

Abstracts of Dissertations

Institute of Information and
Communication Technologies

BULGARIAN ACADEMY OF
SCIENCES



6 / 2017



DISTRIBUTED SYSTEM FOR
TIME SERIES PREDICTION
WITH EVOLUTIONARY
ALGORITHMS AND
ARTIFICIAL NEURAL
NETWORKS

Todor Balabanov

РАЗПРЕДЕЛЕНА СИСТЕМА ЗА
ПРОГНОЗИРАНЕ НА ВРЕМЕНИ
РЕДОВЕ С ЕВОЛЮЦИОННИ
АЛГОРИТМИ И ИЗКУСТВЕНИ
НЕВРОННИ МРЕЖИ

Todor Balabanov

Автореферати на дисертации

Институт по информационни и
комуникационни технологии

БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ

ISSN: 1314-6351

Поредицата „Авториферати на дисертации на Института по информационни и комуникационни технологии при Българската академия на науките“ представя в електронен формат авториферати на дисертации за получаване на научната степен „Доктор на науките“ или на образователната и научната степен „Доктор“, защитени в Института по информационни и комуникационни технологии при Българската академия на науките. Представените трудове отразяват нови научни и научно-приложни приноси в редица области на информационните и комуникационните технологии като Компютърни мрежи и архитектури, Паралелни алгоритми, Научни пресмятания, Лингвистично моделиране, Математически методи за обработка на сензорна информация, Информационни технологии в сигурността, Технологии за управление и обработка на знания, Грид-технологии и приложения, Оптимизация и вземане на решения, Обработка на сигнали и разпознаване на образи, Интелигентни системи, Информационни процеси и системи, Вградени интелигентни технологии, Йерархични системи, Комуникационни системи и услуги и др.

Редактори

Генадий Агре

Институт по информационни и комуникационни технологии, Българска академия на науките
E-mail: agre@iinf.bas.bg

Райна Георгиева

Институт по информационни и комуникационни технологии, Българска академия на науките
E-mail: rayna@parallel.bas.bg

Даниела Борисова

Институт по информационни и комуникационни технологии, Българска академия на науките
E-mail: dborissova@iit.bas.bg

Настоящото издание е обект на авторско право. Всички права са запазени при превод, разпечатване, използване на илюстрации, цитирания, разпространение, възпроизвеждане на микрофилми или по други начини, както и съхранение в бази от данни на всички или част от материалите в настоящето издание. Копирането на изданието или на част от съдържанието му е разрешено само със съгласието на авторите и/или редакторите

*The series **Abstracts of Dissertations of the Institute of Information and Communication Technologies at the Bulgarian Academy of Sciences** presents in an electronic format the abstracts of Doctor of Sciences and PhD dissertations defended in the Institute of Information and Communication Technologies at the Bulgarian Academy of Sciences. The studies provide new original results in such areas of Information and Communication Technologies as Computer Networks and Architectures, Parallel Algorithms, Scientific Computations, Linguistic Modelling, Mathematical Methods for Sensor Data Processing, Information Technologies for Security, Technologies for Knowledge management and processing, Grid Technologies and Applications, Optimization and Decision Making, Signal Processing and Pattern Recognition, Information Processing and Systems, Intelligent Systems, Embedded Intelligent Technologies, Hierarchical Systems, Communication Systems and Services, etc.*

Editors

Gennady Agre

Institute of Information and Communication Technologies, Bulgarian Academy of Sciences
E-mail: agre@iinf.bas.bg

Rayna Georgieva

Institute of Information and Communication Technologies, Bulgarian Academy of Sciences
E-mail: rayna@parallel.bas.bg

Daniela Borissova

Institute of Information and Communication Technologies, Bulgarian Academy of Sciences
E-mail: dborissova@iit.bas.bg

This work is subjected to copyright. All rights are reserved, whether the whole or part of the materials is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in other ways, and storage in data banks. Duplication of this work or part thereof is only permitted under the provisions of the authors and/or editor.



BULGARIAN ACADEMY OF SCIENCES

Abstract of PhD Thesis

DISTRIBUTED SYSTEM FOR TIME SERIES PREDICTION WITH EVOLUTIONARY ALGORITHMS AND ARTIFICIAL NEURAL NETWORKS

Todor Dimitrov Balabanov

Supervisor: Assoc. Prof. Krassimira Genova

Approved by Supervising Committee:

Assoc. Prof. Vladimir Monov
Prof. Maria Nisheva
Prof. Milena Lazarova
Assoc. Prof. Alexander Shikalanov
Assoc. Prof. Krassimira Genova



**INSTITUTE OF INFORMATION AND
COMMUNICATION TECHNOLOGIES**

Department of Modeling and Optimization

The PhD thesis was discussed and allowed to be defended during an extended session of the Modeling and Optimization Department at IICT-BAS, which had been held on September 8, 2017.

The defense of the PhD thesis had been held on December 12, 2017 at 10:30AM in room 507, block 2, IICT-BAS.

The full volume of the dissertation is 135 pages, which includes 25 figures, 30 tables, and 30 pages of references with 240 positions.

Keywords: artificial neural networks, genetic algorithms, differential evolution, financial forecasting, distributed computing, parallel computing, software development, decision support systems

Research Topic and Motivation

A major flaw in the training of artificial neural networks with evolutionary algorithms is the fact that almost there is no implementation of the full-connected artificial neural networks training. The second major drawback is that generally artificial neural networks are used in two phases - training and operation. The problems described by time series is characterized with grow as time progress and naturally artificial neural networks should be re-training, to the new information. Using exact numerical methods (such as the most popular back propagation) is effective, but relatively inapplicable in artificial neural network with full-connected topology. In the literature there are presented numerous hybrid models for training artificial neural networks with exact numerical methods in combination with methods of heuristic optimization. Third, although minor problem is proper selection of appropriate and accurate heuristics for making hybrid training. In the literature there is a relatively small overlap of artificial neural networks trained in a distributed environment so that maximum benefit of parallel implementation of optimization algorithms from the group of evolutionary algorithms and population algorithms.

Aims and Tasks of the Dissertation

To solve the problems identified in the thesis the following objectives are distinguished: 1. Realization of models of full-connected artificial neural networks; 2. Implementation of algorithms for continuous training, even during operation of the artificial neural networks; 3. Implementation of a hybrid combination presented in the literature by exact and heuristic algorithms for artificial neural networks training; 4. Implementation of a

software system as a client-server application running in a distributed computing environment.

To achieve the objectives outlined, group of tasks are targeted. As the main task of the thesis is the realization of a model of artificial neural networks (C ++ and JavaScript implementation) as fully-connected topology. This task consists of two pieces - an implementation of the client application and implementation of the server application. Second in importance is the task of continuous fully-connected artificial neural networks training. The training must take place in a continuous 24/7 cycle. For this purpose algorithms suitable for continuous distribution of tasks from server to client should be available and continuous training of local copies of artificial neural networks that support client machines. Third in importance is the task of appropriately combining algorithms such as genetic algorithms and differential evolution algorithms and back propagation, so that the duration of the training of artificial neural networks to be reduced. The fourth most important task is related to the technical implementation of the system and carrying out the necessary experiments to confirm the effectiveness with which it works.

List of Publications

1. Balabanov, T., Zankinski, I., Simeonova, V., Forecasting Time Series with Artificial Neural Networks and Differential Evolution in Distributed Environment, IIT-BAS Working Papers, IIT/WP-268B, ISSN:1310-652X, 2010.
2. Balabanov, T., Forecasting with Heuristic Approaches in a Distributed Environment, Proceedings of Anniversary Scientific Conference 40 Years Department of Industrial Automation, ISBN:978-954-465-043-8, 163-166, 2011.
3. Balabanov, T., Zankinski, I., Dobrinkova, N., Time Series Prediction by Artificial Neural Networks and Differential Evolution in Distributed Environment, Large-Scale Scientific Computing, Lecture Notes in Computer Science, vol. 7116, ISBN:978-3-642-29842-4, DOI:10.1007/978-3-642-29843-1_22, 198-205, 2011.
4. Balabanov, T., Avoiding Local Optimums in Distributed Population based Heuristic Algorithms, Proceedings of the XXII International Symposium Management of energy, Industrial and Environmental Systems, ISSN:1313-2237, 83 - 86, 2015.
5. Balabanov, T., Zankinski, I., Barova, M., Distributed Evolutionary Computing Migration Strategy by Incident Node Participation. Large-Scale Scientific Computing, Lecture Notes in Computer Science, 9374, Springer International Publishing Switzerland, ISBN:978-3-319-26520-9, DOI:10.1007, 203 - 209, 2015.
6. Balabanov, T., Genova, K., AJAX Distributed System for Evolutionary Algorithms based Artificial Neural Networks Training, Proceedings of the XXIV International Symposium Management of energy, Industrial and Environmental Systems, ISSN:1313-2237, 49 - 52, 2016.

7. Balabanov T., Keremedchiev D., Goranov I., Web Distributed Computing For Evolutionary Training Of Artificial Neural Networks, International Conference InfoTech-2016, Varna - St. St. Constantine and Elena resort, Bulgaria, ISSN:1314-1023, 210-216, 2016.
8. Balabanov, T., Genova, K., Distributed System for Artificial Neural Networks Training Based on Mobile Devices, Proceedings of International Conference Automatics and Informatics'2016, ISSN:1313-1850, 49 - 52, 2016.

Chapter 1 - Overview of Forecasting Time Series Systems

Forecasting time series is a complex task that is used in many areas with a lot of research in the field. To create predictive software systems require multiple skills and knowledge. The importance of the tasks of forecasting is so great that they are used in almost every sphere of human life (temperature, rainfall, power supplies, etc.).

Analysis - problems, goals and tasks

Forecasting of financial time series with the help of heuristics formulate a set of problems that can be pointed as objectives of the thesis, and these goals can be broken down into tasks that will lead to achieving the goals.

A major drawback of the discussed solutions for artificial neural networks training with evolutionary algorithms is the fact that there is almost no occurrence of full-connected artificial neural networks training. The second major drawback in the existing examined solutions is that generally artificial neural networks are used in two phases - training and operation. The problems described in the terms of the time series is characterized with growth and naturally artificial neural networks should be constantly re-trained to fulfill the newly obtained information. Using exact numerical methods (such as the most popular back propagation) is effective, but relatively inapplicable in artificial neural network with full-connected topology. In the literature there are numerous hybrid models for artificial neural networks training with exact numerical methods in combination with heuristic optimization methods. Third, although minor problem is proper selection of appropriate and accurate

heuristics for making hybrid training. In researched information sources relatively little overlap opportunities are seen for artificial neural networks training in a distributed computing environment so that maximum benefit to be obtained by optimization algorithms in the groups of evolutionary algorithms and population algorithms.

The objectives of the thesis are as follows: 1. To implement models of full-connected artificial neural networks; 2. To implement algorithms for continuous training, even during operation of artificial neural networks; 3. To implement the hybrid combination presented in the literature between exact numerical and heuristic algorithms for artificial neural networks training ; 4. To develop a software system as a client-server application running in a distributed computing environment.

To achieve the objectives a series of tasks were formulated. The main task of the thesis is the implementation of a model of artificial neural networks in C ++ and JavaScript as fully-connected topology. This task consists of two pieces - an implementation of the client application and an implementation of the server application. Second in importance is the task of continuous training of fully-connected artificial neural networks. The training must take place in a 24/7 cycle for this purpose there should be available algorithms suitable for continuous distribution of tasks from the server to the client applications and continuous training of local copies of artificial neural networks supported on the client machines. Third in importance is the task of appropriately combining algorithms such as genetic algorithms and differential evolution algorithms with back propagation, so that the duration of the training of artificial neural networks to be reduced. The fourth most important task is related to the technical implementation of the system and carrying out the necessary experiments to confirm the effectiveness with which it works.

Chapter 2 - Model for Time Series Forecasting with Artificial Neural Networks Trained by Evolutionary Algorithms

The complexity of the task of time series forecasting requires the development of faster acting algorithms that lead to predictions with greater accuracy. The basis of artificial neural networks predictions is the idea of training based on a sequence data for the recent period. The choice of training algorithm can be made of two main areas - exact numerical optimization methods and heuristic optimization method. Applying appropriate heuristic algorithms may lead to their use in the distributed environment and significantly speed up the process of training.

Proposed Forecasting System

The thesis is proposec for the purpose of forecasting a mathematical model based on artificial neural networks to be used. The training is done with differential evolution in the form of calculations in a distributed environment. The allowed types of topologies are: 1) Multi-layer perceptron; 2) Multi-layer recurrent; 3) Full-connected networks. The proposed model is based on the classic artificial neural network. As a summation function is proposed the usage of a linear function.



Fig. 2.1 Training system based on DE and ANN in a distributed environment (left) and sharing tasks in a distributed computing system (right).

The training of artificial neural networks are implemented using differential evolution. In the systems for distributed environment calculations the centralized server distributes calculating tasks to the remote computers (Fig. 2.1 - Step 1). At certain moment in time the remote computers perform the task (Fig. 2.1 - Step 2). When the client complete the calculation of the assigned task it returns the result to the server (Fig. 2.1 - Step 3). These three general steps in this process are repeated until there is a working server and client

computers attached to it.

