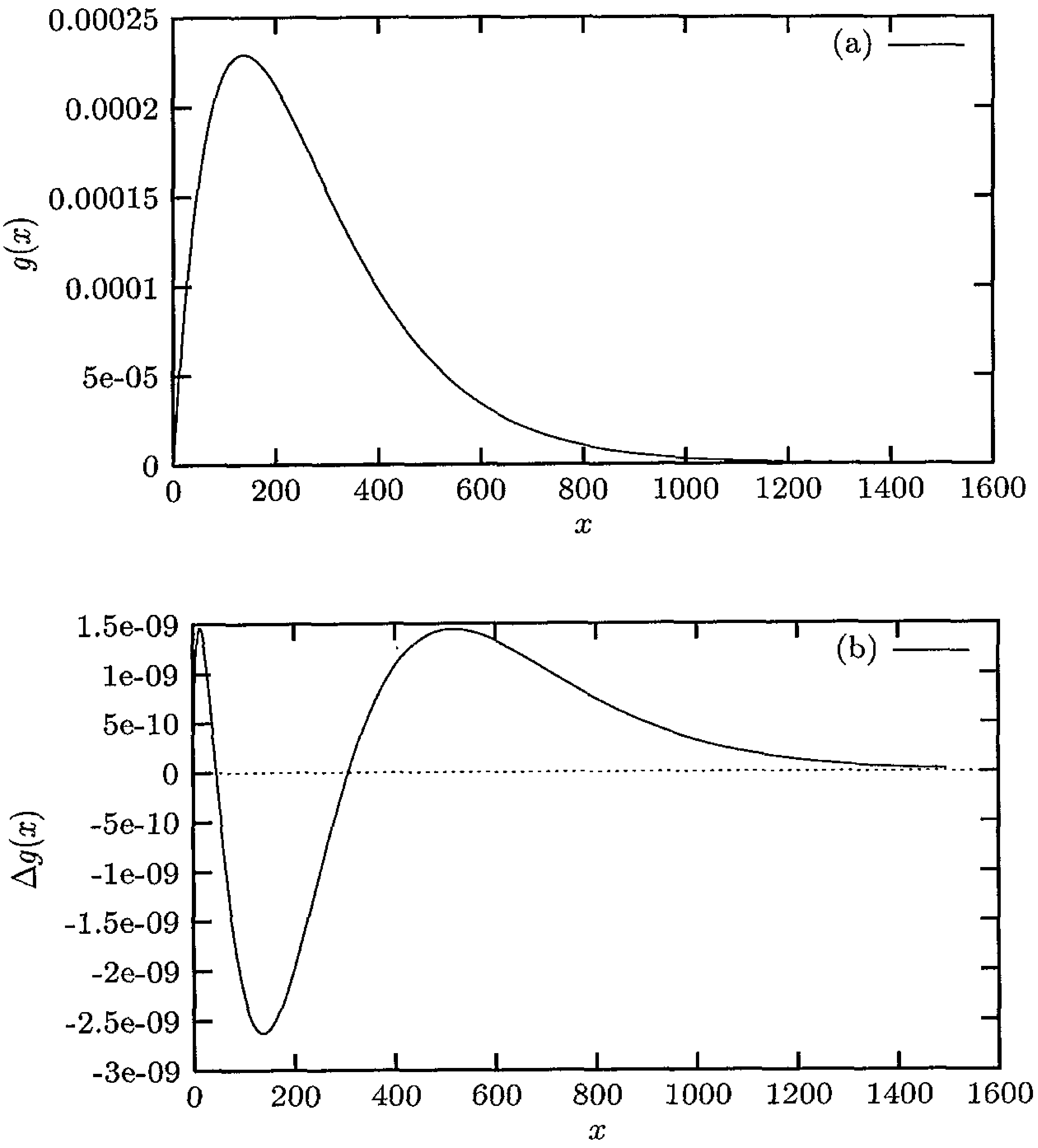

Figure 4.9: The normalized small part of the exponential neural ground state wave function, upper figure (a), of the Dirac-Coulomb problem (α_1), computed with N = 2 and its deviation, lower figure (b), from the analytical one.

