COMPARISON OF TWO NEURAL NETWORK APPROACHES TO MODELING PROCESSES IN A CHEMICAL REACTOR

by

Tatiana SHEMYAKINA, Dmitriy TARKHOV, Alexander VASILYEV* and Yulia VELICHKO

Peter the Great St. Petersburg Polytechnic University, Saint-Petersburg, Russia

Original scientific paper
https://doi.org/10.2298/TSCI19S2583S

In this paper, we conduct the comparative analysis of two neural network approaches to the problem of constructing approximate neural network solutions of non-linear differential equations. The first approach is connected with building a neural network with one hidden layer by minimization of an error functional with regeneration of test points. The second approach is based on a new continuous analog of the shooting method. In the first step of the second method, we apply our modification of the corrected Euler method, and in the second and subsequent steps, we apply our modification of the Störmer method. We have tested our methods on a boundary value problem for an ODE which describes the processes in the chemical reactor. These methods allowed us to obtain simple formulas for the approximate solution of the problem, but the problem is special because it is highly non-linear and also has ambiguous solutions and vanishing solutions if we change the parameter value.

Keywords*: non-isothermal chemical reactor, boundary value problem, ODE, multilayer solution, neural network modeling, global optimization, artificial neural network, artificial neural network adjustment*

Introduction

When you model real objects, you typically encounter two main types of fallacies. The first type of fallacies is an inaccurate description of the modeled object by differential equations and additional (initial, boundary, *etc*.) conditions. The second type of errors is an approximate numerical solution of the equations used. At the moment the main attention in the works on mathematical modeling focused on the second type. When we model complex processes in real objects, the errors of the first type are usually more important because they are more difficult to track and correct.

In this regard, we set ourselves the task of building and developing methods to create a range of approximate functional solutions of differential equations. These solutions should allow for the possibility of refinement according to monitoring data of the object. The complex of our methods for constructing approximate neural network solutions is described and tested on a variety of problems for ODE and PDE, [1-7]. In particular, methods of adjusting models to new data are presented.

* Corresponding author, e-mail: a.n.vasilyev@gmail.com

Our approach differs from others (see [8] and the literature cited there). Because it allows to: build parametric solutions, periodic regeneration of test points and some other useful features. However, our approach [1-7] has certain disadvantages. The main one is a long learning period. In this paper, we conduct a comparative analysis of this approach [1-7] and our new approach [9-11], which allows us to construct approximate neural network solutions of similar accuracy with considerably fewer neuro elements. We give this analysis on the example of the chemical reactor problem [5].

Formally, the problem is written in the form of a nonlinear differential equation with boundary conditions:

$$\frac{\mathrm{d}^2 y}{\mathrm{d}x^2} + \delta \exp(y) = 0, \quad \frac{\mathrm{d}y}{\mathrm{d}x}(0) = 0, \quad y(1) = 0 \tag{1}$$

This problem is solved by a standard method of lowering the order [12]. The peculiarity of the problem is the lack of an accurate solution for parameter values $\delta > \delta^* \approx 0.878458$.

The classical methods for solving such problems [12, 13] are not always easy to apply, and when we solve problems of this type, these methods lead us to a series of problems. So, in work [14] we have an example of non-uniqueness of the stationary combustion mode, as well as a jump-like transition from one mode to another during combustion of condensed systems. We also encounter problems when we apply the method of matched asymptotic expansions [15]. The application of asymptotic methods to problems of this type proved to be difficult, and the results in many cases were not satisfactory: for example, in the article [16] it was necessary to impose additional requirements of the approximation uniformity, in the absence of which the authors receive a contradiction to the condition of non-uniqueness.

Material and methods

We consider the Cauchy problem for a system of homogeneous differential equations:

$$\begin{cases} \mathbf{y}'(x) = \mathbf{f}\left[x, \mathbf{y}(x)\right] \\ \mathbf{y}(x_0) = \mathbf{y}_0 \end{cases} \tag{2}$$

on the interval $D = [x_0; x_0 + a]$, here we enter a vector $\mathbf{y} \subset \mathbb{R}^p$ and a mapping $\mathbf{f} : \mathbb{R}^{p+1} \to \mathbb{R}^p$. When we solve a problem of eq. (2) using classical methods, the segment *D* is split into *n* parts: $x_0 < x_1 < ... < x_k < x_{k+1} < ... < x_n = x_0 + a$, and we apply some iterative formula of the form:

$$\mathbf{y}_{k+1} = A(\mathbf{f}, \mathbf{y}_k, \mathbf{y}_{k+1}, h_k, x_k) \tag{3}$$

where $h_k = x_{k+1} - x_k$, $\mathbf{y}_k$ is an approximation to the exact value of the required solution $\mathbf{y}(z_k)$, *A* is a function that defines the method we use.

