Parameter Estimation of Partial Differential Equations using Artificial Neural Network

Elnaz Jamili and Vivek Dua*

Department of Chemical Engineering, Centre for Process Systems Engineering, University College London, Torrington Place, London, WC1E 7JE, United Kingdom.

*Corresponding author: Vivek Dua

Abstract

The work presented in this paper aims at developing a novel meshless parameter estimation framework for a system of partial differential equations (PDEs) using artificial neural network (ANN) approximations. The PDE models to be treated consist of linear and nonlinear PDEs, with Dirichlet and Neumann boundary conditions, considering both regular and irregular boundaries. This paper focuses on testing the applicability of neural networks for estimating the process model parameters while simultaneously computing the model predictions of the state variables in the system of PDEs representing the process. The capability of the proposed methodology is demonstrated with five numerical problems, showing that the ANN-based approach is very efficient by providing accurate solutions in reasonable computing times.

Key Words: Parameter Estimation; Partial Differential Equation (PDE); Artificial Neural Network (ANN); Irregular Boundaries.

1 Introduction

A wide range of real-world systems in applied sciences and engineering fields belongs to Distributed Parameter Systems (DPS), where pertinent mathematical models often take the form of Partial Differential Equations (PDEs) describing the spatial-temporal dynamics of the system. Developing a reliable parameter estimation method for PDE systems is crucial to obtain accurate parameter values with fast convergence rates for system identification such that the model predictions could confirm the underlying dynamic behaviour of the process.

While previous contributions on the inverse problem of estimating unknown parameters have investigated extensively the parameter estimation properties such as accuracy and computing time; they discuss cases where methods mainly consider functions over a uniform grid discretisation; so, PDE models with irregular boundaries were largely ignored which consequently forms the main objective of this paper. Further advances in terms of estimation accuracy and savings in computation time are the other potential areas of improvements in this context. Several methods can be used for solving a system of partial differential equations, such as the method of weighted residuals (Finlayson and Scriven, 1966), finite difference methods (Smith, 1985; Mazumder, 2015), the numerical Method of Lines (MOL) (Schiesser, 1991), finite element methods (Bathe, 1996), Finite Volume Methods (FVM) (Mazumder, 2015), and artificial neural networks (Lagaris et al., 1998). Xu and Dubljevic (2017) recently developed a methodology based on the Model Predictive Control (MPC) algorithms for linear transport-reaction models. The authors proposed Cayley-Tustin transformation as an exact time discretisation scheme, and then developed a model predictive control formulation to account for the spatial nature of the problem. Irregular boundary conditions were not considered in these works. Applications where irregular boundary conditions are relevant include flow in heterogeneous porous media, neutron transport and biophysics (Berndt et al., 2006). Among the available solution strategies for simulation of PDE models, in this work, an artificial neural network (ANN) was used to solve the partial differential equations because of its excellent performance (Lagaris et al., 1998). ANN-based formulations represent an exciting avenue of research as they offer meshless frameworks to account for irregular boundaries. An ANN model involves parameters such as weight matrices and bias vectors that are adjusted to minimise a suitable error function. The computation of the network parameters in the ANN model forms part of the solution of the PDEs. So, the original parameter estimation problem for PDE systems becomes an optimisation problem in which the objective is to simultaneously approximate the PDE models by computing the ANN network parameters, and estimate the PDE model parameters such that the model predictions are in a good agreement with the measured data (experimental observations). Comprehensive experience in ODE parameter estimation (Dua, 2011; Dua and Dua, 2012) indicates that ANN-based methodology was effectively and successfully tested for ODE systems, and thus is a candidate for parameter estimation of PDEs.

Although a number of recent and related approaches for solving inverse problems have been previously studied, further development for PDEs defined on arbitrarily shaped domains is required. Such recent approaches include works by Bar-Sinai et al. (2019), Brunton et al. (2016) and Raissi et al. (2019). Bar-Sinai et al. (2019) aim to numerically solve PDEs, assisted by neural networks by using the data to train the neural networks and avoid discretising approximate coarse-grained models. Brunton et al. (2016) mainly focus on identifying the fewest terms in the dynamic model that can accurately represent the data. The work of Raissi et al. (2019) has some similarities to our work but differs in how the solution is hypothesised; they approximate a PDE equation by neural network whereas we approximate state variable with the neural network. Also, boundary conditions and irregular boundaries are incorporated in our work.

Classic works used the popular *finite difference method* (FDM) to provide an approximate solution to PDEs and employed the least squares method to estimate the physical properties in the heat conduction equation (Beck, 1970 a, b). The work carried out by Seinfeld and Chen (1971) had looked at the parameter estimation techniques based on the *method of steepest descent*, *quasilinearization*, and *collocation* in the class of PDE problems of chemical engineering interest. Polis et al. (1973) presented a methodology in which *Galerkin's method* had been used to convert the PDEs into a set of ODEs. The authors applied three optimisation schemes including a *steepest descent method*, a *search technique* and *nonlinear filtering*, for estimating the unknown parameters. The purpose of this was to show that the PDE parameter estimation problems could be transformed into a standard optimisation problem in which any optimisation algorithms can be applied. Some earlier reviews were given by Polis and Goodson (1976) and Kubrusly (1977). In the survey by Kubrusly (1977), identification methods for the DPS are classified into three classes: (i) direct method, (ii) reduction to Lumped Parameter Systems (LPS), and (iii) reduction to Algebraic Equations (AE). The direct method utilizes the infinite-dimensional system model to obtain the parameters. The reduction-based method, which is also known as time-space separation, involves spatial discretisation in order to reduce the PDEs into a set of ODEs in time to which estimation methods for LPS can be applied (Hidayat et al., 2017). A number of other related works exist in literature including statistical methods (Banks and Kunisch, 1989; Fitzpatrick, 1991; Xun et al., 2013), Laguerre-polynomial approach (Ranganathan et al., 1984), general orthogonal polynomials (Lee and Chang, 1986), Fourier series method (Mohan and Datta, 1989), singular value decomposition (Gay and Ray, 1995), artificial neural networks coupled with traditional numerical discretisation techniques (Gonzalez-Garcia et al., 1998), and extended multiple shooting method (eMSM) (Muller and Timmer, 2002).

