Multilayer neural network models based on grid methods

T Lazovskaya and D Tarkhov
Peter the Great St-Petersburg Polytechnic University, 29 Politechnicheskaya Str, 195251 Saint-Petersburg, Russia

E-mail: tatianala@list.ru

Abstract. The article discusses building hybrid models relating classical numerical methods for solving ordinary and partial differential equations and the universal neural network approach being developed by D Tarkhov and A Vasilyev. The different ways of constructing multilayer neural network structures based on grid methods are considered. The technique of building a continuous approximation using one simple modification of classical schemes is presented. Introduction non-linear relationships into the classic models with and without posterior learning are investigated. The numerical experiments are conducted.

1. Introduction
Neural network approach to solving differential equations which was proposed by A.N. Vasilyev and D.A. Tarkhov was quite versatile. It could be applied without strong changes for the most extensive range of tasks [1-8]. Classical approaches to solving many of these problems require special artificial techniques. Neural network models were resistant to errors in the given data, they can be built based on the heterogeneous data which may include the results of experiments and have a number of other advantages. At the same time, the neural network approach is not free of a number of disadvantages in comparison to the classical methods of the grids, the finite elements and etc.

First of all, the neural network learning is still very resource consuming procedure. In the second place, the required size of the neural network and its learning time increase sharply with accuracy requirements. This problem is actual when the boundary value problem displays the simulated real object properties with high precision. Third, the neural network approach has not been implemented as a standard software package and it is not integrated at any type of standard package, for example, ANSYS.

In this context, the topical problem is the coupling of the neural networks and classical methods. We assume that obtained hybrid algorithms will be saved the disadvantages of each separate approach while keeping their advantages. We tried three different ways to such coupling.

The first way is that the results of classical methods are used as the additional data at training of the neural network [1,2,5].

The second way consists in the implicit grid methods transforming to the explicit methods using pre-trained neural networks. This approach has been tested for ordinary differential equations in [2].

A third way is to construct multilayer neural network structures based on the grid methods. In the above-cited works on neural network approach, we used neural networks with one hidden layer. For scaling to multilayer networks, it is necessary to determine the network structure (number of layers and the number of elements in each layer), and the initial values of the network parameters (network
weights). This can be done by means evolutionary algorithms [1,6,7], and based on a grid pattern, passing from linear relations to similar nonlinear relations which are typical of neural networks. The result is an analogue of deep learning [2].

In this article, we focus on the second and third approaches due to their principal novelty.

2. Continuous approximations based on classical methods

Before turning from the classical methods to trainable neural networks, we consider one technique of constructing a continuous approximation of the solution of the problem based on the classical Euler method.

We solve the boundary problem

\[
\begin{align*}
y'(t) &= f(t,y(t)), \quad t \in [t_0,T]; \\
y(t_0) &= y_0.
\end{align*}
\]

We construct a solution using the explicit Euler method on the interval \([t_0, t] \in [t_0,T]\) with a variable right endpoint \(x \in [t_0,T]\). The step size is defined by the next expression

\[
h(t) = \frac{x - t_0}{M},
\]

where \(M\) is called the layer number. Then, the approximate solution \(\hat{y}(x)\) is built iteratively and coincides with \(y_M(x)\) in the scheme

\[
\begin{align*}
t_i(x) &= t_0 + h(x)i, \\
y_{i+1}(x) &= y_i(x) + h(x)f(t_i(x), y_i(x)), \quad i = 1, \ldots, M.
\end{align*}
\]

Similarly, we can modify and other classical schemes – the predictor-corrector, the trapezoidal rule, Runge-Kutta, Adams methods and etc.

The result is a continuous function approximating the required solution on all of the interval \([t_0,T]\). The accuracy of the solution, obviously, depends on the layer number \(M\) of considered model. The convergence and the accuracy can be estimated by the error (residual) functional.

Such Euler and Runge-Kutta method modifications were applied to solve the problem with a parameter

\[
\begin{align*}
y'(0) &= \alpha(\cos t - y(t)); \\
y(0) &= 0,
\end{align*}
\]

where \(\alpha \in [0.1, 50]\) and \(t \in [0,1]\).

For large values of the parameter \(\alpha\), we have the classic stiff problem [3, 4] for which explicit methods are numerically unstable.

The solution quality was defined as the mean square residual \(\Delta_{1000}\) at 1000 pairs of random points \((x, \alpha)\) uniformly distributed in the area \([0,1] \times [0.1,50]\). When choosing the number of layers \(M = 30\) a continuous solution based on Euler method was obtained with an error \(\Delta_{1000} = 0.004\), the relative mean square error, in this case, was 0.5 percent.

Note that for \(\alpha = 50\) while the classical scheme gives a pointwise solution oscillating with a large error in the neighborhood of the point \(x = 0\), the errors of continuous approximation of the required function based on the modified methods are rather small. Figure 1 shows errors of all methods mentioned.

Decrease the number of layers to 25 significantly degrades the accuracy of solutions. This is readily illustrated by Figure 2.