In the parallel implementation of differential evolution local copies of the artificial neural network and the differential evolution are created [BAZA2010]. The training is carried out locally by each chromosome in the population indicates a single point (individual) in the search space. While different individuals are grouped relatively close to each other in this space. Upon separation of the calculations on a plurality of computers, like grouping of individuals can result in a local study of the different areas in the space of solutions (the search space) [BATO2011]. In this aspect, the most important is the synchronization policy by the synchronization process includes the broadcasting of the best individuals and their collection in one centralized place (in client-server architecture). This set of globally most fitted individuals can be used when creating of the local sub populations is done. In the presence of a large number of customers, the server load can be lower by reducing the frequency of communication with the clients.

Chapter 3 - Practical Implementation of Time Series Forecasting Software System with Artificial Neural Networks and Evolutionary Algorithms

In MetaTrader 4 trading platform all information is programmatically accessible by means of the built-in programming language called MQL4. The trade terminal MetaTrader 4 provides visualization of information from time series.

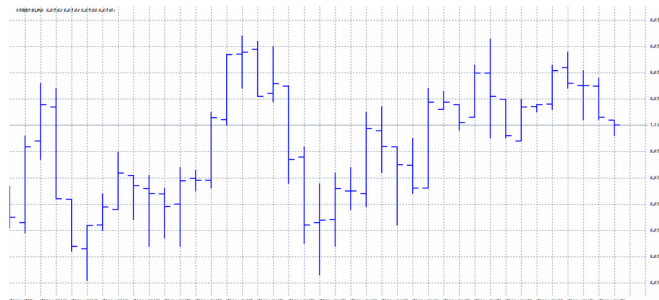


Fig. 3.1 Quotes EUR / USD in MetaTrader 4.

Programming API of MetaTrader 4 allows quotes for each currency pair to be provided outside the trading terminal with normalization applied in advance. The chart specifies the highest level achieved, then the lowest level achieved and the values of each bar normalize in the range 0.0-1.0. This normalization step is necessary because using artificial neural network works with signals in the range 0.0-1.0. The second reason for the performance of normalization is the presence of multiple currency pairs that have different nominal values. For example, the ratio EUR/USD is in the range of 1.13510 (information to date 10.20.2015), while the ratio USD/JPY is around 119 830 (information to date 20.10.2015). Once the artificial neural network offered its forecast the operation of scaling reverses the output to minimum and maximum value on the graph.

Traders receive the currency pair quotes from a centralized server. This organization allows the computer of every trader to launch additional module for calculations (proposed in the thesis) to perform calculations on the artificial neural networks training. The proposed calculation module is a standard executable file that reads the quotations recorded by MQL4 in text files. The training of a specific topology artificial neural networks for specific quotes, in a particular time interval can be distributed among multiple computers.

Backend Module for Monitoring and Tuning of the System

The module for monitoring and system tuning is used to set the parameters under which runs the training of artificial neural networks. The module was developed as a Java Applet and it can be accessed by different computer systems using a standard web browser. The administrative module is intended solely for system operators who are in the role of trusted employees. Additionally, the security level in the DBMS limit access to data only within the local network infrastructure.

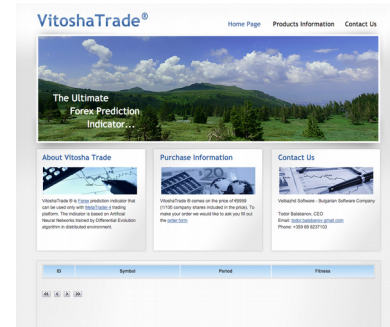
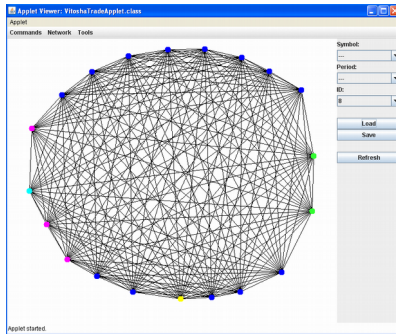


Fig. 3.2 ANN visualization screenshot [NIRA2012] [ANMO2011] (left) and main web page of the public information system (right).

The most significant opportunity of the back-end system is the visualization of an artificial neural network with all connections, as weights and/or activities as well. Neurons in the network are displayed with different colors according to the role held in the network (input, output, or internal shifting). Location of neurons to visual space can be changed by using the mouse and keyboard (Fig. 3.2 – left).

Front-end Module of the Information System

Front-end module is a web site developed with HTML, CSS and PHP. The aim of public module is to present information system, which is for public use. The public information includes a list of various artificial neural networks in the DBMS, different efficiency by which individual networks are trained and their service identifier within the DBMS (Fig. 3.3 - right).

The overall system includes various modules, some of which are available for public use. List of publicly available modules is shown in the additional web page. Also front-end contains a web page with contact information.

Chapter 4 - Application of the Forecasting System and Analysis of the Experimental Results

With increasing number of nodes in artificial neural networks, the number of weights increased significantly, which could further impede the training process, especially for more complex topology of artificial neural networks. The first direction of research is what topology of artificial neural networks would be most effective for the purposes of the forecasting. The classical three-layered artificial neural network and full-connected artificial neural network are examined. Not only the number of layers and the particular set of connections between them are important is also essential the total number of neural nodes.

An important point in the study is the choice of type of input and output data used in the artificial neural networks. In terms of output, it is desirable to obtain directly an estimated value (specific number or the difference of the predicted point relative to the current). In terms of inputs are present the following options: 1. To submit time value; 2. To submit a dimension value of a previous time; 3. To submit a set of values of magnitude in previous moments; 4. To submit differences in the value of the magnitude of previous moments; 5. To submit a combination of previous values of the magnitude and timing information.

Incident nodes participation in a distributed computing system

In the proposed distributed computing system the main population is on a central unit (server). Each client is attached to the system and receives a subset of the global population (Fig. 4.1 - left). After this initial initialization optimization is carried out on a local level, such as the central node is informed only for better individuals from the local population compared with the already known best global achievements.

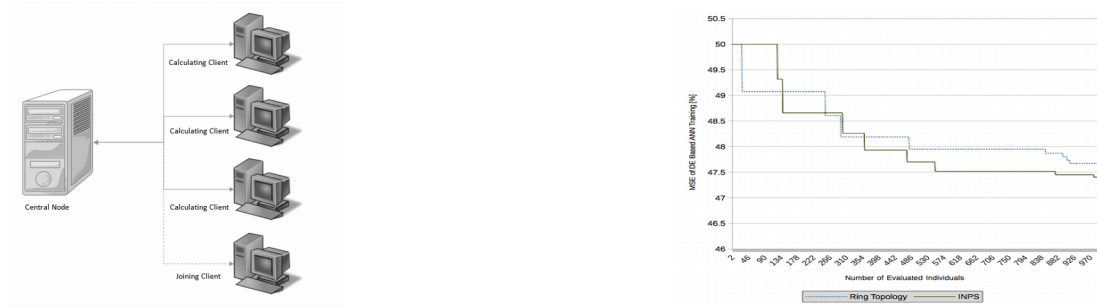


Fig. 4.1 Incident nodes participation in a distributed computing system (left) and comparison between the incident nodes participation and ring topology (right).

The central unit serves as a repository of individuals with the best achieved vitality of local optimization procedures (a form of elitism). This part of the system for the exchange of individuals between computing nodes was presented in [TBIZ2015], as a comparison between incident nodes participation and exchange of information in a ring topology.

From the experiments done it is visible that the incident nodes participation converges better (Fig. 4.1 – right).

Comparison Between C ++ and JavaScript for Forward Pass of ANN Training

In this study [BAGE2016a] a model for artificial neural networks training is presented with the usage of differential evolution as training algorithm. The training of artificial neural networks is implemented in a distributed computing environment and the network is presented in the form of JavaScript code. The communication with the central node is performed by usage of asynchronous AJAX requests.

The performance of JavaScript execution is the main leading argument this language to be avoided in carrying out large-volume calculations or computations that require greater accuracy.



Fig. 4.2 Comparison for performance for information distribution in ANN, at the forward pass of the training between C++ and JavaScript. On the X axis are the number of neurons, and the Y axis is the performance in milliseconds (left) and standard deviation during propagation of information in ANN, at the forward pass between C++ and JavaScript. Along the X axis is the number of neurons, and the Y axis is the standard deviation (right).

In order to check the difference in the performance of the proposed AJAX-JavaScript solution there were made a series of experiments with feedforward pass in an artificial neural network. The feed-forward pass is encoded in two separate modules of the VitoshTrade system - respectively module in C++ and JavaScript module. As is evident from Fig. 4.2 (left), that the performance is comparable to networks of 10 to 100 neurons. Each experiment was executed 30 times and the average values are presented in the figure [BAKE2016].

In terms of performance languages as C++ and JavaScript are comparable, but the stability of the calculation process is better in C++, which can be seen in Fig. 4.2 (right), which reflects the mean square deviation of the time needed for computation. This difference is mainly due to the presence of an interpreter and a web browser, something that is not present in the calculations with C++. In the category languages as C++ program is initially translated to assembly language, and then to machine code.

The implementation of calculations in a distributed environment as AJAX web-based system leads to a very high degree of scalability in the system. Distributed computing can run on any device supporting modern web browser which are capable to run JavaScript and AJAX. Since the calculation is carried out within the web browser, which in turn is a process of the operating system, and it in turn runs on physical hardware, although comparable in performance boost, the calculations are less reliable (presence of interpreter) than they were in the implementation of languages like C/C++ or Assembler.

Comparison Between Different ANN Topologies

In time series forecasting with artificial neural networks the two most recent model networks are classical three-layered network without recurrent connections and network with recurrent connections. The experiments carried out with three-layer networks that have different size of the hidden layer and the fully connected network (each node is connected with each other, including it selves). With the full connectivity the effect of short-term memory in the network is targeted.

For comparison between the full-connected and three-layer network topology of 28-12-5 is selected. Fully connected network consists of 45 nodes. Both networks are trained with genetic algorithms with the following parameters for the genetic algorithm: Population size - 37; Probability of crossover - 0.9; Probability of mutation - 0.03; Title elite elements - 0.1; Arguments for a race selection - 2; Type crossing - binary; Type mutation - random delta vector; Condition for optimization end - 24 hours.

Both networks are trained in equal conditions. The condition for the end of the training is the expiry of a fixed period of time, namely 24 hours. As a result of training, fully-connected network shows better parameters for less epochs of training. For multi-layer perceptron 296037 epochs are needed. The best achieved individual in the population has fitness of 656.1699030329493 and the error of the artificial network is 317.7286234453099, while fully-connected network needed 294003 epochs with best fitness value of 656.169903010489 and network error of 317.0226357675591.

Having more weights in a fully-connected network results in a slow process of training, but the presence of recurrent connections allows the formation of short-term memory in artificial neural networks and at considerably longer training that gives priority to this model.

Comparison Between Three-Layered ANNs with Different Size in the Hidden Layer

The experiments were carried out with multilayer perceptron (3 layers). The input submits 25 values from the past period and on the output are 5 values are expected (the future values). For the purposes of the experiment data are divided into three subsets: 1. Training data; 2. Validation data; 3. Testing data. The tested topologies have size of the hidden layer of 30 to 1 nodes. All experiments were performed at the same intervals. Fig. 4.3 (left) shows the difference in the number of epochs according the size of the hidden layer.



Fig. 4.3 Number training epochs at the same time training interval, but different size of the hidden layer (left) and average error done by ANN with training of network topology of three layers, the size of the hidden layer ranges from 30 to 1 (right).

In the experiments the average error that the networks of varying topology do is evaluated. The aim is to establish in which values of the size of the hidden layer the network is most effective. Error calculation is made only using data from the testing subset. These are data that are not used in the process of training and validation.

Fig. 4.3 (right) clearly shows that the most promising are the models with 22, 12 and 9 elements in the hidden layer, which corresponds to the topologies of networks: 1. 25-22-5; 2. 25-12-5; 3. 25-9-5.

Comparison of Forecasting Effectiveness at Different Input Data Window

For the three most effective model pointed in Fig. 4.3 (right), were carried out further

experiments with the size, which has a time window from past periods. The amount of time window has been studied from 35 to 5 values as the window to predict future results is fixed only to 5 values.



Fig. 4.4 Number training epochs at the same time interval training, but different size of the input layer (left) and average error done by ANN during training of network topology of three layers, where the size of the input layer ranges from 35 to 5 (right).

From Fig. 4.4 (left) it is clear that the size of the input layer almost has no effect on the number of periods during which the training is held in a fixed interval of time (1 hour). Also, it is clearly noticeable that the size of the hidden layer directly affects the number training epoches at a fixed time interval. The slowest training is on hidden layer of 22 elements, the fastest training is on hidden layer of 9 elements.

In Fig. 4.4 (right) clearly appears in which values of the input layer artificial neural network has achieved the lowest error for the proposed estimate. For the size of the hidden layer of 22 elements the most effective input layer has 29 elements. In a hidden layer of 12 elements the most effective input layer has 28 elements. And in the hidden layer of the 9 elements the most effective input layer has 26 elements. Of the three options best results are obtained with a hidden layer of 12 elements, which is about half of the input layer.