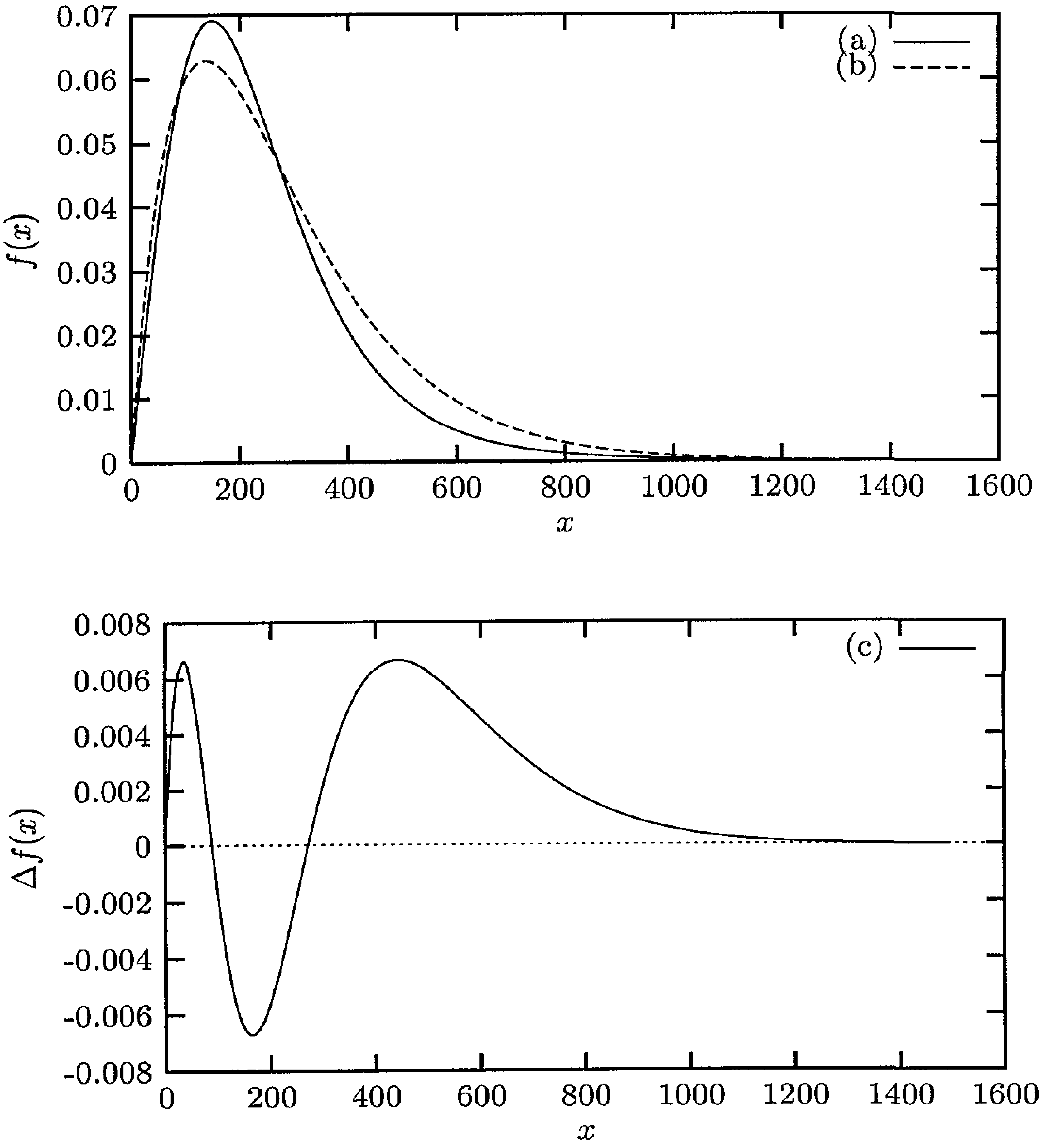

Figure 4.10: The normalized analytical (a), and the sigmoidal derivative neural large part of the ground state wave function (b), of the Dirac-Coulomb problem (α_1), computed with N = 4 and its deviation (c) from the analytical one.

