Appl. Comput. Math. **2** (2003), no. 1, pp. 57-64

NEUROMATHEMATICS: DEVELOPMENT TENDENCIES‡

GALUSHKIN A.I.†, KOROBKOVA. S.V.†, KAZANTSEV P.A.†§

ABSTRACT. This article is the summary of a set of Russian scientists' works, published in the monography [9], in the field of neuromathematics – neural network algorithms for solving mathematical tasks. The general methodic of solving the problems in the neural network logical basis and some unsolved problems that should be considered in the future are presented in this paper.

1. Which tasks are adequate of neurocomputers?

The class of the adequate of neurocomputers is getting wider. It is necessary to form the exact quantitative criteria of the need in the application of neurocomputers in the specified task. All the tasks solved on the computer can be conditionally divided into two groups: formalizable and nonformalizable tasks (fig.1).

The application of the neural network algorithms for the nonformalizable tasks is rather evident at present time. After the design of the neural network algorithm the corresponding computer program is written and the time needed for the task to be solved is analyzed. If the time needed for this task to be solved satisfies the user, the task is solved. If the time of the solution should be reduced, and sometimes it should be considerably reduced (dozens, hundreds, or thousands times) the two ways are possible:

(1) The design of the neuroprocessors – accelerators, based on the certain technology (silicon digital technology, silicon analog or analog-numerical technology, optical technology and others).
(2) Programming the designed neural network algorithm on the cluster computers (several dozens processors) or massive parallel supercomputers (several hundreds processors). These computers will be conditionally called pseudotransputer computers. In addition to that the important and difficult problem of multisequencing the neural network algorithm on the set of the processors of the transputer computer should be solved.

The formalizable tasks which really need to be solved using the neural network technology should be mentioned partially (fig. 1). For the formalizable the algorithm, using the classical methods of the computational mathematics is designed, and the time of its solution on the personal computer is analyzed. If the solution time of the algorithm using the classical methods of the computational mathematics on the personal computer satisfies the user the switch to the neural network algorithms is not necessary. The number of dimensions of most appearing tasks is so large that the time of their solution using the classical methods of the computational mathematics is very large and needs to be greatly reduced. There are two ways in this case (fig. 1):

(1) To design a neural network algorithm as it is shown above.

†Scientific Center of Neurocomputers,(095)2639430,
scn@mail.cnt.ru
‡The publication of this paper was supported by the grant of the Open Society Institute ZUG Foundation.
§*Manuscript received Martch 24, 2003* .