Our approach [9-11] is to use the eq. (3) to construct an approximate solution of the problem (1) on the interval $[x_0; x]$ with a variable upper limit $x \in [x_0; x_0 + a]$. In this case, steps and constructed approximate solutions become functions $h_k = h_k(x)$, $\mathbf{y}_k = \mathbf{y}_k(x)$, $\mathbf{y}_0(x) = \mathbf{y}_0$. In the simplest case of uniform splitting the interval, we obtain $h_k = (x - x_0)/n$, $x_k = x_0 + xk/n$. As an approximate solution of the original problem (1) we propose to use

$\mathbf{y}_n(x)$. If we use an explicit method, *i. e.*, in eq. (3) the function *A* does not depend on the approximation $\mathbf{y}_{k+1}$, then the recurrent eq. (3) allows us to calculate the approximate solution $\mathbf{y}_n(x)$ as an explicit function. If the function *A* depends on the approximation $\mathbf{y}_{k+1}$, the relation (3) can be considered as an equation concerning the approximation $\mathbf{y}_{k+1}$. This equation may admit an exact solution, then instead of the eq. (3) we obtain the relation of the form:

$$\mathbf{y}_{k+1} = B(\mathbf{f}, \mathbf{y}_k, h_k, x_k) \tag{4}$$

Equation (4), as before, allows us to calculate the approximation $\mathbf{y}_n(z)$ and use it as an approximate solution of problem (1). If it is not possible to solve the eq. (3) exactly concerning the expression $\mathbf{y}_{k+1}$, then we can use some approximate method (like Newton's method) or a specially trained neural network to obtain a formula of the form (4).

The most common complication of the problem (2) is the boundary value problem, which has the form:

$$\mathbf{y}'(x) = \mathbf{f}\left[x, \mathbf{y}(x)\right], \quad \tilde{\mathbf{y}}(x_0) = \tilde{\mathbf{y}}_0, \quad \bar{\mathbf{y}}(x_0 + \mathrm{a}) = \bar{\mathbf{y}}_0 \tag{5}$$

this type also includes the problem (1).

Here the vectors $\tilde{\mathbf{y}}, \bar{\mathbf{y}}$ are made up of the co-ordinates of the vector **y**, their total dimension is equal to the dimension of the vector **y**. The boundary value problem can be reduced to a problem with a parameter like:

$$\mathbf{y}'(z) = \mathbf{f}\left[x, \mathbf{y}(x)\right], \quad \tilde{\mathbf{y}}(x_0) = \tilde{\mathbf{y}}_0, \quad \breve{\mathbf{y}}(x_0) = \boldsymbol{\mu} \tag{6}$$

The vector $\breve{\mathbf{y}}$ contains the co-ordinates of the vector **y**, which are not included in the vector $\tilde{\mathbf{y}}$. Equations (3) and (4) allow building the multilayered solution of a problem (6) $\mathbf{y}_n(x, \boldsymbol{\mu})$: from the conditions at the right end we can get the equation $\hat{\mathbf{y}}_n(x_0 + \mathrm{a}, \boldsymbol{\mu}) = \hat{\mathbf{y}}_0$. Solving this equation, we find the parameter **μ**. Our approach can be viewed as a functional variant of the shooting method. We use it further in solving the problem (1).

An essential feature of the problem (1) is that it has a parameter, δ. This feature of the problem does not make our approach much more difficult. In this situation, the parameter will be included in recurrent eqs. (3) and (4), and we will get an approximate solution of the form $\mathbf{y}_n(x, \delta, \boldsymbol{\mu})$. The condition on the right end will be written in the form:

$$\hat{\mathbf{y}}_n(x_0 + \mathrm{a}, \delta, \boldsymbol{\mu}) = \hat{\mathbf{y}}_0 \tag{7}$$