Conclusions

Summary of Dissertation Achievements and Ideas for Future Work

In the dissertation there are presented main results obtained from studies conducted in

methods for predicting time series, algorithms and software systems for solving problems in the field of forecasting. The purpose of this research is the development of heuristic approaches to machine self-adaptive forecasting. Based on these approaches heuristics and algorithms have been proposed to -enhance performance in artificial neural networks training. Accelerated performance in artificial neural networks training is particularly important for predicting time series, as the phase of training is significantly slower than the stage of operating itself. The characteristics of these methods and algorithms, the quality of the proposed design, and good program implementation are the basis for the creation of an effective software system for time series forecasting in the field of currency markets.

The results achieved in the dissertation outlines the following directions for future research, such as study of the proposed heuristic approaches for demand forecasting in other areas, not just in terms of currency markets; Expanding the programming system to include other methods for predicting time series, in order to more efficient comparative analysis; Expanding the capabilities of the web-based interface in the system and even performing calculations in a distributed environment based on calculations in mobile devices.

Contributions

Scientific Contributions

1. A heuristic approach is proposed for training fully-connected neural networks in a distributed environment based on incidental node participation;
2. A heuristic approach is proposed for training fully-connected neural networks in a distributed environment based on differential evolution;
3. A method for machine learning of fully-connected neural networks in a distributed environment based on the proposed heuristic approach for training with incidental node participation was developed;

4. A method for machine learning of fully-connected neural networks in a distributed environment based on the proposed heuristic approach for differential evolution training was developed;

Scientific-Applied Contributions

5. Suggested algorithms for predicting time series using fully-connected neural networks trained by a suggested heuristic approach with incidental node participation;

6. Suggested algorithms for predicting time series using fully-connected neural networks, trained with a proposed heuristic approach with differential evolution;

Applied Contributions

7. The proposed approaches, methods and algorithms for forecasting time series with data from foreign exchange markets have been investigated and experimentally evaluated.

Bibliography

1. [ANMO2011] Anatchkov, M., system visualization and manipulation of artificial neural networks, TUES to TU, 2011.
2. [BATO2011] Balabanov, T. Prediction of heuristic approaches in a distributed environment, Proceedings of Anniversary Scientific Conference 40 Years Department of Industrial Automation, ISBN: 978-954-465-043-8, 163-166, 2011.
3. [BATO2015] Balabanov, T., avoiding local optimum population in heuristic algorithms in a distributed environment, Proceedings of the XXII International Symposium Management of energy, industrial and environmental systems, ISSN: 1313-2237, 83-86, 2015.
4. [BAGE2016a] Balabanov, T., Genova, K., AJAX distributed system for training artificial neural networks with evolutionary algorithms, Proceedings of the XXIV International Symposium Management warm energy facilities and systems, energy management, industrial and environmental systems, ISSN: 1313 -2237, 49-52, 2016.
5. [BAKE2016] Balabanov T., Keremedchiev D., Goranov I., Web Distributed Computing For Evolutionary Training Of Artificial Neural Networks, International Conference InfoTech-2016, Varna - St. St. Constantine and Elena resort, Bulgaria, ISSN:1314-1023, 210-216, 2016.
6. [TBIZ2015] Balabanov, T., Zankinski, I., Barova, M., Distributed Evolutionary Computing Migration Strategy by Incident Node Participation. Large-Scale Scientific Computing, Lecture Notes in Computer Science, 9374, Springer International Publishing Switzerland, ISBN:978-3-319-26520-9, DOI:10.1007, p 203-209, 2015.
7. [BAZA2011] Balabanov, T., Zankinski I., Dobrinkova N., Time Series Prediction by Artificial Neural Networks and Differential Evolution in Distributed Environment, Large-Scale Scientific Computing, Lecture Notes in Computer Science, vol. 7116, ISBN:978-3-642-

29842-4, DOI:10.1007/978-3-642-29843-1_22, 198-205, 2011.

8. [BAZA2010] Balabanov T., I. Zankinski, Simeon C., Forecasting time series with artificial neural networks and differential evolution in distributed environment, preprints of IIT, IIT / WP-268B, ISSN: 1310-652X, 2010.
9. [MTMQ2015] MetaTrader 4, MetaQuotes Software Corp., (2015, October 20) <http://www.metatrader4.com/>
10. [NIRA2012] Nikolova, R. Development of a system for graphing and visual manipulation of artificial neural networks NBU Sofia, 2012.



АВТОРЕФЕРАТ НА ДИСЕРТАЦИЯ

за присъждане на образователна и научна степен “доктор” по
научна специалност “Информатика“

РАЗПРЕДЕЛЕНА СИСТЕМА ЗА ПРОГНОЗИРАНЕ НА ВРЕМЕВИ РЕДОВЕ С ЕВОЛЮЦИОННИ АЛГОРИТМИ И ИЗКУСТВЕНИ НЕВРОННИ МРЕЖИ

Тодор Димитров Балабанов

Ръководител: доц. Красимира Генова,

Научно жури:

Доц. Владимир Монов
Проф. Мария Нишева
Проф. Милена Лазарова
Доц. Александър Шикаланов
доц. Красимира Генова



**Институт по информационни и
комуникационни технологии**

Секция „Моделиране и оптимизация“

Дисертацията е обсъдена и допусната до защита на разширено заседание на секция „Моделиране и оптимизация“ на ИИКТ-БАН, състояло се на 8 септември 2017 г.

Дисертацията съдържа 135 страници, в които 25 фигури, 34 таблици и 30 страници литература, включваща 240 заглавия.

Защитата на дисертацията ще се състои на 12.12.2017 г. от 10:30 часа в зала 507 на блок 2 на ИИКТ-БАН на открито заседание на научно жури в състав:

1. доц. д-р Владимир Монов
2. проф. д-р Мария Нишева
3. проф. д-р Милена Лазарова
4. доц. д-р Александър Шикаланов
5. доц. д-р Красимира Генова

Материалите за защитата са на разположение на интересуващите се в стая 215 на ИИКТ-БАН, ул. „акад. Георги Бончев“, бл. 25А.

Автор: Тодор Димитров Балабанов

Заглавие: РАЗПРЕДЕЛЕНА СИСТЕМА ЗА ПРОГНОЗИРАНЕ НА ВРЕМЕВИ РЕДОВЕ С ЕВОЛЮЦИОННИ АЛГОРИТМИ И ИЗКУСТВЕНИ НЕВРОННИ МРЕЖИ

Общи характеристики на дисертацията

Актуалност на темата

Основен недостатък при обучението на изкуствени невронни мрежи с еволюционни алгоритми е фактът, че почти не се срещат реализации на пълно-свързани изкуствени невронни мрежи. Вторият основен недостатък е, че масово изкуствени невронни мрежи се използват на две фази - обучение и експлоатация. При проблемите, описвани с времеви редове, е характерно с напредване на времето времеви ред да нараства и е съвсем естествено изкуствени невронни мрежи да бъде преобучавани, спрямо новопостъпилата информация. Използването на точни числени методи (като най-популярният обратно разпространение на грешката) е ефективно, но относително неприложимо при изкуствени невронни мрежи с пълно-свързана топология. В литературата са представени множество хибридни модели за обучение на изкуствени невронни мрежи с точни числени методи в комбинация с методи за евристична оптимизация. Трети, макар и второстепенен проблем, е правилното избиране на подходящи точни и евристични методи за извършването на хибридно обучение. В литературата се забелязва относително слабо застъпване на възможностите изкуствени невронни мрежи да се обучават в условията на разпределена среда, така че максимално да се ползват възможностите за паралелна реализация на оптимизационните алгоритми от групата на еволюционните алгоритми и популационните алгоритми.

При търговията с финансови инструменти са важни три основни параметъра: 1. В коя посока ще се промени цената (повишение или намаление); 2. Колко време ще продължи това нарастване или намаляване; 3. Какъв да бъде обемът на дългата или късата поръчка; Тъй като търговията носи своите рискове посредниците/брокерите се налага да правят ежедневни анализи на световната икономическа обстановка. В общия случай на отговорно управление те инвестират средства на свои клиенти (физически

или юридически лица). При подбирането на инструменти в които да се реализират инвестициите, посредниците най-вече се интересуват от посоката в която ще се промени цената. Търгуващите на глобалния пазар условно се разделят на две групи „мечки“ и „бикове“. Метафората за названието на тези две групи идва от факта, че мечката напада противника си опитвайки се да го захлупи с лапите си, а бикът напада противника си опитвайки се да го промуши с рогата си отдолу нагоре. Двете групи търгуващи са в постоянно противопоставяне и по този начин влияят на пазара, така че той да повишава цените си или да намалява цените си в зависимост от това коя от двете групи надделява в конкретния момент. Вторият съществен параметър е колко дълго ще продължи едно повишение или намаляване на цената. След като посредникът е отворил своята позиция и успешно е предвидил посоката на промяна става актуален въпросът колко дълго тази позиция да бъде отворена. Отговорът на този въпрос е пряко свързан с размера на печалбата реализирана от конкретната позиция. Ако позицията бъде затворена твърде рано ще има нереализирана печалба. Ако позицията бъде затворена твърде късно е възможно акумулирана печалба да се стопи. Най-оптималният момент за затваряне на една отворена позиция е когато движението на цената от възходящо стане низходящо или обратното. Последният, трети параметър, е свързан с размера на поемания риск. Опитата на посредниците бързо ги учи, че наличните активи не се инвестират само в един финансов инструмент, а в цяла група инструменти. Разпределянето на риска се постигна с инвестиционен портфейл. Желанието на всеки търгуващ е да инвестира колкото се може по-голяма част от активите в печеливши позиции. Това което ограничава посредниците да инвестират всичко в един конкретен инструмент и една конкретна позиция е рискът да не разпознаят правилно посоката в която ще се промени цената и продължителността на съответното увеличение или намаление. Ако бъде направено погрешно решение за инвестиране (посока, продължителност, обем) могат да последват значителни загуби за съответната позиция. Проблемите при търгуващите на свободните пазари са свързани точно с прогнозиране на посоката и продължителността в промяната на цените. Решението за търгувания обем е задача от областта управление на финансов портфейл

и няма пряка връзка с прогнозирането.

Цели и задачи на дисертацията

За решаване на набелязаните проблеми в дисертационния труд се отличават следните цели: 1. Реализация на модели на пълно-свързани изкуствени невронни мрежи; 2. Реализация на алгоритми за непрекъснато обучение. Обучението да продължава и в процеса на експлоатация на изкуствените невронни мрежи; 3. Реализация на хибридна комбинация от представените в литературата точни и евристични алгоритми за обучение на изкуствени невронни мрежи; 4. Реализация на програмна система. Системата да бъде под формата на клиент-сървър приложение и да работи в среда за разпределени изчисления.

За постигането на целите са очертани група задачи. Основна задача по дисертационния труд е реализацията на модел за изкуствени невронни мрежи със C++ и JavaScript. Да се позволява работата с пълно-свързана топология мрежи. Тази задача се състои от две подзадачи - реализация на клиентско приложение и реализация на сървър приложение. Втора по важност е задачата за непрекъснато обучение на предложени пълно-свързан модел на изкуствени невронни мрежи. Обучението трябва да протича в непрекъснат 24/7 цикъл. Това означава, че трябва да са налични подходящи алгоритми за непрекъснато разпределяне на задачи от сървъра към клиентските приложения и непрекъснато обучение на локалните копия на изкуствените невронни мрежи. Трета по важност е задачата за подходящо комбиниране на алгоритмите генетични алгоритми и диференциална еволюция с алгоритми от вида обратно разпространение на грешката, така че времетраенето на обучението на изкуствени невронни мрежи да бъде намалено. Четвъртата по важност задача е свързана с техническата реализация на системата и провеждането на необходимите експерименти за потвърждаване на ефективността с която тя работи.

Списък с публикации по дисертацията

1. Балабанов Т., Занкински И., Симеонова В., Прогнозиране на времеви редове с изкуствени невронни мрежи и диференциална еволюция в разпределена среда, Работни статии на ИИТ, ИТ/WP-268В, ISSN:1310-652X, 2010.

2. Балабанов. Т., Прогнозиране с евристични подходи в разпределена среда, Proceedings of Anniversary Scientific Conference 40 Years Department of Industrial Automation, ISBN:978-954-465-043-8, 163-166, 2011.

3. Balabanov. T., Zankinski I., Dobrinkova N., Time Series Prediction by Artificial Neural Networks and Differential Evolution in Distributed Environment, Large-Scale Scientific Computing, Lecture Notes in Computer Science, vol. 7116, ISBN:978-3-642-29842-4, DOI:10.1007/978-3-642-29843-1_22, 198-205, 2011.

4. Балабанов, Т., Избягване на локални оптимуми при евристични популационни алгоритми в разпределена среда, Сборник с доклади от XXII Международен симпозиум Управление на енергийни, индустриални и екологични системи, ISSN:1313-2237, 83 - 86, 2015.

