

Multilayer perceptrons and radial basis function neural network methods for the solution of differential equations: A survey

Manoj Kumar, Neha Yadav*

Department of Mathematics, Motilal Nehru National Institute of Technology, Allahabad-211004 (U.P.), India

ARTICLE INFO

Article history:

Received 29 April 2011

Received in revised form 8 August 2011

Accepted 14 September 2011

Keywords:

Differential equations

Neural network

Multilayer perceptron

Radial basis functions

Backpropagation algorithm

ABSTRACT

Since neural networks have universal approximation capabilities, therefore it is possible to postulate them as solutions for given differential equations that define unsupervised errors. In this paper, we present a wide survey and classification of different Multilayer Perceptron (MLP) and Radial Basis Function (RBF) neural network techniques, which are used for solving differential equations of various kinds. Our main purpose is to provide a synthesis of the published research works in this area and stimulate further research interest and effort in the identified topics. Here, we describe the crux of various research articles published by numerous researchers, mostly within the last 10 years to get a better knowledge about the present scenario.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

A series of problems in many scientific fields can be modeled with the use of differential equations such as problems in physics [1–3], chemistry [4–6], biology [7,8], economics [9], etc. Due to the importance of differential equations many methods have been proposed in the relevant literature for their solution such as Runge–Kutta methods [10,11], Predictor–Corrector methods [12,13], Finite Difference methods [14–16], Finite Element methods [17–19], Splines [20–23] and other methods [24–41] etc. These methods require the discretization of domain into the number of finite elements where the functions are approximated locally. Although these methods provide good approximations to the solution, they require a discretization of domain via meshing, which may be challenging in two or higher dimension problems. Also, the approximate solution derivatives are discontinuous and can seriously impact on the stability of the solution. Furthermore, in order to obtain satisfactory solution accuracy, it may be necessary to deal with finite meshes that significantly increase the computational cost.

In spite of the above methods, approximate particular solutions can also be achieved by using multilayer perceptrons, radial basis functions, models based on genetic programming, hybrid approaches based on neural networks, etc. Advantages of these methods are that they involve a single independent variable regardless of the dimension of the problem, and the solution obtained from the neural network methods are differentiable and in closed analytic form. During the last few years much progress has been done in this area. In this article, a wide survey of solution of differential equations using multilayer perceptrons (MLP) and radial basis functions (RBF) along with our own comparative remarks have been presented. We will omit the details of techniques used to solve these models due to length constraints. Interested readers can see the references cited (along with these methods) for more details.

The rest of this article is organized as follows: in Section 2, we present a summary of Multilayer perceptron (MLP) neural network techniques for solving differential equations from various research articles. A brief description of articles published on the Radial Basis function neural network (RBFN) technique for solving differential equations is presented in Section 3. In

* Corresponding author.

E-mail addresses: manoj@mnnit.ac.in (M. Kumar), nehayad441@yahoo.co.in (N. Yadav).

Section 4, a comparison of MLP and RBF techniques for solving differential equations are given. Finally, Section 5 presents major conclusions and further developments.

2. Multilayer perceptron (MLP) neural network techniques to solve differential equations

In this section, we will give a brief description of multilayer perceptron (MLP) neural network techniques for solving differential equations of various kinds. Various research articles considered here are in the ascending order of their years of publication.

In [42], Meadre and Fernandez concentrated on developing a general, numerically efficient, non-iterative method in which the feed forward artificial neural network architecture can be used to model the solution of algebraic and differential equations accurately, using only the equation of interest and the boundary or initial conditions. A feed forward artificial neural network (FFANN) constructed by this non-iterative method is indistinguishable from those methods which are trained using conventional techniques. They borrowed a technique from applied mathematics, known as the method of weighted residuals (MWR), and showed how it can be made to operate directly on the network architecture. The method of weighted residuals is a generalized method for approximating functions, usually from given differential equations. A simple feed forward network using a single input and output neuron with a single hidden layer of processing elements utilizing the hard limit transfer function is constructed to approximate accurately the solution of a first and second order linear ordinary differential equation. They also predicted and controlled the error constructed by feed forward artificial neural networks. The non-iterative approach outlined in the article has allowed suitable algorithms to be devised for the synthesis of FFANNs that approximate non-linear ordinary differential equations. Also the non-iterative algorithm has allowed work to progress in the construction of FFANNs, sigma-Pi, and recurrent networks, to approximate linear and non-linear partial differential equations.

Remark 1. It can be concluded from the above article that it is possible to construct directly and non-iteratively a feed forward neural network to approximate arbitrary linear ordinary differential equations. The methods used are all linear in storage and processing time. The L_2 -norm of the network approximation error decreases quadratically with the increasing number of hidden neurons. The result obtained through the output of network demonstrates the accuracy of approximation.

In the research article [43], Issac Ellias Lagaris and Aristidis Likas presented a new method for solving initial and boundary value problems using artificial neural network. To illustrate their method they took the following general differential equation:

$$G(\vec{x}, \psi(\vec{x}), \nabla \psi(\vec{x}), \nabla^2 \psi(\vec{x})) = 0, \quad \vec{x} \in D \quad (1)$$

subject to the certain boundary conditions which are either Dirichlet or Neumann boundary conditions. Also $\vec{x} = (x_1, x_2, \dots, x_n) \in R^n$ denotes the definition domain and $\psi(\vec{x})$ is the solution to be computed. Collocation method is adopted, to obtain a solution of the above differential equation (1), which assumes the discretization of the domain D and S into \hat{D} and \hat{S} . Solve the above differential equation. They choose the form of trial function $\psi_t(\vec{x})$ such that by construction it satisfies the given boundary conditions. This is obtained by writing the trial function as a sum of two parts

$$\psi_t(\vec{x}) = A(\vec{x}) + F(\vec{x}, N(\vec{x}, \vec{p})) \quad (2)$$

where $N(\vec{x}, \vec{p})$ is a single output feed forward neural network with parameters \vec{p} and n input units fed with the input vector \vec{x} . Term $A(\vec{x})$ contains no adjustable parameters and satisfies the boundary conditions. The second term F employs a neural network whose weights are adjusted to deal with the minimization problem and it is constructed so as not to contribute with boundary conditions. Then the network is trained to satisfy the differential equation. The authors illustrated present method by solving a variety of model problems. The solution such obtained is compared with the solution obtained using the Galerkin finite element method for several cases of partial differential equations and it is found that the method exhibits excellent generalization performance since the deviation at the test points was 'in no case' greater than the maximum deviation at the training points.

Remark 2. Method presented by the authors in [43] is general and can be applied to ordinary differential equations, the system of coupled ordinary differential equations and also to partial differential equations. It is also stressed out that the present method described in [43] can easily be used for dealing with the domains of higher dimensions. The method becomes particularly interesting with the help of neuroprocessor due to expected essential gains in the execution speed.

In article [44], the authors presented a method to solve a class of first order partial differential equation as input to state linearizable or approximate linearizable system. They proposed an extended backpropagation algorithm for training the derivative of a feed forward neural network and further, an approximate solution for a class of partial differential equation is obtained. A feedback control law has been designed to control a class of non-linear systems, based on the approximate solution. Simulation technique is used to demonstrate the effectiveness of the proposed algorithm and it is shown that the method can be very useful for practical applications in the following cases:

I. When a non-linear system satisfies the conditions for input-to-state linearization but the non-trivial solution of the given equation

$$\frac{\partial \lambda(x)}{\partial x} [g(x) \operatorname{ad}_f^1 g(x) \dots \operatorname{ad}_f^{n-2} g(x)] = 0 \quad (3)$$

is hard to find by training the derivative of a neural network. We can seek the approximate solution by the method given in [44]. Since there is no restrictive condition for choosing the basis vector to train the neural network, therefore a simple transformation to construct the basis is recommended.

II. When a non-trivial solution does not exist for the above given equation, we still obtain an approximate solution. If the approximate solution is considered an exact solution for a linearizable feedback system, then the system should approximate the given non-linear system as closely as possible.

Remark 3. It is pointed out that the method presented in [44] can benefit the design of the class of non-linear control systems, when the non-trivial solution of partial differential equations is difficult to find. Simulation examples presented in the article exhibit the advantages and effectiveness of the method. For applications where a large region of operation is required, the control design based on this method cannot result in a satisfactory performance.

Lagaris et al. in [45] studied partial differential equation with the case of complex boundary geometry. They examined the partial differential equations of the form

$$L\Psi = f \quad (4)$$

where L is a differential operator and $\Psi = \Psi(\vec{x}) (\vec{x} \in D \subset R^{(n)})$ with Dirichlet or Neumann boundary conditions. The boundary $B = \partial D$ can be any arbitrary complex geometrical in shape and is defined as a set of points that are chosen so as to represent its shape with reasonable accuracy. Collocation method is adopted and the problem is then transformed into the unconstrained optimization problem. A model based on the synergy of two feed forward neural networks of two different types are developed, which can be written as

$$\Psi_M(\vec{x}, p) = N(\vec{x}, p) + \sum_{l=1}^M q_l e^{-\lambda|\vec{x}-\vec{\alpha}_l+\vec{h}|^2} \quad (5)$$

where $|\cdot|$ denotes the Euclidean norm and $N(\vec{x}, p)$ is a multilayer perceptron with p denoting the set of its weights and biases. The sum in the above equation represents an RBF network with M hidden units that all share a common exponential factor λ . For a given set p of MLP parameters, the coefficients q_l are uniquely determined by requiring that the boundary conditions are satisfied. The authors demonstrated that the model based on the MLP-RBF synergy satisfies exactly the boundary conditions but are computationally expensive since at every evaluation of the model one needs to solve a linear system which may be quite large. Moreover, since many efficient optimization methods need the gradient of the objective function, one has to solve an additional linear system of the same order for each gradient component. On the other hand, penalty method is very efficient but does not satisfy the boundary conditions. Hence they used the combination of these two methods which is profitable and results are quite encouraging. They presented three examples to illustrate their method, the first example is a two dimensional linear Dirichlet problem on a irregular body. The second example is a highly non-linear problem with Dirichlet boundary problem considering star shaped domain, and the third example treats the same PDE with Neumann boundary conditions inside a cardioid family.