In this work, the effectiveness of the proposed methodology is demonstrated through a collection of linear and nonlinear PDEs with different boundary conditions, such as Dirichlet, Neumann and Robin, considering both regular and irregular boundaries. This work is organised as follows: in Section 2, a general formulation of the proposed method is described followed by the numerical case studies which are presented in Section 3 in order to validate the applicability of the methodology, and Section 4 provides a summary of the paper.

2 Parameter Estimation Methodology

The proposed approach in this paper will be illustrated in terms of the partial differential equations under the following assumptions, (i) the PDE model structure of the system to be investigated is pre-selected and known, (ii) the system is identifiable, and (iii) the measured data (experimental observations) are available. Therefore, the main objective is to compute the unknown model parameters while simultaneously providing a solution to the system of PDEs.

Using the Least Squares (LS) objective function, the parameter estimation problem is formulated as follows:

$$\min_{\theta,\Psi(\mathrm{x})} \mathrm{Err}_{PE} = \sum_{p\in P}\left\{\widehat{\Psi}(\mathrm{x}^p) - \Psi(\mathrm{x}^p)\right\}^2 \tag{1}$$

subject to the PDE model taking the form of:

$$\mathcal{J}(\partial^s\Psi, \partial^{s-1}\Psi, \cdots, \partial\Psi, \Psi, \mathrm{x}) = \mathcal{F}_k(\Psi(\mathrm{x}), \theta, \mathrm{x}) \tag{2}$$

and associated boundary conditions, where $\mathcal{J}$ is a given function of the system of PDEs, and $\Psi := \left(\Psi_1(\mathrm{x}), \cdots, \Psi_k(\mathrm{x})\right) \in \mathbb{R}^{n_\Psi}$; $n_\Psi \in \mathbb{N}$, denotes the vector of k unknown functions of state variables in the given system of PDEs. It is assumed that the definition domain, x $:= (x_1, \cdots, x_m) \in \mathbb{R}^{n_\mathrm{x}}$; $n_\mathrm{x} \in \mathbb{N}$, and the right-hand side of the equations, $\mathcal{F}_k(\Psi(\mathrm{x}), \theta, \mathrm{x})$, have been given. If the time is included as one of the independent variables, it can be identified as the zeroth variable, $x_0 = t$. Note that the order of the differential equation is determined by s. $\widehat{\Psi}(\mathrm{x}^p)$ represents the experimental measurements of the state variables at data points x^p; $p \in P \subseteq \mathbb{N}$, and θ is the vector of model parameters to be estimated such that the error, Err_{PE}, between the measured data and the model predictions is minimised.

The methodology proposed in this work involves two main steps: first, approximating the solution by a trial solution, and second, incorporating the boundary conditions within the trial solution, as explained next.

Let $\Psi_k^{ANN}(\mathrm{x})$ denotes the trial solution. The ANN approximation of the model is formulated as follows, and incorporated into the parameter estimation problem:

$$\sum_{p \in P} \sum_{k \in K} \{\mathcal{J}(\partial^s \Psi_k^{ANN}, \partial^{s-1} \Psi_k^{ANN}, \cdots, \partial \Psi_k^{ANN}, \Psi_k^{ANN}, \mathrm{x}^p) - \mathcal{F}_k(\Psi(\mathrm{x}^p), \theta, \mathrm{x}^p)\}^2 \le \varepsilon \tag{3}$$

In the proposed approach, a trial form of the solution (or the neural network approximation of the solution), Ψ^{ANN}, is chosen (by construction) such that the initial/boundary conditions of the differential equation model are satisfied. The trial solution involves a sum of two terms:

$$\Psi^{ANN}(\mathrm{x}) = A(\mathrm{x}) + F\big(\mathrm{x}, N(\mathrm{x})\big) \tag{4}$$

where the first term, $A(\mathrm{x})$, is independent of adjustable parameters so as to satisfy the boundary conditions (BCs), while the term, F, is constructed to employ a feedforward neural network involving adjustable parameters such as weights and biases to deal with the minimisation problem. $N(\mathrm{x})$ represents a single-output feedforward neural network with network parameters and input datasets (Yadav et al., 2015; Lagaris et al., 1998). A systematic way to demonstrate the construction of the trial solution for treating different common case studies in various scientific fields is presented in the appendix.

Different numerical example problems which demonstrate the capabilities of the proposed approach will be presented in the next section. According to the numerical experiments, the ANN-based methodology based upon the formulation presented in this section has been proven to be very effective by providing accurate solutions in reasonable computing times. Moreover, the reported solution accuracy can be improved further by calibration of nodes within the ANN hidden layer in order to compute the optimal ANN topology.