5. Balabanov, T., Zankinski, I., Barova, M.. Distributed Evolutionary Computing Migration Strategy by Incident Node Participation. Large-Scale Scientific Computing, Lecture Notes in Computer Science, 9374, Springer International Publishing Switzerland, ISBN:978-3-319-26520-9, DOI:10.1007, 203 - 209, 2015.

6. Балабанов, Т., Генова, К., AJAX разпределена система за обучение на изкуствени невронни мрежи с еволюционни алгоритми, Сборник с доклади от XXIV Международен симпозиум Управление на топло енергийни обекти и системи, Управление на енергийни, индустриални и екологични системи, ISSN:1313-2237, 49 - 52, 2016.

7. Balabanov T., Keremedchiev D., Goranov I., Web Distributed Computing For Evolutionary Training Of Artificial Neural Networks, International Conference InfoTech-2016, Varna - St. St. Constantine and Elena resort, Bulgaria, ISSN:1314-1023, 210-216, 2016.

8. Балабанов, Т., Генова, К., Разпределена система за обучение на изкуствени невронни мрежи, базирана на мобилни устройства, Proceedings of International Conference Automatics and Informatics'2016, ISSN:1313-1850, 49 - 52, 2016.

Цитирания на публикации по дисертацията

Balabanov, T., Zankinski, I., Barova, M., Strategy for Individuals Distribution by Incident Nodes Participation in Star Topology of Distributed Evolutionary Algorithms, Cybernetics and Information Technologies, Vol. 16 No 1, ISSN: 1314-4081, 2016.

1. Tomov, P., Monov, V., Artificial Neural Networks and Differential Evolution Used for Time Series Forecasting in Distributed Environment, Proceedings of International Conference Automatics and Informatics, Sofia, Bulgaria, ISSN 1313-1850, pp. 129-132, 2016.

Balabanov, T., Zankinski, I., Barova, M., Distributed Evolutionary Computing Migration Strategy by Incident Node Participation, International Conference on Large-Scale Scientific Computing, 10th International Conference, LSSC 2015, Sozopol, Bulgaria, June 8-12, ISBN 978-3-319-26519-3, pp. 203-209, 2015.

2. Tomov, P., Monov, V., Artificial Neural Networks and Differential Evolution Used for Time Series Forecasting in Distributed Environment, Proceedings of International Conference Automatics and Informatics, Sofia, Bulgaria, ISSN 1313-1850, pp. 129-132, 2016.

3. Ташо Ташев, Арсений Баканов, Радостина Ташева. Верхняя граница пропускной способности коммутатора с матричным переключателем для входящего трафика типа модифицированной модели Чанг-а. Доклади на Уиверситетската годишна научна конференция на Национален Военен Университет «В.Левски» 2016, 20-21 Октомври 2016, Велико Търново, България. Издагелски комплекс НВУ, 2016, том 2, стр.107-115. ISSN 1314-1937.

Balabanov, T., Zankinski, I., Dobrinkova, N., Time Series Prediction by Artificial Neural Networks and Differential Evolution in Distributed Environment, International Conference on Large-Scale Scientific Computing, 8th International Conference, LSSC 2011, Sozopol, Bulgaria, June 6-10, ISBN 978-3-642-29842-4, pp. 198-205, 2011.

4. Keremedchiev, D., Barova, M., Tomov, P., Mobile Application as Distributed Computing System for Artificial Neural Networks Training Used in Perfect Information Games, Proceedings of International Scientific Conference UniTech, Gabrovo, Bulgaria, ISSN 1313-230X, vol. 2, pp. 389-393, 2016.

Глава 1 - Обзор на системи за прогнозиране на времеви редове

Прогнозирането на времеви редове е сложна задача. Тя намира приложение в множество области в които по настояще усилено се работи. За създаването на прогнозиращи софтуерни системи се изискват множество знания и умения. Важността на задачите за прогнозиране е толкова голяма, че те намират приложение в почти всяка сфера от човешкия живот (температура, валежи, енергийни запаси и други). Подробен обзор за използването на изкуствени невронни мрежи, генетични алгоритми и размита логика за капиталовите пазари е представен в [SMSK2001]. Обзорът е съпроводен с кратко представяне на трите метода. Класификация на различни методи за прогнозиране на електрическото потребление е представено в [HAMN2002] [JVGM2002]. Авторите разглеждат генетични алгоритми и изкуствени невронни мрежи. Обзор за използването на изкуствени невронни мрежи, генетични алгоритми и размита логика в областта на капиталовите пазари с кратко представяне на методите е достъпен в [ARSH2003]. Обзор за използваните алгоритми в икономиката също е представен и в [EUVO2005]. Направено е представяне на възможностите за прогнозиране с изкуствени невронни мрежи и генетични алгоритми. При обзора на инструментите за прогнозиране на времеви редове с изкуствени невронни мрежи и еволюционни алгоритми е съществен аспектът за използването на съществуващи софтуерни решения описано в [RSQZ2006] и наличието на софтуерен пакети, както описанията в [CDTY2006] (пакет създаден на програмния език C, за операционна система Mac OS X и след това е преработен за операционна система Microsoft Windows). Задълбочен обзор на методите за съставяне на инвестиционен портфейл и начините по които изкуствени невронни мрежи или генетични алгоритми могат да се ползват при тази задача е представен в [MVDL2008]. Обзор на видовете изкуствени невронни мрежи се предлага в [YAHU2009]. Също така се предлагат начини за тяхното приложение в земеделието. В [MSMJ2009] се предлага обзор на видовете изкуствени невронни мрежи и приложението им в геотехническото инженерство. Споменават се практически насоки за избор на размер в скрития слой (примерно 75% от размера на

входния слой). Другата препоръка е размерът на скрития слой да бъде между средното и сумата от размера на входния и изходния слой. Споменава се и стратегия в която прогресивно се започва от малък брой елементи в скрития слой и се увеличава броят докато няма подобрене в параметрите за обучение на изкуствени невронни мрежи. Препоръките също са броят на тренировъчните примери драстично да надхвърля броя на елементите в скрития слой. В [MAWE2010] е изложен изключително подробен обзор на евристичните оптимизационни алгоритми. Описва се паралелна реализация на диференциална еволюция с централен възел. Централният възел синхронизира миграцията между останалите възли. Топологията е кръгова и мигрантите преминават само в посока към следващия пряк съсед. Всичко това се синхронизира от централния възел. На практика такава топологията се превръща в звездовидна. В [JMAМ2012] се излага обзор на пул базираните еволюционни алгоритми за реализацията на разширяеми и асинхронни изчисления в разпределена среда.

Анализ, проблеми, цели и задачи

В резултат на направения обзор могат да се формулират група проблеми. Тези проблеми с адресирани под формата на цели в дисертационния труд. Самите цели се разбиват на задачи. Решението на задачите води до постигане на поставените цели.

Основен недостатък на разгледаните решения за обучение на изкуствени невронни мрежи с еволюционни алгоритми е фактът, че почти не се срещат реализации на пълно-свързани изкуствени невронни мрежи. Вторият основен недостатък на разгледаните съществуващи решения е, че масово изкуствени невронни мрежи се използват на две фази - обучение и експлоатация. За проблемите подлежащи на описване с времеви редове е характерно, че с напредване на времето времевия ред нараства и е съвсем естествено изкуствени невронни мрежи да бъде преобучавани за новопостъпилата информация. Използването на точни числени методи (като най-популярният обратно разпространение на грешката) е ефективно, но относително неприложимо при изкуствени невронни мрежи с пълно-свързана топология. В

литературата са представени множество хибридни модели за обучение на изкуствени невронни мрежи с точни числени методи в комбинация с методи за евристична оптимизация. Трети, макар и второстепенен проблем, е правилното избиране на подходящи точни и евристични методи за извършването на хибридно обучение. В проучените информационни източници се забелязва относително слабо застъпване на възможностите изкуствени невронни мрежи да се обучават в условията на разпределена среда, така че максимално да се ползват възможностите за паралелна реализация на оптимизационните алгоритми от групата на еволюционните алгоритми и популационните алгоритми.

При търговията с финансови инструменти са важни три основни параметъра: 1. В коя посока ще се промени цената (повишение или намаление); 2. Колко време ще продължи това нарастване или намаляване; 3. Какъв да бъде обемът на дългата или късата поръчка. Тъй като търговията носи своите рискове посредниците/брокерите се налага да правят ежедневни анализи на световната икономическа обстановка. В общия случай на отговорно управление те инвестират средства на свои клиенти (физически или юридически лица). При подбирането на инструменти в които да се реализират инвестициите посредниците най-много се интересуват от посоката в която ще се промени цената. Търгуващите на глобалния пазар условно се разделят на две групи „мечки“ и „бикове“. Двете групи търгуващи са в постоянно противопоставяне и по този начин влияят на пазара, така че той да повишава цените си или да намалява цените си. Вторият съществен параметър е колко дълго ще продължи едно повишение или намаляване на цената. След като посредникът е отворил своята позиция и успешно е предвидил посоката на промяна става актуален въпросът колко дълго тази позиция да бъде отворена. Отговорът на този въпрос е пряко свързан с размера на печалбата. Ако позицията бъде затворена твърде рано, то тя ще има нереализирана печалба. Ако позицията бъде затворена твърде късно е възможно акумулирана печалба да се намали. Най-оптималният момент за затваряне на една отворена позиция е когато движението на цената от възходящо стане низходящо или обратното. Последният, трети параметър, е свързан с размера на поемания риск. Стремезът на всеки търгуващ е да инвестира

колкото се може по-голяма част от активите в печеливши позиции. Това което ограничава посредниците да инвестират всичко в един конкретен инструмент и една конкретна позиция е рискът да не разпознаят правилно посоката в която ще се промени цената и продължителността на съответното увеличение или намаление.

Ако бъде направено погрешно решение за инвестиране (посока, продължителност, обем) могат да последват значителни загуби за съответната позиция. Проблемите са свързани с прогнозиране на посоката и продължителността в промяната на цените.

Целите на дисертационния труд са както следва: 1. Да се реализират модели на пълно-свързани изкуствени невронни мрежи; 2. Да се реализират алгоритми за непрекъснато обучение, дори в процеса на експлоатация на изкуствените невронни мрежи; 3. Да се реализира хибридна комбинация от представените в литературата точни и евристични алгоритми за обучение на изкуствени невронни мрежи; 4. Да се реализира програмна система (клиент-сървър приложение) работеща в среда за разпределени изчисления.

За постигането на набелязаните цели са формулирани серия задачи. Основна задача по дисертационния труд е реализацията на модел за изкуствени невронни мрежи, под формата на C++ и JavaScript, като пълно-свързана топология. Тази задача се състои от две подзадачи - реализация на клиентско приложение и реализация на сървър приложение. Втора по важност е задачата за непрекъснато обучение на предложения пълно-свързан модел на изкуствени невронни мрежи. Обучението трябва да протича в непрекъснат 24/7 цикъл, като за тази цел трябва да са налични подходящи алгоритми за непрекъснато разпределяне на задачи от сървъра към клиентските приложения и непрекъснато обучение на локалните копия на изкуствените невронни мрежи, които клиентските машини поддържат. Трета по важност е задачата за подходящо комбиниране на алгоритми като генетични алгоритми и диференциална еволюция с алгоритми, като обратно разпространение на грешката, така че времетраенето на обучението на изкуствени невронни мрежи да бъде намалено. Четвъртата по важност задача е свързана с техническата реализация на системата и

провеждането на необходимите експерименти за потвърждаване на ефективността с която тя работи.

Глава 2 - Модел за прогнозиране на времеви редове с изкуствени невронни мрежи, обучавани с еволюционни алгоритми

Настоящото изложение представя модел за прогнозиране на времеви редове, базиран на изкуствени невронни мрежи, обучавани с алгоритъм за диференциална еволюция, в разпределена среда. Сложността на задачата за прогнозиране на времеви редове изисква разработването на по-бързо действащи алгоритми, които водят до прогнози с по-голяма точност. В основата на изкуствените невронни мрежи за прогнозиране е заложена идеята за обучение, на база последователност от данни за изминал период. Изборът на обучаващ алгоритъм може да бъде направен от две основни направления - точни оптимизационни числени методи и евристични оптимизационни метод. Прилагането на подходящи евристични алгоритми може да доведе до използването им в разпределена среда и значително ускоряване на процеса за обучение.

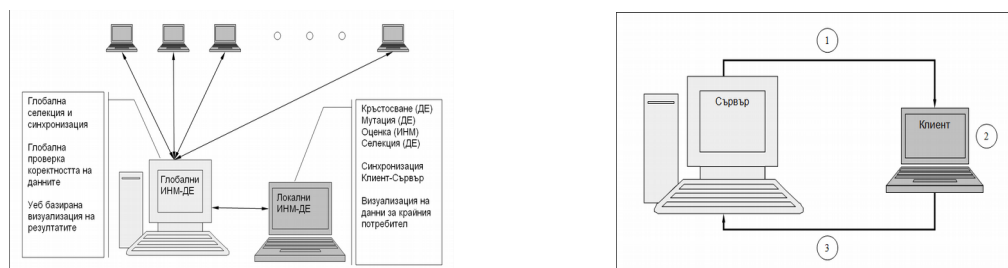
Прогнозирането на времеви редове е задача, в която по зададени хронологични данни, трябва да се направи прогноза за данни в предстоящ период от времето. Когато става въпрос за финансови инструменти дял от науката, наречена „Иконометрия”, се занимава с разработка на алгоритми, финансови индикатори и осцилатори, с помощта на които да се анализират данните. Създаването на ефективни и надеждни средства за финансово прогнозиране е трудоемка и сложна задача. Поради тази причина много актуална е областта за разработка на самоорганизиращи се или самообучаващи се системи за прогнозиране.