Remark 4. Solutions obtained by the given approach show that the method is equally effective, and retains its advantage over the Galerkin Finite element method. It also provides accurate solutions in a closed analytic form that satisfy the boundary conditions at the selected points. The method is quite general and can be used for a wide class of linear and non-linear partial differential equations.

In article [46], the authors described a method to solve partial differential equation and its boundary or initial conditions by using neural networks. They used the fact that multiple input, single output and single hidden layer feed forward networks with a linear output layer with no bias are capable of arbitrarily well approximating arbitrary functions and its derivatives. To clarify the working of their method, they apply the following differential equation with two of its initial conditions:

$$\frac{d^2 y}{dt^2} + y = 0, \quad t \in [0, 1] \quad (6)$$

$$\frac{dy}{dt} = 1, \quad y(0) = 0. \quad (7)$$

Then, they obtain networks of which some are specifically structured. To find the solution of the partial differential equation and its boundary and initial conditions they trained all obtained networks simultaneously as a consequence of their interrelationship. For this they implemented evolutionary algorithm to train the networks. They noticed that good results can be obtained if they restrict the values of the variables to the interval $[-5, 5]$.

Remark 5. The authors presented the knowledge about the partial differential equations and its boundary and/or initial conditions to incorporate into the structures and the training sets of several neural networks and found that the results for one and two dimensional problem are very good in respect of efficiency, accuracy, convergence and stability.

Alli et al. in [47] solved a mathematical model of vibration control of flexible mechanical systems, using multilayer perceptron technique. Because of non-linearity and complex boundary conditions, their numerical solutions always have some drawbacks such as numerical instability. That is why, they proposed an alternative method using feed forward artificial neural networks. Initially a general formula for the numerical solutions of n th order initial value problem is developed using ANN. Then the method is applied to many controlled and non-controlled vibration problems of flexible structures whose dynamics are represented by ODE's and PDE's, for e.g., they considered mass-damper-spring system, whose mathematical model is given by:

$$m \frac{d^2 \psi}{dt^2} + c \frac{d\psi}{dt} + k\psi = 0 \quad (8)$$

where the initial conditions of the systems are: $\psi(0) = 1$ and $\frac{d\psi(0)}{dt} = 0$ with $t \in [0, 2]$. The authors also considered the second and fourth order PDEs that are the mathematical models of the control of longitudinal vibrations of rods and lateral vibration of beams. To test their method, they also obtain the solutions of the same problems by using analytical and Runge–Kutta method. The obtained figures show that the results are very closed to the Runge–Kutta and analytical method.

Remark 6. It has been pointed out that the method presented in [47] can be used for a wide class of linear and non-linear PDEs with complex boundary conditions and the results obtained are very good. It has been also observed that the presented method also success outside the training points when the neuron numbers in the hidden layer are increased.

Smaoui et al. in [48] analyzed the dynamics of two non-linear partial differential equations known as the Kuramoto–Sivashinsky (K–S) equation and the two dimensional Navier–Stokes (N–S) equation using the combination of Karhunen–Loeve (K–L) decomposition and artificial neural networks. K–S equation is solved using pseudo spectral Galerkin method at bifurcation parameters $\alpha = 17.75$. K–L decomposition was thus applied on the numerical simulation data to extract coherent structures of the dynamical behavior represented by heteroclinic connection. The author then used the artificial neural network to model and predict P times steps into the future dynamical behavior at $\alpha = 17.75$ for different values of P . They found that the neural network model is able to capture the dynamics, and observed that as P increases, the model behavior degrades. And for the two dimensional N–S equation, a quasiperiodic behavior represented in phase space by a torus is analyzed at $Re = 14 : 0$. Applying the symmetry observed in the two-dimensional N–S equations on the quasiperiodic behavior, eight different tori were obtained. They showed that by exploiting the symmetries of the equation and using K–L decomposition in conjunction with neural networks, a smart neural model can be obtained.

Remark 7. In [48], the authors presented a hybrid method which is the combination of K–L decomposition and artificial neural network, to solve the partial differential equations. They established a foundation for modeling the two dimensional N–S equation using neural network for all values of Reynolds numbers and for different values of wave numbers.

In [49] Malek et al. investigated a novel hybrid method based on optimization techniques and neural network methods for the solution of high order ordinary differential equations. The authors considered the general initial/boundary value problem of the form:

$$\begin{cases} D \left[x, y(x), \frac{dy}{dx}, \frac{d^2y}{dx^2}, \dots, \frac{d^ny}{dx^n} \right] = 0 & x \in [a, b] \\ C \left[x, y(x), \frac{dy}{dx}, \frac{d^2y}{dx^2}, \dots, \frac{d^ny}{dx^n} \right] = 0 & x = a \text{ and/or } b \end{cases} \quad (9)$$

where D is a differential operator of degree n , C is an initial/boundary operator, x is an independent variable belonging to $[a, b]$ and y is an unknown dependent variable to be calculated. For problem formulation, they assume that a general approximation solution to Eq. (9) is of the form $y_T(x, P)$ where y_T is a dependent variable to x and P , where P is an adjustable parameter involving weights and biases in the structure of three layer feed forward neural networks and satisfies the following optimization problem:

$$\begin{cases} \text{Min}_P \int_a^b \left\| D \left[x, y_T(x, P), \frac{dy_T(x, P)}{dx}, \dots, \frac{d^ny_T(x, P)}{dx^n} \right] \right\|^2 dx \\ C \left[x, y_T, \frac{dy_T(x, P)}{dx}, \dots, \frac{d^ny_T(x, P)}{dx^n} \right] = 0, & x = a \text{ and/or } b. \end{cases} \quad (10)$$

In order to deal with Eq. (10) it is simpler to deal the following constrained optimization problem of the form $y_T(x, P) = \alpha(x) + \beta[x, N(x, P)]$, where first term involve adjustable parameters and satisfy initial or boundary conditions and second term is feed forward three layer perceptron. Minimization in Eq. (10) has been done which is considered as a training process for the proposed neural network. The error $E(x)$ corresponding to every entry x has to be minimized. The authors described,

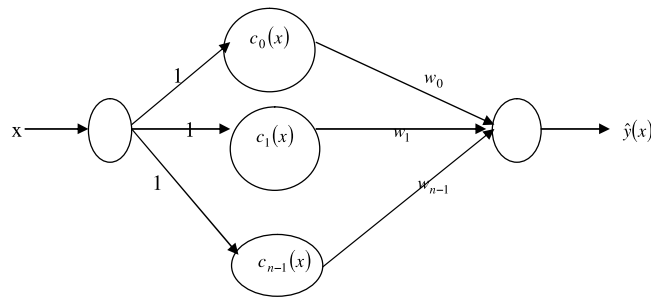


Fig. 1. Neural Network based on cosine based model.

the general formulation of initial or boundary value problems by taking differential equations of various order. They developed programs with MATLAB 6.5.1 to demonstrate behavior and properties of the proposed method and concluded that the proposed novel solution act as a good interpolation technique as well as a good extrapolation method for the close enough point outside the boundary points of the interval.

Remark 8. The main contribution of this article [49] is the potential of the hybrid technique to deal with differential equation of higher orders as well as low order. This hybrid technique is applicable for partial differential equation as well as ordinary differential equation and numerical simulation indicates that the proposed novel model solution is stable and computes the approximate solution in few numbers of iterations. Moreover, it acts as a good extrapolation technique for a good range of points outside the interval $[-1, 1]$.

An algorithm for the selection of both input variables and a sparse connectivity of the lower layer of connections in feed forward neural network of multilayer perceptron with one layer of hidden non-linear single linear output node is presented by Saxen et al. in [50]. Initially random weights in the lower layer of connections are used as a starting point and the complexity of this part of the network is gradually decreased. At every step of the algorithm, least significant connection is eliminated by a procedure, where the lower layer weights are zeroed, in turn, and the network corresponding to minimum approximation error is selected. In carrying out the comparison between the networks, the upper layer connection weights are determined by linear least squares, this feature made the method efficient and robust. The results of the algorithm are saved in a book-keeping matrix that can be interpreted in retrospect to suggest a suitable set of inputs and connectivity of the lower part of the network. Various test examples are presented by the authors to illustrate that the proposed algorithm is a valuable tool for the users in extracting relevant inputs from a set of potential ones.

Remark 9. This method is a systematic method that can guide the selection of both input variables and sparse connectivity of the lower layer of connections in feed forward neural networks of multi-layer perceptron type with one layer of hidden non-linear units and a single linear output node and the algorithm developed for the method is efficient, rapid and robust.

In article [51], the authors emphasized on solving the first order initial value problem in ordinary differential equations. They used the cosine function as the transfer function of neural network and supposed the error function is $e(x) = f(x, \hat{y}(x)) - \hat{y}'(x)$; and made $e(x) \rightarrow 0$ by training neural network. Thus, the $\hat{y}(x)$ approximate to $y(x)$. The model of neural network based on the cosine basis function is shown in Fig. 1.