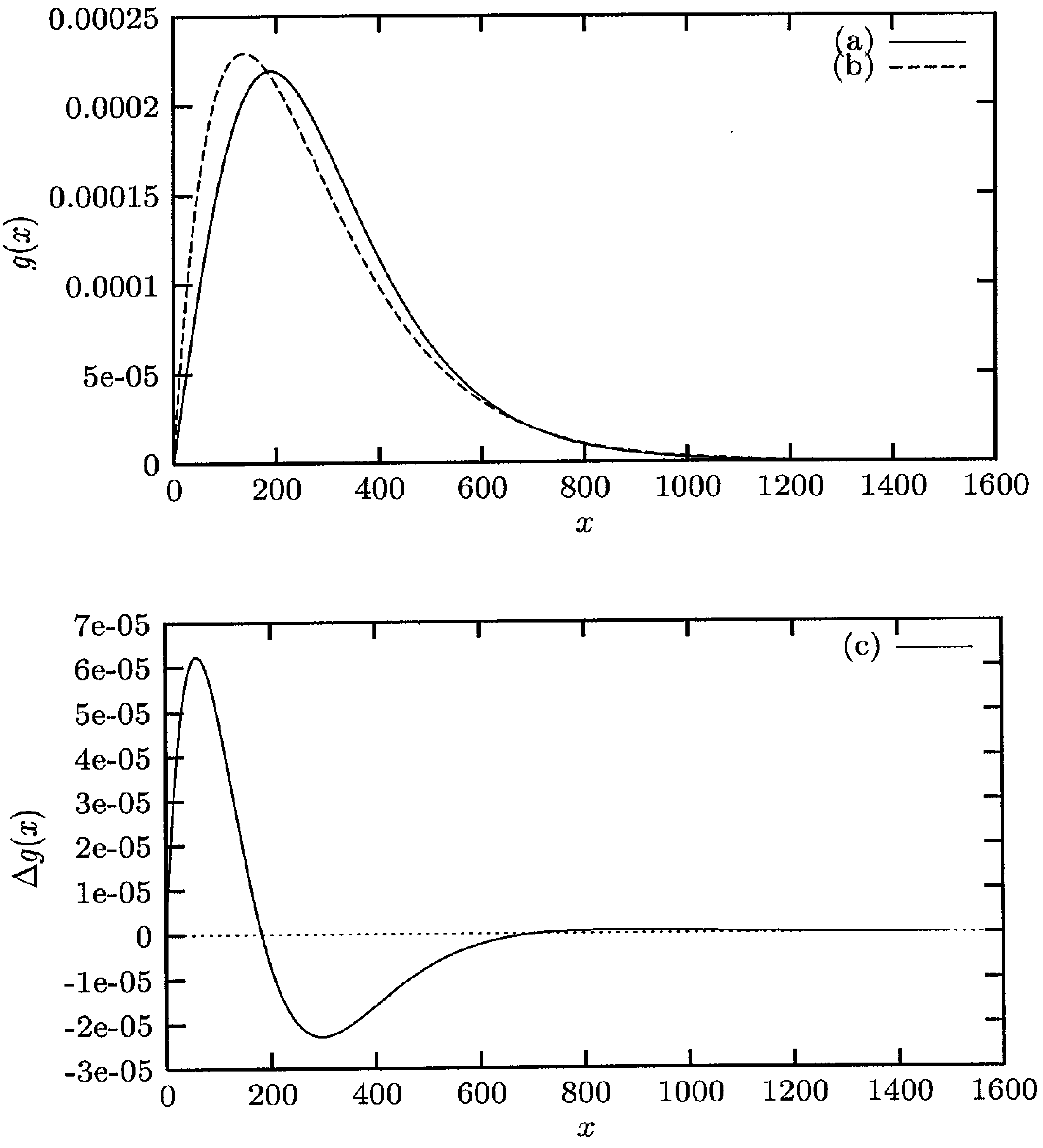

Figure 4.11: The normalized analytical (b), and the sigmoidal derivative neural small part of the ground state wave function (a), of the Dirac-Coulomb problem (α_1), computed with N = 4 and its deviation (c) from the analytical one.