Приложението на изкуствени невронни мрежи при прогнозирането на времеви редове в икономиката се разглежда от различни автори, някои от които са Dunis [DUWI2002], Giles [GILA2001] и Moody [MOOD1995]. Като основен модел се използват така наречените Feed Forward Neural Networks (виж Haykin [HAYK1999]).

Мрежите от тип FFNN са много ефективни, но страдат от основен недостатък, свързан с липсата на кратковременна памет. Избягването на този проблем се осъществява с използването на така наречените Recurrent Neural Networks. Основната трудност при RNN идва от невъзможността за тяхното обучение да се използват точни градиентни методи (виж Werbos [WERB1990]). Комбиниран подход за обучение на изкуствени невронни мрежи с помощта на еволюционни алгоритми е предложен от Yao [YAOX1999] и други автори. Еволюционните алгоритми дават значително по-добри резултати за търсене на оптимуми в сложни многомерни пространства с наличие на множество локални оптимуми, в които градиентните методи не дават добри резултати (виж Holland [HOLL1975]).

Предложена система за прогнозиране

В дисертационния труд е предложено за целите на прогнозирането да се използва математически модел базиран на изкуствени невронни мрежи, чието обучение се извършва с диференциална еволюция, под формата на изчисления в разпределена среда. Топологията на изкуствените невронни мрежи представлява изследователски интерес и за това в софтуерната система топологията се определя от оператор на отдалечения сървър. Допустимите видове топологии са: 1) Многослойни; 2) Многослойни с обратна връзка; 3) Пълно-свързани. Предложеният модел е базиран на класическа мрежа, каквито се използват в моделите с обратно разпространение на грешката. Като сумираща функция е прието да се използва линейна функция.



Фиг. 2.1 Система за обучение на ИИМ с ДЕ, в разпределена среда (ляво) и обмен на задачи в системата за ИПС (дясно).

Обучението на изкуствени невронни мрежи се осъществява с помощта на диференциална еволюция. В рамките на алгоритъма диференциална еволюция всяка хромозома представя един комплект тегла за конкретна топология изкуствени невронни мрежи, към която се отнасят определен брой обучаващи примери. Последователно, отделните хромозоми (комплекти тегла) се зареждат в структурата на изкуствената невронна мрежа, след което се подават обучаващите примери, изчислява се грешката, която изкуствената невронна мрежа допуска за всеки пример, сумират се грешките от отделните примери и се определя коефициент за жизнеспособност на хромозомата (комплект тегла). Начинът, по който се определя коефициентът на жизнеспособност, е един от най-ключовите проблеми, от които зависи успехът на цялостното прогнозиране. При времевите редове е необходимо обучаващите примери да се подават в хронологична последователност, което създава затруднения при изкуствени невронни мрежи, обучавани с алгоритъм за обратно разпространение на грешката. Също така, отдалечените във времето обучаващи примери е необходимо да оказват по-малко влияние при формирането на коефициента за жизнеспособност. След като бъдат оценени отделните хромозоми те влизат в изчислителната схема на диференциалната еволюция като над тях се извършва селекция, кръстосване и мутация.

При системите за изчисления в разпределена среда централизиран сървър разпределя задания за изчисление на отдалечени компютри (Фиг. 2.1 - стъпка 1). В някакъв период от време отдалечените компютри изпълняват заданието (Фиг. 2.1 - стъпка 2). Заданията могат да са оформени като пакети за пресмятане в рамките на няколко минути, но може да са оформени и в пакети за пресмятане в рамките на седмица или месец. Периодът за продължителност на пресмятането силно зависи от естеството на изчисленията, които трябва да се извършат. Чрез подходящо регулиране на периода за пресмятане може да се постигне оптимално натоварване на сървъра и максимална актуалност на информацията, която сървърът съхранява, с цел синхронизиране на разпределения изчислителен процес. Когато клиентът приключи изчислението на възложеното задание, връща резултата на сървъра (Фиг. 2.1 - стъпка

3). Трите основни стъпки в този процес се повтарят, докато има работещ сървър и клиентски компютри, закачени за него.

При паралелния вариант на алгоритъма за диференциална еволюция се създават локални копия на изкуствената невронна мрежа и на диференциалната еволюция [BAZA2010]. Обучението протича локално, като всяка хромозома в популацията обозначава отделна точка (индивид) в пространството на търсене. Макар и различни, индивидите се групират относително близко един до друг в това пространство. При разделяне на изчисленията върху множество изчислителни машини, подобно групиране на индивиди може да доведе до локално изследване на различни области от пространството на решенията (пространството на търсене) [BATO2011]. В този аспект най-съществена е политиката за синхронизиране, като процеса на синхронизиране включва излъчването на най-добрите индивиди и събирането им на едно централизирано място (при клиент-сървър архитектура). Това множество на глобално най-устойчивите индивиди може да се използва при създаването на следващи локални ареали. Освен клиент-сървър архитектура е възможно да се реализира peer-to-peer решение, при което отсъства централизиран сървър, а всяко локално работещо приложение се грижи за комуникацията с други локално работещи приложения. Основно предимство на предложената клиент-сървър система за изчисления в разпределена среда е нейната изключително висока степен на скалируемост (изключително улеснено включване на допълнителни компютърни системи в системата за разпределени изчисления) [BAZA2011]. Честотата на комуникация между клиентите и сървъра може да се регулира прецизно, като за интервали може да се ползват часове, седмица и дори месеци. При наличие на голям брой клиенти, натоварването на сървъра може да се компенсира с намаляване честотата на комуникация с клиентите.

Първото предимство на предложения модел се състои в използването на диференциална еволюция за обучението на изкуствени невронни мрежи, като по този начин се избягва опасността от преобучение [BATO2015]. Второто предимство е възможността, чрез диференциална еволюция, да се обучават изкуствени невронни

мрежи с обратни връзки. Третото предимство е възможността, чрез диференциална еволюция, да се обучават изкуствени невронни мрежи, без да е от значение редът по който се подават обучаващите примери. Четвъртото предимство е възможността да се обучават различни копия на изкуствени невронни мрежи и това да става паралелно, водещо до подобряване на бързодействието и по-добро покритие на пространството за търсене. При евентуални злонамерени клиенти, които изпращат некоректна информация към сървър (комплекти тегла), то тази информация може да се провери лесно. От друга страна, некоректната информация, получена от злонамерени клиенти, може да се използва в неподправен вид за обогатяване генетичния фонд на сървър. Реално, злонамерените клиенти не представляват проблем за предложения модел, дори ако не се вземат специални мерки за борба с тях.

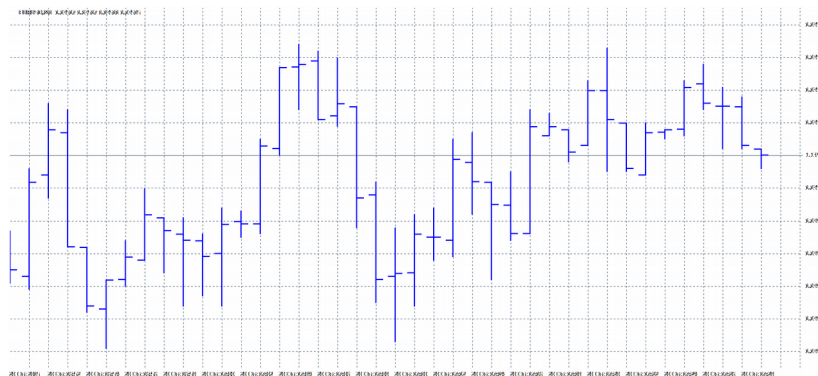
Освен предимства предложеният модел има и следните недостатъци: Използването на изкуствени невронни мрежи е свързано с бавно обучение. Макар и много ефективни, изкуствените невронни мрежи изискват големи количества изчислителни ресурси, основно във фазата на обучение. Използването на диференциална еволюция значително забавя процеса на обучение, в сравнение с алгоритъма за обратно разпространение на грешката, но поради множеството изброени предимства диференциална еволюция е предпочитан пред обратно разпространение на грешката. Интересен би бил модел в който диференциална еволюция се комбинира с обратно разпространение на грешката, макар и това да не може да се постигне без някакъв компромис по отношение топологията на изкуствените невронни мрежи. Разработването на програми за изчисления в разпределена среда е по-сложно от писането на строго последователни или паралелни програми.

Глава 3 - Софтуерна система за прогнозиране на времеви редове с изкуствени невронни мрежи и еволюционни алгоритми

Основен фактор при практическата реализация е наличието на подходящи експериментални данни. Развитието на финансовите пазари, през последните две

десетилетия, дава изключително добър избор от времево зависима информация. Цената на националните валути се оповестява на така наречения *foreign exchange market* (FOREX). Търгуването на валутните пазари изключително се улеснява с услугите, които предлагат инвестиционните посредници. Самата търговия се извършва с помощта на търговски софтуер наречен търговски терминал. За Източна Европа най-разпространената търговска платформа е *MetaTrader 4*, произведена от фирмата *MetaQuotes Software Corp.* [MTMQ2015].

В платформата *MetaTrader 4* валутната информация е представена под формата на времеви ред, онагледен графично с нива на отваряне, нива на затваряне, нива за най-висока достигната цена и нива на най-ниска достигната цена (Фиг. 3.1). Цялата тази информация е програмно достъпна със средствата на вградения в *MetaTrader 4* програмен език, наречен *MQL4*. Търговският терминал на *MetaTrader 4* дава възможност за визуализиране на информацията от времевия ред на няколко различни, равни по дължина, времеви интервала, като следва: *M1* - всеки бар отразява една минута, *M5* - всеки бар отразява пет минути, *M15* - всеки бар отразява петнадесет минути, *M30* - всеки бар отразява тридесет минути, *H1* - всеки бар отразява един час, *H4* - всеки бар отразява четири часа, *D1* - всеки бар отразява един ден, *W1* - всеки бар отразява една седмица, *MN* - всеки бар отразява един месец. От ляво, на всеки бар (Фиг. 3.1) присъства хоризонтална черта, която символизира нивото при което барът е бил отворен. От дясно на всеки бар има втора хоризонтална черта, която символизира нивото на което е бил затворен. Долният край на всеки бар показва най-ниското ниво, постигнато в рамките на времевия интервал, за който се отнася барът. Горният край на всеки бар показва най-високото ниво, постигнато в рамките на времевия интервал, за който се отнася барът.



Фиг. 3.1 Котировки EUR/USD в MetaTrader 4.

Програмните средства на MetaTrader 4, чрез вградения език MQL4, позволяват котировките за всяка валутна двойка да бъдат предоставени извън рамките на търговския терминал. Извеждането на информацията може да стане по няколко различни начина, като най-удачен за дисертационния труд е записът на информацията в текстов файл. Преди информацията за всеки бар от графиката с котировките да бъде изведена от търговския терминал информацията се нормализира. На графиката се определя най-високото постигнато ниво, след това най-ниското постигнато ниво и стойностите на всеки бар се нормализират в диапазона 0.0-1.0. Тази стъпка за нормализация се налага, тъй като използваната изкуствена невронна мрежа работи със сигнали нормализирани в диапазона 0.0-1.0. Втората причина за извършването на нормализация е наличието на множество валутни двойки, които имат различни номинални стойности. Примерно, отношението EUR/USD е от порядъка на 1.13510 (информация към дата 20.10.2015), докато отношението USD/JPY е от порядъка на 119.830 (информация към дата 20.10.2015). След като изкуствената невронна мрежа предложи своята прогноза се извършва обратната операция по мащабиране на изхода, спрямо минималната и максималната стойност на графиката.

По своята същност, MetaTrader 4 е търговски терминал със софтуерна архитектура от тип „клиент-сървър“. Търгуващите клиенти (хора, наречени на професионален жаргон трейдър) получават котировките на валутните двойки от централизиран сървър, управляван от инвестиционния посредник. Тази организация позволява на компютъра, на всеки трейдър да се стартира допълнителен модул за пресмятания (модул предложен в дисертационния труд), който да извършва

изчисления по обучението на изкуствени невронни мрежи. Предложеният изчислителен модул представлява стандартен изпълним файл, който прочита котировките, записани от MQL4, в текстовите файлове. Обучението на конкретна топология изкуствени невронни мрежи, за конкретни котировки, на конкретен времеви интервал може да се разпредели между множество компютри.

Развойни средства

Тъй като системата е разделена на няколко логически обособени части, за всяка част са подходящи различни развойни инструменти.

