

Abstracts of Dissertations

Institute of Information and
Communication Technologies

BULGARIAN ACADEMY OF
SCIENCES



8 / 2017



POST-LEARNING
STRATEGY AND
EVOLUTIONARY
ARCHITECTURE IN
NEURAL NETWORKS

Kristina Karanova

НОВИ ОПТИМИЗАЦИОННИ
СТРАТЕГИИ И
ЕВОЛЮЦИОННИ
АРХИТЕКТУРИ ЗА ОБУЧЕНИЕ
НА НЕВРОННИ МРЕЖИ

Кристина Капанова

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

POST-LEARNING STRATEGY AND EVOLUTIONARY ARCHITECTURE IN NEURAL NETWORKS

Kristina Georgieva Kapanova

**Supervisors: Prof. Ivan Dimov,
Assoc. Prof. Jean Michel Sellier**

Approved by Supervising Committee:

Acad. Ivan Popchev
Cor. Member Krassimir Atanassov
Prof. Ivan Dimov
Prof. Stefka Fidanova
Assoc. Prof. Olimpia Roeva



**INSTITUTE OF INFORMATION AND
COMMUNICATION TECHNOLOGIES**

Department of Parallel Algorithms

The PhD thesis was discussed and allowed to be defended during an extended session of the Department of Parallel Algorithms at IICT-BAS, which had been held on November 15, 2016.

The defense of the PhD thesis had been held on February 14, 2017 at 14:00 in Room 218, Block 25A, IICT-BAS. The full volume of the dissertation is 120 pages. It consists of six chapters, including Introduction and conclusion. It includes also an appendix. The list of references contains 180 titles. The text of the dissertation includes 5 tables and 29 figures. These lists are given at the end of the dissertation.

Keywords: artificial neural networks, supervised learning, optimization strategies, evolutionary algorithms, post-learning strategy, sensitivity analysis

Introduction

In this thesis, the subject of research is artificial neural networks. Many aspects of artificial neural networks have been studied since their inception, with over 40 different paradigms proposed and used for various task related problems. Nevertheless, the field faces many problems such as the inherent instability of the system depending on the chosen model. One of the most fundamental problems in the field is the existence of local minima and the possibility of the network to reach it during training without a potential escape from it. The main objectives of the PhD thesis are from scientific and application perspective including several important tasks. First, the development of a novel post-learning algorithm, which once implemented allows for the network to escape from possible local minima or saddle points reached during the optimization process. Secondly, the aim is to implement sensitivity analysis indicators to study the perturbation of the weights in the network, affected by noise in the system. The study of sensitivity analysis could provide beneficial to the area of hardware based neural networks, where noise from various sources influences the network's performance. Third, the development and implementation of a novel automatic evolutionary architecture of a neural network through a hybrid genetic algorithm. As such, trial and error approaches to network design are discarded for an automatic discovery of network topology appropriate for specific tasks. The aim is to assist research with the decision making possibilities and computational complexity involved in designing neural networks.

To address these objectives the following tasks were constructed:

1. Development of a new optimization algorithm initialized once the training process of a network is completed.
2. Selection and practical implementation of sensitivity analysis benchmarks to investigate levels of noise in the system in real time.
3. Development of a hybrid evolutionary algorithm, employed towards the automatic selection of a neural network architecture dependent on specific task.
4. Program implementation in C a neural network architecture, including the developed algorithms.

1. Methodology

The methodology of the thesis is based on research from fields such as artificial intelligence, artificial neural networks, optimal optimization methods, and quantum physics. The work combines approaches based on the laws of physics, and statistical analysis and machine learning. Several mathematical indicators related to sensitivity analysis to determining the level of noise in the system have been utilized based on hardware and software implementation of neural networks. New hybrid evolutionary algorithm has been developed for the automatic design of topological structures of neural networks. Numerical experiments have been carried out and analyzed. The proposed algorithms and indicators have been implemented in the development of a neural network in C language.

Artificial neural networks incorporate a network of simple processing units, interconnected and some cases running in parallel. Each unit contains an activation value, which communicates with other units along connections of different strengths. Those connections determine how input to the processors is transformed into output. Significantly, the knowledge in a connectionist structures is encoded in the strength of the connections between the units and not in the symbolic structure of the machine. The parameters of the connections are found adaptively through a learning strategy until an acceptable solution is found. The particular learning paradigm, known as supervised learning, introduces to the system a training set of an input-output mapping. In the course of the training, the difference between actual output and the desired output (predicted by the model) is calculated and

the connections between nodes are adjusted according to predefined principle in order to minimize the error between desired and actual output. The procedure is repeated multiple times, often randomizing the order of the training samples presented to the network until no considerable changes in the synaptic connections are observed. Thus the connections between the neurons determine the functioning of the artificial neural network. This adaptability of the neurons is essential characteristic of ANNs to adapt to various environments - including the ability to perform in real-time conditions with continuous change of information.

In 1940s W.S. McCulloch, a neuroscientist, and W. Pitts, a logician, encouraged by Rashevsky, proposed in a paper [1] a mathematical description of an extremely simplified model of a neuron called "threshold logic unit" for complex pattern recognition task. Their neuron either produces an output or not. For a neuron to fire, more than 1 synaptic weight needs to be excited in a certain time interval. If instead an inhibitory synapse is active, it prevents the firing of the neuron at specific time period. Later, the set of processing nodes (neurons) that are connected to one another has been described as an artificial neural network. The McCulloch-Pitts model was capable of converting a certain logical proposition in terms of an artificial neural network.

In 1958 Frank Rosenblatt introduced one of the first training algorithms for learning in neural networks in [2] and proving the convergence of the perceptron rule. Rosenblatt suggested the innovation of numerical weights, the computing units which serve as threshold elements and an interconnection pattern. In his model, the learning is achieved by adaption of the weights of the network through a predefined algorithm. The initial experimentation with the Rosenblatt's model brought an important question about the possible limitations that might be observed in various pattern recognition tasks and the ability of the network to solve them effectively. Some early experiments with the Rosenblatt's perceptron produced unrealistic expectations in the connectionism field.

Many aspects of artificial neural networks have been studied since their inception, with over 40 different paradigms proposed and used for various task related problems. Nevertheless, the field faces many problems such as the inherent instability of the system depending on the chosen model. One of the most fundamental problems in the field is the existence of local minima and the possibility of the network to reach it during training without a potential escape from it. To deal with those inherent problems, various combinatorial algorithms have been proposed. Unfortunately not always a substantial improvement have been introduced due to slow convergence problems and the presence of multiple saddle points. The growth of network parameters provides additional burden to the convergence speed of algorithms and increase of the computational complexity of the system. The network parameters are generally based on manual design. The decision about the number of input and output neurons depends on the specific problem at hand. A developer, knowledgeable about the specific task, selects the number of hidden layers, number of neurons in every hidden layer, the training algorithm, as well as the type of activation function. Those components are crucial for the computational performance, efficiency and accuracy of the network but are often established through a trial and error basis and on the personal experience of the researchers.

One of the first network architectures is introduced in 1958s by F. Rosenblatt, when he incorporated a perceptron in a simple neural network to classify linearly separable patterns [2]. The Rosenblatt perceptron consists of a single neuron, with adjustable weights and bias (w_0). Rosenblatt has introduced for the first time also a novel algorithm applied to the free parameters of the system to adjust their values.

Major drawback of this model is its convergence only when the system is presented with patterns from two linearly separable classes (known as perceptron convergence theorem). The Rosenblatt's perceptron is one of the fundamental parts of every neural network. While it provides an oversimplification to the biological neuron, it adheres to the following association of the input/output

parameters. Let us define the number of inputs as $(x_1, x_2, x_3, \dots, x_n)$, and the weight as w_i and the connection from neuron to neuron as w_{ij} . The output (Out) of a neuron is seen as

$$Out = \sum_{i=1}^N w_i x_i \quad (1.1)$$

constituting the weighted sum of the incoming inputs and weights. If a predefined threshold (θ) is exceeded, the neuron produces an output Out. As part of the training algorithm is the introduction of bias (w_0), which can be represented in the network as

$$Out = w_0 + \sum_{i=1}^N w_i x_i \quad (1.2)$$

A variety of approaches to the design of neural networks have been considered in the literature, with few distinct considerations [3], [4], [5], [6]:

- The selection of the number of hidden layers;
- The number of neurons to be represented in every hidden layer;
- The selection of a globally optimal solution, which avoids local minima;
- Convergence time should be reasonable;
- Validation of the network for under or over fitting;
- Consideration for the interaction of hidden neurons and their interaction on a global level.
- The ability to balance the considerations described above will provide a network preserving its function approximation capabilities without decrease of efficiency [7].