Here, w_n is weight vector, $c_n(x) = \sum_{n=0}^{N-1} \cos(nx)$ is transfer function of the hidden layer neuron, $x \in [0, \pi]$, weight matrix vector is $W = [w_0, w_1, \dots, w_{n-1}]^T$ and transfer function matrix as $c(x) = (c_0(x), c_1(x), \dots, c_{n-1}(x))^T$, N is the number of hidden layer neurons and Neural network output is given as ;

$$\hat{y}(x) = \sum_{n=0}^{N-1} \omega_n \cos nx. \tag{11}$$

They developed a neural network algorithm, by considering the initial value problems in ordinary differential equations, for which weight is adjusted as;

$$w_n(k + 1) = w_n(k) - \mu e(k)[n \sin nx_k + f_y(x, \hat{y}(x_k)) \cos nx_k] \quad n = 0, 1, \dots, N - 1 \tag{12}$$

where μ is learning rate and $0 < \mu < 1$. They also described convergence criteria for neural network algorithm in terms of the following theorem:

Theorem. Suppose that μ is the learning rate, N is the number of hidden layer neurons, L is the upper limit of $\frac{\partial f(x,y)}{\partial y}$. Then

$$\left| \frac{\partial f(x,y)}{\partial y} \right| \leq L. \text{ If learning rate satisfies } 0 < \mu < \frac{12}{N(2N^2+6L^2-3N+1)}, \text{ then the algorithm is convergent.}$$

Then, the author tested the validity of the algorithm on a plenty of examples and compare the results with the other existing methods. They found that the algorithm is more precise, it provides a new approach on numerical computation, also it can give results at an arbitrary x between two adjacent nodes.

Remark 10. As per the results given in the article [51], it has been observed that the proposed algorithm is more precise than the Euler Method, Modified Euler Method and Heun's Method. However, compared to Runge–Kutta method of order four, the results of calculation are less precise but method is more flexible than any other difference methods.

In [52] the authors presented the numerical solution of non-linear Schrodinger equation by feed forward neural network. They considered the following time independent Schrodinger equation of the motion of a quantum particle in a one dimensional system:

$$\hat{H}\psi(x) \equiv \left(-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x) \right) \psi(x) = E\psi(x) \tag{13}$$

where m and $V(x)$ are the mass of a particle and the potential function, respectively. \hat{H} , $\psi(x)$, and E denote the system Hamiltonian, Eigen function and the Eigen value, respectively. The identity of quantum mechanical state can be represented by the wave function $\psi(x) = A(x) \cdot S(x)$ in the coordinate system. The authors represented the wave function by the feed forward artificial neural network, in which the coordinate value is regarded as an input while the networks output are assigned to two separate parts. They obtained energy function of the artificial neural network from the Schrodinger equation and its boundary conditions and used unsupervised training method for training the network. The present method is applied to the quantum infinite square well and one dimensional quantum harmonic oscillator systems and the result proves its wide applicability. The authors compared the accuracy of their method by comparing the results to the results that are analytically known and also by the Runge Kutta method of order four.

Remark 11. It has been pointed out that the method presented in [52] can solve the Schrodinger equation effectively with small number of unknown parameters in comparison with the other conventional methods e.g. Runge Kutta, Finite difference etc. The method can be used for dealing with the higher dimensions also.

In [53], the author proposed a hybrid method based on artificial neural networks, minimization techniques and collocation method to determine a related approximate solution in a closed analytic form of the time independent partial differential equations containing several arbitrary and boundary conditions of the form:

$$\begin{cases} \forall i_1 = 1, \dots, p_1 : D_{i_1} \left[t, x, \dots, \frac{\partial^{\alpha_0 + \alpha_1 + \dots + \alpha_n}}{\partial t^{\alpha_0} \partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}} y_i(t, x), \dots \right] = 0, & t \in [t_0, t_{\max}], x \in \Omega \\ \forall i_2 = 1, \dots, p_2 : I_{i_2} \left[t_0, x, \dots, \frac{\partial^{\alpha_0 + \alpha_1 + \dots + \alpha_n}}{\partial t^{\alpha_0} \partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}} y_i(t_0, x), \dots \right] = 0, & x \in \Omega, 1 \leq i \leq m \\ \forall i_3 = 1, \dots, p_3 : B_{i_3} \left[t, x, \dots, \frac{\partial^{\alpha_0 + \alpha_1 + \dots + \alpha_n}}{\partial t^{\alpha_0} \partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}} y_i(t, x), \dots \right] = 0, & t \in [t_0, t_{\max}], x \in \Omega \end{cases} \tag{14}$$

where the real valued multivariable functions D_{i_1} , I_{i_2} and B_{i_3} represents the known and non-linear time dependent system of partial differential equations, initial and boundary conditions, respectively, t is the time variable, x is the real valued spatial variable, $\Omega \subseteq R^n$ is a bounded domain, $(\alpha_0, \alpha_1, \dots, \alpha_n) \in N_0^{n+1}$ ($N_0 = N \cup \{0\}$) is multi index variable. They establish a trial approximate solution $y_T(t, x, P) = [y_{T_1}(t, x, P_1), \dots, y_{T_m}(t, x, P_m)]$ according to Kolmogorov and Cybenko theorems [54,55] that includes m three layered feed forward neural networks and contains adjustable parameters for the solution. The authors also proposed a specific trial solution for the bi-harmonic problem in order to simplify the corresponding energy function that must be minimized; they obtained mixed partial derivatives for the output of neural networks using multi index representation. The method is illustrated on various examples that involve a direct technique of the Nelder–Meed method [56] for optimization and the approximate solution obtained is in closed analytic form which works well for the points inside and outside the problem domain near boundary points.

Remark 12. The method proposed by the authors in the article [53] provides the approximate solution in a closed analytic form. Some advantages of this approach are that, it can solve the time dependent systems of partial differential equations, the method is generalized for solving the higher order and non-linear problems, it deals with a few number of parameters, solution is fast evaluated, the method can be applied to initial and two point boundary value problem for ordinary differential equations, and it uses parallel processing. Unlike the other methods, there is no ill conditioning of the concluded linear system in the expansion methods or the necessity of making a special relation among the step size for different axis in the finite difference method.

In [57], the author implemented multilayer perceptron and radial basis function neural networks methods to solve non-linear Schrodinger equation in Hydrogen atom and triangle shaped quantum well. They considered the following differential equation for illustrating the method:

$$\begin{aligned} H\psi(r) &= f(r), & \text{in } D \\ \psi(r) &= 0, & \text{on } \partial D \end{aligned} \tag{15}$$

where H is a differential operator, $f(r)$ is a unknown function, $D \subset R^3$ and ∂D is a boundary of Multilayer perceptron neural network method with gradient descent backpropagation algorithm was employed to solve the above differential equation.

The wave function was represented by the multilayer perceptron and radial basis function neural networks, in which the coordinate values are regarded as inputs while the network output is assigned to the Eigen value function and the energy function of the network is obtained from the differential equation and its boundary conditions with unsupervised training method for training the network. Eigen values and function are obtained by training the network. The results obtained by the given approach are compared with the Runge–Kutta method. This comparison shows the efficiency of the artificial neural network method for solving the differential equation. The present method is applied to the Hydrogen atom and two dimensional triangle shaped quantum well. The results proved wide applicability of the method and it is also shown that the present method can solve the differential equation with smaller number of unknown parameters with the conventional methods.

Remark 13. The proposed method provides more accurate results as compared to analytical solutions and two well known numerical methods Runge–Kutta and Finite Element Method. Also this method can be applied to solve the differential equations with smaller number of unknown parameters, whereas conventional methods require large number of unknown parameters.

In article [58], Tsoulos et al. used a hybrid method utilizing constructed feed-forward neural networks by grammatical evolution and a local optimization procedure, in order to solve ordinary differential equations, system of ordinary differential equations and partial differential equations. The construction of neural network with grammatical evolution was introduced by Tsoulos et al. [59]. They utilized the well established grammatical evolution technique [60] to evolve the neural network topology along with the network parameters. The authors applied the constructed neural network methodology on a series of differential equation while preserving the initial or boundary conditions using penalization. Considering the differential equation problem, a series of experiments in 19 well known problems was conducted, which shows that the proposed method can solve all the problems. The proposed method does not require the user to enter any information regarding the topology of the network and the advantage of using an evolutionary algorithm is that the penalty function can be incorporated easily into the training process.

Remark 14. The main advantage of the proposed method is that it has very less execution time and does not require a user to enter any parameter. This method can be easily parallelized, since it is based on the genetic algorithms and can be extended by using different BNF grammars for the constructed neural networks with different topologies or different activation functions.

Filici, in [61], presented a method of error estimation for the neural approximation of the solution of an ordinary differential equation. The author adopted the ideas presented by Zadunaisky [62,63] in order to provide a method that can estimate the errors in the solution of an ordinary differential equation by means of a neural network approximation. Firstly neural approximation to the ordinary differential equation problem is computed, and then neural neighboring problem is solved and the true error \bar{e} is estimated. A bound on the difference between the errors e and their estimations \bar{e} is derived, which is used to provide an heuristic criterion for the validity of the error estimation under some assumptions. A set of examples are presented by the author to show that the method can provide reasonable estimation of true errors. These examples also show that the criterion of validity works well in assessing the performance of the method.

Remark 15. In article [61], the author presented a novel method, by which the order of magnitude of the error can be correctly estimated in the neural approximation of solution of ordinary differential equation.

In [64], the authors attempted to present a novel method for solving fuzzy differential equations. Initially they replaced the fuzzy differential equation with the system of ordinary differential equations and obtained a trial solution of the system of ordinary differential equation using multilayer perceptron neural network technique. Initial condition is satisfied by the construction of trial solution and the network is trained to satisfy the differential equations. The author compared their result with the other existing numerical methods and showed that the use of neural network provides solution with very good generalization and high accuracy. They illustrated their method with the help of several examples.

Remark 16. It has been pointed out that the approximated solution obtained by proposed method in [64] is very close to the real solutions and has small errors. After solving a problem with the given approach, the solution of fuzzy differential equation is available for each arbitrary point in the training interval.