За разработката на програмния код се използват група инструменти за разработка. От страна на MetaTrader 4, естественият програмен език е вградения в търговската платформа MQL4. Малък софтуерен под модул, написан на MQL4, има за задача да извежда информацията за времевите редове в текстови файлове и да чете прогнозите генерирани от изкуствените невронни мрежи. Също така, MQL4 под модулята има задача да визуализира прогнозата в подходяща форма за разчитане от крайните потребители.

По отношение на разработения в този дисертационен труд софтуерен модул, съществуват набор от алтернативи за избор на програмен език. Най-популярните програми езици са: C/C++, Java, C#. За разработката е избран C/C++, като аргументите в негова полза са, че това е програмен език, който не е доминиран от конкретна корпоративна фирма, има изключително добра поддръжка на различни операционни системи (MS Windows, Mac OS X, Linux). Изборът на C/C++ е аргументиран и с това, че езикът има добра поддръжка за супер компютри (Open MPI), както и за графични ускорители от тип GPU. Основна характеристика на C/C++ е относителното бързодействие с което се изпълнява програмния код, тъй като се свежда до асемблера на конкретната хардуерна архитектура. Не на последно място, C/C++ е основният език в платформата BOINC (Berkeley Open Infrastructure for Network Computing), която е една от най-разпространените платформи за изчисления в разпределена среда.

Придържайки се към концепцията за пресмятания в BOINC е възможно да се създадат собствени решения за пресмятане в разпределена среда, както е описано в [SINI2011] [HREL2011].

Тъй като основната идея за пресмятане в разпределена среда е изчисленията основно да се извършват на клиентските машини, то за сървър е удачно да се подберат най-олекотените технологии и инструменти за разработка. Съобщенията, разменяни между клиентските компютри и сървъра са в структуриран формат JSON. След 1992 година един от най-широко разпространените комуникационни протоколи е HTTP и това е причината той да бъде избран в дисертационния труд. За обработката на HTTP заявки от страната на сървъра е необходимо постоянно да работи софтуерно приложение наречено уеб сървър. Най-популярните възможности за уеб сървър са Apache, Tomcat и Microsoft IIS. С цел намаляване на финансовите разходи от страна на сървъра, за нуждите на дисертационния труд, е избран уеб сървър Apache. Уеб сървърът Apache може да се инсталира на множество операционни системи, тъй като представлява софтуер с отворен код.

Изборът на Apache уеб сървър е обвързано и с избора на програмен език от страна на сървъра. Най-разпространените програмни езици за сървърни приложения са PHP, JSP и ASP. В дисертационния труд е избран PHP, тъй като е език с основно предназначение за разработка на сървърни скриптове.

Слоят за трайно съхранение на данните, от страна на сървъра, е представен от релационна система за управление на бази от данни (СУБД). Най-популярните възможни алтернативи са Oracle, MS SQL Server, PostgreSQL и MySQL. Изборът е на MySQL защото е значително по-лесна за използване в сравнение с останалите. Компанията Oracle притежава правата над MySQL, което е своеобразна гаранция за стабилното поддържане на тази СУБД.

Моделiranje на структурите от данни в системата

Програмният код, работещ от страната на клиентските компютри, представлява

група класове, компилиращи се до самостоятелен изпълним файл. От класовете, в паметта на компютъра, се създават обекти. По време на работа, на системата вътрешното състояние на обектите се променя. Тази промяна на обектите се съхранява в енергийно независима памет, под формата на релационна база данни. За целта се извършва обектно-релационно напасване. За онагледяване на обектно-ориентираната структура, всеки клас е представен със CRC карта (Class Responsibility Collaboration).

ANN е основният клас в програмата и описва изкуствена невронна мрежа, което включва вътрешно представяне на мрежата и алгоритми за обработка на постъпващата и излизащата информация. ANNIO е за входно-изходната информация и представлява вектор от реални числа, а който е предназначен отделен базов клас. ANNInput отговаря за входната информация и е вектор от реални числа и представлява част от входно-изходен вектор а ANNOutput за изходната информация и е вектор от реални числа и представлява част от входно-изходен вектор. В ActivityMatrix е представена матрицата за активностите на връзките в изкуствените невронни мрежи и е двумерен масив от реални числа, който определя матрицата на съседство. Използва се за определяне на силата, на връзките в изкуствените невронни мрежи, като по този начин определя топологията, която мрежата има. Chromosome описва хромозомите, които са обекти, които изграждат популацията на диференциалната еволюция. Всяка хромозома представлява комплект тегла за изкуствена невронна мрежа и има допълнителна характеристика за жизнена стойност на хромозомата. Communicator е за комуникацията с отдалечения сървър. Поверена е на група класове, които са организирани в йерархия от наследявания. Този подход е избран за да бъде възможно разширяване на системата с допълнителни комуникационни протоколи. Communicator е базов клас, наследен от по-специфични реализации на комуникационната функционалност. В Counter за да се събира статистика по време на работа за системата и е представен от контейнер с броячи. На база, на информацията от броячите се прави статистическа оценка за ефективността на обучението и точността на прогнозите. С CrossoverType в алгоритъма за диференциална еволюция могат да се приложат различни типове кръстосване, което често зависи от избраната топология на

изкуствената невронна мрежа. Алгоритъмът за диференциална еволюция е реализиран в отделен клас DE, който има отговорност да извършва селекцията, рекомбинацията и оценката на отделните индивиди в популацията. Тъй като теглата и активностите на връзките в изкуствените невронни мрежи се моделират с матрица на съседство е удачно залагането на базов клас GraphMatrix, който да бъде наследен. Един от най-разпространените комуникационни протоколи е HTTP, което е основна аргументация в системата да бъде предоставена възможност за обмен на съобщения, между клиента и сървъра HttpCommunicator, точно на основата, на HTTP протокола. При зареждане на системата, без наличие на връзка към сървъра е важно параметрите за обучение на изкуствените невронни мрежи да бъдат инициализирани и поради тази причина е въведена възможността инициализацията да се извърши от инициализационен файл Init. При комуникация по протокола HTTP е възможно информацията да бъде предавана в текстов вид, без да бъде структурирана, но много по-ефективно и защитено от грешки е, ако информацията бъде пакетизирана в JSON съобщения. В този аспект е добавена възможност за комуникация между клиента и сървъра на база HTTP+JSON в JsonHttpCommunicator. Параметрите на всеки модел (ИНМ+ДЕ+ВР) се обработват от помощен клас ModelParameters, така че методите да получават само един обект, а не серия от параметри. Отделен клас Neuron моделира поведението на невроните. Методите на този клас имат задача да извършват пресмятанията и да запазват вътрешното състояние на неврона. Основните типове неврони, описани в NeuronType, биват - обикновени, входящи, изходящ и bias (отместващи). Според своя тип, всеки неврон изпълнява различна роля в мрежата. Възможно е неврони да съчетават повече от една функция (примерно неврон, който е входен, а в същото време и изходен за мрежата). Всяка мрежа е представена неврони и тегла за връзките между тях. В паметта на компютъра се използва списък от неврони NeuronsList, като контейнер за обектите от тип неврон. Популацията е представена в паметта на компютъра с клас Population като контейнер от обекти тип хромозома. Времеви ред за стойността на валутните двойки се представя под формата на графика с барове и клас RateInfo. Всеки бар носи информация за котировката на валутите в конкретен момент

от времето. В програмата MetaTrader4 времевите редове са организирани на равни интервали от време. Програмата дава възможност финансовата информация да бъде представяна във времеви редове с различен период, между отделните отчитания. Времеви периоди са изброен тип TimePeriod както следва: M1 (една минута), M5 (пет минути), M15 (петнадесет минути), M30 (тридесет минути), H1 (един час), H4 (четири часа), D1 (един ден), W1 (една седмица), MN1 (един месец). Множеството от обекти за изкуствена невронна мрежа, диференциалната еволюция, обучаващо множество, комуникационен обект и броячи се помества в обект от класа за трениране на мрежата. Задачата на клас Trainer е да служи като контейнер от високо ниво, за останалите обекти. Тренировъчният пример е представен с клас TrainingExample и е стойност на невроните, както следва: входни сигнали в изкуствени невронни мрежи, изходни сигнали от изкуствени невронни мрежи и очаквани изходни сигнали от изкуствени невронни мрежи. Обучаващото множество е клас TrainingSet и е последователност от три паралелни масива, както следва: вход в изкуствени невронни мрежи, изход на изкуствени невронни мрежи и очаквана стойност на изхода, на изкуствени невронни мрежи. Матрицата за теглата на връзките в изкуствена невронна мрежа представена от WeightsMatrix е двумерен масив от реални числа, който определя матрицата на съседство. Използва се за определяне на силата, на сигналите между невроните в изкуствени невронни мрежи, като по този начин определя знанието, което мрежата има.

Съхранението на информацията в системата се извършва на централизирания сървър с помощта на СУБД, в която данните са моделирани под формата на същности и релации. В термините на СУБД същностите и релациите се представят под формата на таблици в базата данни.

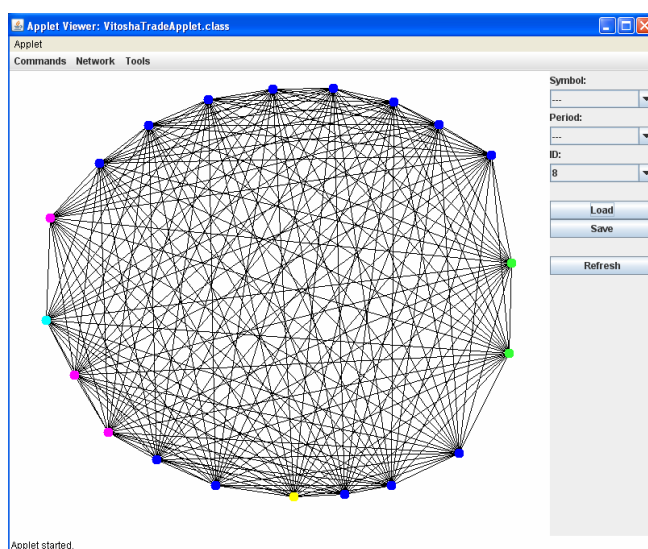
В таблица app всяка изкуствена невронна мрежа е описана с матрица на съседство (полето wights), която съдържа теглата на връзките в мрежата. Различните мрежи имат различен комплект тегла и съответно биха дали различна точност при прогнозирането. Общата допусната грешка при прогнозирането се използва за жизнена стойност в алгоритъма за диференциална еволюция (полето fitness). Различните

изкуствени невронни мрежи се характеризират с тип на мрежата и това се отразява с помощна таблица, връзката към която е полето `ann_kind_id`. В таблица `ann_kind` видовете изкуствени невронни мрежи основно се описват с топология на възлите и силата на връзките между отделните възли (полето `activities`). Полето `activities` също представлява матрица на съседство и определя дали между два възела (включително и примките) съществува връзка и колко силна е тази връзка. Всеки вид изкуствена невронна мрежа има определен брой неврони (полето `number_of_neurons`) и информация какъв тип е всеки от невроните (полето `flags`). Тъй като не се търсят връзки между отделни валутни двойки, всеки вид изкуствена невронна мрежа се отнася само за конкретна валутна двойка и конкретен период на времевия ред (външна връзка през полето `currency_pairs_id`). Чрез таблица `currency_pairs` всеки времеви ред в системата отразява информацията за една валутна двойка. Валутните двойки се характеризират със символно название (примерно EURUSD, полето `symbol`). Втората характеристика на времевия ред е периодът (външна връзка през полето `period_id`). В таблица `neurons_coordinates`, за целите на системната администрация, всяка изкуствена невронна мрежа се визуализира в административен панел. На практика се изобразяват невроните, като възли в граф, и теглата като ребра на графа. За да се осъществи визуализацията, всеки неврон се описва с координати в двумерната равнина (полето `coordinates`). В програмата MetaTrader4 са заложили няколко стандартни времеви интервала, за времевите редове, отразени в таблица `time_periods`. Интервалите се описват със символно название (полето `period`) и брой минути, на които отговаря интервалът (полето `minutes`). Чрез таблица `training_options` системата би могла да определя параметри за обучението на изкуствени невронни мрежи с отделна таблица в реляционната база данни. Като параметри на обучението се използват броят индивиди в диференциалната еволюция, броят барове за вход в мрежата, броят барове за изход от мрежата и кой вид мрежа ще се обучава. От таблица `training_set` тренировъчните примери за изкуствената невронна мрежа представляват самият времеви ред, който постъпва в системата от изхода на програмата MetaTrader4. Времевият ред съдържа информация за валутната двойка, периода на времевия ред, колко стойности от

времеви ред са представени и група паралелни масиви, които имат следното значение: time - момент във времето за който се отнася стойността, open - ниво на отваряне, за съответния интервал, close - ниво на затваряне, за съответния интервал, low - най-ниско постигнато ниво, за съответния интервал, high - най-високо ниво, постигнато за съответния интервал, volume - изтъргуван обем в конкретния времеви интервал.