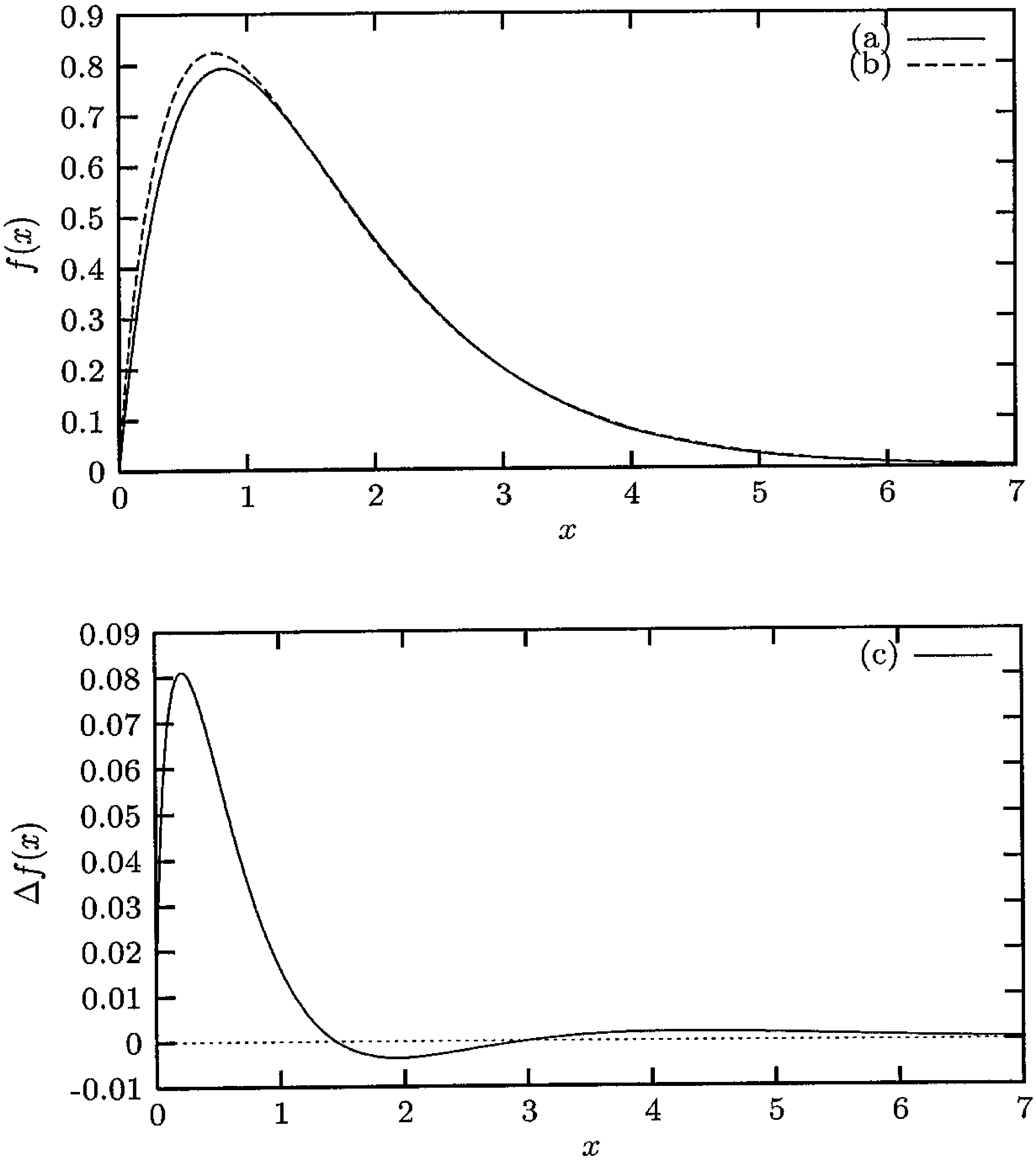

Figure 4.12: The normalized analytical (b), and the exponential neural large part of the ground state wave function (a), of the Dirac-Coulomb problem (α_2), computed with N = 1 and its deviation (c) from the analytical one. Distances were measured in fermi (fm).