Note that it is possible to obtain uniform estimates for the continuous approximate solutions on the interval based on the known estimates for the initial methods residuals at the finite point of the interval.
3. Introduction nonlinear relationships into the classic models

If in the previous section as a result of modification of the classical method of Euler, we obtained an approximate solution in the form of a function not associated with neural networks, then below we build in a similar manner an approximate solution of the problem (1) most closely to possible neural network.

The idea is to replace the increment linear expression with the nonlinear relationship which behaves almost linearly for small values of the argument. We can get an initial approximation in the form of the neural network choosing the functions which are typical for a neural network basis as such nonlinear dependencies.

If it is necessary to find approximate solutions more accurate (in the terms of the residual or error functional) the constructed neural network can be trained by fitting the weight during the minimization of the corresponding error functional. Below we present particular examples.

4. Nonlinear Euler method

Solving the problem (1) we transform the already modified scheme (2) of the Euler method to the next form

\[
\begin{aligned}
    t_i(x) &= t_0 + h(x)i, \\
    y_{i+1}(x) &= y_i(x) + \theta[h(x)f(t_i(x), y_i(x))], \\
    i &= 1, \ldots, M.
\end{aligned}
\]  

(4)
The number $M$ still determines the layer number of the model. As will be seen, a sufficiently large value of $M$ provides constructing the model of a good accuracy without recourse to further learning and fitting of parameters.

We continue to solve the problem (3) using the scheme (4) and build a multi-layer perceptron network. Even when the number of layers $M$ is equal to 25 we get a quite good approximation (Figure 3).

![Figure 3](image)

**Figure 3.** Error Graphs for the continuous approximation of the solution of problem (3) with $\alpha = 50$ (a) and $\alpha = 0.1$ (b) obtained by using the scheme (4) in the case of the layer number $M = 25$.

The comparison with Figure 2 shows a significantly higher accuracy of the solution obtained based on the scheme (4). With the layer number increasing the calculation time cost and the complexity of the constructed solution increase too.

5. Trainable neural networks based on the Euler method

Here, we build a multi-layer neural network by using the previous section methods and choose a small number of layers. So, we are able to train obtained network at the lowest cost. Network parameters are introduced by the addition of linear factors in the argument of the sigmoid function ($\tanh(\cdot)$) included in the solution constructed.

The three-layer parameterized neural network corresponded to the problem (3) and constructed by the scheme (4) has the following form

$$y(x, \alpha; b, c, d) = \tanh(bx\alpha) + \tanh[cx\alpha\cos\frac{x}{3} - \tanh(bx\alpha)] + \tanh[dx\alpha\cos\frac{2x}{3} - \tanh(bx\alpha)] - \tanh[cx\alpha\cos\frac{x}{3} - \tanh(bx\alpha)]].$$

Further, the network weights $b, c$ and $d$ are fitted during the minimization of the corresponding error functional. We have used the functional in the discrete form

$$\sum_{j=1}^{m} (y_j + \alpha_j(y_j - \cos x_j))^2,$$

and test points $(x_j, \alpha_j) \in [0; 1] \times [\alpha_{\min}; \alpha_{\max}]$ are regenerated after every 3-5 training steps. We have used the normalized functional too in the form

$$\sum_{j=1}^{m} (y_j + \alpha_j(y_j - \cos x_j))^2 / \alpha_j^2.$$

The parameter values $b = c = d = 1/3$ from the scheme (4) were used as a quite good initial approximations. Optimization was carried out using a combination of the RProp and the Particle swarm algorithms.

Neural network learning allowed to get an acceptable accuracy in the case the significantly smaller number of layers (Figure 4). Note that the methods [3] give similar accuracy for the single-layer network with 20 neurons.
6. Nonlinear modification of the implicit Euler method vs the classical methods

We solve the problem (3) for various fixed values of the parameter $\alpha$ using different approaches: classic explicit and implicit (8) Euler methods, trapezoidal rule (9) and the method modified by introducing a sigmoid nonlinearity (10). We remind that depending on the alpha the problem cannot be solved by explicit classical methods. For considered problem, implicit methods equations are solved simply. Let $h=1/M$ is a step size in terms of classical methods ($M$ is the layer number). Corresponding schemes will be of the next forms

$$y_{i+1} = \frac{1}{1+\alpha h} y_i + \frac{\alpha h}{1+\alpha h} \cos ih; \quad (8)$$

$$y_{i+1} = \frac{1-0.5\alpha h}{1+0.5\alpha h} y_i + \frac{0.5\alpha h}{1+0.5\alpha h} (\cos ih + \cos(i+1)h); \quad (9)$$

$$y_{i+1} = \frac{1 - \text{th}(\alpha h)}{1 + \text{th}(\alpha h)} y_i + \frac{\text{th}(\alpha h)}{1 + \text{th}(\alpha h)} \cos(i+1)h, \quad (10)$$

where $y_0 = 0$, $i = 0, \ldots, M$.

In the case of a non-stiff problem statement, all methods work equally well, the function $\text{th}(x)$ is almost linear for small values of the argument.