In article [65], the authors considered optimal control problems of discrete non-linear systems. They considered an invariant system; that is function f is independent from the time k . Moreover, the cost increment function r is also considered as independent of the time k . Taking horizon N as infinite, they deduced a equation:

$$\frac{\partial r}{\partial u} + \left(\frac{\partial f}{\partial u} \right)^T \cdot \frac{\partial J}{\partial x} [f(x, u)] = 0. \quad (16)$$

Eq. (16) is difficult to solve and analytical solutions are not usually possible to obtain, due to the non-linearity of the problem. Thus, the author used neural networks to solve the problem. They proposed to use two coupled neural networks to solve the Hamilton–Jacobi–Bellman equation in order to approximate nonlinear feedback optimal control solutions. Neural network weights have been updated using a modified version of the supervised backpropagation algorithm. In order to test the robustness of the proposed method, they consider two kinds of perturbations. The first type is obtained by variations on

the parameters of the system and in the second type of perturbation, they considered random noises caused by the sensor imperfections which affect the measured state variables. Simulation results are obtained, which show the performance and robustness of the proposed approach.

Remark 17. In article [65], the authors presented a new method to use two coupled neural networks for solving optimal control problems, whose analytical solution is not easy to obtain. The results obtained by the proposed method are highly acceptable and is robust towards parametric variations and measurement noises.

Dua, in [66], proposed a new method for parameter estimation of ordinary differential equations, which is based upon decomposing the problem into two sub problems. The first sub problem generates an artificial neural network model from the given data and then the second sub problem uses the artificial neural network model to obtain an estimate of the parameters of the original ordinary differential equation model. The analytical derivatives from the artificial neural network model obtained from the first sub problem are used for obtaining the differential terms in the formulation of the second sub problem. The author considered a problem $P1$ as:

$$\varepsilon_1 = \min_{\theta, z(t)} \sum_{i \in I} \sum_{j \in J} (\hat{z}_j(t_i) - z_j(t_i))^2 \quad (17)$$

subject to:

$$\frac{dz_j(t)}{dt} = f_j(z(t), \theta, t) \quad j \in J, \quad z_j(t=0) = z_{0j} \quad j \in J, \quad t \in [t_0, t_f]$$

and constructed the sub problems $P2$ and $P3$ of $P1$. Problem $P3$ involves only algebraic variable, θ , and therefore it can be solved to global optimality more efficiently than the original problem $P1$ involving differential as well as algebraic variables, z and θ . The author recognized that a simpler sub problem $P3$ is obtained by solving the first sub problem, to obtain the artificial neural network model. The proposed approach is tested on the various example problems and encouraging results have been obtained.

Remark 18. The author presented a very novel approach in [66] for parameter estimation using artificial neural networks. The main advantage of the proposed method is that requirement of high computational resources is avoided for computing the solution of a high optimization problem and instead of that two sub problems are solved. This approach is particularly useful for large and noisy data sets and nonlinear models where artificial neural networks are known to perform well.

3. Radial Basis Function (RBF) neural network techniques to solve differential equations

In the present section, a brief description of the research articles published recently for the solution of differential equations using radial basis function neural networks is given. Research articles presented here are arranged in the ascending order of their years of publication.

Mai-Duy and Tran-Cong [67] presented a mesh free procedure for solving linear differential equations based on multiquadric radial basis function networks (RBFNs). They developed direct and indirect radial basis function network procedures for solving differential equations by considering 2D Poisson equation over the domain Ω

$$\nabla^2 u = p(x), \quad x \in \Omega \quad (18)$$

where ∇^2 is the Laplacian operator, x is the spatial position, p is a known function of x and u is the unknown function of x to be found subject to Dirichlet or Neumann boundary conditions over the boundary Γ .

$$\begin{aligned} u &= p_1(x), \quad x \in \Gamma_1 \\ n \cdot \nabla u &= p_2(x), \quad x \in \Gamma_2 \end{aligned} \quad (19)$$

where n is the outward unit normal vector; ∇ is the gradient operator; Γ_1 and Γ_2 are the boundaries of the domain such as $\Gamma_1 \cup \Gamma_2 = \Gamma$ and $\Gamma_1 \cap \Gamma_2 = \phi$, p_1 and p_2 are known functions of x . They proposed that the solution u and its derivative can be approximated in terms of basis functions. In the present method [67], the width of the RBFs is the only adjustable parameter according to $a^{(i)} = \beta d^{(i)}$, where $d^{(i)}$ is the distance from the i th center to the nearest center. This approach is demonstrated on various numerical examples and solution obtained shows that the indirect radial basis function network (IRBFN) method produces results of several order magnitude than those associated with the direct radial basis function network (DRBFN) method. They compared the DRBFN method with the IRBFN and concluded that IRBFN methods are more accurate than the DRBFN over a wide range of β and hence the choice of RBF width is less critical. Different combination of RBF centers and collocation points are tested on both regularly and irregularly shaped domains.

Remark 19. Unlike the feed forward neural network (FFNN) method, the present procedure in [67] is non-iterative and hence it is more efficient. The IRBFN method with the optimal β value made the method very attractive in comparison to conventional methods such as FDM, FEM, FVM and BEM. Both regularly shaped and irregularly shaped domains can be handled by the method.

In [68], Jianyu et al. presented a multiquadric radial basis function network for solving linear ordinary differential equations (ODEs). The proposed method to solve differential equation relies on the whole domain and the whole boundary instead of the data set. Initially, the authors presented a method for approximation of the function and its derivatives. The RBFN procedure is described by the authors by considering the 2D Poisson equation as given in Eq. (17). It is proposed in the method that the solution u and its derivatives can be approximated in terms of basis function, and the design of network is based on the information provided by the given differential equation and its boundary conditions. Here the model u being decomposed into m basis functions in a given family and unknown parameters, are obtained by minimizing the error term. They considered linear and nonlinear examples of ordinary differential equations to illustrate their method. All the parameters can be determined at the same time without a learning process.

Remark 20. The advantage of the given technique in [68] is that, it does not need sufficient data and based on the domain and the boundary only. The result obtained through this method is more attractive than the previous method given in [68]. ODEs as well as the PDEs can be solved by the proposed method and the boundaries are not necessarily strict.

In [69], the authors presented a combination of new mesh free radial basis function network (RBFN) methods and a domain decomposition (DD) technique for approximating functions and solving Poisson's equation. Since the IRBFN procedure achieves greater accuracy than DRBFN over a wide range of RBF widths for function approximation [70], therefore the IRBFN method is considered in conjunction with a domain decomposition technique for approximation of functions and solving partial differential equations particularly Poisson's equation. They developed a new feature of the IRBFN method for the approximation, so that the difficulties related to solving big matrices can be overcome by using a subregioning technique. Each sub region is approximated by a separate RBFN and the network is trained independently and, if desired in parallel. Subregioning of the domain provides an effective means of keeping the size of the system matrices down while improving accuracy with increasing data density. Authors developed the boundary integral equations based domain decomposition method for the estimation of boundary conditions at interfaces in solving given Poisson's equation of potential problem.

$$\nabla^2 u = b, \quad x \in \Omega \quad (20)$$

$$u = \bar{u}, \quad x \in \delta\Omega_u \quad (21)$$

$$q = \frac{\partial u}{\partial n} = \bar{q}, \quad x \in \partial\Omega_q \quad (22)$$

where u is potential, q is the flux across the surface with unit normal, n , \bar{u} and \bar{q} are the known boundary conditions, b is a known function of position and $\partial\Omega = \partial\Omega_u + \partial\Omega_q$ is the boundary of the domain Ω . In their method, the interface boundary conditions are first estimated by using boundary integral equations (BIEs) at each iteration and sub domain problems are then solved by using the RBFN method. Also the volume integrals in standard integral equation representation (IE), which usually require volume discretization, are completely eliminated in order to present the mesh free RBFN method. The convergence rate of the approach can be affected by the element type used to compute BIEs.

Remark 21. The numerical examples show that the RBFN methods in conjunction with domain decomposition technique not only achieve a reduction of memory requirement but also a high accuracy of the solution. The results obtained by the method [69] are excellent on both rectangular and non rectangular domains. The boundary integral equation based domain decomposition method is very suitable for coarse-grained parallel processing and can be extended to those problems whose governing equation can be expressed in terms of integral equations such as viscous flow problems.

Mai-Duy and Tran-Cong in [70] presented a numerical approach, based on radial basis function of one, two and three variables and its derivatives. They considered the problem that a set of data points whose elements consist of paired values of the independent variables (a vector x) and the dependent variable (a scalar y), denoted by $\{x^{(i)}, y^{(i)}\}_{i=1}^n$, where n is the number of input points and $x = [x_1, x_2, \dots, x_p]^T$, where p is the number of dimension and the superscript T denotes the transpose operation. For which we have to find a closed form approximate function f of the dependent variable y and its closed form approximate derivative functions. The direct radial basis function network (DRBFN) as well as indirect radial basis function network (IRBFN) method is proposed to approximate the function and its derivative. Following are some common type of RBFNs that are of particular interest in the study, e.g.

$$\text{Multiquadrics-}\Phi^{(i)}(r) = \Phi^{(i)}(\|x - c^{(i)}\|) = \sqrt{r^2 + a^{(i)2}} \quad (23)$$

$$\text{Inverse Multiquadrics-}\Phi^{(i)}(r) = \Phi^{(i)}(\|x - c^{(i)}\|) = \frac{1}{\sqrt{r^2 + a^{(i)2}}}, \quad (24)$$

$$\text{Gaussians-}\Phi^{(i)}(r) = \Phi^{(i)}(\|x - c^{(i)}\|) = \exp\left(\frac{-r^2}{a^{(i)2}}\right), \quad (25)$$

for some $a^{(i)} > 0$, where $a^{(i)}$ is usually referred to as the width of the i th basis function and $r = \|x - c^{(i)}\| = \sqrt{(x - c^{(i)}) \cdot (x - c^{(i)})}$. In the DRBFN method, the closed form approximating function is first obtained from a set of training points and the derivative functions are then calculated directly by differentiating such closed form RBFNs. In the case

of IRBFN, the formulation of the problem starts with the decomposition of the derivative of the function into RBF's. The derivative expression is then integrated to yield an expression for the original function, which is then solved via the general least squares principle which gives an appropriate set of discrete points. Both the DRBFN and IRBFN methods are able to offer better results in comparison with the conventional methods using linear shape functions. The presented RBFN methods also eliminate the need for FE-type discretization of the domain of analysis, since the input data consist of a set of unstructured discrete data points. The authors found that among the RBF's considered, MQRBF offers the best performance in accuracy in the case of both the DRBFN and IRBFN method. The results obtained are compared with the feed forward neural network and it was found that the method is appropriate.