Thus, the problem boils down to find from eq. (7) the dependence of the form $\boldsymbol{\mu}(\delta)$. Next, we present the results of the method with the steps of the same length. In the first step, we apply the corrected Euler method [13] $y(x, \delta, p, 1) = p - \delta x^2 \exp(p)/8$, we have designated the only co-ordinate of the vector **μ** through *p*. In the following steps, we use the Störmer method:

$$y(x, \delta, p, k+1) = 2y(x, \delta, p, k) - y(x, \delta, p, k-1) - \frac{\delta x^2}{4n} \mathrm{e}^{y(x, \delta, p, k)}, \quad k = 1, \ldots n, \quad y(x, \delta, p, 0) = p$$

As a result, we obtain the function $y_n(x, \delta, p) = y(x, \delta, p, n)$ as an approximate solution to the problem. The parameter *p* we find from the condition on the right end:

$$y_n(1, \delta, p) = 0 \tag{8}$$

We are looking for an approximate solution of this equation in the form of neural network decomposition:

$$p(\delta) = \sum_{i=1}^{N} c_i v(\delta, \mathbf{a}_i) \tag{9}$$

As the base neuro elements, we use the function such as:

$$v(\delta, a_i, b_i) = th\left(a_i \delta + b_i\right), \quad i = 1, ..., n .$$

The weights of the neural network are linearly incoming parameters c_i and non--linearly incoming parameters a_i, b_i. They are determined in the process of learning the network based on minimization of an error functional. This process involves a residual in satisfying the condition (8):

$$J = \sum_{j=1}^{M} \left\{ y_n \left[1, \delta_j, \sum_{i=1}^{N} c_i v_i(\delta_j, \mathbf{a}_i) \right] \right\}^2 \tag{10}$$

For $n = 2$ we obtain an approximate solution of the form:

$$y_2(x, \delta, p) = p - \frac{\delta x^2}{4} \left[\mathrm{e}^p + \exp\left(p - \frac{\delta x^2 \, \mathrm{e}^p}{8} \right) \right]$$

For $n = 3$ we get a more complex formula:

$$y_3(x, \delta, p) = p - \mathrm{e}^{p - \mathrm{e}^p x^2 \delta / 18} \left[\frac{2}{9} + \frac{1}{6} \mathrm{e}^{\mathrm{e}^p x^2 \delta / 18} + \frac{1}{9} \mathrm{e}^{-\mathrm{e}^p \left(1 + 2\mathrm{e}^{-\mathrm{e}^p x^2 \delta / 18} \right) x^2 \delta} \right] x^2 \delta$$

As the number of layers increases, we obtain similar but more complex formulas.

Results and discussion

We compare methods of this work with methods of the article [5] for 100 neurons without additional data. At the same time, the neural network is trained on the interval $\delta \in [0.1; 1]$. We have given results for the values of the parameter, δ = 0.2 and δ = 0.8, figs. 1-8.

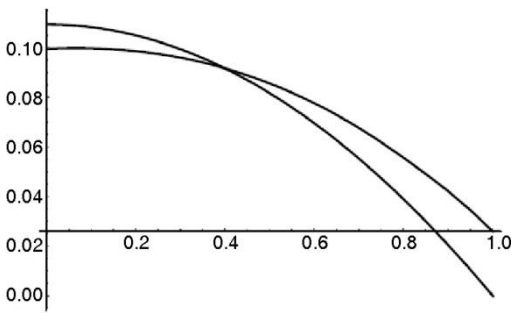

Figure 1. Graph of approximate solution, which is constructed by applying the methods of the article [5] for 100 neurons without additional data, and the exact solution of the problem for the values of the parameter δ = 0.2

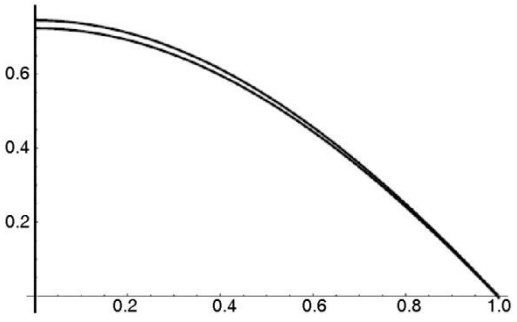

Figure 2. Graph of approximate solution, which is constructed by applying the methods of the article [5] for 100 neurons without additional data, and the exact solution of the problem for the values of the parameter δ = 0.8.

The method which is based on the application of eqs. (3), eq. (8) has given results: For $n = 2$ and neural network (9) composed of 5 neurons.