Before proceeding with the numerical analysis, it is worth noticing that the generic mathematical formulation of the parameter estimation problem involves minimisation of the LS objective function, Equation (1), subject to the PDE model, Equation (2), and associated BCs, and the ANN model, Equations (3)-(4).

3 Numerical Case Studies

In this section, a number of case studies will be presented to demonstrate the advantages of the proposed modelling framework for the parameter estimation of partial differential equations. To computationally test and illustrate the performance of the proposed methodology for estimating unknown parameters in PDE models, the following example problems will be treated. The first problem seeks to estimate the diffusivity in the heat equation; the second one considers a linear Poisson equation with *Dirichlet* BCs while the third one studies the linear Poisson equation with mixed BCs; the fourth example problem examines a non-linear Poisson equation with mixed BCs; and the last one treats a highly non-linear problem with an irregular boundary. In all models with orthogonal box boundaries, the domain was taken to be $[0,1] \times [0,1]$ considering both uniform and non-uniform grid discretisation. A summary of the problems and the solutions obtained is given in Table 1.

All the optimisation problems were formulated as NLPs and solved using GAMS 24.7.1 (Rosenthal, 2008) on a Dell workstation with 3.00 GHz processor, 8GB RAM, and Windows

7 64-bit operating system. It should be noted that the main difficulty with the parameter estimation arises from the non-convexity of the non-linear objective function, as minimisation of such functions may result in different local optimal solutions. For this reason, the parameter estimation results may change for various NLP solvers and initial parameter guess values used for the solvers. Each solver can handle certain model types and one has to choose an appropriate solver that allows for optimal solutions to be computed in reasonable CPU times. To this end, the optimisation problems corresponding to the PDE models with orthogonal box boundaries were modelled in GAMS 24.7.1 and solved using SNOPT, while those corresponding to the PDE models with irregular boundaries were solved using KNITRO.

3.1 Problem 1

A numerical example is presented for the estimation of the diffusivity in the heat equation with *Dirichlet* BCs. The model is a linear PDE of parabolic type in one dimension of time and one space dimension.

3.1.1 Parameter Estimation using Uniform Grid

Consider the following partial differential equation with associated boundary and initial conditions, representing a mathematical model for a system governed by the heat equation (Seinfeld and Chen, 1971):

$$\theta \frac{\partial^2 \Psi}{\partial x^2} = \frac{\partial \Psi}{\partial t}$$

$$\Psi(0, x) = \sin \pi x \qquad 0 \leq x \leq 1$$

$$\Psi(1, x) = 0$$

$$\Psi(t, 0) = \Psi(t, 1) = 0 \qquad 0 \leq t \leq 1 \tag{17}$$

in which $\Psi = \Psi(t, x)$ denotes the state variable representing the temperature profile, x is the space coordinate, t is the time, and the model parameter $\theta \in \mathbb{R}^{n_\theta}$; $n_\theta \in \mathbb{N}$, stands for the thermal diffusivity which is unknown throughout the parameter estimation problem.

For this example problem, PSE's gPROMS® advanced process modelling platform was used for the generation of the simulated measurement data. The PDE model (Equation 17) was numerically solved by setting the actual value of the unknown parameter as $\theta = 1$. The model was implemented in gPROMS while the partial differential equation describing the heat transfer process was simulated using Orthogonal Collocation on Finite Elements (OCFE) scheme. To obtain a precise numerical solution, both time and space domains were to be handled using third order orthogonal collocation over ten finite elements.

Having simulated measurement data, the parameter estimation problem was formulated and solved in GAMS using ANN model. Note that to approximately solve the heat equation using an ANN, the trial form of the solution must be written as:

$$\Psi^{ANN}(t,x) = (1-t)\sin\pi\, x + t\,(1-t)\,x\,(1-x)\,N(t,x) \tag{18}$$

As discussed earlier in the previous, the trial solution is chosen such that the initial/boundary conditions of the PDE model are satisfied. Therefore, by incorporating the four boundary points given in Equation (17), into Equation (10), $\lambda_1 = \lambda_2 = 1$ is obtained, while $A(t,x) = (1-t)\sin\pi\, x$ is found by direct substitution in the general form given by Equation (11). Considering a uniform square discretisation of the domain $[0,1] \times [0,1]$, solving the parameter estimation problem gives $\mathrm{Err}_{PE} = 6.3643 \times 10^{-6}$ and $\theta = 0.98863$ as the parameter estimate.

3.1.2 Parameter Estimation using Non-Uniform Grid

To show the ability of the ANN-based simultaneous formulation for estimating unknown parameters, a non-uniform grid discretisation is now investigated in this section. A desirable feature of the ANN-based approach is that random points of each variable can be chosen over the domain resulting in a non-uniform grid. This could be useful in PDE models with irregular boundaries in which more sample points might be required in some regions of the domain.

The network architecture is now considered to be an ANN with two inputs $\mathrm{x}^p := (x^p, t^p)$, one hidden layer and twenty nodes in the hidden layer. For performing training, a total of 121 data points, $p := (1, 2, \cdots, 11)$, are obtained by considering nine random points of the domain $(0, 1)$ of each variable and four boundary points as: $x^1 = 0$, $x^{11} = 1$, $t^1 = 0$ and $t^{11} = 1$. Solving the parameter estimation problem for this case study gives $\mathrm{Err}_{PE} = 1.82757$ and $\theta = 0.99603$ as the parameter estimate. Computational times for the obtained results are approximately 40.5 seconds for the uniform grid and 226.8 seconds for the non-uniform grid.