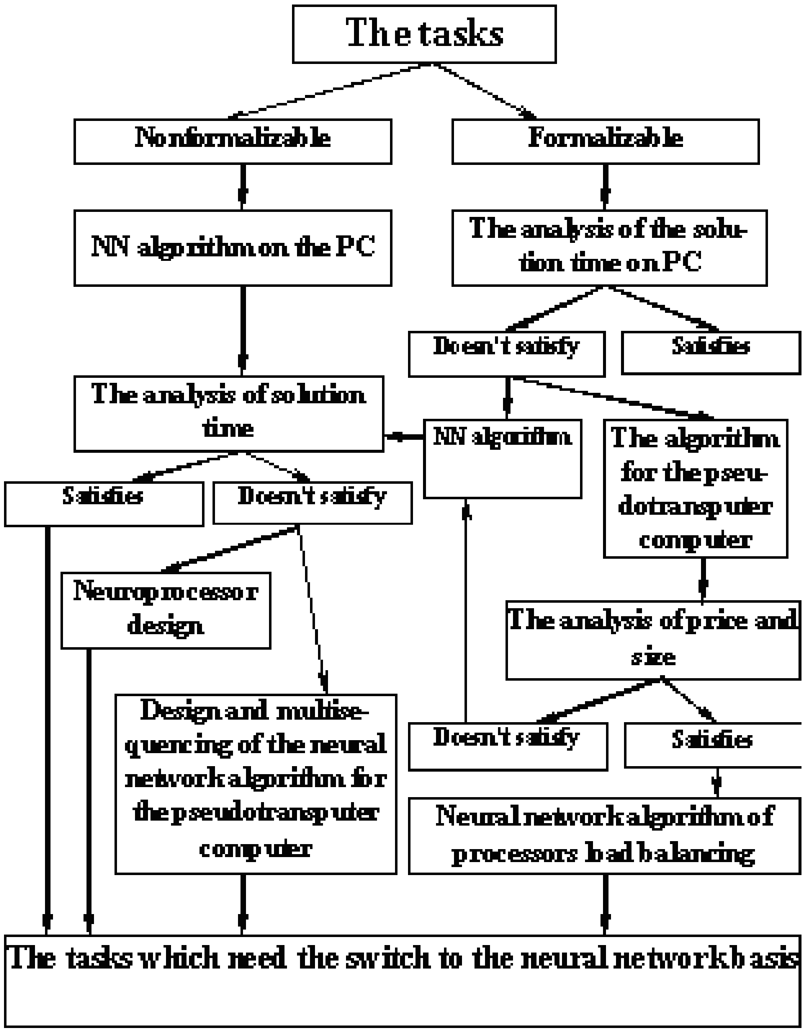

Figure 1. The tasks adequate of the neurocomputers.

(2) To design the algorithm of the solution using the classical computational mathematics with the multisequencing this algorithm on the set of processors. After that we should analyze, if the chosen computation system satisfies the user with its weight and size. If the computational system doesn't satisfy the user with its weight and size, it is natural to return to the design of the neural network algorithm to solve the task. (fig. 1).

If the computational system satisfies the user with its weight and size, some new neural network tasks may appear, for example the task of the design of the neural network algorithm of the optimization of the processor load balancing in the pseudotransputer computational system.

2. The unified method of solving tasks in the neural network logic basis.

The unified method of the solving tasks in the neural network logic basis was first described in [3]. The neural networks algorithms of solving the tasks have the uniform structure, which is determined by the methodic of the synthesis of the multilayer neural networks. The structure has the following phases:

- Physical and geometrical task definition;
- Mathematical task definition;

- Neural network task definition.

Neural network task definition in its turn contains:
o The description of the initial data;
o The definition of the input (n)of the neural network;
o Forming of the primary optimization functional of the neural network in this task;
o The definition of the output (n) of the neural network;
o The definition of the desired output of the neural network;
o The definition of the vector signal of the neural network error in this task;
o Forming of the secondary optimization functional of the neural network in this task;
o The analytic definition of the transformation performed by the neural network; the choice of the specific neural network structure;
o Finding the analytic expression for the secondary optimization functional adjusted parameter gradient;
o Forming the algorithm of the neural network training in the task;
o The choice of the initial conditions before training the neural network;
o The choice of the typical input signals to test the process of the solution of the task;
o The design of the experiment plan.

The creation of neuromathematics was initialized not by the mathematics, but by the specialists in the control theory and neurocomputers while the particular specialists in computational mathematics just silently watched the process. The base of the methodic of the design of neural networks algorithms are the methods of control theory, analytical selftraining systems, adaptive filtration and others. Neural network algorithms are quite the same for many different mathematical tasks, because while designing neural network algorithms there is a considerable amount of common problems in the methodic of the solution of these tasks. Most of them are either not regarded at all or passed over in silence while designing the classical solution algorithms.