By choosing the number of layers $M = 30$ we consider the problem (3) in a fairly stiff statement at $\alpha = 100$. Figure 5 shows that while the explicit method diverges schemes (8)-(10) give error increase in the neighborhood of the point $x = 0$ but sigmoidal scheme accuracy is an order of magnitude more.

With further increase of the parameter $\alpha$ sigmoidal modification (10) shows the best result.

Figure 5. Error graphs of the problem (3) solution obtained based on schemes (8) - (10) in the case of $\alpha = 100$ and the layer number $M = 30$.

7. Introduction of neural network parameters to original functional object

We have already discussed above construction of continuous approximations based on classical methods. We showed how you can select the network parameters for further learning. At the same
time, the interior point positions were prefixed on the assumption of the uniform decomposition of an interval. Another way to organize the algorithm is to make steps variables.

The use of explicit Euler method with one intermediate point to the problem (3) gives the following formula

\[
y(x, \alpha; b) = abx \cos(bx) + \alpha(1 - b)x(\cos x - abx \cos(bx)).
\]

Figure 6 shows the acceptable result obtained by the use of formula (11).

8. Extension of new approaches to partial differential equations

It is possible to offer different ways to extend the proposed approaches to partial differential equations. The simplest of these is the method of lines [4].

We present a simple example of this scheme for the heat equation

\[
\frac{\partial y}{\partial t} = \frac{\partial^2 y}{\partial x^2}, \quad (t, x) \in [0,1] \times [0,1],
\]

with boundary conditions \( y(0, x) = 1 - x^2, \ y'(t, 0) = 0 \) and \( y(t, 1) = 0 \). The explicit scheme of the methods is considered in the form

\[
G(i, j) = G(i, j - 1) + R\left[ h\left( G(i-1, j-1) - G(i, j-1) \right) \right] + R\left( h\left( G(i+1, j-1) - G(i, j-1) \right) \right) + R\left( h\left( G(i, j+1) - G(i, j) \right) \right).
\]

If \( R(x) = x \) we obtain a simple explicit method. If we choose sigmoid function \( R(x) = \frac{x}{1 + |x|} \) the scheme (13) can be considered as the neural network model. Figure 7 is illustrated the results in the case when we divide the segment by 10 parts with respect to \( x \) and by 180 parts – to \( t \).

Figure 7. Error graphs at \( x = 0 \) of the approximate solution of the heat equation problem obtained by regular explicit scheme (a), the scheme (13) (b) and the scheme (13) with averaged solution at the current and the previous points (c).
9. Conclusions
In this paper, we made the first attempts to bridge the gap between classical numerical methods for solving differential equations and neural network approach. We have considered the possibility of constructing continuous approximations based only on classical numerical schemes, the introduction of non-linear relationships both on searching pointwise solutions and in constructing the continuous approximations. We have shown the possibility of learning of the obtained models through the introduction of the corresponding parameters. The calculations based on considered algorithms can be speed up by several orders of magnitude in the case of using specialized neuron chips.

This area is promising for the further investigation and theoretical study results.

Acknowledgments
The work is supported by the Russian Foundation for Basic Research (Grant no. 14-01-00660).

References
[1] Tarkhov D and Vasilyev A 2014 Mathematical Models of Complex Systems on the Basis of Artificial Neural Networks Nonlinear Phenomena in Complex Systems 17 pp 327-335
[2] LeCun Y, Bengio Y and Hinton G 2015 Deep learning Nature 521 553 pp 436-444
[3] Lazovskaya T and Tarkhov D 2015 Fresh approaches to the construction of parameterized neural network solutions of a stiff differential equation St. Petersburg Polytechnical University Journal: Physics and Mathematics http://dx.doi.org/10.1016/j.spjpm.2015.07.005
[4] Hairer E and Wanner G 2010 Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems (Springer) 614 p
[5] Budkina E, Kuznetsov E, Lazovskaya T, Leonov S, Tarkhov D and Vasilyev A 2016 Neural Network Technique in Boundary Value Problems for Ordinary Differential Equations ISNN 2016 LNCS 9719 pp 277-283
[6] Gorbachenko V, Lazovskaya T, Tarkhov D, Vasilyev A and Zhukov M 2016 Neural Network Technique in Some Inverse Problems of Mathematical Physics ISNN 2016 LNCS 9719 pp 320-316
[7] Shemyakina T, Tarkhov D and Vasilyev A 2016 Neural Network Technique for Processes Modeling in Porous Catalyst and Chemical Reactor ISNN 2016 LNCS 9719 pp 547-554
[8] Tarkhov D and Vasilyev A 2005 New neural network technique to the numerical solution of mathematical physics problems. I: Simple problems Optical Memory and Neural Networks (Information Optics) 14 pp 59-72
[9] Tarkhov D and Vasilyev A 2005 New neural network technique to the numerical solution of mathematical physics problems. II: Complicated and nonstandard problems Optical Memory and Neural Networks (Information Optics) 14 pp 97-122
[10] Kainov N, Shemyakina T and Tarkhov D 2014 Application of neural network modeling to identification and prediction problems in ecology data analysis for metallurgy and welding industry Nonlinear Phenomena in Complex Systems 17 (1) pp 57-63