3.2 Problem 2

Consider the following Poisson equation with Dirichlet BCs, which is a partial differential equation of elliptic type (Lagaris et al., 1998):

$$\nabla^2\Psi(x, y) = e^{-x}(x - \theta_1 + y^3 + \theta_2 y) \tag{19}$$

$$\Psi(0, y) = y^3$$

$$\Psi(1, y) = (1 + y^3)e^{-1}$$

$$\Psi(x, 0) = xe^{-x}$$

$$\Psi(x, 1) = e^{-x}(x + 1)$$

where the actual values of the parameters are $\theta = [\theta_1 \quad \theta_2] = [2 \quad 6]$, and $x, y \in [0, 1]$. The analytical solution for the above PDE model is as follows:

$$\Psi_{analytic}(x,y) = e^{-x}(x + y^3) \tag{20}$$

To illustrate the performance of the proposed methodology, the vector of parameters in Equation 19 is assumed to be unknown and must be estimated by formulating and solving the parameter estimation problem. The domain $[0,1] \times [0,1]$ was taken with a uniform grid discretisation considering a mesh of 36 points obtained by subdividing the interval in five equal subintervals corresponding to six equidistant points in each direction. Using Equation (10), the trial solution of the PDE model must be written as $\Psi_k^{ANN}(x,y) = A(x,y) + x(1-x)y(1-y)\ N\ (x,y)$. The term, $A(x,y)$, can be obtained by direct substitution in the general form given by Equation (11):

$$\begin{aligned} A(x,y) = (1-x)\ y^3 + x\ (1+y^3)\ e^{-1} + (1-y)\ x\ (e^{-x} - e^{-1}) \\ + y\ [(1+x)e^{-x} - (1 - x + 2\ x\ e^{-1})] \end{aligned} \tag{21}$$

Equation 21 incorporates the BCs given in Equation 19. Parameter estimation problem was modelled and solved in GAMS. Solving the parameter estimation problem for the uniform grid discretisation provides $\mathrm{Err}_{PE} = 2.7615 \times 10^{-6}$ and $\theta = [\theta_1 \quad \theta_2] = [2.03029 \quad 6.00006]$, and required only 8.6 seconds of computation time. The computational experiment was carried out for ten nodes in the hidden layer.

It is interesting to explore the advantage of ANN-based framework for estimating the model parameters over a non-uniform grid, when a small number of points is available for performing training. A non-uniform grid was generated by considering four random points of the domain $(0,1)$ of each variable and four boundary points as the following: $x^1 = 0$, $x^6 = 1$, $y^1 = 0$ and $y^6 = 1$. Using 7 nodes in the hidden layer, we obtained $\theta = [\theta_1 \quad \theta_2] = [2.00926 \quad 5.99466]$, an error of $\mathrm{Err}_{PE} = 1.919 \times 10^{-4}$ and it took approximately 22 seconds to converge.

3.3 Problem 3

Let us consider a PDE model representing a Linear Poisson Equation with mixed BCs as stated as follows (Lagaris et al., 1998):

$$\nabla^2\Psi(x,y) = (2-\theta^2\, y^2)\sin(\pi\, x) \quad (22)$$

$$\Psi(0,y) = 0$$

$$\Psi(1,y) = 0$$

$$\Psi(x,0) = 0$$

$$(\partial\Psi(x,1)\,/\,\partial y) = 2\sin(\pi\, x)$$

where the actual value of the parameter is $\theta = \pi$, and $x, y \in [0, 1]$. As before, a uniform grid discretisation is first studied; hence, training was performed using a mesh of 121 points obtained by considering eleven equidistant points of the domain $[0, 1]$ of each variable. For constructing the ANN topology, one hidden layer with ten hidden nodes were used for this case study.

The analytical solution of the given PDE model (Equation 22) is stated as follows:

$$\Psi_{analytic}(x,y) = y^2\sin(\pi\, x) \quad (23)$$

Using Equation (13), the trial solution of the PDE model must be written as $\Psi^{ANN}(x,y) = B(x,y) + x\,(1-x)\,y\left[N(x,y) - N(x,1) - \frac{\partial N(x,1)}{\partial y}\right]$. The term, $B(x,y)$, can be achieved by direct substitution in the general form given by Equation (15):

$$B(x,y) = 2\,y\sin(\pi\, x) \quad (24)$$

Solving the parameter estimation problem provides an error of $\mathrm{Err}_{PE} = 0.01657$ in about 258.8 seconds, and the computed parameter estimate is $\theta = 3.14123$. For a non-uniform grid of nine random points in $(0, 1)$, we obtained $\mathrm{Err}_{PE} = 0.01167$ and $\theta = 3.14325$ for ten nodes in the hidden layer and it took about 84 seconds for the convergence of the algorithm.

3.4 Problem 4

A nonlinear PDE problem (Lagaris et al., 1998) with the same mixed BCs as in Problem 3, is treated in this section. The analytical solution and the neural network approximation of the solution are the same with those of Problem 3. However, the mathematical model is given by:

$$\nabla^2\Psi(x, y) + \Psi(x, y)\frac{\partial}{\partial y}\Psi(x, y) = \sin(\pi\, x)\left(2 - {\theta_1}^2\, y^2 + \theta_2\, y^3 \sin(\pi\, x)\right) \tag{25}$$

where the actual values of the parameters are $\theta = [\theta_1 \quad \theta_2] = [\pi \quad 2]$, and $x, y \in [0, 1]$.