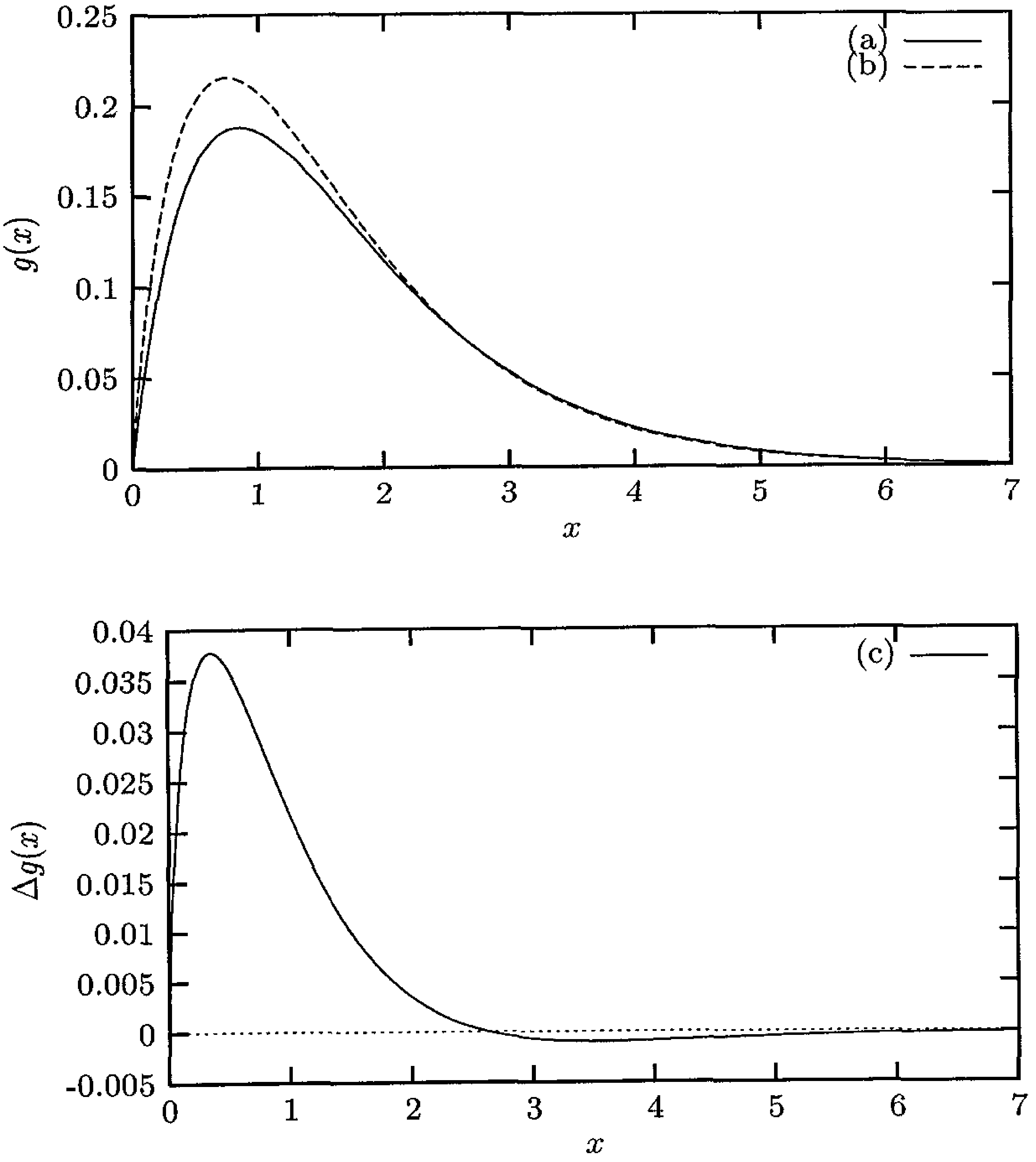

Figure 4.13: The normalized analytical (b), and the exponential neural small part of the ground state wave function (a), of the Dirac-Coulomb problem (α_2), computed with N = 1 and its deviation (c) from the analytical one. Distances were measured in fermi (fm).