Remark 22. The DRBFN as well as the IRBFN method are presented by the author in [70]. The results obtained by the method show that IRBFN methods achieve greater accuracy than DRBFN in the approximation of both functions and its derivatives and this superior accuracy is maintained over a wide range of RBF widths ($0.2 < \beta > 9$).

Jianguo et al. [71] in 2003 defined a neural network for solving PDE in which the activation function of the hidden nodes are the RBF, whose parameters are determined by the two stage gradient descent strategy. They described different methods for approximation of the function and its derivatives. The authors illustrated their method by considering a 2D Poisson equation

$$\Delta u = P(x), \quad x \in \Omega \quad (26)$$

where Δ is Laplace operator, x is the spatial function, P is known function of x and u is the unknown function of x to be found subject to the Dirichlet and Neumann boundary conditions over boundary

$$\begin{aligned} u &= P_1(x), \quad x \in \delta\Omega \\ n \times \nabla u &= P_2(x), \quad x \in \delta\Omega_2 \end{aligned} \quad (27)$$

n is the outward unit normal, ∇ is gradient operator, $\delta\Omega_1, \delta\Omega_2$ is the boundary of domain such that $\delta\Omega_1 \cup \delta\Omega_2 = \delta\Omega$ and $\delta\Omega_1 \cap \delta\Omega_2 = \phi$ and P_1, P_2 are known functions of x . They introduced a new incremental algorithm for growing RBF networks and a two-stage learning strategy for training the network parameters. The novelty of the algorithm has different aspects. First of all, all learning procedures are accomplished by a two-stage gradient method which not only optimizes the weights of the network, but also optimizes the centers and widths of the neurons. The learning strategy is able to save computational time and memory space because of the selective growing of nodes whose activation functions consist of different radial basis functions. Moreover, the activation function of the added neuron can be a different radial basis function. The results proved that the strategy is better than the algorithm whose neurons are the same RBF type.

Remark 23. In this article [71], the authors focused on how to choose the activation function and how to increase the number of nodes as little as possible. The network here allows us to obtain fast and accurate solutions of differential equations starting from the randomly sampled data sets. The incremental architecture allows us to achieve good approximation results without wasting memory space and computational time.

Kansa et al. [72] proposed a finite volume analog of the meshless RBF method for the solution of system of nonlinear time dependent partial differential equations. They employed physical domain decomposition over piecewise continuous sub domains that are bounded by shocks, contact surfaces, or rarefaction fans. The authors converted the set of nonlinear multidimensional partial differential equations into a set of ordinary differential equations, by a series of rotational and translational transformations. They introduced an additional local transformation that maps these ordinary differential equations into compatibility or eigen vector ordinary differential equations that propagate at 'characteristic' velocities, thereby decoupling the compatibility ordinary differential equations. By writing the compatibility variables as a series expansion of RBF's with time dependent expansion coefficients, the time advanced problem is converted into a method of lines problem. The volume integration of the RBF's is performed in parallel by integrating over each sub domain Ω_i separately, and normalizing the results. They tracked strong shocks, capture weak shocks using artificial viscosity methods to dampen them and used Riemann solvers and shock polar method for shock wave interactions. When pairs of knots coincide, they introduced a discontinuous surface at the coincidence loci.

Remark 24. In [72], the authors presented a volumetric integral RBF formulation for the time dependent partial differential equations. Since volumetric integral formulation of time dependent conservation equations increases the convergence rates of radial basis function approximates, therefore fewer number of knots are required to discretize the domain.

Zou et al. [73] in 2005 presented a new kind of RBF neural network method based on Fourier progression, by adopting the trigonometric function as basis function. They used $\hat{W}B(x)$ to approximate a unknown function $f(x)$ where,

$$\begin{aligned} \hat{W} &= [\hat{w}_1, \hat{w}_2, \dots, \hat{w}_n, \hat{w}_{n+1}, \dots, \hat{w}_{2n}, \hat{w}_{2n+1}] \\ B(x) &= [\sin x, \dots, \sin nx, \cos x, \dots, \cos nx, c] \end{aligned} \quad (28)$$

with the Fourier progression theory that every continuous function $f(x)$ can be expressed as follows, $f(x) = c + \sum_{n=1}^{\infty} a_n \sin nx + \sum_{n=1}^{\infty} b_n \cos nx$, and constructed a neural network which is dense for continuous function space. They constructed an optimal weight matrix by assuming that, a function vector $h : \Omega \rightarrow R^p$, for any $\sigma > 0$, there always exist a

function array $B : R^m \rightarrow R^l$ and an optimal weight matrix W^* such that $\|h(x) - W^{*T}B(x)\| \leq \sigma, \forall x \in \Omega$, where Ω is a tight set of R^m and $h(x) - W^{*T}B(x) = \Delta h(x)$, $\hat{W} = \hat{W} - W^*$, where $\hat{W} \in R^{l \times 3}$ is used to estimate value of W^* . To apply the neural network to a practical system, a class of nonlinear systems was considered by the authors. Then it is used in a class of high order system with all unknown control function matrices. The adaptive robust neural controller is designed by using back stepping method and effectiveness of the method is presented by simulation study.

Remark 25. It has been pointed out that by adopting the trigonometric function as the basis function, the input need not be force between -1 and 1 , and there is no need to choose the center of basis function. Use of differential reconstruction in the above article [73] made the selection of simulation of parameter easier, which increased the damp of the system.

In article [74], the author presented a meshless method based on the radial basis function networks for solving high order ordinary differential equations directly. Two unsymmetric RBF collocation schemes, named the usual direct approach based on a differentiation process and the proposed indirect approach based on an integration process, are developed to solve high order ordinary differential equations. They considered the following initial value problem governed by the following p th order ordinary differential equation

$$y^{(p)} = F(x, y, y', \dots, y^{(p-1)}) \quad (29)$$

with initial conditions $y(a) = \alpha_1, y'(a) = \alpha_2, \dots, y^{(p-1)}(a) = \alpha_p$, where $a \leq x \leq b, y^{(i)}(x) = \frac{d^i y(x)}{dx^i}$, F is a known function and $\{\alpha_i\}_{i=1}^p$ is a set of prescribed conditions. Like other meshless numerical methods, the direct RBF collocation approach is based on the differential process to represent the solution. In the proposed RBF collocation approach, the closed forms representing the dependent variable and its derivatives are obtained through the integration process. In the case of solving high order ODEs, difficulties to deal with multiple boundary conditions are naturally overcome with integrating constants. Analytical and numerical techniques for obtaining new basis functions from RBF's are discussed. Among RBFs, multiquadrics are preferred for practical use. Numerical results show that the proposed indirect approach performs much better than the usual direct approach.

Remark 26. High convergence rates and good accuracy are obtained with the proposed method using a relatively low number of data points. The proposed unsymmetric IRBF collocation method can be extended to the solution of high order PDEs.

In [75], the authors presented a new indirect radial basis function collocation method for numerically solving bi-harmonic boundary value problem. They considered the bi-harmonic equation:

$$\frac{\partial^4 v}{\partial x_1^4} + 2 \frac{\partial^4 v}{\partial x_1^2 \partial x_2^2} + \frac{\partial^4 v}{\partial x_2^4} = F \quad (30)$$

in the rectangular domain Ω with F being a known function of x_1 and x_2 , which can be reduced to a system of two coupled Poisson's equations

$$\frac{\partial^2 v}{\partial x_1^2} + \frac{\partial^2 v}{\partial x_2^2} = u, \quad x \in \Omega, \quad \frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} = F, \quad x \in \Omega \quad (31)$$

since, in the case when boundary data are $\{v = r(x), \partial^2 v / \partial n^2 = s(x), x \in \partial \Omega\}$ the use of two Poisson equations is preferred as each equation has its own boundary condition. In this research article [75], the authors described the indirect radial basis function networks and proposed a new technique of treating the integrating constant for bi-harmonic problems, by eliminating integration constant point wise subject to the prescribed boundary conditions. It overcomes the problem of increasing size of conversion matrices caused by scattered points and provides an effective way to impose the multiple boundary conditions. Two types of boundary conditions $\left\{v, \frac{\partial^2 v}{\partial n^2} / u\right\}$ and $\left\{v, \frac{\partial v}{\partial n}\right\}$ are considered. The integration constant is excluded from the networks and employed directly to represent given boundary conditions. For each interior point, one can form a square set of k linear equations with k being the order of PDE's, from which the prescribed boundary conditions are incorporated into the system via integration constants. This is advancement in the indirect radial basis function collocation method for the case of discretizing the governing equation in a set of scattered data points.

Remark 27. The proposed new point wise treatment in article [75] overcomes the problem of increasing size of conversion matrices, and provides an effective way to implement the multiple boundary conditions without the need to use fictitious points inside or outside the domains or to employ first order derivatives at grid points as unknowns. This method is truly a meshless method, which is relatively easy to implement as expression for integration constants are given explicitly and this represents a further advancement in the case of IRBFN for the case of discretizing the governing equations on a set of scattered data points. Numerical results presented in this article show that the method achieves a high degree of accuracy and high rate of convergence.