The network was first trained using a uniform grid of six equidistant points in $[0, 1]$. Parameter estimation problem was solved for twelve hidden nodes for a uniform grid to give $\mathrm{Err}_{PE} = 4 \times 10^{-5}$ and $\theta = [3.11367 \quad 1.97794]$. By considering seventeen nodes in the hidden layer for a non-uniform grid, $\mathrm{Err}_{PE} = 1 \times 10^{-10}$ and $\theta = [3.22134 \quad 1.97967]$ were obtained. Convergence was achieved in 492 and 68 CPU seconds for uniform and non-uniform grid, respectively.

3.5 Problem 5

Consider the following highly nonlinear problem (Lagaris et al., 2000) with a star-shaped domain as shown in Figure 2.

$$\nabla^2\Psi(x, y) + e^{\Psi(x,y)} = 1 + x^2 + y^2 + \frac{4}{(\theta + x^2 + y^2)^2} \tag{26}$$

where the actual value of the model parameter is $\theta = 1$ and $x, y \in [-1, 1]$.

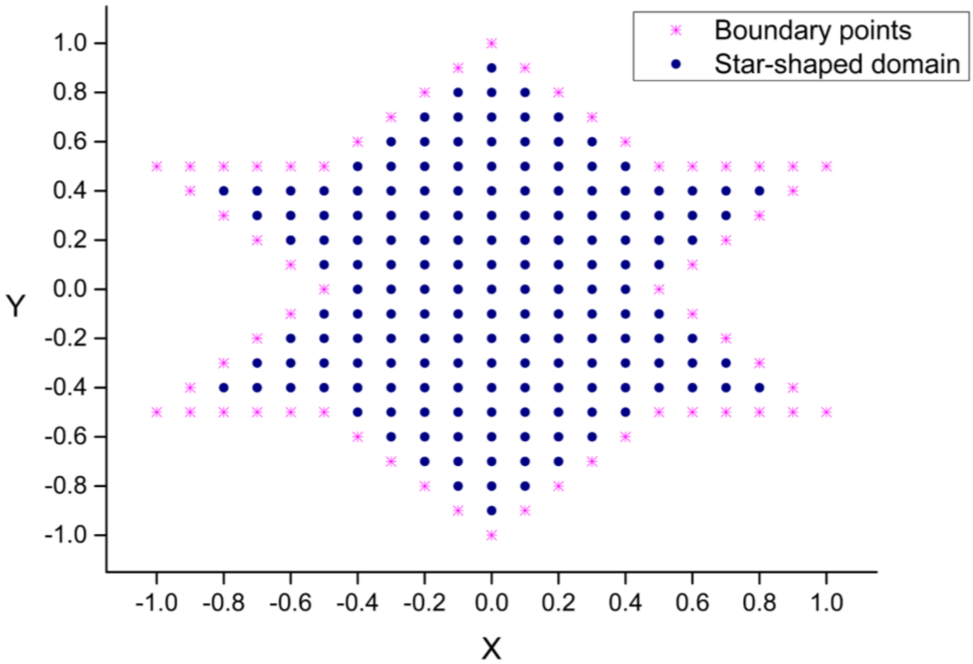

Figure 2: The star-shaped domain (171 points) and the boundary points (60 points) corresponding to Problem 5.

The star-shaped boundary has twelve vertices and sides. The boundary points (x, y) on the definition domain are considered by picking points on the interval $[-1, 1]$ on the x axis and y axis, respectively. The total number of points taken on the boundary is 60, and a total of 171 points were taken within the star-shaped domain. Using the analytical solution, $\Psi_{analytic}(x, y) = \log(1 + x^2 + y^2)$, the values of the state variable at the boundary points were computed and have been used in the training.

The unknown model parameter can be estimated while simultaneously computing the model predictions for the state variable. Solving the parameter estimation problem using an ANN with nineteen hidden nodes for the above PDE model yields $\mathrm{Err}_{PE} = 1.2749 \times 10^{-4}$ and $\theta = 1.57098$, and required 158.48 seconds of computation time. The proposed approach for parameter estimation works well for PDE models with arbitrarily complex boundaries. As indicated here, a close estimate of the parameter is made and the approximate solution is of high accuracy since there is a good match between the exact solution and the model predictions.

4 Concluding Remarks

A computationally efficient parameter estimation framework based on the artificial neural network (ANN) approximations was developed for PDE models and tested extensively on different example problems. To evaluate the performance of the suggested methodology, we experimented five numerical examples with a mesh-grid of small and moderate size, considering different distributions (uniform and non-uniform) with boundary conditions (*Dirichlet* and *Neumann*) defined on boundaries with simple and complex geometry. A summary of the results obtained from solving the parameter estimation problem using the ANN scheme is presented in Table 1. Based upon our experience, the proposed methodology worked better than conventional techniques.

Table 1: Example problems 1 – 5.