Nowadays, the renewed interest in the field of artificial neural networks and their applicability for many problem solving tasks has led to their pervasive use in many research fields and application domains. One of the field of robotics, where the networks are used as a way to direct manipulators to hold objects, based on some sensory data. Neural networks are applied in robot steering and path planning [8], [9] as well as object manipulation. Deep neural networks like Supervision, consisting of more than 650000 neurons and utilizing more than 60 million parameters during the training and recognize objects in particular categories [10], Neural networks has been essential in the object recognition research, where generating precise and detailed object recognition in real time environment becomes crucial. Neural networks have been employed with considerable success in the financial/banking sector. They are used in daily basis in measuring credit card risk for applicants, for loan evaluation procedure and loan rates prediction, mortgage screening, real estate appraisal, financial market and exchange rate forecasting as well as corporate financial analysis [11], [12] Another field where artificial neural networks has found incredible usage in the last 15 years in in manufacturing. Such activities include planning and management, process control, product analysis, machine diagnosis and analysis, quality analysis for different products (from the food, chemical, chip industry, etc) [13]. The ability of artificial neural networks to learn have been of great use in various scheduling activities - vehicle transportation scheduling, and routing system of various kinds. The swiftkey (a keyboard app for Android phones) is also utilizing neural networks to predict and correct language. Nowadays Facebook has been using GPU based neural networks to recognize images, but have also used ANNs to create automatic pictures based on specific understanding of what objects look like, especially in the production of thumbnail images [14]. On the other hand, Google has used neural networks and let them produce visual elements the network wants to emphasize on [15].

Hardware implementation of neural networks, the SyNAPSE project, a collaboration between IBM and Hewlett-Packard, has been one of the leading research initiatives in the field [16]. The project has received over thirty million dollars (27 million Euros) grant from DARPA to create this "cognitive computer" which is essentially made up of thousands of parallel CPUs, each a square micron in size representing the neurons and connection of a neural network. DARPA is not limiting its financial focus only on this project. In fact, the annual budget of the organization is over sixty one million dollars (55 million Euros) for Machine Learning and almost fifty million dollars (or 45 million Euros) for Cognitive Computing [17].

3. Training of neural networks

The field of artificial neural networks has attempted to emulate certain learning capabilities through the applicability of symbolic, data-intensive and statistical approaches. Nowadays there are myriad of algorithms and techniques applied in various domains in the field of neural network training (i.e. learning). Any attempt to accomplish an overview of the methods and applications used for training would be incomplete. One may classify the learning algorithms as supervised and unsupervised learning, being differentiated through their learning structure. Some consider reinforcement learning as a separate learning category, or as a subcategory of supervised/unsupervised learning [18]. The basis for many neural network learning algorithms is structural inferences focusing on the relationship among the variables and the data structure. We can formulate any learning task for a neural network in terms of a variational problem. Data modeling types like function regression, pattern recognition and time series prediction, could also be presented as a variational problems and are the most common problems encountered in neural networks.

For the training of a network to find the optimal set of weights and therefore produce the desired output we need to define the technique through which this is achieved. Therefore the training of artificial neural networks may be represented as an optimization task, i.e. the selection of efficient strategy to discover the minimum or the maximum values of a function, which may consists of many independent variables. The selection of appropriate error function depends on the particular optimization algorithm, as well as on the topology of the neural network [18], [19]. Some of the frequently utilized cost functions include: linear, Quadratic, Sum of squares (The least squares), Smooth nonlinear, Nonsmooth.

The supervised learning algorithms applied to the training of neural networks are focused on the minimization of the cost function in order to solve a particular task. The algorithms need to possess several important characteristics [18]:

- Need to be able to solve a task with minimum computational burden.
- The algorithm should be robust in the presence of noise.
- The algorithm should be able to produce similar outcomes independent of the initial conditions used during the initialization.
- The generalization capabilities should provide adequate outputs when the neural network is provided with data different from the training set.
- The computational burden of the algorithm should not be strongly dependent on the dimensionality and size of the problem and training data. It should be scalable. This problem is more pronounced currently in big data.
- Convergence requirements.
- Setting appropriate stopping criteria in order to find the best possible outcome.

The field of artificial intelligence has been a place where arguments have taken place as to whether our world is relying on deterministic casual events, where one uses probabilities to calculate missing knowledge. The dissenting opinion states that our world is essentially indeterministic with we use

probabilities to measure the random events. As described in the previous chapter, optimization problems (training algorithms) are broadly classified as deterministic and stochastic. In this classical approach all weights transformation is carried through during the optimization process. A step further, one can say that the training methods from the previous chapter are constituted in terms of classical (deterministic) physics. But if one sees the world as essentially indeterministic, an interesting possibility to exploit is quantum mechanical effects in a ANNs. The idea has been suggested for the first time (to the best of the limited information available) in [20] where the authors propose that biological expressions of quantum phenomena are related to the activation points in the brain activity. All of this is described as a nerve impulse firing achieved by the movement of a quantum particle in the vicinity of a potential energetic barrier. The availability of an action potential is proposed to play a significant role in the information processing functions of the brain since there effects such as tunneling arises. On the other hand Penrose [21] suggests the concept of the existence of micro-tubules, capable of maintaining a macroscopic coherent superposition.

The possibility of biological neurons to exploit quantum effects provides an encouragement of introducing randomness in a neural network implementation with the goal of achieving certain computational advantages [20]. The proposal for the implementation of quantum mechanical laws inside every neuron correlates to the condition of the numerical simulation of the time-dependent Schrödinger's equation (or any other equivalent formalism like Feynman, Wigner). This is necessary to quantitatively determine the eventual tunnelling effects. Therefore, the network needs to numerically simulate the following time-dependent partial differential equation [22] for every neuron:

$$i\hbar \frac{\partial \Phi}{\partial t} (\mathbf{r}, t) = \left(-\frac{\hbar^2 \nabla^2}{2m} + V(\mathbf{r}) \right) \Phi(\mathbf{r}, t), \quad (1.3)$$

where i is the imaginary unit, $\Phi(\mathbf{r}, t)$ is the wave function defined over space and time, \hbar is the reduced Planck's constant, \mathbf{r} is the position of the particle, t is the time, m is the mass of the particle, ∇^2 is the Laplacian operator, and $V(\mathbf{r}, t)$ is the potential energy acting on the particle.

The method is modeled by analogy of a biological neuron as a semiconductor heterostructure that consists of an energetic barrier (e.g. AlGaAs) that is situated between two energetically lower areas (e.g. GaAs). [20] gives as the possibility to consider the activation function of an artificial neuron as one or more particles interacting with the barrier (see Fig. 3.7) once they enter the heterostructure.

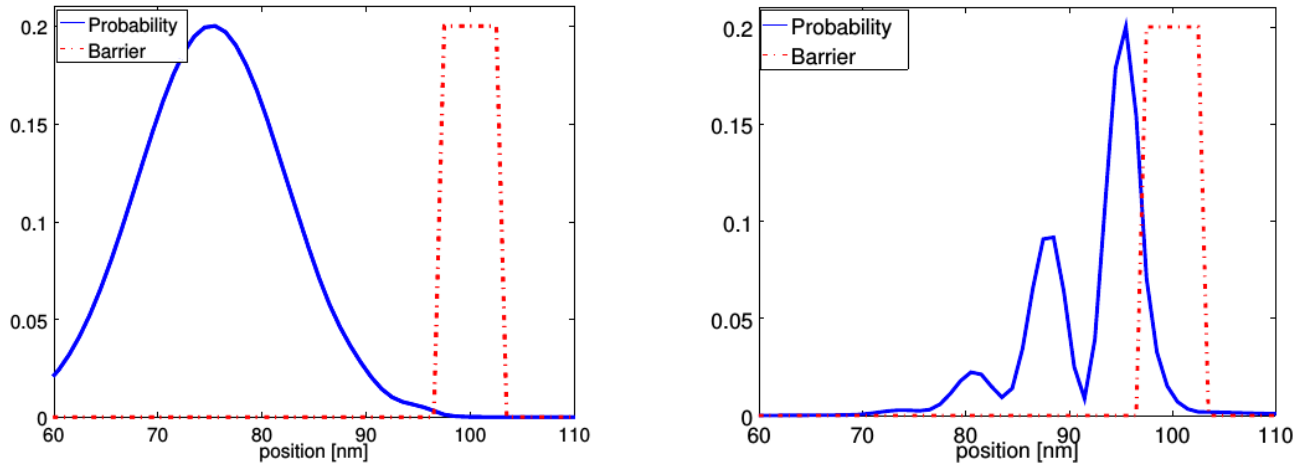


Fig 1 Left plot: a Gaussian wave-packet, (blue) continuous line, is travelling against an energetic potential barrier, (red) dashed line. Right plot: after a certain time, the wave-packet is interacting with the barrier. Part of the packet is scattering back while the rest is tunnelling.