Административен модул за наблюдение и настройване на системата

Модулът за наблюдение и настройване на системата се ползва за задаване на параметри, при които протича обучението на изкуствени невронни мрежи. Модулът е разработен под формата на Java Applet, за да бъде достъпен от различни компютърни системи, използващи стандартен уеб браузър. Модулът е реализиран под формата на двуслойна софтуерна архитектура (потребителски интерфейс - база данни). Системата за сигурност (автентификация и авторизация) е изцяло базирана на системата за сигурност, която предоставя СУБД. Административният модул е предназначен единствено за операторите на системата, които са в ролята на доверени лица. Допълнително, нивото на сигурност в СУБД ограничава достъпа до данните само в рамките на локалната мрежова инфраструктура.



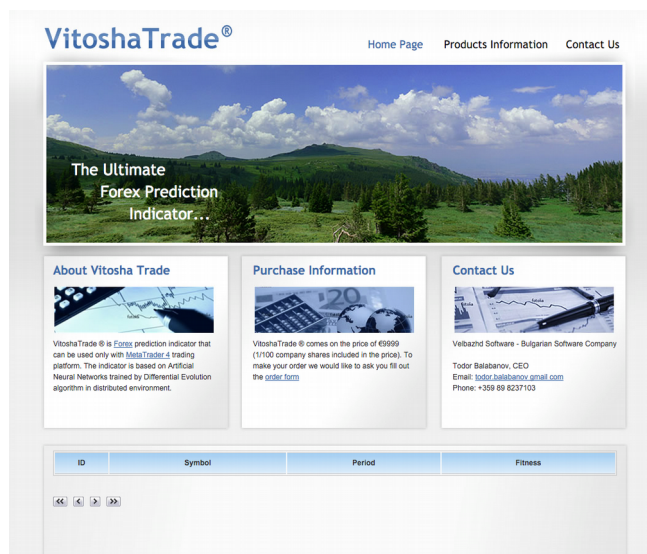
Фиг. 3.2 Екран за визуализация на ИИМ [NIRA2012][ANMO2011].

Най-съществената възможност на административната система е визуализирането

на изкуствени невронни мрежи с всички връзки, както под формата на тегла, така и под формата на активности. Невроните в мрежата се изобразяват с различни цветове, спрямо ролята, която изпълняват в мрежата (входни, изходни, вътрешни или отместващи). Разположение на невроните върху визуалното пространство може да се променя, с помощта на мишката и клавиатурата (Фиг. 3.2).

Публичен модул за информация от системата

Публичният модул представлява уеб сайт, разработен с HTML, CSS, PHP. Целта на публичния модул е да представя информация от системата, която е за публично ползване. Публичната информация включва списък на различните изкуствени невронни мрежи в СУБД, различната ефективност до която отделните мрежи са обучени и техния служебен идентификатор в рамките на СУБД (Фиг. 3.3).



Фиг. 3.3 Основна уеб страница за публикуване на публично достъпна информация от системата.

Цялостната система включва различни модули, част от които са достъпни за публично ползване. Списък с публично достъпните модули е представен в допълнителна уеб страница. Също така публичният модул съдържа и уеб страница с информация за контакт.

Глава 4 - Приложение на системата за прогнозиране и анализ на експериментални резултати

Сложността при прогнозирането на валутните пазари се дължи на динамиката в изменението на пазарните процеси и множеството фактори, оказващи влияние върху цената. Предложеният теоретичен модел и конкретната му софтуерна реализация ще послужат за изследване на проблемите, разгледани по-долу в текста.

При нарастване броя на възлите в изкуствените невронни мрежи, броят на теглата се увеличава значително, което може допълнително да затрудни процеса на обучение, особено при по-сложна топология на изкуствени невронни мрежи. Първото направление за изследване е каква топология на изкуствени невронни мрежи би била най-ефективна за целите на прогнозирането. Разгледани са класическа трислойна изкуствена невронна мрежа, многослойна изкуствена неврона мрежа с обратни връзки и пълно-свързана изкуствена невронна мрежа. Освен броят слоеве и конкретният набор връзки между тях от съществено значение е и общият брой неврони.

Съществен момент в изследването е изборът на вида на входно-изходните данни при работа с изкуствени невронни мрежи. По отношение на изхода, желателно е да се получава директно прогнозна стойност (конкретно число или разлика на прогнозирания момент, спрямо текущия). По отношение на входните данни са на лице следните възможности: 1. Да се подава стойност за време; 2. Да се подава стойност на величината от предходен момент; 3. Да се подава съвкупност от стойности на величината в предходни моменти; 4. Да се подават разлики в стойността на величината от предходни моменти; 5. Да се подава комбинация от предходни стойности на величината и информация за времето.

Освен с данни за време и стойност на величина, изкуствените невронни мрежи може да се обучава за класифициране на „истински” и „фалшиви” сигнали, подавани от финансови индикатори и/или осцилатори. При този вариант самото прогнозиране се получава от комбинация индикатор/осцилатор и изкуствени невронни мрежи, която потвърждава достоверността на подаваните сигнали.

Инцидентно участие на възли в система за разпределени изчисления

В предложената система за разпределени изчисления, основната популация се намира на един централен възел (сървър). Всеки клиент се закача към системата и получава подмножество от глобалната популация (Фиг. 4.1 - ляво). След тази първоначална инициализация оптимизацията протича на локално ниво, като към централния възел се изпращат само индивиди от локалната популация, които имат по-добра жизнена стойност спрямо вече известната най-добра глобална жизнена стойност.



Фиг. 4.1 Инцидентно включване на възли в разпределената система (ляво) и сравнение между инцидентно включване на възли и кръгова топология (дясно).

На практика, централният възел служи за хранилище на индивидите с най-добра постигната жизненост от локалните процедури за оптимизация (своеобразна форма на елитаризъм). Тази част от системата, за обмен на индивиди между отделните изчислителни възли е разгледана, в изолиран вариант в [TBIZ2015], като е направено сравнение между възли с инцидентно включване и изключване, и обмен на информация по кръгова топология.

От проведените експерименти се вижда, че при инцидентно включване на възлите сходността на оптимизационният процес е по-добра (Фиг. 4.1 – дясно).

Сравнение между C++ и JavaScript, при правия пас на ИИМ

В настоящото изследване [BAGE2016a] е представен модел за обучение на

изкуствени невронни мрежи с помощта на обучаващ алгоритъм - диференциална еволюция, който е от групата на еволюционните алгоритми. Обучението на изкуствени невронни мрежи се осъществява в разпределена среда, като мрежата е представена във вид на JavaScript програмен код, а комуникацията с централния възел се извършва с помощта на AJAX асинхронни заявки. Целта за обучение на изкуствени невронни мрежи е прогнозиране на стойностите за различни валутни двойки, на глобалния валутен пазар.

Бързодействието, при изпълнението на JavaScript, е основният аргумент езикът да се избягва при извършването на големи по обем изчисления или при изчисления, които изискват голяма точност. JavaScript попада в групата на скриптовите езици, чийто код не се компилира до инструкции за процесора, а се интерпретира от програмни модули наречени интерпретатори.



Фиг. 4.2 Сравнение в бързодействието при разпространение на информацията в ИНМ, по време на правия пас, между C++ и JavaScript. По оста X са броят неврони, а по оста Y е бързодействието в милисекунди (ляво) и стандартно отклонение по време при разпространение на информацията в ИНМ, по време на правия пас, между C++ и JavaScript. По оста X е броят неврони, а по оста Y е стандартното отклонение (дясно).

Във всеки един момент интерпретаторът може да прекрати изпълнението на програмния код и поради тази причина не може да се гарантира достатъчна надеждност на изчисленията. За да се провери разликата в бързодействието на предложеното AJAX-JavaScript решение са извършени серия експерименти, с разпространение на информацията в изкуствена невронна мрежа при правия пас. Алгоритъмът за правия пас е кодиран в два отделни модула на системата VitoshaTrade - съответно модул на C++ и модул на JavaScript. Както е видно от Фиг. 4.2 (ляво), бързодействието е съизмеримо, за мрежи от 10 до 100 неврона. Всеки експеримент е

изпълнен 30 пъти и средните стойности са представени на фигурата [BAKE2016].

По отношение на бързодействието езиците C++ и JavaScript са съизмерими, но устойчивостта на изчислителния процес е по-добре при C++, което е видно на Фиг. 4.2 (дясно), която отразява средното квадратично отклонение на времето нужно за изчисление. Тази разлика се дължи основно на наличието, на интерпретатор и уеб браузър, нещо което не присъства при изчисленията в C++. При езици от категорията на C++ програмата първоначално се транслира до асемблерен език, а след това и до машинен код.

Реализацията на изчисления в разпределена среда, под формата на AJAX уеб базирана система води до много висока степен на разширяемост в системата. Практически, разпределените изчисления могат да се стартират на всяко устройство поддържащо съвременен уеб браузър, способен да стартира JavaScript и AJAX. Тъй като изчислението се извършва в рамките на уеб браузъра, който от своя страна е процес в адресното пространство на операционната система, а тя от своя страна се изпълнява на физическия хардуер, макар и съизмерими по бързодействие, изчисленията са по-ненадеждни (наличие на интерпретатор), отколкото биха били при реализация на езици, като C/C++ или Assembler.

Сравнение между различни топологии на ИНМ

При прогнозирането на времеви редове с изкуствени невронни мрежи двата най-актуални модела мрежи са класическа трислойна мрежа, без обратни връзки и мрежа с обратни връзки. Проведените експерименти са с трислойни мрежи, които имат различен размер на скрития слой и пълно свързани мрежи (всеки възел е свързан с всеки останал, включително и със себе си). При пълната свързаност се цели ефект на кратковременна памет в мрежата.

За сравнението между пълно-свързана и трислойна мрежа е избрана трислойна топология 28-12-5. Пълно свързаната мрежа се състои от 45 възела. И двата вида мрежи са обучени с генетични алгоритми при следните параметри за генетичния

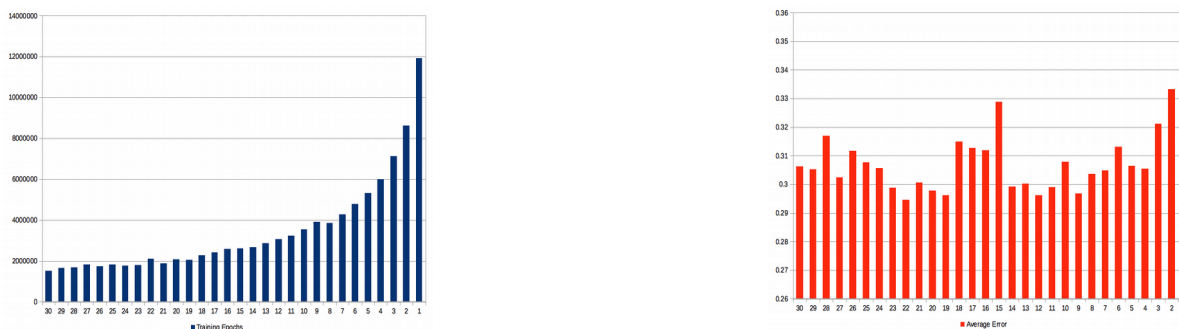
алгоритъм: Размер на популацията - 37; Вероятност за кръстосване - 0.9; Вероятност за мутация - 0.03; Дял елитни елементи - 0.1; Аргументи за състезателна селекция - 2; Вид кръстосване - бинарно; Вид мутация - случаен делта вектор; Условие за край - 24 часа.

И двете мрежи са обучени, при равни други условия. Условието за край на обучението е изтичането на фиксиран период от време, а именно 24 часа. В резултат на обучението, пълно-свързаната мрежа показва по-добри параметри, за по-малко епохи обучение. За трислойната мрежа са нужни 296037 епохи, постига се жизнена стойност на най-добрия индивид в популацията от 656.1699030329493 и допуснатата грешка от изкуствената мрежа е 317.7286234453099, докато при пълно-свързаната мрежа са нужни 294003 епохи, най-добра жизнена стойност 656.169903010489 и допуснатата грешка от 317.0226357675591.

Наличието на повече тегла при една пълно-свързана мрежа води до по-бавен процес на обучение, но наличието на обратни връзки позволява формирането на кратковременна памет в изкуствените невронни мрежи и при значително по-дълго обучение това дава предимство на този модел.

Сравнение между трислойни ИНМ с различен размер в скрития слой

Проведените експерименти са извършени с трислойни мрежи, като на входа се подават 25 стойности от изминал период, а на изхода се очакват 5 стойности в бъдещ период. За целите на експеримента данните условно са разделени в три под множества: 1. Данни за обучение; 2. Данни за валидация; 3. Данни за тестване. Изпробвани са топологии с размер на скрития слой от 30 възела до 1 възел. Всички експерименти са извършени при еднакъв интервал от врем. Фиг. 4.3 (ляво) показва разликата в броя епохи, спрямо размера на скрития слой.