Problem	**Grid discretisation**	**Parameter**	**Actual value**	**Estimate**	**Error ($\mathbf{Err}_{PE}$)**	**CPU time (s)**
Problem 1	Uniform	θ	1	0.98863	6.3643×10^{-6}	40.5
	Non-uniform	θ	1	0.99603	1.82757	226.8
Problem 2	Uniform	θ_1	2	2.03029	2.7615×10^{-6}	8.6
		θ_2	6	6.00006		
	Non-uniform	θ_1	2	2.00926	1.919×10^{-4}	22
		θ_2	6	5.99466		
Problem 3	Uniform	θ	π	3.14123	0.01657	258.8
	Non-uniform	θ	π	3.14325	0.01167	84
	Uniform	θ_1	π	3.11367		492

		θ_2	2	1.97794	4×10^{-5}	
Problem 4	Non-uniform	θ_1	π	3.22134	1×10^{-10}	68
		θ_2	2	1.97967		
Problem 5	Non-uniform	θ	1	1.57098	1.2749×10^{-4}	158.48

Varying the ANN topology will have different computational demands such as the prediction accuracy and the central processing unit (CPU) times for estimating parameters. A trade-off between the solution accuracy and the computational time is required to land on an optimal configuration of the ANN model. The highest prediction accuracy with minimum computational time was achieved using a single hidden layer ANN model. The computational demands required to converge to the optimal solution are presented in Table 1. The illustrative examples provided in this paper demonstrate that the ANN-based approach is very efficient as it provides accurate solutions in reasonable computing times.

Declarations of interest: none

Appendix

Figure 1 aims to demonstrate the structure of an ANN with m inputs, a single hidden layer, h nodes in the hidden layer and one linear output. The output of the network, for a given input vector x $:= (x_1, \cdots, x_m)$, is given by:

$$N_k = \sum_{j=1}^{h} \nu_{jk}\sigma_j \tag{5}$$

where

$$\sigma_j = \frac{1}{1 + e^{-a_j}} \tag{6}$$

where

$$a_j = \sum_{i=1}^{m} \omega_{ij}x_i + b_j \tag{7}$$

ω_{ij} denotes the weight from the input $i = 1, \cdots, m$ to the hidden node $j = 1, \cdots, h$, ν_{jk} represents the weight from the hidden node j to the output, b_j is the bias of hidden unit j, and σ_j stands for the sigmoid transfer function. There are several possibilities of using transfer functions of different types, such as linear, sign, sigmoid and step functions (Yadav et al., 2015); here we consider the sigmoid transfer function (Lagaris et al., 1998).

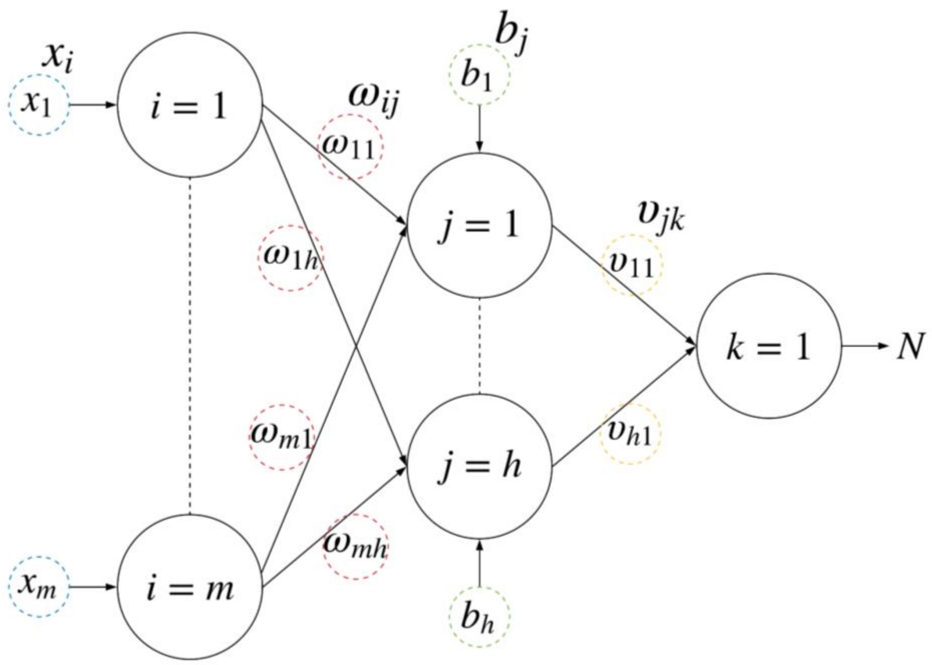

Figure 1: An Artificial Neural Network (ANN) with $\boldsymbol{m}$ inputs, one hidden layer, $\boldsymbol{h}$ nodes in the hidden layer and one linear output.

The l^{th} derivative of the output with respect to the i^{th} input, takes the form:

$$\frac{\partial^l N}{\partial x_i^l} = \sum_{j=1}^{h} \nu_{jk} \omega_{ij}^{l} \sigma_j^{(l)} \tag{8}$$

where $\sigma_j^{(l)}$ represents the l^{th} derivative of the sigmoid function.

After establishing the network structure and assuming the required conditions, the objective function is minimised. In this study, nonlinear programming (NLP) optimisation problems were implemented and solved in GAMS using SNOPT and KNITRO as solvers.

It must be noted that in the present work, two-dimensional second-order PDE problems will be treated; however, the methodology can be extended to more dimensions and derivative orders.

