Comparative testing of algorithms for constructing an approximate neural network and finite-element solutions of the model boundary value problem

E A Varshavchik, A R Galyautdinova, J S Sedova, D A Tarkhov and A M Kobicheva

Peter the Great St.Petersburg Polytechnic University, 29, Polytechnic St., St. Petersburg, 195251, Russia

E-mail: aliyagalyautdinova@gmail.com

Abstract. In this paper we study two methods for solving the boundary value problem for the Laplace equation in the unit square. The first is based on the application of the square of the discrepancy to satisfy the Laplace equation, the second is based on the application of the energy functional. We compare the results obtained using functions characteristic of the finite element method and neural networks. Also, we implement a quality assessment and comparison of the approximate solution methods for different sampling point selection methods.

1. Introduction
Neural networks are increasingly used for approximate solutions of problems for differential equations (ordinary and partial derivatives) - in references we listed some of these numerous works. At the same time, there were some fundamental points left behind. In this paper, we focused on two of them.
Firstly, we investigate the potential application of the neural network functional basis and the functional basis that is characteristic of the finite element method. However, we do not use the solution of a linear system, although the linear problem is solved. This is due to the fact that we want to use a simple example to reveal the potential capabilities of the bases in question in order to get an idea of what can be expected when considering nonlinear problems using nonlinear optimization methods for determining approximation parameters. In the case of applying linear finite elements, the energy functional is minimized, and in the case of quadratic finite elements, the functional based on the application of the squared residual of satisfaction of the Laplace equation is minimized.
Secondly, we compare the variant with the constant test points, which is typical for most papers [1-5] with our approach based on complete regeneration of test points [6-9].

2. Material and methods
We consider the results of computational experiments for the Dirichlet problem for the Laplace equation $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$ in the unit square $[0;1] \times [0;1]$. As boundary conditions, we chose conditions on the sides of the square $u=0$ for $x=0$ or $y=0$ and $u=1$ for $x=1$ or $y=1$. We implemented the task by two methods.
We search the solution in the method using neural networks in the form of an RBF network \( y = \sum_{j=1}^{m} \psi_j \| x - c_j \| \). As basis functions we used standard Gaussian \( \psi_i(x,y,a_i,x_i,y_i) = \exp[-a_i((x-x_i)^2 + (y-y_i)^2)] \). The paper considers results with different numbers of neurons \( n \).

The solution is selected by optimizing the error functional \( J = J_1 + \delta J_2 \). We investigated the solutions and further compared them with the use of two functionals.

In the first series of computational experiments, as the first term of \( J_1 \), which is responsible for the execution of the equation, we used \( \sum_{j=1}^{M} (\Delta u(x'_j,y'_j))^2 \), where \((x'_j,y'_j)\) are test points randomly uniformly distributed inside the square \([0;1] \times [0;1]\). In this case, the quadratic finite elements were used.

In the second series of computational experiments, as the first term of \( J_1 \) we used the energy functional \( \sum_{j=1}^{M} (\Delta u(x'_j,y'_j))^2 + (\frac{\partial u(x'_j,y'_j)}{\partial x})^2 \). In this case, linear finite elements were used.

As the second term of \( J_2 \), which is responsible for the fulfillment of the boundary conditions, we used

\[
\sum_{j=1}^{M_1} (u(x''_j,0))^2 + \sum_{j=1}^{M_1} (u(0,y'_j))^2 + \sum_{j=1}^{M_1} (u(x''_j,1) - 1)^2 + \sum_{j=1}^{M_1} (u(1,y''_j) - 1)^2,
\]

where the test points in which the values of the function are calculated, we choose on the boundaries of the square.

The optimization was carried out using the RProp method. The experiment was implemented at the value of the number of test points \( M = 100 \), in which the operator is calculated and \( M_1 = 10 \) test points on each of the boundaries located at equal intervals. The test points inside the square were taken as random randomly distributed. In the case of partial regeneration of points, points with maximal discrepancies do not regenerate.

The results are compared with an approximate solution

\[
v(x,y) = \frac{4}{\pi} \sum_{i=1}^{100} sh[\pi(2i - 1)x] \sin[\pi(2i - 1)y] + sh[\pi(2i - 1)y] \sin[\pi(2i - 1)x]
\]

based on the Fourier series.