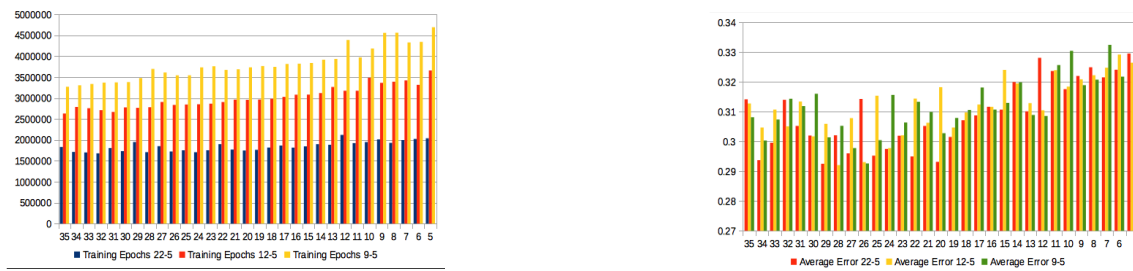
Фиг. 4.3 Брой епохи за обучение, при еднакъв времеви интервал, за обучението, но различен размер на скрития слой (ляво) и средна грешка, допускана от ИНМ, при обучение на мрежи с топология от три слоя, в които размерът на скрития слой варира от 30 до 1 (дясно).

В експериментите е оценена средната грешка, която допускат мрежите с различна топология. Целта е да се установи при кои стойности за размера на скрития слой мрежата е най-ефективна. Изчисляването на грешката е извършено само с използване на данните от подмножеството за тестване. Това са данни, които не са използвани в процеса за обучение и валидация.

От Фиг. 4.3 (дясно) ясно се вижда, че най-перспективни са моделите с 22, 12 и 9 елемента в скрития слой, което съответства на топологии мрежи: 1. 25-22-5; 2. 25-12-5; 3. 25-9-5.

Сравнение за ефективност на прогнозирането при различен прозорец от входни данни

За трите най-ефективни модела, определени на Фиг. 4.3 (дясно), са извършени допълнителни експерименти с изменение на размера, който има времеви прозорец от отминалите периоди. Размерът на времеви прозорец е изследван от 35 до 5 стойности като прозореца за прогнозиране на бъдещите резултати е фиксиран само на 5 стойности.



Фиг. 4.4 Брой епохи за обучение, при еднакъв времеви интервал, за обучението, но различен размер на входния слой (ляво) и средна грешка, допускана от ИНМ, при обучение на мрежи с топология от три слоя, в които размерът на входния слой варира от 35 до 5 (дясно).

От Фиг. 4.4 (ляво) ясно се вижда, че размерът на входния слой почти не оказва влияние в броя епохи, през които протича обучението, за фиксиран интервал от време (в случая 1 час). Също така, ясно се забелязва, че размерът на скрития слой оказва пряко влияние на броя епохи, за обучение, при фиксиран интервал от време. Най-бавно протича обучението при скрит слой от 22 елемента, а най-бързо протича обучението при скрит слой от 9 елемента.

На Фиг. 4.4 (дясно) ясно се вижда при кои стойности на входния слой изкуствената невронна мрежа е постигнала най-малка грешка за предложената прогноза. За размер на скрития слой от 22 елемента, най-ефективният входен слой е 29 елемента. При скрит слой от 12 елемента, най-ефективният входен слой е 28 елемента. И при скрит слой от 9 елемента, най-ефективният входен слой е 26 елемента. От трите възможности най-добри резултати се получават при скрит слой от 12 елемента, което е около половината от входния слой.

Заключение

Резюме на постигнатото и идеи за бъдеща работа

В дисертационната работа са представени основни резултати, получени от проведените изследвания в областта на методите за прогнозиране на времеви редове, алгоритми и програмни системи за решаване на задачи в областта на прогнозирането.

Целта на тези изследвания е разработката на евристичните подходи за машинно самообучение, разширяващи възможностите за прогнозиране на времеви редове. На основата на тези евристични подходи са предложени евристични методи и алгоритми, подобряващи бързодействието в машинното самообучение на изкуствени невронни мрежи. Ускоряването на бързодействието в обучението на изкуствени невронни мрежи е особено важно при прогнозирането на времеви редове, тъй като фазата на обучение е значително по-бавна отколкото фазата на самото прогнозиране. Характеристиките на тези методи и алгоритми, качеството на предложения дизайн, както и добрата програмна реализация, са в основата за създаването на една ефективна програмна система за прогнозиране на времеви редове, в сферата на валутните пазари.

Постигнатите резултати в дисертационната работа очертават следните насоки за бъдещи изследвания, като: изследване на предложените евристични подходи за търсене на прогнози в други области, а не само по отношение на валутните пазари; Разширяване на програмната система, като се включат и други методи за прогнозиране на времеви редове, с цел по-ефективен сравнителен анализ; Разширяване на възможностите на уеб базирания интерфейс в системата и дори осъществяване на изчисления в разпределена среда, базиращи се на пресмятания върху мобилни устройства.

Авторска справка за приносите в дисертацията

Научни приноси

1. Предложен е евристичен подход за обучение на пълно свързани невронни мрежи в разпределена среда, базиран на инцидентно включване на възли;
2. Предложен е евристичен подход за обучение на пълно свързани невронни мрежи в разпределена среда, базиран на диференциална еволюция;

3. Разработен е метод за машинно обучение на пълно свързани невронни мрежи в разпределена среда, базиран на предложения евристичен подход за обучение с инцидентно включване на възли;

4. Разработен е метод за машинно обучение на пълно свързани невронни мрежи в разпределена среда, базиран на предложения евристичен подход за обучение с диференциална еволюция;

Научно-приложни приноси

5. Предложени са алгоритми за прогнозиране на времеви редове с използване на пълно свързани невронни мрежи, обучавани с предложия евристичен подход с инцидентно включване на възли;

6. Предложени са алгоритми за прогнозиране на времеви редове с използване на пълно свързани невронни мрежи, обучавани с предложия евристичен подход с диференциална еволюция;

Приложни приноси

7. Изследвани са и са експериментално оценени предложените подходи, методи и алгоритми за прогнозиране на времеви редове с данни от валутни пазари.

Библиография

- 1 [HAMN2002] Alfares, H., Nazeeruddin, M., Electric load forecasting: literature survey and classification of methods, International Journal of Systems Science, vol. 33, Num. 1, p.23-34, 2002.
2. [ANMO2011] Аначков, М., Система за визуализиране и манипулиране на изкуствени невронни мрежи“, ТУЕС към ТУ-София, 2011.
3. [ARSH2003] A. Shapiro , Capital Market Applications of Neural Networks, Fuzzy Logic and Genetic Algorithms, 13th International AFIR Colloquium, Vol. 1, p.493-514, 2003.
4. [BATO2011] Балабанов. Т., Прогнозиране с евристични подходи в разпределена среда, Proceedings of Anniversary Scientific Conference 40 Years Department of Industrial Automation, ISBN:978-954-465-043-8, 163-166, 2011.
5. [BATO2015] Балабанов, Т., Избягване на локални оптимуми при евристични популационни алгоритми в разпределена среда, Сборник с доклади от XXII Международен симпозиум Управление на енергийни, индустриални и екологични системи, ISSN:1313-

2237, 83 - 86, 2015.

6. [BAGE2016a] Балабанов, Т., Генова, К., AJAX разпределена система за обучение на изкуствени невронни мрежи с еволюционни алгоритми, Сборник с доклади от XXIV Международен симпозиум Управление на топло енергийни обекти и системи, Управление на енергийни, индустриални и екологични системи, ISSN:1313-2237, 49 - 52, 2016.
7. [BAKE2016] Balabanov T., Keremedchiev D., Goranov I., Web Distributed Computing For Evolutionary Training Of Artificial Neural Networks, International Conference InfoTech-2016, Varna - St. St. Constantine and Elena resort, Bulgaria, ISSN:1314-1023, 210-216, 2016.
8. [TBIZ2015] Balabanov, T., Zankinski, I., Barova, M., Distributed Evolutionary Computing Migration Strategy by Incident Node Participation. Large-Scale Scientific Computing, Lecture Notes in Computer Science, 9374, Springer International Publishing Switzerland, ISBN:978-3-319-26520-9, DOI:10.1007, p 203-209, 2015.
9. [BAZA2011] Balabanov. T., Zankinski I., Dobrinkova N., Time Series Prediction by Artificial Neural Networks and Differential Evolution in Distributed Environment, Large-Scale Scientific Computing, Lecture Notes in Computer Science, vol. 7116, ISBN:978-3-642-29842-4, DOI:10.1007/978-3-642-29843-1_22, 198-205, 2011.
10. [BAZA2010] Балабанов Т., Занкински И., Симеонова В., Прогнозиране на времеви редове с изкуствени невронни мрежи и диференциална еволюция в разпределена среда, Работни статии на ИИТ, ИТ/WP-268B, ISSN:1310-652X, 2010.
11. [CDTY2006] C. Dawson, T. Young, J. Allen, Machine Intelligence Genetic Algorithm Neural Network System for the Engineered Wood Industry, USDA SBIR Phase-I Final Report, Jul 2006.
12. [DUWI2002] Dunis, C.L. & Williams, M., Modelling and trading the eur/usd exchange rate: Do neural network models perform better? Derivatives Use, Trading and Regulation, 8(3), pp. 211–239, 2002.
13. [EUVO2005] E. Vogel, Complex Indicators: Nonlinear Pricing and Reflexivity, Department of Informatics, St. Petersburg State University of Economics and Finance, 191023, Saint-Petersburg, Russia, 2005.
14. [GILA2001] Giles, C.L., Lawrence, S. & Tsoi, A.C., Noisy time series prediction using a recurrent neural network and grammatical inference. Machine Learning, 44(1/2), pp. 161–183, 2001.
15. [YAHU2009] Huang, Y., Advances in Artificial Neural Networks – Methodological Development and Application, Algorithms, Vol.2, No.3, p.973-1007, 2009.
16. [HAYK1999] Haykin, S., Neural Networks, A Comprehensive Foundation. Prentice-Hall, Inc., 2nd edition, 1999.
17. [HOLL1975] Holland, J., Adaptation In Natural and Artificial Systems. The University of Michigan Press, 1975.
18. [HREL2011] Христова, Е., Мултимедиен скрийнсейвър за Linux, подходящ за вграждане в проекти за разпределени изчисления, ТУЕС към ТУ-София, 2011.
19. [JMAM2012] Merelo, J., Mora, A., Fernandes, C., Arenas, M., A. Esparcia-Alcazar, Using pool-based evolutionary algorithms for scalable and asynchronous distributed computing, In Proceedings of the 12th International Conference on Parallel Problem Solving From Nature (PPSN 2012), Sep 2012.
20. [MTMQ2015] MetaTrader 4, MetaQuotes Software Corp., (2015, October 20) <http://www.metatrader4.com/>
21. [MOOD1995] Moody, J.E., Economic forecasting: Challenges and neural network solutions. Proceedings of the International Symposium on Artificial Neural Networks, Hsinchu, Taiwan, 1995.
22. [SMSK2001] Moon, SW., Kong, SG., Block-Based Neural Networks, IEEE Transactions On Neural Networks, Vol. 12, No. 2, Mar 2001.
23. [NIRA2012] Николова, Р., Разработка на система за графично изобразяване и визуално манипулиране на изкуствени невронни мрежи, НБУ-София, 2012.
24. [RSQZ2006] Segall, R., Zhang, Q., Applications of Neural Network and Genetic Algorithm Data Mining Techniques in Bioinformatics Knowledge Discovery - A Preliminary Study, Proceedings of the Thirty-seventh Annual Conference of the Southwest Decision Sciences Institute, p. 278-285, 2006.
25. [MSMJ2009] Shahi, M., Jaksa, M., Maier, H., State of the Art of Artificial Neural Networks in Geotechnical Engineering, Advances in Artificial Neural Systems (Bouquet 08), Vol. 2009, Article No. 5, 2009.
26. [JVGM2002] Valdes, J., Mateescu, G., Time Series Model Mining with Similarity-Based Neuro-Fuzzy Networks and Genetic Algorithms: A Parallel Implementation, In Proceedings of the RSCTC'02, Special Session on Distributed and Collaborative Data Mining, Pennsylvania. October 2002. NRC 44933.

27. [MVDL2008] Vasquez, M., Losada, D., Osorio, F., Exploiting Stock Data: A Survey Of Computational Techniques for Generating Portfolio Beliefs, Aceptado para publicar en Vol. 28, No. 1. Revista Ingenieria e Investigacion Facultad de Ingenieria, Apr 2008.
28. [MAWE2010] Weber, M., Parallel Global Optimization, Structuring Populations in Differential Evolution, Dissertation in Faculty of Information Technology of the University of Jyvaskyla, Nov 2010.
29. [SINI2011] Симеонов, Н., Мултимедиен скрийнсейвър за Windows, подходящ за вграждане в проекти за разпределени изчисления, ТУЕС към ТУ-София, 2011.
30. [WERB1990] Werbos, P., Backpropagation through time: what it does and how to do it. Proceedings of the IEEE, 78(10), pp. 1550–1560, 1990.
31. [YAOX1999] Yao, X., Evolving artificial neural networks. Proc of the IEEE, 87(9), pp. 1423–1447, 1999.

Abstracts of Dissertations

Number 6, 2017

INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGIES
BULGARIAN ACADEMY OF SCIENCES

БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ
ИНСТИТУТ ПО ИНФОРМАЦИОННИ И КОМУНИКАЦИОННИ ТЕХНОЛОГИИ

Брой 6, 2017

Автореферати на дисертации