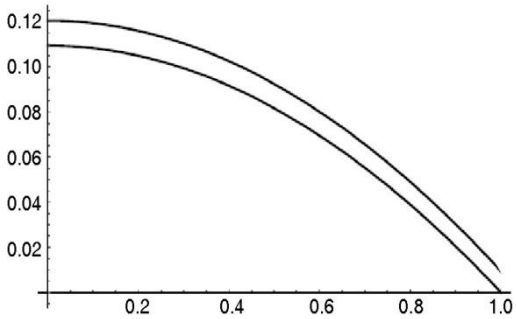

Figure 3. Graph of approximate solution $y_2(x,\delta,p)$ and the exact solution of the problem for the values of the parameter $\delta = 0.2$

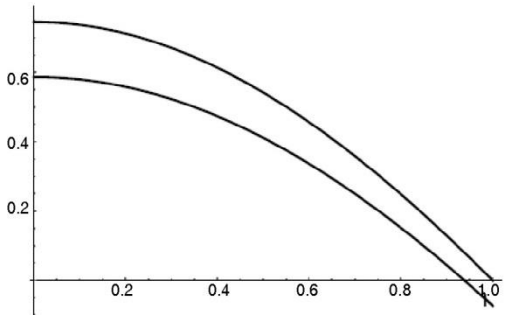

Figure 4. Graph of approximate solution $y_2(x,\delta,p)$ and the exact solution of the problem for the values of the parameter $\delta = 0.8$

For $n = 3$ and neural network (9) composed of 15 neurons.

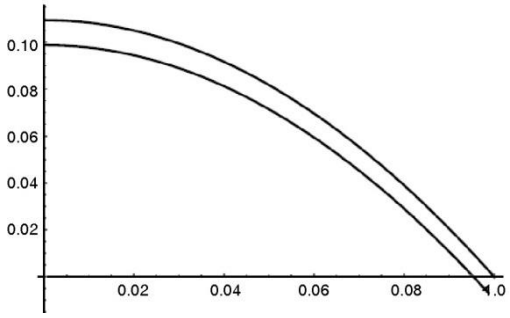

Figure 5. Graph of approximate solution $y_3(x,\delta,p)$ and the exact solution of the problem for the values of the parameter $\delta = 0.2$

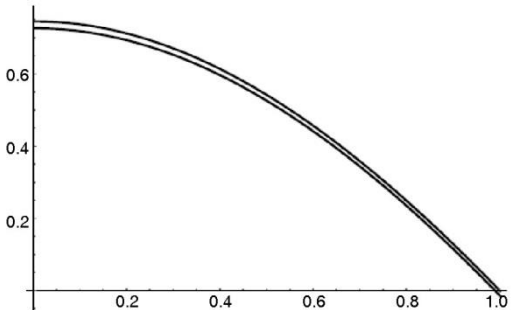

Figure 6. Graph of approximate solution $y_3(x,\delta,p)$ and the exact solution of the problem for the values of the parameter $\delta = 0.8$

For $n = 4$ and neural network (9) composed of 30 neurons:

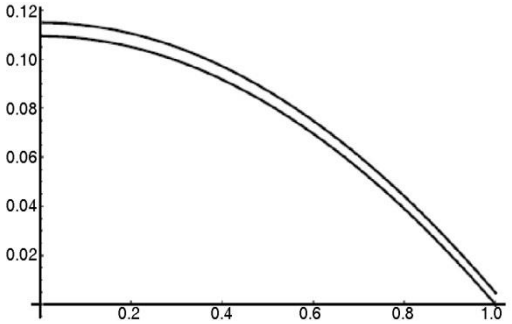

Figure 7. Graph of approximate solution $y_4(x,\delta,p)$ and the exact solution of the problem for the values of the parameter $\delta = 0.2$

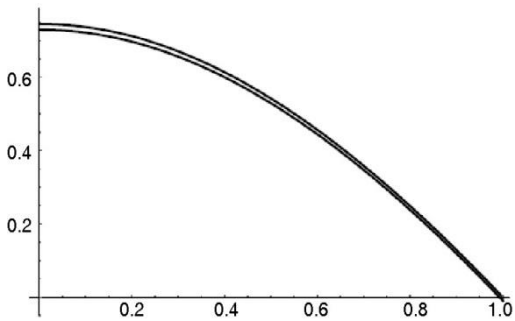

Figure 8. Graph of approximate solution $y_4(x,\delta,p)$ and the exact solution of the problem for the values of the parameter $\delta = 0.2$

Conclusion

The methods that we have considered in this paper allowed us to obtain simple formulas for the approximate solution of problem (1). These formulas are considerably simpler than those which were obtained by the methods of [5]. The accuracy of the method, which is based on the corrected Euler method and the Störmer method for $n = 2$, is acceptable for the parameter values, δ, and far from critical. When we approach the critical value of the parameter, the quality of the solution slightly deteriorates. The accuracy of the method at $n = 3$ is significantly higher near its critical parameter values. An even more accurate solution can be obtained if we increase the number of layers n and the number of neurons in the network (9). But at the same time, the complexity of the formulas will rapidly increase and their specification according to experimental data becomes complicated. Considered a model problem supports the conclusion that the layered methods provide an opportunity to build more simple models without losing accuracy, which simplifies their use and the parameters are the input variables.