The Numerical experiments of the suggested post-learning strategy is presented. The implementation of the neural network works to fit two known functions - a polynomial of second degree ($f(x) = x^2$) and the square root a polynomial ($f(x) = \sqrt{x}$), with three training data points provided. In both experiments the network architecture is identical - with one input node, 4 hidden neurons and one output neuron. During the numerical experiments, we have deliberately hindered the network by stopping the algorithm at an arbitrary local minimum. The task is accomplished by means of the temperature rate of the algorithm, which we decrease in a non-optimal fashion. Through the strategy, the goal is to clearly show that our technique is able to provide a further solution to improve the training even after the optimization process is completed.

The first numerical experiment depicts the fitting of a polynomial of second degree. The reader might observe that indicative to the amount of noise applied is the dispersement of both the quantum output and error throughout the plot in settings where one successively increases the amplitude of the noise. As such, a moderate increase of noise to 1% leads to a situation where the network output in the middle of the solutions decreases its accuracy, while still retaining better results in the upper and lower bound of the curve.

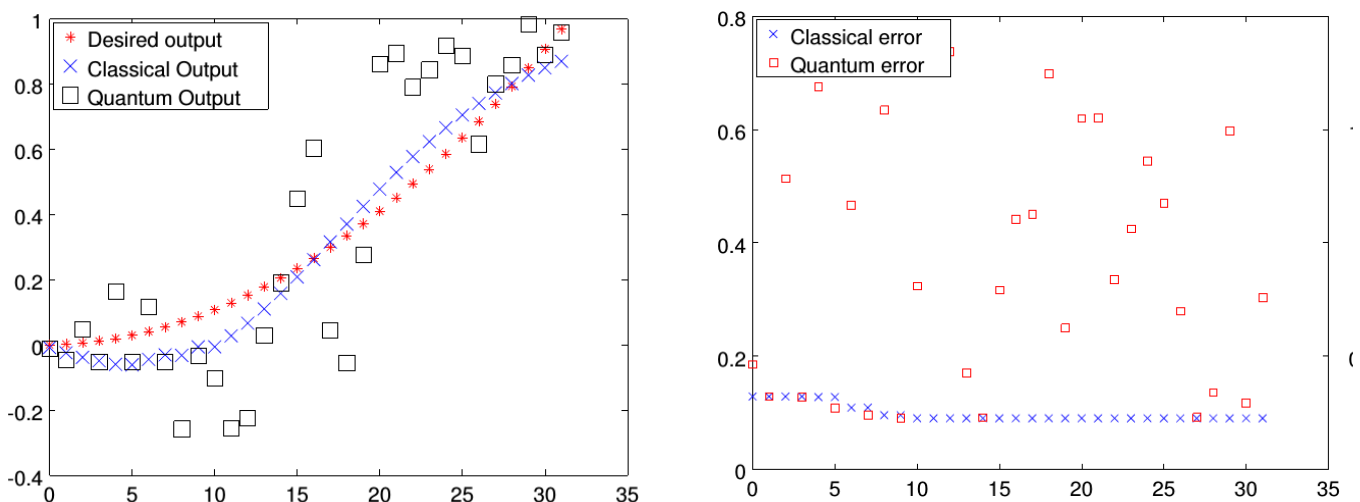


Fig 2: Left plot represents the network's output. Right plot represents the network's error when 1% noise is applied
Abstracts of Dissertations 8 (2017) 3-16

With the addition of more noise to the system, the quantum network outperforms the classical one in few cases. The increase from 1% to 2% and 4% noise, leads to an overall gradual decrease in error . We have performed a second test in fitting a square root of a polynomial, taking into account the assumption of the capacity of the novel technique to potentiality operate in a distinctive mode when computing different functions. The efficiency of the quantum network is in point of fact increased in a situation where 2% noise is added. The network produces as much as twice as many quantum outcomes, as when imposing 0.5% or 1% noise. Moreover, we observe the presence of lower errors . There is significant divergence from the initial experiment after the early steep error reduction from the quantum part. Here, the clustering of the quantum error points is found near the classical ones.

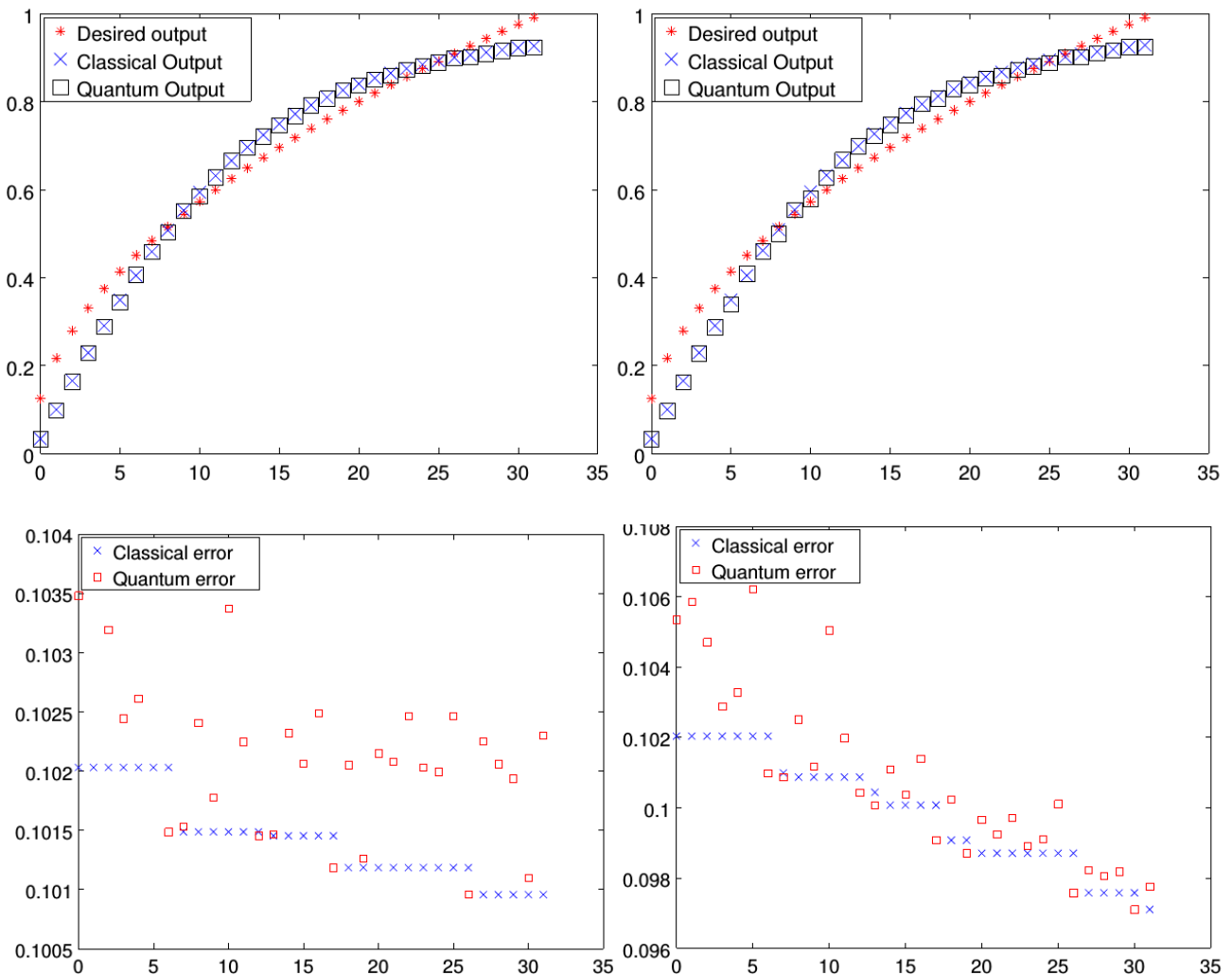


Fig 3: Upper left and right plots represent the network’s output in 1% and 2% noise in the case of square root of polynomial. Lower left and right plots represent the network’s error when 1% and 2% noise is applied respectively

4. Sensitivity Analysis in Neural Networks

We introduce a sensitivity analysis tool to establish the level of noise in the network. Other research have been carried out on the "random or unpredictable fluctuations and disturbances" [23] and the possible benefits noise in neural systems can introduce [24]. For instance in engineering, the noise is identified as a detrimental to the system and the quality of the output. In biological neural systems, on the other hand, noise is naturally existing, often providing certain benefits to information processing [23, 24, 25, 26]. Currently, we believe no definitive explanation has been reached about how naturally occurring fluctuations in neural systems could be exploited, why they do in fact occur and what could be the potential advantages. While adding certain amounts of noise could provide benefits from the network oscillations, in other cases this oscillation could be detrimental [27].