The neural network model was evaluated in two ways. The first is based on the square of errors of satisfaction of the equation within a square, over 10,000 random points (in the notation of \( J_1 \)). The second is for all 4 boundaries, and the points are distributed at regular intervals on each of its boundaries (in the notation of \( J_2 \)). In each method, for a given functional, the process was started 10 times. Based on the results obtained, the mean values, mathematical expectation and variance of \( J_1 \) and \( J_2 \) were calculated.

3. Results

In the first series of computational experiments, we compare the application of two kinds of functions with the use of the functional \( J_1 \), based on the calculation of the Laplace operator with consideration of a different number of basis functions. In this case, the number of chosen parameters is taken equal and the regeneration of all test points is used.

According to Table 1, we can say that both for points taken inside the square and for points on the boundary of the square, the neural network model yielded the best result, the error in this case is an order of magnitude smaller, but the variance shows that the finite element model gives a more stable decision.
Table 1. Evaluation of the quality of the finite element and neural network model

|                  | Finite Elements | Neural networks |                  |
|------------------|-----------------|-----------------|------------------|
|                  | Number of       |                 | n               |                 |
|                  | parameters      | $10^{-4}J_1$   | $10^{-4}J_2/4$  | $10^{-4}J_1$   | $10^{-4}J_2/4$ |
| minimum          | 36              | 0.711           | 0.0577          | 9               | 0.0948         | 0.0234          |
| average          |                 | 1.51            | 0.0647          |                 | 0.201          | 0.0316          |
| dispersion       |                 | 0.219           | 2.91E-05        |                 | 0.00539        | 3.11E-05        |
| minimum          | 121             | 2.21            | 0.0465          | 30              | 0.0622         | 0.0206          |
| average          |                 | 2.87            | 0.0511          |                 | 0.138          | 0.0257          |
| dispersion       |                 | 0.857           | 7.50E-06        |                 | 0.00418        | 2.20E-05        |

We illustrate the results with graphs of a finite-element and neural network approximate solution for a given functional.

![Graph](image)

Fig. 1. Comparison of the approximate solutions $u(x, x)$ and the approximate solution $v(x, x)$ for two cases: (a) – the finite element method for 36 chosen parameters, (b) – method with a neural network usage for $n=9$

![Graph](image)

Fig. 2. Comparison of the approximate solutions $u(x, x)$ and the approximate solution $v(x, x)$ for two cases: (a) – the finite element method for the 121 chosen parameters, (b) – method with a neural network usage for $n=30$

According to the graphs, we can conclude that the most accurate solution is obtained in the case of using neural networks. The finite element method gives an exact solution only at the boundaries of the
subregions, at other points the error is significant. This is due to the less efficient operation of the optimization method for functions of small smoothness.

Further, we present the data of the second series of computational experiments, which is different in that in this case the energy functional is minimized, that is, the sum of squares along the test points of the first derivatives of the sought function, and different amounts of basis functions are also considered.

Table 2. The quality evaluation of the finite element and neural network model

|                | Finite Elements | Neural networks |
|----------------|-----------------|-----------------|
| **Number of parameters** | 10^{-4}J_1 | 10^{-4}J_2/4 | n | 10^{-4}J_1 | 10^{-4}J_2/4 |
| **minimum**    | 5               | 0.0123          | 0.0575 | 9            | 0.00425          | 0.0317          |
| **average**    | 0.0661          | 0.0794          | 0.00208 | 0.00568          | 0.0343          |
| **dispersion** | 0.0113          | 0.0620          | 30 | 1.23E-06       | 3.68E-06       |
| **minimum**    | 10              | 0.0136          | 0.0620 | 0.00507          | 0.0313          |
| **average**    | 0.0372          | 0.0781          | 0.00631 | 0.00631          | 0.0349          |
| **dispersion** | 0.0017          | 0.000162        | 5,70E-07 | 4,69E-06       |

The results of the calculations presented in Table 2 show that, as in the first series of computational experiments, as within the square and on its boundaries, a much more accurate result is obtained when solving the problem posed by the use of neural networks. But in this case, using the energy functional in the work, the finite element method was not distinguished by the greater stability of the solutions, a smaller dispersion was also revealed when solving with neural networks.

We give graphs of the finite-element and neural network approximate solutions for the energy functional.