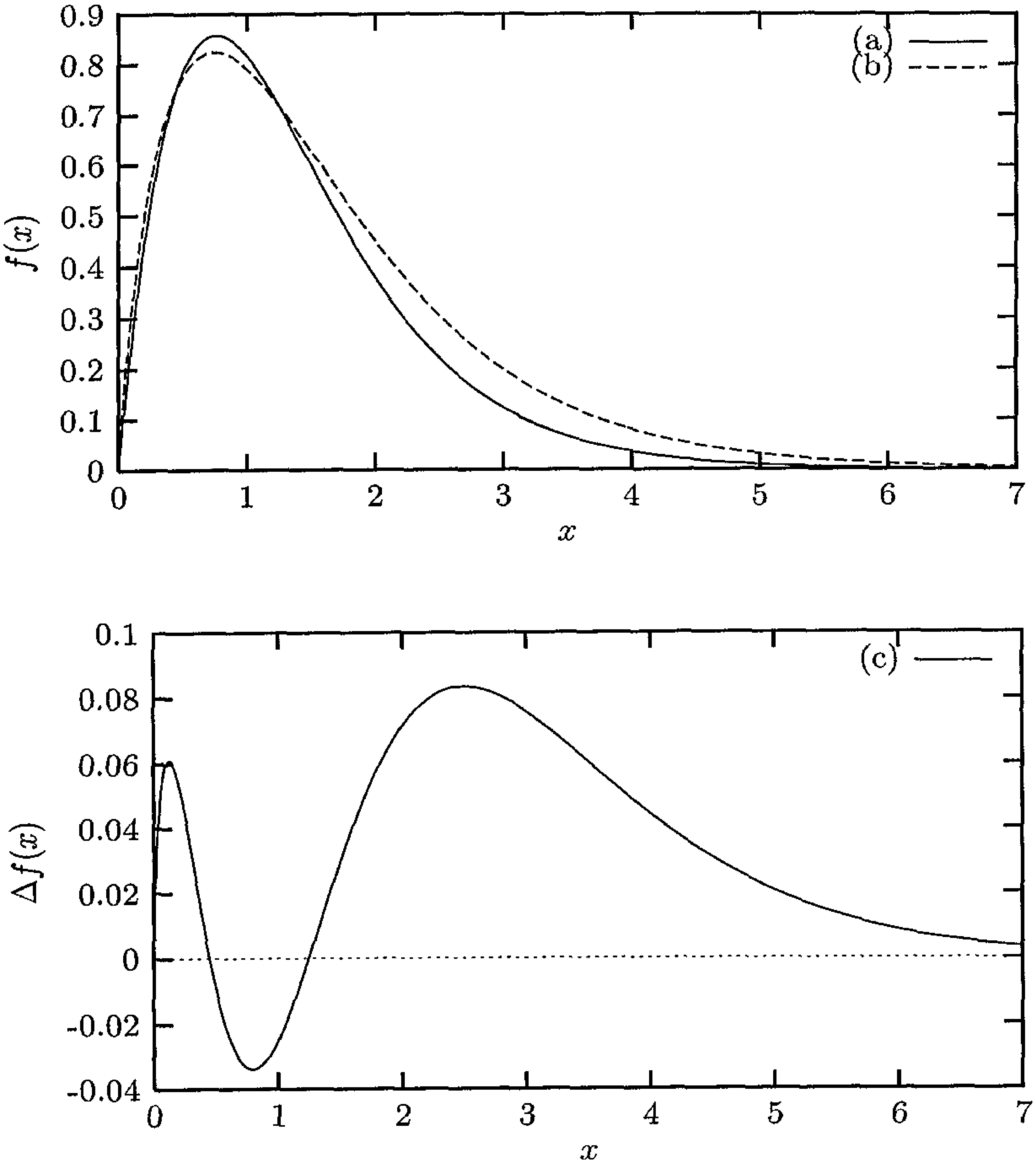

Figure 4.14: The normalized analytical (b), and the sigmoidal derivative neural large part of the ground state wave function (a), of the Dirac-Coulomb problem (α_2), computed with N = 1 and its deviation (c) from the analytical one. Distances were measured in fermi (fm).