Please note that the following description, based on the work of Lagaris et al. (1998) is presented for the sake of completeness. Consider the following mathematical model of a PDE problem with *Dirichlet* boundary conditions (BCs), in which $s = 2$ and $\mathrm{x} := (x_1, x_2)$ where

$\mathrm{x} \in [\mathrm{x}^{LO}, \mathrm{x}^{UP}]$.

$$\mathcal{J}(\partial^2\Psi, \partial\Psi, \Psi, \mathrm{x}) = \mathcal{F}_k(\Psi(\mathrm{x}), \theta, \mathrm{x}) \tag{9}$$

$$\Psi(x_1^{LO}, x_2) = \mathcal{F}_k^0(x_2) \qquad k \in K$$

$$\Psi(x_1^{UP}, x_2) = \mathcal{F}_k^1(x_2) \qquad k \in K$$

$$\Psi(x_1, x_2^{LO}) = \mathcal{g}_k^0(x_1) \qquad k \in K$$

$$\Psi(x_1, x_2^{UP}) = \mathcal{g}_k^1(x_1) \qquad k \in K$$

The ANN network structure can be established for the above single PDE system, resulting in: $k = 1$, $l = 2$, and $m = 2$. The two input units of the network are assumed to be: $x_1 = x$ and $x_2 = y$. The form of the trial solution for the PDE model represented by Equation (9) is formulated as follows:

$$\Psi_k^{ANN}(x, y) = A(x, y) + x\ (\lambda_1 - x)\ y\ (\lambda_2 - y)\ N(x, y) \tag{10}$$

where an ANN model, $N(x, y)$, is considered for each trial solution $\Psi_k^{ANN}(x, y)$. The term $A(x, y)$ is then formulated as:

$$\begin{aligned} A(x, y) = {} & (1 - \zeta_1\ x)\mathcal{F}^0(y) + \zeta_2\ x\ \mathcal{F}^1(y) \\ & +(1 - \zeta_3\ y)\{\mathcal{g}^0(x) - [(1 - \zeta_1 x)\mathcal{g}^0(0) + \zeta_2\ x\ \mathcal{g}^0(1)]\} \\ & +\zeta_4\ y\ \{\mathcal{g}^1(x) - [(1 - \zeta_1 x)\mathcal{g}^1(0) + \zeta_2\ x\ \mathcal{g}^1(1)]\} \end{aligned} \tag{11}$$

Note that $\Psi^{ANN}(x, y)$, $A(x, y)$, λ_1, λ_2, ζ_1, ζ_2, ζ_3 and ζ_4 satisfy the *Dirichlet* BCs of the PDE model given by Equation (9). This therefore facilitates the numerical solution of the PDE model for given values of θ, which can be obtained by minimising the error quantity formulated as the following NLP problem (Lagaris et al., 1998):

$$\mathrm{Err}_{PDE} = \min_{\Psi^{ANN},N,\sigma,\omega,\nu,a,b} \sum_{p\in P} \sum_{k\in K} \{\mathcal{J}(\partial^s \Psi_k^{ANN}, \partial^{s-1}\Psi_k^{ANN}, \cdots, \partial\Psi_k^{ANN}, \Psi_k^{ANN}, \mathrm{x}^p) - \mathcal{F}_k(\Psi(\mathrm{x}^p), \theta, \mathrm{x}^p)\}^2 \quad (12)$$

If the PDE model given by Equation (9) is reformulated with mixed boundary conditions, the neural network approximation of the solution, where $x_1 = x$, $x_2 = y$, $x, y \in [0,1]$ and $k = 1$, is written as (Lagaris et al., 1998):

$$\Psi^{ANN}(x,y) = B(x,y) + x\,(1-x)\,y\left[N(x,y) - N(x,1) - \frac{\partial N(x,1)}{\partial y}\right] \quad (13)$$

Mixed BCs, which involve *Dirichlet* on part of the boundary and *Neumann* elsewhere, is of the form:

$$\Psi(0,y) = \mathcal{F}^0(y) \quad (14)$$

$$\Psi(1,y) = \mathcal{F}^1(y)$$

$$\Psi(x,0) = g^0(x)$$

$$(\partial\Psi(x,1) \,/\, \partial y) = g^1(x)$$

The term $B(x,y)$, of the trial solution (Equation (13)) is chosen to satisfy the mixed BCs (Lagaris et al., 1998):

$$\begin{aligned} B(x,y) = {} & (1-x)\mathcal{F}^0(y) + x\,\mathcal{F}^1(y) + g^0(x) \\ & -[(1-x)g^0(0) + x\,g^0(1)] \\ & +y\,\{g^1(x) - [(1-x)g^1(0) + x\,g^1(1)]\} \end{aligned} \quad (15)$$

The trial solutions presented above allow us to treat PDE models with orthogonal box boundaries. It however poses a challenge when the aim is to deal with realistic problems whose the boundaries are highly irregular. One of the key contributions of this paper is to develop a meshless methodology for parameter estimation, capable of dealing with any arbitrarily complex geometrical shape. This is achieved by choosing a trial solution in such a way so as to satisfy the differential equation. More specifically, the boundary conditions can be exactly satisfied by picking points on the boundary and hence the network is trained to satisfy the differential equation. The model suitable for this case can be written as:

$$\Psi_k^{ANN}(x,y) = N_k(x,y) \tag{16}$$

References

Banks, H. T. & Kunisch, K. (1989) Estimation techniques for distributed parameter systems. Boston: Birkhäuser Basel.