Golbabai et al. in [76] implemented the RBF neural network method for solving linear integro-differential equations. They proposed the approach by considering the following equation:

$$Dy(x) - \lambda \int_{\Gamma} k(x, t)y(t)dt = g(x), \quad \Gamma = [a, b] \quad (32)$$

with the supplementary conditions as follows:

$$\begin{cases} Dy(x) = y'(x) + A_1(x)y(x), \\ y(\alpha_1) = \gamma_1, \end{cases} \quad (33)$$

$$\begin{cases} Dy(x) = y''(x) + A_1(x)y'(x) + A_2(x)y(x), \\ y(\alpha_1) = \gamma_1, \quad y'(\alpha_2) = \gamma_2 \end{cases} \quad (34)$$

where D is the differential operator, λ , γ_1 , and γ_2 are constants, $\alpha_1, \alpha_2 \in \Gamma$, A_1, A_2, g and k are known functions and y is the unknown function to be determined. For illustrating the method they rewrite Eq. (34) in the following operator form

$$Dy - \lambda Ky = g, \quad (35)$$

where, $(Ky)(x) = \int_{\Gamma} k(x, t)y(t)dt$, and used the collocation method which assumes discretization of the domain into a set of collocation data. They assumed an approximate solution $y_p(x)$ such that it satisfies the supplementary conditions and the quasi-Newton Broyden–Fletcher–Goldfarb–Shanno (BFGS) method is used for training the RBF network. The authors also described an algorithm which is used in their experiment. The main attraction of their algorithm is that it starts with a single training data and with a single hidden layer neuron, then continues the training patterns one by one and allows the network to go. Various numerical examples are considered to demonstrate the proposed idea and method. Golbabai et al. also described the radial basis function neural network method for solving the system of non linear integral equations in [77].

Remark 28. The result obtained by this approach [76] proves that the RBF neural network with quasi-Newton BFGS technique as a learning algorithm provides a high accuracy of the solution. Also the approach is quite general and appears to be the best among approximation methods used in the literature. This method is recommended by the author for use in solving a wide class of integral equations because of its ease of implementation and high accuracy. Moreover, the reported accuracy can be improved further by increasing the number of training data and the number of hidden units in the RBF network to some extent.

In [78], the authors presented a new Cartesian-grid collocation method based on radial basis function networks (RBFNs) for numerically solving elliptic partial differential equations (PDEs) in irregular domains. The proposed method combines the efficiency of a Cartesian-grid method and the property of high-order convergence of a 1D-integrated RBF interpolation scheme. Poisson equations with Dirichlet boundary conditions, biharmonic equations with Dirichlet boundary conditions and Poisson equations with Dirichlet and Neumann boundary conditions are investigated by the authors. One-dimensional integrated RBFNs are employed to represent the variable along each line of the grid, resulting in a significant improvement of computational efficiency. It does not require complicated interpolation techniques for the treatment of Dirichlet boundary conditions in order to achieve a high level of accuracy.

Remark 29. The proposed method requires much less computation effort than the 2D-IRBFN method. Numerical tests show that the method yields fast convergence, i.e. third and fifth-order accuracy with respect to grid size for the solution of second and fourth order PDE's.

The research article [79] introduced a variant of direct and indirect radial basis function networks for the numerical solution of Poisson's equation. In this method they initially described the DRBFN and IRBFN procedure described by Mai Duy and Tran Cong in [69] for the approximation of both functions and their first and higher order derivatives. The authors illustrated the method by considering a numerical example of a two-dimensional Poisson equation:

$$\nabla^2 u = \sin(\pi x_1) \sin(\pi x_2) \quad (36)$$

where, $0 \leq x_1 \leq 1$, and $0 \leq x_2 \leq 1$ with $u = 0$ on whole boundary points. They consider 20 points, 11 of those were boundary points and 9 were interior points, and used multiquadric radial basis function method which Mai Duy and Tran Cong in [67] had used. Then they computed the approximate solution by converting the Cartesian coordinate into polar:

$$\nabla^2 u = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2}. \quad (37)$$

They found that the approximated solution of this new method is better than both DRBFN and IRBFN method on the Cartesian ones. Further, they applied this method to the two-dimensional Poisson equation in the elliptical region and achieved better accuracy in the terms of root mean square error.

Remark 30. In the above approach [79], it has been shown that transformation of Poisson's equation into polar coordinates can achieve a better accuracy than the DRBFN and IRBFN methods on the Cartesian ones. Also, the accuracy of the IRBFN method is influenced by the width parameter of the radial basis functions such that this parameter must be in the special range and as it increases the condition number increases too, but in this method, variations of the width parameter of a basis function do not influence the accuracy of the numerical solution. Hence the condition number is small and the obtained system is stable.

Chen et al. in [80] proposed a method that develops a mesh free numerical method for approximating and solving PDEs, based on the integrated radial basis function networks (IRBFNs) with adaptive residual sub sampling training scheme. An integrated radial basis function network for approximating and solving PDEs is described initially. In this article, the authors adopted the residual sub sampling scheme suggested in [81] to train the IRBF network. In the training process, neurons are added and removed based on the residuals evaluated at a finer point set, and the shape parameter adjusting scheme is modified to suit the IRBF neuron behavior which is different from the DRBF network. They simply considered the shape parameters by multiplying the distances between two neighboring neurons with a fixed coefficient and the multiquadric function is taken as the transfer function of the neurons. The adaptive process for determining the locations of neurons in integrated radial basis function networks for approximating a one-dimensional function is described by the training procedure. During the training procedure, two neurons whose centers are end points which are always kept fixed. Numerical examples are conducted to show the effectiveness of the method.

Remark 31. Since IRBFNs are capable of smoothing the derivative errors for solving PDEs, therefore with the proposed adaptive procedure, IRBFNs require less neurons to attain accuracy than DRBFN. Approximation based on smooth IRBFNs is highly effective in approximating smooth functions, even if the neuron sets are relatively coarse. The adaptive method applied for training in this article is an effective technique for dealing with the steep and corner feature of the PDE solutions; and the IRBF networks contribute to improve the accuracy of solving PDEs. Hence a combination of IRBF and adaptive algorithm is a promising approach for mesh free solutions of PDEs. This method can easily be applied for solving higher dimension problems and time dependent nonlinear equations.

4. Comparison of MLP and RBF techniques for solving differential equations

In this section, we will discuss some research articles in which comparison of MLP and RBF techniques for solving differential equations have been presented.

Choi and Lee in [82] presented an article to give a comparison on solving differential equation using backpropagation and radial basis function neural networks. An already developed approach for solving differential equation using neural networks described in [43,45] is used by the authors to illustrate the Backpropagation algorithm for solving the differential equations. They considered an example of an ordinary differential equation given in [83] and solved it using MLP and RBF neural networks.

$$\frac{d\psi}{dx} + \frac{1}{5}\psi = e^{-x/5} \cos(x). \quad (38)$$

In order to compare the performance of reformulated RBFN and Backpropagation specified for solving the differential equation problem, they performed some experiments using a network with a 1-2-1 and 1-5-1 structure. Each experiment was performed 20 times and averaged with randomly selected initial parameters, weight vectors or centers with values ranged from -1 to 1 , learning from 0.01 to 0.001 and momentum from 0.1 to 0.9 . They summarized and compared the result using four types of networks: backpropagation with sigmoid activation function, backpropagation with tangent hyperbolic activation function, the general RBFN with exponential RBF and the reformulated RBFN with the simplified cosine RBF.

Remark 32. The authors finally concluded that the reformulated RBFN with adaptation to the gradient descent algorithm have more generalized accuracy and alleviation of the local minima problem than the backpropagation algorithm.

In [84], the authors presented a comparative study of RBF and MLP neural networks for predicting critical heat flux. Critical heat flux (CHF) is an important parameter for the design of nuclear reactors and much experimental and theoretical research has been performed but there is not a single correlation to predict CHF because it is influenced by many parameters. Experimental data generally have a limited range of system parameters such as pressure, mass flux, etc., and also there are more than 500 correlations available for water flow inside tubes but each correlation has a limited range of application and accuracy over a range of databases. Hence they imposed the RBF and ANN technique to evaluate CHF using experimental databases. The overall procedure for the CHF prediction is given in four steps: (i) prepare the input data, (ii) normalize inputs, (iii) train the network and (iv) predict the CHF. They predicted the CHF with MLP and RBF in fixed inlet and outlet conditions and compared the parametric trends. In the case of inlet conditions: (a) The CHF increase with increasing mass flux, but in the constant pressure it is stronger at lower L/D . (b) The CHF decreases with increasing L/D , so RBF has a better performance for this parameter. (c) The pressure effect is more complex than other parameters. Overall, the CHF increases with pressure at low pressure, passes through a maximum, and then decreases at higher pressures. Whereas, in the case of outlet conditions: (a) CHF increases with increasing mass flux (b) CHF decreases with increasing L/D (c) The CHF increases with pressure at low pressure, passes through a maximum, and then decreases at higher pressures. They found that the results on fixed outlet conditions with MLP shows some complexity but RBF results have a better performance.

Remark 33. Finally they concluded from the overall study that ANN is much faster and easy to make predictions with high accuracy and can be applied successfully to predict CHF. RBF networks give better results against the other ANN structures like MLP and can be used for other parameters like CHF. They also used a new method for input data normalization and found that RBF predicts CHF with root mean square (RMS) errors of 0.24%, 7.9%, 0.16% and MLP predicts CHF with RMS errors of 1.29%, 8.31% and 2.71%, in fixed inlet conditions, local conditions and fixed outlet conditions, respectively. The results show that neural networks with RBF structure have superior performance in CHF data prediction over MLP neural networks.