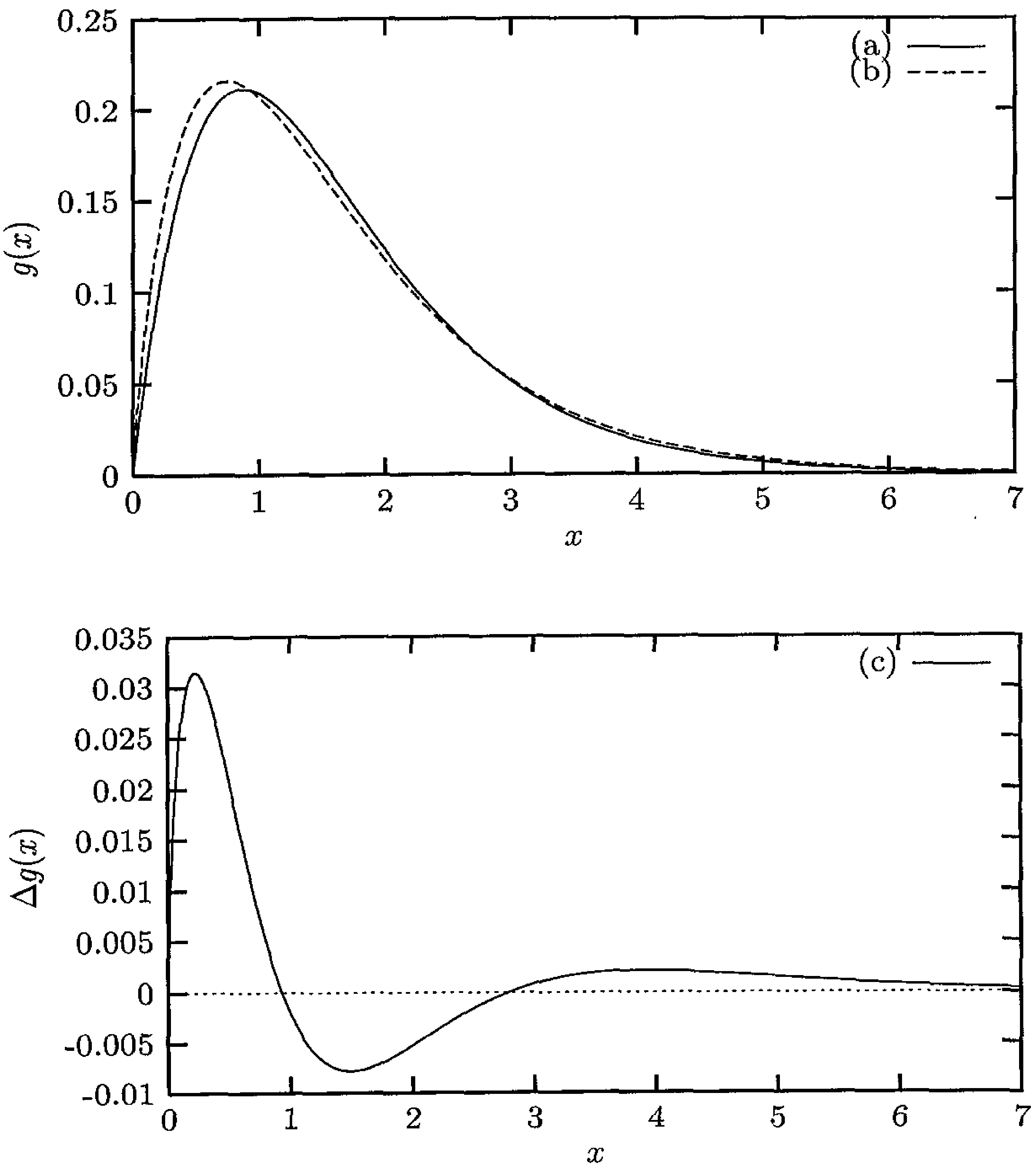

Figure 4.15: The normalized analytical (b), and the sigmoidal derivative neural small part of the ground state wave function (a), of the Dirac-Coulomb problem (α_2), computed with N = 1 and its deviation (c) from the analytical one. Distances were measured in fermi (fm).

Chapter 5

Conclusions

We have used the artificial neural network method of Ref. [1] to determine bound state solutions to selected two-body potentials. Because of the known approximation capabilities of feedforward multilayer perceptrons, we employed a three-layer perceptron in our calculations. We focused mainly on the effect of the size of the hidden layer on the accuracy of the results. The dependence of the accuracy on the form of the trial solution was also investigated. Our conclusions can be summarized as follows:

- In the case of potentials that satisfy the conditions

  $$\lim_{r\to 0} r^2\, V(r) = 0 \quad \text{and} \quad \lim_{r\to\infty} r\, V(r) = 0\,,$$

  a three-layer perceptron with a higher number of neurons in the hidden layer generally yields very accurate results. This was explicitly demonstrated by the results for the simple harmonic oscillator and Coulomb-plus-power law potentials. This inference is in line with the findings of Ref. [2].

- The accuracy achieved by the three-layer perceptron is greatly enhanced by explicitly incorporating characteristic features and properties of the expected solution when constructing a trial solution. This observation was demonstrated by the results obtained for the simple harmonic oscillator potential. For example, the first excited state could be constructed using the orthogonality condition. However, such a construction would have resulted in less accurate results as is clearly seen from the results of the second excited state.

- In the variational method, when the numerical energy is correct to second order then the corresponding wave function is generally correct to first order [28]. The results of the neural network method used in this work did not display a definite relationship between the deviations in the energies and the deviations in the corresponding wave functions. Although the accuracy in the numerical wave functions was lower than that of the corresponding energies, this accuracy was generally higher than the first order predictions of the variational method. The observed order of accuracy of the eigenvalues and the corresponding eigenfunctions emanates from the fact that, unlike the variational method which minimizes the functional $\langle H \rangle$, the ANN method minimizes the variance $\langle [H - \langle H \rangle]^2 \rangle$.

- The accuracy of the results obtained for the Coulomb, Yukawa, and exponential potentials showed little dependence on the size of the hidden layer. For a hidden layer with more than three neurons, the results referred to do not indicate any pattern that suggests that a more dense hidden layer will yield more accurate results. Instead, the accuracy of the results depended on other factors, for example, the range of integration. However, the relativistic treatment of the Coulomb potentials indicate that certain transfer functions will produce better results than others. Therefore, for these potentials, a network with three hidden neurons will yield very accurate results.

- The factoring-out of the asymptotic behavior of the bound state wave functions in the formulation of the trial solution has certain advantages. However, if the form of the chosen factor differs greatly from the exact one, then this technique will inhibit the process of obtaining better results. This observation is demonstrated by the results of the simple harmonic oscillator potential, where factorizing produced exceptionally accurate results, and the results of the Woods-Saxon potential, where factoring was not so effective. Incorporating the asymptotic behavior of the bound state wave functions in the formulation of the trial solution, by an appropriate choice of the transfer function, also has advantages. For instance, not only is the resulting trial solution simpler, but it also requires fewer minimization parameters than the corresponding factored form.

- The artificial neural network method can be readily generalized to higher dimensions. Therefore, the findings of this work are applicable also for systems consisting of more than two particles such as in the bound state of a three-particle system. Since the corresponding networks for higher dimensions are relatively large, they will require comparatively larger computer memory and longer computation times.