The goal of those sensitivity analysis techniques is to provide us with an accurate mathematical approach to determine the noise levels in a network. Our aim is with those tools to establish whether the noise can be utilized for improving network performance and when it is detrimental and acts as a source of fluctuations. Furthermore, with the sensitivity analysis tools we could achieve a better understanding of the noise variation on the network's behavior and the sensitivity of the output relation to the noise fluctuation. The utilized sensitivity coefficients indicate what is the possible variance of the outputs of the network due to fluctuations in the weight parameters of the system. To the best of our knowledge, no study has been accomplished to understand the level of influence of the noise in the network and what type of consequences such noise present for the network's performance. The functionality of a network could be considerably altered with the modification of noise in the weights and therefore there would be considerable sensitivity to the system. Vice versa, with negligible sensitivity, the performance of the network should not be substantially altered.

Those sensitivity tools provide a quantitative representation of the fact that moderate amount of noise do in fact improve the network's performance. For this sensitivity analysis, we have implemented three measurements to establish the scope of the solutions: the Euclidean distance (L_2), the maximum norm (L_∞) and the cosine-similarity (L_{cos}). The current sensitivity analysis implements two numerical experiments - for convex and concave non decreasing functions with the goal of avoidance of biased results.

Considering the Euclidean distance parameter (fig. 4 upper left plot), we observe the presence of a distinctive peak around 40% noise, which indisputably indicates improvement of the network. As such the quantity of noise is improving the solution drastically. In contrast, the third parameter provides a snapshot of a deterioration of the network's behavior when more noise is applied. Moreover, in terms of L_2 and L_∞ , their decreasing amounts represent a good solution to the problem, in L_{cos} we observe oscillations and notice that near 1 the network provides a sufficient solution, and away from it there is no such available good solution.

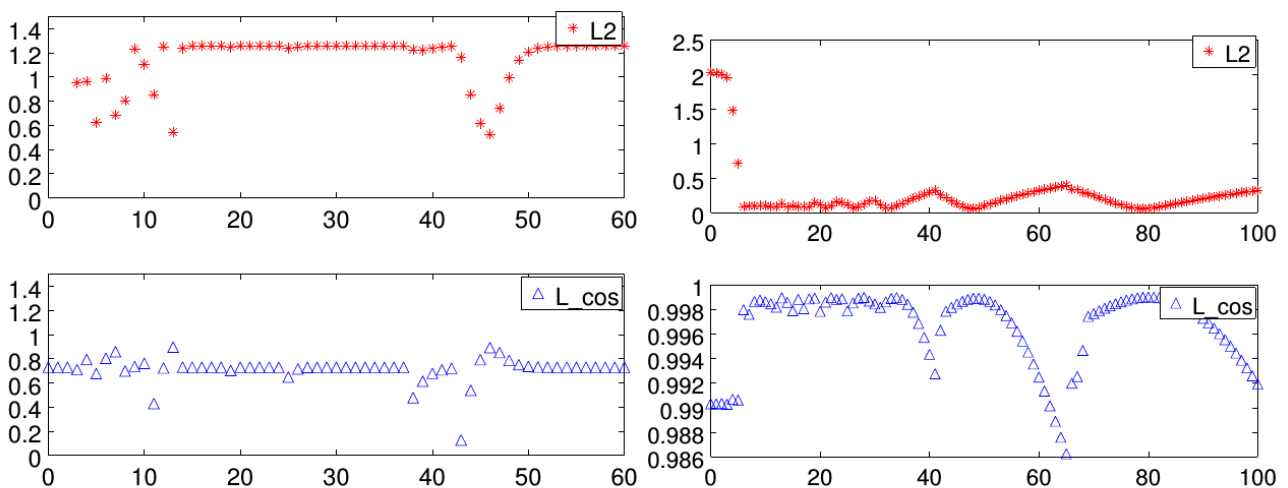


Fig 4 The left side plots describe the comparison between the Euclidean distance (L_2) indicator and the cosine similarities in absolute value for the case of a polynomial of second degree ($f(x) = x^2$). The right side plots illustrate the comparison between the Euclidean distance (L_2) indicator and the cosine similarities in absolute value for the case of the square root of a polynomial ($f(x) = \sqrt{x}$)

5. Evolutionary ANN Architecture

We introduce a novel approach to the evolution of the neural network architecture in order to provide an automatic and computationally feasible self-organization of an artificial neural network layout to solve a given problem. A hybrid stochastic optimization and genetic algorithm, allowing the network to choose almost every possible structure in the implementation of the network in a given space of possible architectures is discussed. Through this evolving architecture, we obtain a method which is easily parallelized and is capable of escaping to overfitting of the output data when the minimal amount of training data is available.

Our algorithm allows for the randomization of the number of neurons, number of hidden layers, types of synaptic connections, use of transfer functions and type of training algorithm, when selecting the best possible topology. The recombination of all those components establishes a multidimensional space of possible structures, while capable of achieving sufficient performance at affordable computational resources. The training process is based on the simulated annealing method [29].

The number of layers, the amount of neurons per hidden layer and the available connections are the first three criteria used in the neural network architecture method developed in this work. The model operates first by restricting the available choices to a minimum and maximum value allowed for layers and neurons for each run of the network. There is no specific implementation in the code for the addition or removal of neurons during the process.

The network is not limited to a type of connections between neurons. The evolutionary algorithm can choose to develop a sequential neuron connection (in a feedforward manner), or non-sequentially - connecting neurons irrespective of their position between layers. The evolutionary algorithm provides the best possible solution by comparison between elements of various generations. One could restrict the number of inputs per neuron allowed in the architectures depending on the specific task that needs to be solved (i.e. to further control dimensions of the space of solutions).

The genetic algorithm has been one of the most widely used approaches in studies of evolution of neural network architectures [30], [33], [29], [32], [31]. In this work, we introduce a new hybrid genetic approach through the application of an additional stochastic layer, running simultaneously to the genetic one. The latter combines mutation and crossover approach, through which the algorithm evolves the potential topological combinations. Then we select the offspring with the best fitness.

Fitness and Selection Strategy. The algorithm initialization assigns fitness scores each individual in the population, indicating the quality of the specific network topology. In this work, we have utilized the L2 – norm fitness function (although we are not limited to only this function). Each time step, the algorithm accepts further evolution an individual with the best fitness. The process is similar to Tournament selection method [34].

Training Process. At each time step, every individual from the population is trained by the simulated annealing method described above. In this regard, one might consider the combination of the architecture selection and the training process two nested optimization problem. Therefore, one might speculate the possibility both strategies influencing one another.