The work presented in [85] is an attempt to resolve the crucial issues which arise during the solution the inverse kinematic (IK) problems in the 6R serial robot, by proposing a fusion approach that makes use of RBF–ANN for prediction of joint angles and forward kinematics relations to determine the deviation of the pose of end-effector obtained from predicted joint angles from the desired pose. The data required for training of ANN is proposed to be derived from the forward kinematics relations instead of deriving a set of complex inverse kinematics equations. The scope of the author's work is limited to computation of inverse kinematics solutions at the various poses of the path obtained in Cartesian coordinates from the high-end path planner. The proposed fusion approach which makes use of RBF–ANN for prediction of IK solutions is implemented to obtain the IK solutions for five poses of the path. They obtained the inverse kinematic solutions with RBF and MLP–ANN techniques and concluded that the IK solution obtained with the RBF–ANN fusion process is superior to the solutions obtained with other approaches. They also calculated the IK solution for RBF and MLP techniques separately and compared them.

Remark 34. In this article, the authors noticed that a large error occurs with MLP networks to the premature termination of the network training process (approximately after training for 26 h). However, RBF network training is continued until the convergence condition is reached. On the contrary, MLP shows a better performance when used with the proposed fusion process. We refer to Refs. [86–90] for a detailed comparison between the MLP and RBF techniques for solving differential equations arising in real life problems.

5. Conclusion and direction for further development

In the present paper, the computational methods based on artificial neural networks for solving differential equations are briefly discussed and a huge amount of literature related to problems of differential equations of different types from various fields was analyzed. This paper is devoted to the crux of various research articles published in refereed journals to get a better view of the subject. This survey article may be an excellent reference for researchers to offer a state-of-the-art of the most active recent developments of methods for solving differential equations with their applications and remaining challenges in artificial neural networks.

Finally, we conclude that a large amount of work has been done in the respective field. It is well known that many phenomena in biology, chemistry, engineering and physics can be described by various types of differential equations or systems. When we associate an artificial neural network (ANN) technique to the phenomenon, it adds many attractive features to the solution that is the solution obtained via ANN is differentiable, in a closed analytic form and can be used in any subsequent calculation. Moreover, the employment of neural networks provides a solution with very good generalization properties compared to FEM. By comparing two techniques of ANN that is MLP and RBFNN, it can also be concluded that the RBFNN technique provides more accurate solutions to the MLP neural network and provides excellent generalization.

We are particularly interested to motivate the researchers to solve differential equations of various types arising in science and engineering, on which a significant amount of work is currently being done in the literature dedicated to multilayer perceptron (MLP) neural networks and radial basis function neural networks (RBFNN).

Acknowledgments

The authors extend their appreciation to anonymous reviewers for their valuable suggestions in revising this paper.

References

- [1] A.R. Its, A.G. Izergin, V.E. Korepin, N.A. Slavnov, Differential equations for quantum correlation functions, *International Journal of Quantum Physics B* (1990) 1003–1037.
- [2] A.V. Kotikov, Differential equations method: the calculation of vertex-type Feynman diagrams, *Physics Letters B* 259 (1991) 314–322.
- [3] Y.Z. Peng, Exact solutions for some nonlinear partial differential equations, *Physics Letters A* 314 (2003) 401–408.
- [4] J.G. Verwer, J.G. Blom, M.V. Loon, E.J. Spee, A comparison of stiff ODE solvers for atmospheric chemistry problems, *Atmospheric Environment* 30 (1996) 49–58.
- [5] J. Behlke, O. Ristau, A new approximate whole boundary solution of the Lamm differential equation for the analysis of sedimentation velocity experiments, *Biophysical Chemistry* 95 (2002) 59–68.
- [6] U. Salzner, P. Otto, J. Ladik, Numerical Solution of partial differential equation system describing chemical kinetics and diffusion in a cell with the aid of compartmentalization, *Journal of Computational Chemistry* 11 (1990) 194–204.
- [7] R.V. Culshaw, S. Ruan, A delay differential equation model of HIV infection of CD4 + T-cells, *Mathematical Biosciences* 165 (2000) 27–39.
- [8] G.A. Bocharov, F.A. Rihan, Numerical modeling in biosciences using delay differential equations, *Journal of Computational and Applied Mathematics* 125 (2000) 183–199.
- [9] R. Noberg, Differential equations for moments of present values in life insurance, *Mathematics and Economics* 17 (1995) 171–180.
- [10] J.C. Butcher, *The Numerical Analysis of Ordinary Differential Equations: Runge Kutta and General Linear Methods*, Wiley Interscience, New York, NY, USA, 1987.

- [11] J.G. Verwer, Explicit Runge–Kutta methods for parabolic partial differential equations, *Applied Numerical Mathematics* 22 (1996) 359–379.
- [12] J. Douglas, B.F. Jones, On predictor–corrector methods for nonlinear parabolic differential equations, *Journal of the Society for Industrial and Applied Mathematics* 11 (1963) 195–204.
- [13] R.W. Hamming, Stable predictor corrector methods for ordinary differential equations, *Journal of the ACM* 6 (1959) 37–47.
- [14] M. Petracic, G. Kuo Petracic, K.V. Roberts, Automatic production of programmes for solving partial differential equations by finite difference methods, *Computer Physics Communications* 4 (1972) 82–88.
- [15] A.A. Tseng, S.X. Gu, A finite difference scheme with arbitrary mesh system for solving high order partial differential equations, *Computers & Structures* 31 (1989) 319–328.
- [16] B. Wu, R.E. White, One implementation variant of finite difference method for solving ODEs/DAEs, *Computers and Chemical Engineering* 28 (2004) 303–309.
- [17] V.G. Korneev, Iterative methods of solving system of equations of the finite element method, *USSR Computational Mathematics and Mathematical Physics* 17 (1977) 109–129.
- [18] M. Kumar, P. Kumar, Computational method for finding various solutions for a quasilinear elliptic equation of Kirchhoff type, *Advances in Engineering Software* 40 (2009) 1104–1111.
- [19] V. Thomee, From finite differences to finite elements: a short history of numerical analysis of partial differential equations, *Journal of Computational and Applied Mathematics* 128 (2001) 1–54.
- [20] M. Kumar, P.K. Srivastava, Computational techniques for solving differential equations by cubic, quintic and sextic spline, *International Journal for Computational Methods in Engineering Science & Mechanics* 10 (2009) 108–115.
- [21] S. Sallam, W. Ameen, Numerical solution of general n th order differential equations via splines, *Applied Numerical Mathematics* 6 (1990) 225–238.
- [22] H.M. El-Hawary, S.M. Mahmoud, Spline collocation methods for solving delay differential equations, *Applied Mathematics and Computation* 146 (2003) 359–372.
- [23] M. Kumar, Y. Gupta, Methods for solving singular boundary value problems using splines: a survey, *Applied Mathematics and Computation* 32 (2010) 265–278.
- [24] N. Shawagfeh, D. Kaya, Comparing numerical methods for the solutions of systems of ordinary differential equations, *Applied Mathematics Letters* 17 (2004) 323–328.
- [25] J. Liu, G. Hou, Numerical solutions of the space- and time-fractional coupled Burgers equations by generalized differential transform method, *Applied Mathematics and Computation* 217 (2011) 7001–7008.
- [26] P. Darnia, A. Ebadian, A method for the numerical solution of the integro-differential equations, *Applied Mathematics and Computation* 188 (2007) 657–668.
- [27] M. Kumar, Parul, Methods for solving singular perturbation problems arising in science and engineering, *Mathematical and Computer Modelling* 54 (2011) 556–575.
- [28] A. Saadatmandi, M. Dehghan, Numerical solution of higher order linear Fredholm integro-differential difference equation with variable coefficients, *Computers & Mathematics with Applications* 59 (2010) 2996–3004.
- [29] A. Tocino, R. Ardanuy, Runge Kutta methods for the numerical solution of stochastic differential equations, *Journal of Computational and Applied Mathematics* 138 (2002) 219–241.
- [30] S. Abbasbandy, A. Taati, Numerical solution of nonlinear Volterra integro-differential equations with nonlinear differential part by the operational Tau method and error estimation, *Journal of Computational and Applied Mathematics* 231 (2009) 106–113.
- [31] M. Kumar, N. Singh, A collection of computational techniques for solving singular boundary value problems, *Advances in Engineering Software* 40 (2009) 288–297.
- [32] S. Momani, K. Al-Khaled, Numerical solutions for systems of fractional differential equations by the decomposition method, *Applied Mathematics and Computation* 162 (2005) 1351–1365.
- [33] B.P. Sommeijer, P.J. Vander Houwen, B. Neta, Symmetric multistep methods for second-order differential equations with periodic solutions, *Applied Numerical Mathematics* 2 (1986) 69–77.
- [34] M. Ghasemi, M.T. Kajani, E. Babolian, Numerical solutions of the nonlinear integro-differential equations: wavelet Galerkin method and homotopy perturbation method, *Applied Mathematics and Computation* 188 (2007) 450–455.
- [35] M. Kumar, P. Kumar, Computational method for finding various solutions for a quasilinear elliptic equation of Kirchhoff type, *Advances in Engineering Software* 40 (2009) 1104–1111.
- [36] K. Maleknejad, A. Arzhang, Numerical solution of the Fredholm singular integro-differential equation with Cauchy kernel by using Taylor-series expansion and Galerkin method, *Applied Mathematics and Computation* 182 (2006) 888–897.
- [37] M. Kumar, H.K. Mishra, P. Singh, A boundary value approach for singularly perturbed boundary value problems, *Advances in Engineering Software* 40 (2009) 298–304.
- [38] Q. Wang, D. Cheng, Numerical solution of damped nonlinear Klein–Gordon equations using variational method and finite element approach, *Applied Mathematics and Computation* 162 (2005) 381–401.
- [39] M. Kumar, N. Singh, Modified Adomian decomposition method and computer implementation for solving singular boundary value problems arising in various physical problems, *Computers and Chemical Engineering* 34 (2010) 1750–1760.
- [40] X. Yang, Y. Liu, S. Bai, A numerical solution of second order linear partial differential equation by differential transform, *Applied Mathematics and Computation* 173 (2006) 792–802.
- [41] V.S. Erturk, S. Momani, Solving system of fractional differential equations using differential transform method, *Journal of Computational and Applied Mathematics* 215 (2008) 142–151.
- [42] A.J. Meadre, A.A. Fernandez, The numerical solution of linear ordinary differential equations by feed forward neural networks, *Mathematical and Computer Modelling* 19 (1994) 1–25.
- [43] I.E. Lagaris, A. Likas, Artificial neural networks for solving ordinary and partial differential equations, *IEEE Transactions on Neural Networks* 9 (1998) 987–1000.
- [44] S. He, K. Reif, R. Unbehauen, Multilayer neural networks for solving a class of partial differential equations, *Neural Networks* 13 (2000) 385–396.
- [45] I.E. Lagaris, A.C. Likas, Neural network methods for boundary value problems with irregular boundaries, *IEEE Transactions on Neural Networks* 11 (5) (2000) 1041–1049.
- [46] L.P. Aarts, P.V. Veer, Neural network method for solving partial differential equations, *Neural Processing Letters* 14 (2001) 261–271.
- [47] H. Alii, A. Ucar, Y. Demir, The solution of vibration control problems using artificial neural networks, *Journal of the Franklin Institute* 340 (2003) 307–325.
- [48] N. Smaoui, S. Al-Enezi, Modelling the dynamics of nonlinear partial differential equations using neural networks, *Journal of Computational and Applied Mathematics* 170 (2004) 27–58.
- [49] A. Malek, R.S. Beidokhti, Numerical solution for high order differential equations using a hybrid neural network-optimization method, *Applied Mathematics and Computation* 183 (2006) 260–271.
- [50] H. Saxen, F. Petterson, Method for the selection of inputs and structure of feed forward neural networks, *Computers and Chemical Engineering* 30 (2006) 1038–1045.
- [51] X. Li-Ying, W. Hui, Z. Zhe-Zhao, The algorithm of neural networks on the initial value problems in ordinary differential equations, in: 2nd IEEE Conference on Industrial Electronics and Applications, 2007, pp. 813–816.
- [52] Y. Shirvany, M. Hayati, R. Moradian, Numerical solution of the nonlinear Schrodinger equation by feedforward neural networks, *Communications in Nonlinear Science and Numerical Simulation* 13 (2008) 2132–2145.