Fig. 3. Comparison of the approximate solutions $u(x, x)$ and the approximate solution $v(x, x)$ or two cases: (a) – the finite element method for 36 selected parameters, (b) – using a neural network for $n=9$ when using energy functional.
Fig. 4. Comparison of the approximate solutions \( u(x, x) \) and the approximate solution \( v(x, x) \) or two cases: (a) – the finite element method for 121 selected parameters, (b) – using a neural network for \( n=30 \) when using energy functional

Analyzing these graphs, we can conclude that the most accurate solution is obtained in the case of the use of neural networks. This method of approximation is more smooth.

It is important to investigate how the regeneration of points affects the error of the results. The main difference of our works [6-9] from the bulk of the papers [1-5] is that we apply periodic regeneration of test points. After every 5 steps of the non-linear optimization of functional \( J \), the test points are regenerated. Moreover, a certain fraction \( d_t \) of these points, in which the expression corresponding to the term \( J_1 \) is maximal, does not regenerate. In [1-5], test points do not regenerate, which corresponds to the case \( d_t=1 \). We also study the case of partial regeneration of points, when \( d_t=0.5 \). In this case, the study was carried out by the method of approximation by neural networks using the energy functional.

Table 3. Estimation of the point regeneration's effect on the result for a neural network model for parameters \( M=100, M_1 = 10, n = 9 \).

|        | \( d_t = 0 \) | \( d_t = 0.5 \) | \( d_t = 1 \) |
|--------|---------------|---------------|---------------|
|        | \( 10^{-4}J_1 \) | \( 10^{-4}J_2/4 \) | \( 10^{-4}J_1 \) | \( 10^{-4}J_2/4 \) |
| minimum | 0.00425 | 0.0317 | 0.00545 | 0.0304 | 0.00474 | 0.0258 |
| average | 0.00568 | 0.0343 | 0.00622 | 0.0365 | 0.00798 | 0.0332 |
| dispersion | 1.23E-06 | 3.68E-06 | 6.70E-07 | 7.90E-06 | 8.01E-06 | 0.000188 |

According to the data in Table 3, we can state that a complete regeneration of the points gives a significant decrease in the error within the square. In addition, the solution in this case becomes much more stable.

4. Conclusion

Comparing the data given in Tables 1 and 2, as well as graphs, we can conclude that the results obtained using the energy functional are more accurate. In addition, in both series of computational experiments, the method of approximation by neural networks is optimal.

In addition, in solving such problems it is more prudent to resort to the use of complete regeneration of points in order to achieve a more accurate and sustainable result.

Thus, the use of neural networks in solving a model boundary value problem for the Laplace equation in the unit square gives a smaller error than the finite element method, which allows us to recommend it for use in solving boundary value problems for partial differential equations.
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] Lagaris I E, Likas A and Fotiadis D I 1998 Artificial Neural Networks for Solving Ordinary and Partial Differential Equations *IEEE Transactions on Neural Networks* **9** (5) 987–1000

[2] Dissanayake M W M G and Phan-Thien N 1994 Neural-network-based approximations for solving partial differential equations *Communications in Numerical Methods in Engineering* **10** (3) 195–201

[3] Fasshauer G E 1999 Solving differential equations with radial basis functions: multilevel methods and smoothing *Adv. in Comp. Math.* 11 139–159

[4] Fornberg B and Larsson E 2003 A Numerical Study of some Radial Basis Function based Solution Methods for Elliptic PDEs *Computers and Mathematics with Applications* **46** 891–902

[5] Gorbachenko V I and Zhukov M V 2017 Solving Boundary Value Problems of Mathematical Physics Using Radial Basis Function Networks *Computational Mathematics and Mathematical Physics* **57** (1) 145–155

[6] Tarkhov D A and Vasilyev A N 2005 New neural network technique to the numerical solution of mathematical physics problems. I: Simple problems *Optical Memory and Neural Networks (Information Optics)* (Allerton Press, Inc.) **14** (1) 59–72

[7] Lazovskaya T V, Tarkhov D A and Vasilyev A N 2017 Parametric Neural Network Modeling in Engineering *Recent Patents on Engineering* **11** (1) 10–15

[8] Gorbachenko V I, Lazovskaya T V, Tarkhov D A, Vasilyev A N and Zhukov M V 2016 Neural Network Technique in Some Inverse Problems of Mathematical Physics *Springer International Publishing Switzerland* 310–316

[9] Budkina E M, Kuznetsov E B, Lazovskaya T V, Tarkhov D A, Shemyakina T A and Vasilyev A N 2017 Neural Network Approach to Intricate Problems Solving for Ordinary Differential Equations *Optical Memory and Neural Networks* **26** (2) 96–109