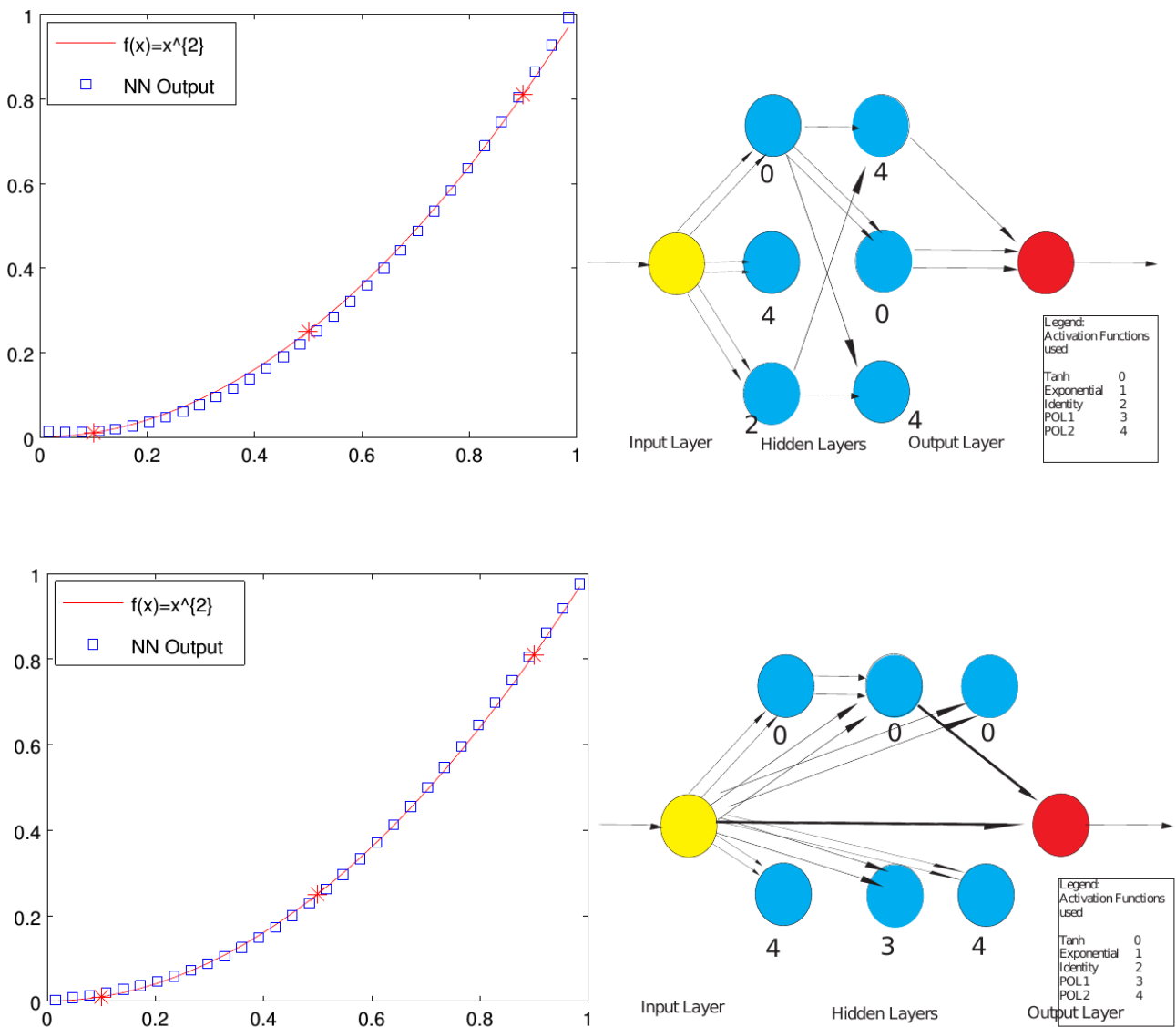
Initialization of the Technique. The algorithm is provided with the ability to choose from four specified parameters described in the previous subsection: number of possible hidden layers, number of possible neurons in every hidden layer, number of connections per neurons, and type of activation function when choosing topology parameters.

Termination. To be terminated the algorithm needs to meet one of two conditions. It needs to reach the maximum assigned number of generations or the fitness error to fall below the assigned threshold

value after r consecutive generations.

The work presents the results from a numerical experiment of fitting a known function $f(x) = x^2$ in order to corroborate the ability of the proposed method. Several examples are provided, presenting the method's outcomes with restrictions in the number of hidden layers and neurons in them. This work presents architectures from 4 scenarios - implementing architectures with 3, 4, 5, and 6 hidden layers and different number of neurons allowed for each hidden layer 2, 3, 4, 5, 6. For the current numerical validation, the amount of input connections per neuron is limited to the maximum- 2. Those restrictions are only executed in the current experiment and are not in any way intrinsic to the effectual work of the proposed hybrid evolutionary technique. Similar to the training data utilized in the previous chapters, the training data is limited to three points - (0.1, 0.1), (0.5, 0.5) and (0.9, 0.9).

In accordance to the discussed numerical experiments, one may conclude in certain situations a sequentially connected network might not be superior in certain tasks. Moreover, in situations where the proposed hybrid evolutionary technique utilizes more than one hidden layer with several neurons, the algorithm frequently performs a model of a neuron removal. As such it discontinues redundant calculations. In those instances, the network accomplishes sufficient results according to a user defined cost function. Conjointly, the neurons connected to the output layer consistently use the logistic activation function. Therefore, inherent consequence bigger architectures, which allow for expanded search space of possible solutions, the hybrid algorithm reduces the architecture by providing several layers, though leaving only one neuron from them is connected to the output layer.



Abstract of Dissertation S8 (2017) 8-16
 The upper plots represent the Scenario with 4 layers, with left plot – showing the output of network and right plot representing the network architecture. Lower plots represent the Scenario with 5 layer, with left plot – showing the output of network and right plot representing the network architecture.

Contribution summary

The contribution of the thesis can be summarized as follows.

1. This dissertation has introduced a new post-learning technique to assist the supervised learning algorithm of an artificial neural network to escape a local minima or a saddle point if such has been reached during the optimization process.
2. We have implemented a mathematical tool to quantitatively measure the influence of noise in the perturbed weights on the performance of the network. We demonstrated numerical results indicating the effects of noise on the network and its performance sensitivity to such process.
3. We have introduced a hybrid genetic algorithm for the automatic evolution of neural network architectures. This method, although providing a search in a multi-modal search space for all possible solutions is capable with minimal training data to produce an output with small or no overfitting of the output data, which is a common problem in the field of neural networks.

References:

- [1] Warren S McCulloch and Walter Pitts. „A logical calculus of the ideas immanent in nervous activity“. In: The bulletin of mathematical biophysics 5.4 (1943), pp. 115–133
- [2] Frank Rosenblatt. „The perceptron: a probabilistic model for information storage and organization in the brain.“ In: Psychological review 65.6 (1958), p. 386
- [3] Gaurang Panchal, Amit Ganatra, YP Kosta, and Devyani Panchal. „Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers“. In: International Journal of Computer Theory and Engineering 3.2 (2011), pp. 332–337
- [4] Kurt Hornik. „Approximation capabilities of multilayer feedforward networks“. In: Neural networks 4.2 (1991), pp. 251–257
- [5] Kurt Hornik. „Some new results on neural network approximation“. In: Neural Networks 6.8 (1993), pp. 1069–1072
- [6] Věra Krková. „Kolmogorov’s theorem and multilayer neural networks“. In: Neural networks 5.3 (1992), pp. 501–506
- [7] Ken-Ichi Funahashi. „On the approximate realization of continuous mappings by neural networks“. In: Neural networks 2.3 (1989), pp. 183–192
- [8] Travis Dierks, Bryan Brenner, and Sarangapani Jagannathan. „Neural network-based optimal control of mobile robot formations with reduced information exchange“. In: Control Systems Technology, IEEE Transactions on 21.4 (2013), pp. 1407–1415
- [9] Shuai Li, Sanfeng Chen, Bo Liu, Yangming Li, and Yongsheng Liang. „Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks“. In: Neurocomputing 91 (2012), pp. 1–10
- [10] Jia Deng, Wei Dong, Richard Socher, et al. „Imagenet: A large-scale hierarchical image database“. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE. 2009, pp. 248–255
- [11] Stjepan Oreski, Dijana Oreski, and Goran Oreski. „Hybrid system with genetic algorithm and artificial neural networks and its application to retail credit risk assessment“. In: Expert systems with applications 39.16 (2012), pp. 12605–12617
- [12] Arash Bahrammirzaee. „A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems“. In: Neural Computing and Applications 19.8 (2010), pp. 1165–1195
- [13] George A Rovithakis and Manolis A Christodoulou. Adaptive control with recurrent high-order neural networks: theory and industrial applications. Springer Science & Business Media, 2012
- [14] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. „Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks“. In: arXiv preprint arXiv:1506.05751 (2015)