Acknowledgment

This paper is based on research carried out with the financial support of the grant of the Russian Scientific Foundation (project №18-19-00474).

References

[1] Tarkhov, D. A., Vasilyev A. N., Mathematical Models of Complex Systems on the Basis of Artificial Neural Networks, *Nonlinear Phenomena in Complex Systems, 17* (2014), 3, pp. 327-335
[2] Kainov, N. U., *et al*., Application of Neural Network Modeling to Identification and Prediction in Ecology Data Analysis for Metallurgy and Welding Industry, *Nonlinear Phenomena in Complex Systems*, *17* (2014), 1, pp. 57-63
[3] Budkina, E. M., *et al.*, Neural Network Technique in Boundary Value Problems for Ordinary Differential Equations, *Lecture Notes in Computer Science*, *9719* (2016), July, pp. 277-283
[4] Gorbachenko, V. I., *et al.*, Neural Network Technique in Some Inverse Problems of Mathematical Physics, *Lecture Notes in Computer Science*, *9719* (2016), July, pp. 310-316
[5] Shemyakina, T. A., *et al.*, Neural Network Technique for Processes Modeling in Porous Catalyst and Chemical Reactor, *Lecture Notes in Computer Science*, *9719* (2016), July, pp. 547-554
[6] Lazovskaya, T. V., *et al.*, Parametric Neural Network Modeling in Engineering, *Recent Patents on Engineering*, *11* (2017), 1, pp. 10-15
[7] Lozhkin, V., *et al.*, Differential Neural Network Approach in Information Process for Prediction of Roadside Air Pollution by Peat Fire, *Proceedings*, IOP Conf. Series: Materials Science and Engineering, Kazan, Russia, 2016, Vol. 158
[8] Neha, Y., *et al*., An Introduction to Neural Network Methods for Differential Equations, Springer, New York, London, 2015, p. 115
[9] Lazovskaya, T. V., Tarkhov, D. A., Multilayer Neural Network Models Based on Grid Methods, *Proceedings*, IOP Conf. Series: Materials Science and Engineering, Kazan, Russia, 2016, Vol. 158
[10] Vasilyev, A. N., *et al.*, Approximate Analytical Solutions of Ordinary Differential Equations (in Russian), *Proceedings*, Selected Papers of the XI International Scientific-Practical Conference Modern Information Technologies and IT-Education (SITITO 2016), Moscow, Russia, 2016, pp. 393-400
[11] Lazovskaya, T. V., *et al*., Multi-Layer Solution of Heat Equation, Advances, in: *Neural Computation, Machine Learning and Cognitive Research*, (Eds. Kryzhanovski, B., *et al*.), Springer, Moskow, Russia, 2017, Oct., pp. 18-19
[12] Hairer, E., *et al*., *Solving Ordinary Differential Equations I: Nonstiff Problem*, Springer-Verlag, Berlin, 1987, p. 480

[13] Na, T. Y. *Computational Methods in Engineering Boundary Value*, Academic Press, Dearborn, Mich., USA, 1979, p. 296
[14] Hall, A. R., Wolfhard, H. G., Multiple Reaction Zones in low Pressure Flames with Ethyl and Methyl Nitrate, Methyl Nitrate and Nitromethane, *Proceedings*, VI Symp. (Intern.) on Combustion, New York, USA, 1957, pp. 190-199
[15] Bush, W. B., Fendell, F. E., Asymptotic Analysis of Laminar Flame Propagation for General Lewis Numbers, *Comb Sci. Techn., 1* (1970), 6, pp. 421-428
[16] Fendell, F. E., Asymptotic Analysis of Premixed Burning with Large Activation Energy, *J. Fluid Mech., 56* (1972), 1, pp. 81-95

Paper submitted: September 10, 2018
Paper revised: November 11, 2018
Paper accepted: November 30, 2018