- [53] R.S. Beidokhti, A. Malek, Solving initial boundary value problems for system of partial differential equations using neural network and optimization techniques, *Journal of the Franklin Institute* 346 (2009) 898–913.
- [54] A.N. Kolmogorov, On the representation of continuous functions of several variables by superpositions of continuous functions of one variable and addition, *American Mathematical Society Translations* 28 (1963) 55–59.
- [55] G. Cybenko, Approximation of superpositions of a sigmoidal function, *Mathematics of Control, Signals, and Systems* 2 (1989) 304–314.
- [56] J.A. Nelder, R. Mead, A simplex method for function minimization, *The Computer Journal* 7 (1965) 308–313.
- [57] Y. Shirvany, M. Hayati, R. Moradian, Multilayer perceptron neural network with novel unsupervised training method for the solution of partial differential equations, *Applied Soft Computing* 9 (2009) 20–29.
- [58] I.G. Tsoulos, D. Gavrilis, E. Glavas, Solving differential equations with constructed neural networks, *Neurocomputing* 72 (2009) 2385–2391.
- [59] I. Tsoulos, D. Gavrilis, E. Glavas, Neural network construction and training using grammatical evolution, *Neurocomputing* 72 (2008) 269–277.
- [60] M. O'Neill, C. Ryan, Grammatical evolution, *IEEE Transactions on Evolutionary Computation* 5 (2001) 349–358.
- [61] C. Filici, Error estimation in the neural network solution of ordinary differential equations, *Neural Networks* 23 (2010) 614–617.
- [62] P.E. Zadunaisky, On the estimation of errors propagated in the numerical integration of ordinary differential equations, *Numerische Mathematik* 27 (1976) 21–39.
- [63] P.E. Zadunaisky, On the accuracy in the numerical solution of the N -body problem, *Celestial Mechanics* 20 (1979) 209–230.
- [64] S. Effati, M. Pakdaman, Artificial neural network approach for solving fuzzy differential equations, *Information Sciences* 180 (2010) 1434–1457.
- [65] N.K. Masmoudi, C. Rekkik, M. Djemel, N. Derbel, Two coupled neural network based solution of the Hamilton–Jacobi–Bellman equation, *Applied Soft Computing* 11 (2011) 2946–2963.
- [66] V. Dua, An artificial neural network approximation based decomposition approach for parameter estimation of system of ordinary differential equations, *Computers and Chemical Engineering* 35 (2011) 545–553.
- [67] N. Mai-Duy, T. Tran-Cong, Numerical solution of differential equations using multiquadric radial basis function networks, *Neural Networks* 14 (2001) 185–199.
- [68] L. Jianyu, L. Siwei, Q. Yingjian, H. Yaping, Numerical solution of differential equations by radial basis function neural networks, in: *Proceedings of International Joint Conference on Neural Networks, 2002*, pp. 773–777.
- [69] N. Mai-Duy, T. Tran-Cong, Mesh-free radial basis function network methods with domain decomposition for approximation of functions and numerical solution of Poisson's equation, *Engineering Analysis with Boundary Elements* 26 (2002) 133–156.
- [70] N. Mai-Duy, T. Tran-Cong, Approximation of function and its derivatives using radial basis function networks, *Applied Mathematical Modelling* 27 (2003) 197–220.
- [71] L. Jianyu, Luo Siwei, Qi Yingjian, Huang Yaping, Numerical solution of elliptic partial differential equation by growing radial basis function neural networks, in: *Proceedings of the International Joint Conference on Neural Networks, 2003*, pp. 85–90.
- [72] E.J. Kansa, H. Power, G.E. Fasshauer, L. Ling, A volumetric integral radial basis function method for time dependent partial differential equations I formulation, *Engineering Analysis with Boundary Elements* 28 (2004) 1191–1206.
- [73] H. Zou, J. Lei, C. Pan, Design of a new kind of RBF neural network based on differential reconstruction, in: *International Joint Conference on Neural Network and Brain*, vol. 1, 2005, pp. 456–460.
- [74] N. Mai-Duy, Solving high order ordinary differential equations with radial basis function networks, *International Journal for Numerical Methods in Engineering* 62 (2005) 824–852.
- [75] N. Mai-Duy, T. Tran-Cong, Solving biharmonic problems with scattered-point discretization using indirect radial basis function networks, *Engineering Analysis with Boundary Elements* 30 (2006) 77–87.
- [76] A. Golbabai, S. Seifollahi, Radial basis function networks in the numerical solution of linear integro-differential equations, *Applied Mathematics and Computation* 188 (2007) 427–432.
- [77] A. Golbabai, M. Mammadov, S. Seifollahi, Solving a system of nonlinear integral equations by an RBF network, *Computers and Mathematics with Applications* 57 (2009) 1651–1658.
- [78] N. Mai-Duy, T. Tran-Cong, A Cartesian-grid collocation method based on radial-basis-function networks for solving PDEs in irregular domains, *Numerical Methods for Partial Differential Equations* 23 (2007) 1192–1210.
- [79] A. Aminataei, M.M. Mazarei, Numerical solution of Poisson's equation using radial basis function networks on the polar coordinate, *Computers and Mathematics with Applications* 56 (2008) 2887–2895.
- [80] H. Chen, Li Kong, W. Leng, Numerical solution of PDEs via integrated radial basis function networks with adaptive training algorithm, *Applied Soft Computing* 11 (2011) 855–860.
- [81] S. Sarra, Integrated radial basis functions based differential quadrature method and its performance, *Computers & Mathematics with Applications* 43 (2002) 1283–1296.
- [82] B. Choi, J. Lee, Comparison of generalizing ability on solving differential equations using backpropagation and reformulated radial basis function network, *Neurocomputing* 73 (2009) 115–118.
- [83] K. Valasoulis, D.L. Fotadis, I.E. Lagaris, A. Likas, Solving differential equations with neural networks: implementation on a DSP platform, in: *14th International Conference on Digital Signal Processing*, vol. 2, 2002, pp. 1265–1268.
- [84] N. Vaziri, A. Hojabri, A. Erfani, M. Monsefi, B. Nilforooshan, Critical heat flux prediction by using radial basis function and multilayer perceptron neural networks: a comparison study, *Nuclear Engineering and Design* 237 (2007) 377–385.
- [85] S.S. Chiddarwar, N.R. Babu, Comparison of RBF and MLP neural networks to solve inverse kinematic problem for 6R serial robot by a fusion approach, *Engineering Applications of Artificial Intelligence* 23 (2010) 1083–1092.
- [86] A.S. Yilmaz, Z. Ozer, Pitch angle control in wind turbines above the rated wind speed by multi-layer perceptron and radial basis function neural networks, *Expert Systems with Applications* 36 (2009) 9767–9775.
- [87] M. Kashaninejad, A.A. Dehghani, M. Kashiri, Modeling of wheat soaking using two artificial neural networks (MLP and RBF), *Journal of Food Engineering* 91 (2009) 602–607.
- [88] B. Suchacz, M. Wesolowski, The recognition of similarities in trace elements content in medicinal plants using MLP and RBF neural networks, *Talanta* 69 (2006) 37–42.
- [89] J.S. Chandnok, I.N. Kar, S. Tuli, Estimation of furnace exit gas temperature (FEGT) using optimized radial basis and back-propagation neural networks, *Energy Conversion and Management* 49 (2008) 1989–1998.
- [90] A. Afkhami, M.A. Tarighat, M. Bahram, Artificial neural networks for determination of enantiomeric composition of α -phenylglycine using UV spectra of cyclodextrin host–guest complexes: comparison of feed-forward and radial basis function networks, *Talanta* 75 (2008) 91–98.