- [15] Jon Fingas. Facebook and Google get neural networks to create art. 2015. URL : <http://www.engadget.com/2015/06/20/facebook-and-google-ai-image-creation/>
- [16] Sung Hyun Jo, Ting Chang, Idongesit Ebong, et al. „Nanoscale memristor device as synapse in neuromorphic systems“. In: *Nano letters* 10.4 (2010), pp. 1297–1301
- [17] James Barrat. *Our final invention: Artificial intelligence and the end of the human era*. Macmillan, 2013
- [18] Simon S Haykin, *Neural networks and learning machines*. Vol. 3. Pearson Education Upper Saddle River, 2009
- [19] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995
- [20] Friedrich Beck and John C Eccles. „Quantum aspects of brain activity and the role of consciousness“. In: *How the SELF Controls Its BRAIN*. Springer, 1994, pp. 145–165
- [21] Roger Penrose. *The emperor’s new mind: Concerning computers, minds, and the laws of physics*. Oxford University Press, 1999
- [22] Abraham Goldberg, Harry M Schey, and Judah L Schwartz. „Computer-generated motion pictures of one-dimensional quantum-mechanical transmission and reflection phenomena“. In: *American Journal of Physics* 35.3 (1967)
- [23] A Aldo Faisal, Luc PJ Selen, and Daniel M Wolpert. „Noise in the nervous system“. In: *Nature Reviews Neuroscience* 9.4 (2008), pp. 292–303
- [24] G Bard Ermentrout, Roberto F Galán, and Nathaniel N Urban. „Reliability, synchrony and noise“. In: *Trends in neurosciences* 31.8 (2008), pp. 428–434
- [25] Gustavo Deco, Viktor Jirsa, AR McIntosh, Olaf Sporns, and Rolf Kötter. „Key role of coupling, delay, and noise in resting brain fluctuations“. In: *Proceedings of the National Academy of Sciences* 106.25 (2009), pp. 10302–10307
- [26] Richard B Stein, E Roderich Gossen, and Kelvin E Jones. „Neuronal variability: noise or part of the signal?“. In: *Nature Reviews Neuroscience* 6.5 (2005), pp. 389–397
- [28] Costas A Anastassiou, Rodrigo Perin, Henry Markram, and Christof Koch. „Ephaptic coupling of cortical neurons“. In: *Nature neuroscience* 14.2 (2011), pp. 217–223
- [29] Scott Kirkpatrick, C Daniel Gelatt, Mario P Vecchi, et al. „Optimization by simulated annealing“. In: *science* 220.4598 (1983), pp. 671–680
- [30] Pavel Kordík, Jan Koutník, Jan Dřchal, et al. „Meta-learning approach to neural network optimization“. In: *Neural Networks* 23.4 (2010), pp. 568–582
- [31] Vittorio Maniezzo. „Genetic evolution of the topology and weight distribution of neural networks“. In: *Neural Networks, IEEE Transactions on* 5.1 (1994), pp. 39–53
- [32] Kenneth O Stanley and Risto Miikkulainen. „Efficient evolution of neural network topologies“. In: *Evolutionary Computation, 2002. CEC’02. Proceedings of the 2002 Congress on*. Vol. 2. IEEE. 2002, pp. 1757–1762
- [33] Xin Yao and Yong Liu. „A new evolutionary system for evolving artificial neural networks“. In: *Neural Networks, IEEE Transactions on* 8.3 (1997), pp. 694–713
- [34] Dirk Thierens and David Goldberg. „Convergence models of genetic algorithm selection schemes“. In: *Parallel Problem Solving from Nature — PPSN III: International Conference on Evolutionary Computation The Third Conference on Parallel Problem Solving from Nature Jerusalem, Israel, October 9–14, 1994 Proceedings*. Ed. by Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 119–129



БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ

АВТОРЕФЕРАТ НА ДИСЕРТАЦИЯ

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

НОВИ ОПТИМИЗАЦИОННИ СТРАТЕГИИ И ЕВОЛЮЦИОННИ АРХИТЕКТУРИ ЗА ОБУЧЕНИЕ НА НЕВРОННИ МРЕЖИ

Кристина Георгиева Капанова

**Ръководители: Проф. Иван Димов,
Доц. Жан Мишел Селие**

Научно жури:

**Акад. Иван Попчев
Чл. Кор. Красимир Атанасов
Проф. Иван Димов
Проф. Стефка Фиданова
Доц. Олимпия Роева**



**Институт по информационни и
комуникационни технологии
Секция „Паралелни алгоритми“**

Дисертационният труд е обсъден и допуснат до защита на разширено заседание на секция "Паралелни алгоритми" на ИИКТ-БАН, състояло се на 15 ноември 2016 г.

Настоящата дисертация се състои от 6 глави. Дисертационният труд е оформен в 120 страници, включващи 29 фигури, 5 таблици и 180 цитирани източника.

Защитата на дисертацията ще се състои на ... 2017г. от 14:00 часа в зала 218 на Института по Информационни и комуникационни технологии - БАН на открито заседание на научно жури в състав:

1. Акад. Иван Попчев
2. Чл. кор. Красимир Атанасов
3. Проф. дтн. Иван Димов
4. Проф. Стефка Фиданова
5. Доц. д-р Олимпия Роева

Материалите за защитата са на разположение на интересующите се в стая 111А на ИИКТ-БАН, ул. „Акад. Г. Бончев“, блок 25А град София.

Автор: Кристина Георгиева Капанова

Заглавие: Нови оптимизационни стратегии и еволюционни архитектури за обучение на невронни мрежи

Обща характеристика на дисертационния труд

Актуалност и съвременно състояние на темата

Обект на изследванията в дисертационния труд са изкуствените невронни мрежи, които могат да бъдат описани като разпределен успореден процесор, който съдържа елементи за обработка на информация. Първоначалните изследвания за изкуствени невронни мрежи са стимулирани от човешкия мозък, който може да бъде разгледан като комплексна и паралелна, информационно-обработваща нелинейна система. Невронната мрежа има две основни сходства с човешкия мозък: знанието, необходимо за функционирането на мрежата е придобито от околната среда в процеса ѝ на обучение, а за съхраняване на тези знания са пригодени синапсови тегла. Методът, използван за процеса на обучение е наричан обучаващ алгоритъм.

Невронните мрежи могат да бъдат както линейни, така и нелинейни. Възможността за нелинейност се отличава с това, че е разпределена по мрежата, и отговаря за начина на формиране на входният сигнал, който също може да е нелинеен. Основно преимущество в работата на невронните мрежи е обучението с учител (supervised learning), при който се описва входната информация в изходна (input-output mapping). За постигането се променят синаптичните тегла на основата на определен набор от учебни примери. В това отношение, всеки учебен пример се състои от входен сигнал и съответстващия му желан отговор. По време на обучителния процес мрежата изменя синаптичните тегла с цел намаляване на отклонението на желания изходен сигнал съгласно предварително избрани статистически критерии. Обучението с учител може да бъде разгледано и като оптимизационна задача.

Способността за адаптиране на теглата на връзките между невроните според околната среда посредством обучителен алгоритъм допринасят за функционирането на мрежите в различни среди благодарение на адаптивността на основните параметри. Тази способност за адаптация е особено важна и при изпълнението на невронните мрежи под формата на електроника. По този начин, дори и при увреждане на определен брой неврони или синаптични връзки, производителността пада незначително. Тяхната успоредна структура също така потенциално може да ускори решението на някои видове задачи.

Качество на невронните мрежи е техният универсален механизъм при обработката на информацията, което позволява една архитектура на невронната мрежа да се използва в множество различни области.

Днес изкуствени невронни мрежи биват прилагани в различни изследователски и приложни сфери. Един от тях е в областта на роботиката, където мрежите се използват като начин да се ориентира манипулацията на обекти, въз основа на някои сетивни данни. Невронни мрежи се прилагат и за планиране на движения и управление на роботи. В областта на машинното зрение невронните мрежи се прилагат за извличане информация за околната среда чрез обработката на данни от колекция на изображения. Невронни мрежи са прилагани със значителен успех във финансовия и банков сектор. Те се използват за измерване на кредитния риск на кандидатите, за процедурата за оценка на кредита и прогнозиране на кредитните ставки, оценка на недвижими имоти, прогнозиране на финансовия пазар и валутния курс, както и за корпоративен финансов анализ.

В сферата на медицината, невронни мрежи са прилагани за анализ на раковите клетки, ЕЕГ и ЕКГ анализ, както и за контрол на качеството в болници.

Цели и задачи на дисертационния труд

Целта на настоящата дисертация е създаване на нов алгоритъм за обучение на изкуствени невронни мрежи, който веднъж изпълнен, позволява на мрежата да излезе от локален минимум или седловинни точки, в които е попаднал алгоритъма. Дисертацията представя и приложение на методи за анализ на чувствителността за изучаване на смущаването на теглата в мрежата, засегнати от шум в системата. Изследването на анализ на чувствителността може да е изключително полезно при хардуерно изпълнение на невронни мрежи, при които шум от различни източници влияе на производителността на невронната мрежа. Дисертационният труд цели и представянето на нов еволюционен алгоритъм за автоматичното генериране на архитектури на невронни мрежи чрез представянето на хибриден генетичен алгоритъм. По този начин, подходи на проба и грешка при създаването на топологии на невронни мрежи се заменя с автоматичното откриване на архитектура на невронни мрежи, подходящи за специфични задачи. Целта е подпомагането на процеса при създаването на изкуствени невронни мрежи и намаляването на човешка грешка в този цикъл.

Конкретните задачи за постигането на тези цели са:

1. Създаването на нов оптимизационен алгоритъм, инициализиран след приключване на обучителния процес на невронната мрежа;
2. Селектиране и практическо приложение на инструменти за анализ на чувствителността при наличие на шум в невронната мрежа;
3. Създаване на нов хибриден еволюционен алгоритъм, с цел автоматизирано изграждане на невронен модел, според определена задача;
4. Програмно осъществяване алгоритмите и инструментите;

Методология на изследването

Методологията на настоящите резултати се базират на изследвания в областите на изкуствения интелект, изкуствените невронни мрежи, оптимизационни методи, както и квантовата физика. Комбинирани са подходи, базирани на закони от физиката, както и на статистически анализ и машинно самообучение.