Bar-Sinai, Y., Hoyer, S., Hickey, J., & Brenner, P. (2019) Learning data-driven discretizations for partial differential equations. Proc. Natl. Acad. Sci. U.S.A. 116, 15344–15349.

Bathe, K. J. (1996) Finite element procedures. New Jersey: Prentice Hall.

Beck, J. V. (1970a) Nonlinear estimation applied to nonlinear inverse heat conduction problem. International Journal of Heat and Mass Transfer, 13(4), 703-716.

Beck, J. V. (1970b) Sensitivity coefficients utilized in nonlinear estimation with small parameters in a heat transfer problem. Journal of Basic Engineering, 92(2), 215-221.

Berndt, M., Manteuffel, T.A., McCormick, S.F., Starke, G. (2006) Analysis of First-Order System Least Squares (FOSLS) for Elliptic Problems with Discontinuous Coefficients: Part I. SIAM Journal on Numerical Analysis, 43(1), 386-408. http://www.jstor.org/stable/4101267.

Brunton, S.L., Proctor, J.L., & Kutz, J.N. (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Natl. Acad. Sci. U.S.A. 113, 3932–3937.

Dua, V. (2011) An Artificial Neural Network approximation based decomposition approach for parameter estimation of system of ordinary differential equations. Computers & Chemical Engineering, 35(3), 545-553.

Dua, V. & Dua, P. (2012) A simultaneous approach for parameter estimation of a system of ordinary differential equations, using artificial neural network approximation. Industrial & Engineering Chemistry Research, 51(4), 1809-1814.

Finlayson, B. A. & Scriven, L. E. (1966) The method of weighted residuals - a review. Applied Mechanics Reviews, 19(9), 735-748.

Fitzpatrick, B. G. (1991) Bayesian-analysis in inverse problems. Inverse Problems, 7(5), 675-702.

Gay, D. H. & Ray, W. H. (1995) Identification and control of distributed-parameter systems by means of the singular-value decomposition. Chemical Engineering Science, 50(10), 1519-1539.

Gonzalez-Garcia, R., Rico-Martinez, R. & Kevrekidis, I. G. (1998) Identification of distributed parameter systems: a neural net based approach. Computers & Chemical Engineering, 22, S965-S968.

Hidayat, Z., Babuska, R., Nunez, A. & De Schutter, B. (2017) Identification of distributed-parameter systems from sparse measurements. Applied Mathematical Modelling, 51, 605-625.

Kubrusly, C. S. (1977) Distributed parameter system identification - survey. International Journal of Control, 26(4), 509-535.

Lagaris, I. E., Likas, A. & Fotiadis, D. I. (1998) Artificial neural networks for solving ordinary and partial differential equations. IEEE Transactions on Neural Networks, 9(5), 987-1000.

Lagaris, I. E., Likas, A. C. & Papageorgiou, D. G. (2000) Neural-network methods for boundary value problems with irregular boundaries. IEEE Transactions on Neural Networks, 11(5), 1041-1049.

Lee, T. T. & Chang, Y. F. (1986) Analysis and identification of linear distributed systems via double general orthogonal polynomials. International Journal of Control, 44(2), 395-405.

Mazumder, S. (2015) Numerical methods for partial differential equations: finite difference and finite volume methods, 1st ed. Academic Press.

Mohan, B. M. & Datta, K. B. (1989) Identification via Fourier-series for a class of lumped and distributed parameter-systems. IEEE Transactions on Circuits and Systems, 36(11), 1454-1458.

Muller, T. G. & Timmer, J. (2002) Fitting parameters in partial differential equations from partially observed noisy data. Physica D-Nonlinear Phenomena, 171(1-2), 1-7.

Polis, M. P. & Goodson, R. E. (1976) Parameter identification in distributed systems - synthesizing overview. Proceedings of the IEEE, 64(1), 45-61.

Polis, M. P., Goodson, R. E. & Wozny, M. J. (1973) Parameter identification for distributed systems using Galerkins criterion. Automatica, 9(1), 53-64.

Raissi, M., Perikaris, P. & Karniadakis, G.E. (2019) Physics-informed neutral networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics 378, 686-707.

Ranganathan, V., Jha, A. N. & Rajamani, V. S. (1984) Identification of linear distributed systems via Laguerre-polynomials. International Journal of Systems Science, 15(10), 1101-1106.

Rosenthal, R. E. (2008) GAMS — A User’s Guide. Washington, DC: GAMS Development Corporation.

Schiesser, W. E. (1991) The numerical method of lines: integration of partial differential equations. San Diego: Academic Press, Inc.

Seinfeld, J. H. & Chen, W. H. (1971) Estimation of parameters in partial differential equations from noisy experimental data. Chemical Engineering Science, 26(6), 753-766.

Smith, G. D. (1985) Numerical solution of partial differential equations: finite difference methods, 3rd ed. Oxford: Clarendon Press.

Xu, Q. Q. & Dubljevic, S. (2017) Linear model predictive control for transport-reaction processes. AICHE Journal, 63(7), 2644-2659.

Xun, X. L., Cao, J. U., Mallick, B., Maity, A. & Carroll, R. J. (2013) Parameter estimation of partial differential equation models. Journal of the American Statistical Association, 108(503), 1009-1020.

Yadav, N., Yadav, A. & Kumar, M. (2015) An introduction to neural network methods for differential equations. Netherlands: Springer.