- The accuracy of the artificial neural network method also depends on a number of factors, some of which are not directly related to neural networks. For example, the accuracy of the estimations of integrals. Although the Gauss-Legendre integration technique is comparatively accurate, integration methods based on adaptive mesh size may prove to be better choices. The effect of the number of hidden layers on the accuracy of the solution approximation for multi-layer perceptrons and other network architectures need to be investigated also.

Bibliography

[1] I. E. Lagaris, A. Likas, and D. I. Fotiadis, *IEEE Trans. on Neural Networks*, **9**, 987 (1998).

[2] K. Hornik, M. Stinchcombe, and H. White, *Neural Networks*, **2**, 359 (1989).

[3] P. Arena, L. Fortuna, G. Muscato, and M. G. Xibilia, *Lecture Notes in Control and Information Sciences* **234**, (Springer, London, 1998).

[4] K. I. Funahashi, *Neural Networks*, **2**, 183 (1989).

[5] R. C. Williamson and U. Helmke, *IEEE Trans. on Neural Networks*, **6**, 1 (1995).

[6] R. Yentis, Jr. and M. E. Zaghoul, *IEEE Trans. on Circuits and Systems-I*, **43** 8, 687 (1996).

[7] P. Bhat, AIP Conference Proceedings: *Advanced Computing and Analysis Techniques in Physics Research*, (Edited by P. C. Bhat and M. Kasemann), 583 (2000). p22-p30.

[8] L. Dudko, AIP Conference Proceedings: *Advanced Computing and Analysis Techniques in Physics Research*, (Edited by Bhat P. C. and Kasemann M.), 583 (2000), p83-p85.

[9] M. Takeda and J. Goodman, *Appl. Opt.*, **25** 18, 3033 (1996).

[10] E. Barnard and D. Casasent, *Opt. Comput.*, **963** 88, 537 (1988).

[11] H. Lee and I. Kang, *J. Comput. Phys.*, **91**, 110 (1990).

[12] L. Wang and J. M. Mendel, *IEEE Int. Joint Conference on Neural Networks*, **2**, 125 (1990).

[13] A. J. Meade Jr and A. A. Fernadez, *Math. Comput. Mod.*, **19** 12, 1 (1994), *Math. Comput. Mod.*, **20** 9, 19 (1994).

[14] I. E. Lagaris, A. Likas, and D. I. Fortiadis, *Comput. Phys. Commun.*, **104**, 1 (1997).

[15] I. E. Lagaris, A. Likas, and D. G. Papaggeorgiou, *IEEE Trans. on Neural Networks.* **11**, 1041 (2000).

[16] E. Merzbacher, *Quantum Mechanics* 2nd. Ed. (Wiley, New York, 1970).

[17] W. Greiner, *Relativistic Quantum Mechanics* : Wave Equations, (Springer, Berlin, 2000).

[18] Y. Suzuki and K. Varga, *Stochastic Variational Approach to Quantum-Mechanical Few-Body Problems* **m 54**, (Springer, Berlin, 1998).

[19] V. I. Korobov, *Phys. Rev.* A **61**, 064503 (2000).

[20] J. Giammarco and J. Franklin, *Phys. Rev.* A **36** 12, 5839 (1987).

[21] D. G. Papageorgiou, I. N. Demetropoulos, and I. E. Lagaris, *Comput. Phys. Commun.*, **109**, 227 (1998).

[22] C. M. Bishop, *Neural Networks for Pattern Recognition*, (Clarendon, Oxford, 1998).

[23] S. Haykin, *Neural Networks* : A Comprehensive Foundation, (Macmillan, New York, 1994).

[24] R. Williams and D. Zipser, Neural Computation **1**, 270 (1989).

[25] G. L. Payne, *Lecture Notes in Physics.* **273**, 64 (1987).

[26] L. I. Schiff *Quantum Mechanics*, 2nd Edition, (McGraw Hill, New York, 1955).

[27] Gh. Adam, L. Gr. Ixaru, and A. Corciovei, *J. Comput. Phys.*, **22**, 1 (1976).

[28] J. Killingbeck and S. Galicia, *J. Phys.* A: Math. Gen. **13**, 3419 (1980).

[29] S. A. Rakityansky, S. A. Sofianos, and K. Amos, *Nuovo Cimento*, **111** B 3, 363 (1996).

[30] W. E. Burchman *Elements of Nuclear Physics*, (Longman, New York, 1979).

[31] H. A. Bethe and E. E. Salpeter, *Quantum Mechanics of One- and Two-Electron Systems*, (Springer-Verlag, Berlin, 1957).