Използвани са математически индикатори, свързани с анализ на чувствителността при определянето нивото на шум в система, базирана на софтуерно или хардуерно осъществяване на невронна мрежа.

Създаване на хибриден метаевристичен алгоритъм за автоматична конструкция на невронни мрежи, позволяваща в някои случаи избягването на проблема за обобщението на информацията.

Проведени са числени експерименти, чиито резултати са представени и анализирани.

Осъществена е реализация на предложените алгоритми в този дисертационен труд на езика C, част от която е описана в Приложението на разработката.

Апробация на резултатите

Резултати, включени в дисертацията, са докладвани на:

- Семинар на секцията "Паралелни алгоритми 2016";

- Докторантски форум, център за обучение-БАН, 2016; (презентацията за хибридни еволюционен алгоритъм е отличена с награда на форума от организаторите)

Част от резултатите са представени на следните конференции:

- 10th International Conference on „Large-Scale Scientific Computations“ (LSSC'15), Созопол, България, 2015;
- 10th IMACS Seminar on Monte Carlo Methods (MCM 2015), Линц, Австрия, 2015;
- LinuxCon 2015, Дъблин, Ирландия, 2015;

Резултатите от дисертацията са публикувани в 3 статии в международни списания с импакт фактор.

Списък на публикациите по дисертацията

1. K.G. Karanova, I.T. Dimov, J.M. Sellier, On randomization of neural networks as a form of post-learning strategy, *Soft Computing* (2015). *doi* : 10.1007/s00500 – 015 – 1949 – 1, (IF.1.63).
2. K.G. Karanova, I.T. Dimov, J.M. Sellier, A Neural Network Sensitivity Analysis in the Presence of Random Fluctuations, *Neurocomputing* (2016), *doi* : 10.1016/j.neucom.2016.10. (IF.2.392)
3. K.G. Karanova, I.T. Dimov, J.M. Sellier, A genetic approach to automatic neural network architecture optimization, *Neural Computing and Applications* (2016), *doi* : 10.1007/s00521 – 016 – 2510 – 6, (IF.1.492)

Съдържание на дисертацията

Настоящата дисертация се състои от 6 глави. Дисертационният труд е оформен в 120 страници, включващи 29 фигури, 5 таблици и 180 цитирани източника.

Глава 1. Увод

В уводната глава на дисертацията е направен кратък исторически анализ на областта на изкуствения интелект, и специфично на невронните мрежи. Описани са основни разлики между невронните мрежи и изчислителните архитектури.

Глава 2. Основи на изкуствените невронни мрежи

Втора глава от настоящата дисертация цели въвеждането в основните понятия и принципи на работа на изкуствените невронни мрежи. Зараждането на идеята и описанието на изкуствена невронна мрежа за пръв път е публикувано през четиридесетте години на 20 век, представяйки опростен модел за работата на човешкия неврон. МакКълук и Питс [22], описват в статия математическата структура на опростен невронен модел, разгледан като прагов логически елемент (ПЛЕ).

В настоящата разработка, под невронни мрежи (НМ) се разбира математически модел, който се състои от отделни елементи, симулиращи в ограничен модел работата на естествения неврон в мозъка на човека. Невронните мрежи са изградени от отделни елементи, които обработват входящите сигнали, формирайки съответна изходна стойност (изходен сигнал), която се разпространява към елементите, с които е свързан. Можем да определим три основни елемента:

- множество от връзки (синапси), характеризирани посредством теглови коефициент w_{ij} , които могат да приемат положителни или отрицателни стойности.
- суматор, който сумира входните сигнали, умножени по съответния теглови коефициент. Най-широко разпространеният суматор е линейният.
- активационна функция (функция на изхода), чрез която се преобразува получения сумарен вход в изхода (изходния сигнал) на съответния неврон.

В математически вид, функционирането на неврон се описва по следния начин. Определяме $(x_1, x_2, x_3, \dots, x_i)$ като входни сигнали, w_i синаптически тегла на неврона, а w_i като връзка между невроните. Изходните данни (z) на неврона се определят като

$$z = \sum_i w_i x_i \quad (1)$$

и са определени като линейна сума на входните сигнали и синаптичните връзки. Като част от алгоритъма може да бъде определен и прагов елемент (bias), който се означава със символа (w_0), и може да бъде описан като:

$$z = w_0 + \sum_{i=1}^n w_i x_i \quad (2)$$

Използването на прага в този смисъл, предпазва ефекта на афинното преобразуване (affine transformation) на изходния линейен суматор.

В глава 2 са описани основните характеристики на невронните мрежи: параметрите на обработващите елементи, типът на входните и изходните стойности, активационните функции, както и топология на мрежата. Най-често срещаните типове активационни функции, които определят изходния сигнал на неврона са функция на единичен скок или прагова функция (threshold function)

$$f(z) = \begin{cases} 1 & : z \geq 0 \\ 0 & : z < 0 \end{cases} \quad (3)$$

както и Сигмоидална функция (sigmoid function), която може да бъде описана по следния начин:

$$y(z) = \frac{1}{1 + e^{\beta z}} \quad (4)$$

Сигмоидала е най-често използван при мрежи с обратно разпространение на грешката, където функцията, която е бързонарастваща, работи за поддържането на баланса между линейно и нелинейно поведение. За преодоляването на определени недостатъци на сигмоидалната функция, може да бъде използвана хиперболична функция (hyperbolic tangent-tanh), която функционира в по-голямо пространство $[-1, 1]$:

$$f(z) = \begin{cases} 1 & : z > 0 \\ 0 & : z = 0 \\ -1 & : z < 0 \end{cases} \quad (5)$$

Като част от втора глава са описани различни популярни топологии на невронни мрежи, като са представени и илюстрации за дизайна на невронни мрежи. Важна характеристика на невронните мрежи е тяхната мрежова архитектура (структура). Могат да се разграничат няколко фундаментално различни класа от мрежови архитектури. В най-опростения модел, мрежата има входен слой (input layer) на възлите на източника, който предава информацията на изходният слой (output layer) на невроните (изчислителният възел) но не и обратно. Тази мрежа е от право-разпространяващ тип.

Многослойна мрежа Многослойната невронна мрежа е вид невронна мрежа за пряко разпространение на сигнала и се характеризира с наличието на един или няколко скрити слоя (hidden layers), а невроните, групирани в тези слоеве се наричат скрити неврони (hidden neurons) или скрити елементи (hidden units) [3]. Под термина "скрит" се разбира слой или неврон, който не се вижда пряко от входа или изхода на мрежата, като целта на скритите елементи е да допринасят за способността на мрежата да извлече от изхода статистики от по-висок ред. Връзките между невроните във слоевете могат да бъдат описани като насочен граф и обикновено са напълно свързани [12], [6]. Насоченият граф описва не само протичането на сигнала между невроните, но и как сигнала се предава в съответния неврон. Невронната мрежа за пряко разпространение може да бъде описана по следния математически модел, където l обозначава броя слоеве в мрежата, net_j^l описва j -ия неврон в слоя l . Синаптичните тегла от i -ия неврон в слой $l - 1$ до j -ия неврон на слой l са представени като $w_{ij}^{l-1,l}$, а изходната стойност на j -ия неврон в слой l -ия е представена като $f(net_j^l)$ активационна функция:

$$net_j^l = \sum_{i=1}^{n_{l-1}} w_{ij}^{l-1,l} y_i^{l-1}, y_j^l = f(net_j^l) \quad (6)$$

Определянето на броя скрити неврони е от изключително значение при структурирането на невронната мрежа, тъй като прекалено голямо количество неврони намаляват точността на обобщението на данните. Допълнителен проблем се състои от факта, че голямото количество неврони служи за автоматичното запомняне на обучителните данни, което премахва способността за качествено обобщение на информацията. От друга страна, недостатъчен брой неврони допринася за намаляване на обучаващата способност на мрежата и откриването на комплексни гранични стойности на проблема. Точният брой неврони във мрежата, както и във всеки скрит слой не е обоснован и е строго зависим от специфичната задача, обучителен алгоритъм и от специфични решения на учения.