3. A neural network is the best function approximator.

The statement that a neural network is the best function approximator can be explained the following way. For the classical approximation theory, which is the base of computational mathematics for the single processor and multiprocessor fon Newman computers, the function which should be approximated can be written this way:

$$y(x) = \sum_i \alpha_i \psi_i(x)$$

In this formula the set of functions ψ_i, is usually chosen a priori, considering the interests of the author of the specific method of approximation and some features of the functions, which are proved during the approximation. And for the neural networks the basic expression is quite different, for example for the threelayer feedforward neural network:

$$y(x) = \psi(\sum_i \alpha_i \psi(\sum_j \alpha_{ij} \psi(\sum_k \alpha_{kj} x_k)))$$

In this expression the structure of the equivalent basic functions ψ_i, is in the neural network basis with the set of the factors, which are trained during the search of the best approximator. This representation of the approximated function can be complicated if needed by the increase of the number of layers and changing the number of neurons in each layer of the neural network, introduction of the cross-links into the structure of the neural network (or as they are called recurrent neural networks). The advantages of the neural network representation of the approximated function:

- More flexibility of the basic functions ψ_i (x), because of the adaptation of the neural network to the input data;

- High potential parallelism of the neural network operations;
- Uniformity of the basic neural network operations (multiplication, addition, counting the simple nonlinear function ψ);
- Technical simplicity of performing these simple operations in the different physical realizations;
- The possibility of the control of the number of sum elements for $j$and $k$for each i, which also can be adapted in the search of the best approximator;
- The possibility to use the neural network structures for the a priori representation of the function $y(x)$.

4. The basic tasks of neuromathematics.

The basic mathematical tasks that are the subject of neuromathematics at the moment are listed below [9]:

- Arithmetic and logical operations on neural networks.
- Linear algebraic equation systems.
- Matrix operations.
- Non-linear algebraic equation systems.
- Linear inequality systems.
- Computational geometry.
- Minimum and maximum function searching.
- Function approximation.
- Function extrapolation.
- Optimization tasks.
  o General chapter.
  o Linear programming.
  o Square programming.
  o Non-linear programming.
  o Combinatorics.
  o Commivoyager task.
  o Timetable generation tasks.
  o Dynamic programming.
- Graph theory.
  o General works.
  o Path searching in graphs.
  o Loops in graphs.
  o Graph splitting.
  o Graph factions.
  oGraph imbrications.
  o Graph apportionment.
  oPetri networks.
- Random number generation.
- The tasks in finite fields and rings.
- Theorem proving.
- Sorting.
- Differential games.
- Ordinary differential equations.
- Integral equations.
- Differential equations in partial derivatives.
  o General chapter.
  o Grid generation.
  o Electrodynamics tasks.

o Gidrodynamics tasks.
o Aerodynamics tasks.
o Schrödinger equation.

- Chaos, attractors and neural networks.

5. New tasks of the computational mathematics, appearing during the consideration of the neural network solution algorithms.

Neuromathematics poses new problems in computational mathematics, which either haven't been solved yet or have been not enough thorough solved. The following tasks can be placed among such tasks:

- The choice of the initial conditions;
- Universalization of the algorithms of solving different tasks, and the related opportunity of the unbiased comparison of different algorithms by the quantitative features;
- The choice of the classes of the typical input signals for testing the quality of the algorithms (verification);
- Selection and getting rid of the false task definitions;
- The study and control of the dynamics of the solution process;
- The problem of the single and multiple solution of the tasks, which appears because of the multiextremality of the optimization functional.

6. The choice of the initial conditions.

The problem of the effective neural network solutions of the mathematical tasks creates new methods of the choice of the initial conditions while designing the iterative algorithms. The first steps in neuromathematics have shown that the choice of the initial conditions when training neural networks should be done separately for each task. The idea of the choice of the initial conditions for the pattern recognition and selfteaching was shown in [2, 5]. In our opinion the choice of the initial conditions of the neural network if the task of filtration or extrapolation should be done by the neural network realizing the equivalent optimal linear filter [1]. Generally the solution of any tasks is started with the neural network realization of the equivalent linear variant of the solution of the task and further training the neural network on the nonlinear features.

7. False task definitions.

While considering the neural network algorithms of solving tasks the problem of false task definitions appears. Let's give the example. The solution of the linear differential equation systems in the classical computational mathematics is sometimes equivalent to the matrix inversion. It is important to mention that the neural network algorithms of solving these two tasks – solving linear differential equation systems and matrix inversion – are completely different. So solving the tasks of matrix inversion becomes unnecessary, the task becomes false. The same is with the ordinary differential equation systems. These equations are the means of formalizing the behavior of the physical objects, the general description of which is in the form of ordinary differential equation systems, the factors of which are obtained during the experiment. Then there is the special computational procedure of solving these equations to obtain the output signals from the input ones. In fact the use of neurocomputers leads to the fact that the formal description and solving the ordinary differential equation systems becomes unnecessary, making the neural network of the special structure, the factors of which are obtained while training the neural network on the experimental data from which the factors of the nonlinear differential equation system were obtained before, the means of the formal description of the physical objects. That's why the themes of neurocontrol considerably reduces the interest in the problem of the solution of the ordinary nonlinear differential equations.

8. The control of the dynamics of the process of the solution of the task.

Just during the design of the neural network algorithms of the task solution, besides the effective choice of the initial conditions, there are several methods of the control of the dynamics of the task solution process, including the convergence time:

- The control of the parameters (the choice of modifications) of the iterational procedure of the search of the multiextremal optimization functional in the task solution;
- Filtering and extrapolating the signal corresponding to the optimization functional gradient while solving the task [1];
- Tuning the steepness factors of the activation functions in the multilayer neural networks;
- The choice and the analysis of the special structures of neural networks, adequate of the solved task class (cellular neural networks, continual neural networks, neural networks with lateral, random and feedback connections, neural networks with variable structure [2, 5]);

The possibility of varying the types of optimization functional by changing the structure of the neural network with forward connections can be illustrated with the specific example (fig. 2).

For the first variant of the structure – the single neuron, the optimization functional in the selfteaching mode has four local extremes with the equal values of the functional in the extremum points.

For the second variant of the structure of the neural network with forward connections represented as a threelayer neural network with four neurons in the first layer the optimization functional in the selfteaching mode has a global extremum with the hiperplanes distribution, realized by the neurons of the first layer, besides a great number of local extremums.

The problem of single and multiple solutions in important in the design of the algorithms. In the case of neural networks this problem is closely connected with the problem of multiextremality of the optimization functional, and also with the possibility of varying the multiextremal optimization functional by varying the structure of the neural network with forward connections, solving this task. For some tasks the multisolution problem is natural, for example the task of solving the linear inequality system, where we should find any point in the area boarded by the hiperplanes.

There was the attempt to represent the well-known classical algorithms of solving the algebraic equation systems in the neural network basis. The limited nature of the equivalent neural network structures, corresponding to the classical algorithms, and the algorithms of their adaptation and the necessity of the switch to the more complex neural network structures and more complex adaptation algorithms, which was impossible in the classical computational mathematics, oriented on the fon Newman computational systems structures.

9. The problems of multisequencing the neural network algorithms.

Just like the problem of multisequencing the classical algorithms, oriented on the fon Newman architectures is the subject of computational mathematics, the subject of neuromathematics is the task of multisequencing the neural network algorithms on a number of nodes, which are the components of the commutation environment of the modern neurocomputers [6-8]. The application mathematician – the developer of the neural network algorithm of solving the task, requiring hardware support to accelerate the solution of the task, after developing the main algorithm, has to solve the task of multisequencing it on the set of the processing nodes. Each of these nodes is a fragment of the neuroaccelerator. The same task of multisequencing the neural network algorithm appears when the neural network algorithm is emulated on the modern supercomputers and cluster computers. As it had been mentioned in [7, 8], the neural network algorithms being multisequenced on a number of commutation system nodes of the supercomputer, cluster computer or neurocomputer provide more even loading of the commutation system and processor nodes and allow the revision of neural network algorithm (the choice of the structures adequate of the commutation system), on the specific commutation system, providing the specified transfer quality. Parallel algorithms for the transputerlike and transputer computers

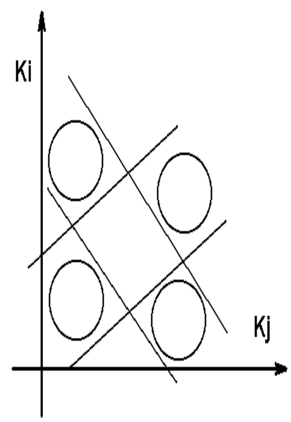

FIGURE 2. The illustration of the possibility control of optimization functional by changing the structure of the neural network with forward connections. (the circles are the lines of equal values for the probability density distribution function)

have been designed because of the algoritmist's desire to reduce the information interchange between the processors as much as possible. There is almost no load balancing at the same time of the processors. In the case of multisequencing on the set of nodes of the commutation environment of neurocomputer load balancing becomes possible with the choice of the neural network structure adequate of the commutation environment. But the losses due to the limited structure of the chosen neural network are possible in this case.

10. Conclusions.

(1) The bibliography of the scientific works in the field of neural network algorithms for solving mathematical tasks, which have been published in the recent 10-12 years, is quite a proof of the emerging of a new section of the computational mathematics, which was called neuromathematics in the late 80s.
(2) The natural development of neuromathematics is closely connected with people's natural desire to solve difficult tasks cheaper and faster. though we should mention, that it is in the beginning of its evolution and, of cause, it needs young specialists' concern.
(3) The range of mathematical tasks for which the neural network algorithms have been developing the recent years is extremely wide. It covers almost all mathematical tasks, which are difficult to solve using the ordinary computers. The limits are the solution time and the size of the computational system.
(4) It has been mentioned above that monography [9] is devoted to neuromathematics, i.e. the neural network algorithms for solving mathematical tasks, formulated in the well-known classical terms without any application specification. The only exception in [9] are the chapters of electro, gas and aerodynamics, where the classical differential equations in partial derivatives are solved in the neural network logic basis for the specific equation systems. That was done because of the extraordinary importance of these chapters for the future superneurocomputers, oriented on the scientific researches similar to the ones characteristic of the classical supercomputers.

References

[1] Galushkin A.I.,*The calculation of the optimal linear discrete filters. // Automatics and computational technics*- Moscow, "Mashinostroenie", N9, 1968.
[2] Galushkin A.I.,*The synthesis of multilayer pattern recognition systems*, Moscow, "Energy", 1974 (368p).
[3] Galushkin A.I., Sudarikov V.A.,Shabanov E.N. *Neuromathematics: the methods of solving the tasks on the neurocomputers. – Mathematical modeling* - Moscow., 1991,- vol. 3, N8, p.101-106.
[4] Galushkin A.I., *Neural Network Theory and Neuromathematics.* IJCNN'99, Washington.
[5] Galushkin A.I., *Neural network theory* – Moscow, IPRGR, 2000.
[6] Galushkin A.I., *Neurocomputers.*- Moscow, IPRGR, 2000.
[7] Galushkin A. I.,*Supercomputers and neurocomputers.* // Neurocomputers: design and applications, N.Y., Begell House Publ., 2000. N6.
[8] Galushkin A.I.,*Superneurocomputers and neurocomputers. // Neurocomputers: design and applications.*- Moscow, 2001, N6.
[9] *Neuromathematics* (collective monography), Moscow., IPRGR, 2002.
[10] Galushkin A.I.,*Neuromathematics: development problems. Neurocomputers: design and applications*, Moscow, Radiotechnika, 2003, 1 (p. 3-40).

Galushkin A.I., Korobkova S.V., Kazantsev P.A.
Scientific Center of Neurocomputers
Moscow, Russia Phone: (095)2639430, E-mail: scn@mail.cnt.ru