Общоприет подход към проблема е прилагането на теоремата на Колмогоров [20], в която се посочва, че за изчислението на произволна непрекъсната функция, трябва да се осигури брой неврони, равен на два пъти броя на входните възли плюс още един. Допълнителен метод е изчислението на броя на скритите неврони, описани чрез $НМ$, при наличие на брой неврони във входен слой k и изходен слой v , където

$$НМ = \frac{1}{2}(k + v). \quad (7)$$

Рекурентни мрежи За разлика от невронна мрежа с право разпространение на сигнала, рекурентните мрежи имат поне една обратна затваряща се връзка [11]. Този тип мрежи могат да бъдат съставени и само от един слой неврони, които връщат

изходните сигнали обратно на входовете на всички неврони в мрежата. При наличие на самозахранваща се обратна връзка, изходът на неврона е върнат обратно на собствения вход.

Дълбоки мрежи Описаните до тук невронни мрежи, са със сравнително опростена архитектура, към която понякога се отнася термина плитка архитектура (shallow architecture), съставена от множество неврони в няколко невронни слоя. Увеличението на наличните данни и данни от по-висок ред предразполага към създаването на дълбоки невронни мрежи (deep neural networks-DNN), които прилагат повече от два скрити слоя на конвенционална многослойна мрежа [24].

В настоящата работа е приложена многослойна невронна мрежа с пряко разпространение на сигнала. Обобщението на този раздел налага и до разглеждането на един от обектите на този дисертационен труд: обучителния процес при невронните мрежи.

Глава 3. Проблемът на обучение на невронна мрежа и изпълнение на нова стратегия след обучение

В глава 3 обучителният процес на невронните мрежи е разгледан като оптимизационен процес и е предложен нов пост-обучителен алгоритъм. Главата е структурирана в два основни раздела. В първия раздел са описани основни оптимизационни алгоритми за обучението на мрежи. Във втората част е представен нов модел на алгоритъм, както и описание на проведените числени експерименти.

В първата част на трета глава са описани основни обучителни стратегии за обучение на невронни мрежи. Основно свойство на невронните мрежи е способността им да се обучават (learn) на основа на предоставени данни, в резултат на което да подобряват своята производителност по време на процеса посредством определени правила. Съществуват различни алгоритми за обучение, всеки от който, има предимства и недостатъци. Цялостна класификация на методите и приложенията използвани за обучение би била непълна. Различните видове алгоритми могат да бъдат разпределени в две основни групи: обучение с учител и обучение без учител [13]. В първата група попадат алгоритми, които следят за разликата между получения и очаквания изход на мрежата (в този случай очаквания изход на мрежата е предварително зададен от учителя), като итеративно системата извършва корекции на синаптичните тегла съответстващо на предварително избран оптимизационен процес. За разлика от алгоритмите с учител, тези без учител не разчитат на предварителни данни за точния изход на мрежата. Промяната на синаптичните тегла се осъществява по начин, който да позволи на данните да бъдат представени така, че да отговарят на предварително зададен критерии за качеството на представянето и способа на моделиране на параметрите. Могат да бъдат определени няколко последователни стъпки по време на обучителния процес: в невронната мрежа постъпват данни от околната среда в резултат на което се променят свободните параметри на невронната мрежа. Това допринася за изменение на възбуждането на невронната мрежа в следващи стъпки.

В тази глава обучителните стратегии са разгледани като оптимизационен проблем, при който се избира ефективна стратегия за намиране на минималните или максималните стойности на функция, състояща се от много независими променливи. При невронните мрежи тази функция е целевата функция (e.g. cost function, energy or objective function) и пряко зависи от предварителната конфигурация на различни

части от мрежата. В следствие, оптимизационния алгоритъм се стреми да достигне най-доброто състояние на системата, в съответствие с обективната функция. Целият процес се повтаря, докато системата е неспособна да намери следващо подобрене на резултата или е достигнала максималния размер на позволените итерации. При обучението с учител, проблемът се определя като минимизиране на функцията, при което се цели грешката да е близост до или равна на нула.

При стартиране на процеса на оптимизация, синаптичните тегла са произволно определени според предварително зададена стратегия. Следващата стъпка въвежда обучаващото множество (x_i, y_i) , където x_i представлява входящ сигнал, а y_i очакван изходен сигнал. Мрежата изчислява решението z , което първоначално значително се различава от желания резултат, дефиниран чрез y_i . По време на обучителния период, теглото на синаптичните връзки се променя, за да се сведе грешката до минимум.

Описани са няколко често използвани целеви функции, които са описани в таблица).

Name	Cost Function
L_2	e^2
L_1	$ e $
L_p	$\frac{1}{p} \cdot e ^{\frac{1}{p}}$
<i>Logistic</i>	$\frac{1}{\alpha} \cdot \log(\cosh(\alpha \cdot e))$

Най-използваните функции на грешката. α контролира достоверността на големите различия в стойностите, желания сигнал е обозначен с d , действителното резултат от z и сигнал за грешка като e , при $e = d - z$

Алгоритмите за обучение на невронни мрежи с учител са съсредоточени върху минимизирането на целевата функция за решаването на определена задача. За успешното прилагане на обучителни алгоритми те трябва да функционират с минимална изчислителна тежест. Необходимо е те да са стабилни в присъствието на шум в системата, да са в състояние да произведат резултати, слабо зависими от началните условия. Желателно е изчислителната сложност на алгоритъма да не бъде силно зависима от размерността и размера на данните приложени по време на обучението. Под внимание трябва да се вземат наличие на подходящи критерии за спиране на процеса, за откриването на подходящ резултат и възможна сходимост.

Настоящата глава описва различни видове обучителни алгоритми. Първият от тях е с обратно разпространение на грешката. При този вид алгоритъм, множество итерации са необходими за достигане на решение, в рамките на всяка от които, се разглеждат един или всички примери от обучаващото множество. За всеки такъв пример (x_i, y_i) , се постига право разпространение на активационните стойности и следва обратен анализ на грешките на изхода и разпространение на корекциите на теглата на връзките между елементите от различните слоеве. Този алгоритъм има ниска степен на сходимост, която съответно е и с ниска скорост на обучение, както и необходимост от голям брой обучаващи примери.

При този метод е налична многомерна повърхност за грешки (error surface) при координатите на свободните параметри. Реалната повърхност на грешките се усреднява (averaged) по всички възможни обучителни примери, представени във вид на

двойки „вход–изход“ (x_i, y_i) . Повишаването на производителността на мрежата във времето се осъществява при изместване към страната на максимума на повърхността на грешките. Под максимум се разглежда както локален, така и глобален. За да се постигне е необходимо системата да има информация за градиента на повърхността на грешките, който отговаря на текущото поведение на мрежата. Градиента на повърхността за грешки във всяка точка е вектор определящ най-бързото спускане на тази повърхност [23].

Генетичен алгоритъм - адаптивен евристичен алгоритъм за търсене, основан на еволюционните идеи за естествен подбор [14]. Алгоритъмът представя възможност за използване на случайно търсене в дефинираното пространство на търсене за решаването на даден проблем. Алгоритъмът се стартира с множество от решения, наречени популация. На всяка стъпка, решенията от една популация се вземат и използват от новата популация. Основните параметри на генетичните алгоритми включват вероятност за кръстосване и вероятност за мутация. Първата е свързана с честотата на извършване на кръстосването. Втората взема под внимание колко често части от хромозома ще мутират. Основната функция на мутацията е да предпази алгоритъма от попадане в локален екстремум, но трябва да се прецени честота на използване, тъй като по този начин се променя произволното търсене. Основен проблем е решението за размера на популацията или колко хромозоми да бъдат включени в едно поколение. При липса на достатъчно хромозоми, алгоритъма е с лимитирани възможности за извършване на кръстосване, което води до намаляване на обсега за изследване на пространството на търсене. От друга страна, при прекалено голямо количество хромозоми, алгоритъма ще забави скоростта на сходимост.

Симулирано закаляване - алгоритъма има своето начало от статистическата механика (алгоритъм Метрополис) и използва явна стратегия за избягване от локалните минимума [18]. Алгоритъмът действа чрез симулация процеса на закаляване на метал или стъкло, при което при подходящ план на охлаждане конфигурацията има ниска енергия. За избягването на локалните минимума алгоритъма позволява ходове към решения с по-високи стойности на обектната функция, като такъв ход се счита за “изкачване на хълма” (uphill move). Вероятността за ход към дадена точка се определя чрез

$$Pr[accept] = e^{-\frac{\Delta E}{T}}, \quad (8)$$

където ΔE е разликата между реалната енергия и тази преди хода, и T е ефективната температура на системата. Следователно ходът бива приет при $[0, 1] \ni R < Pr[accept]$.

Други два евристични алгоритъма, използвани за обучителни стратегии, които са разгледани в дисертационния труд включват алгоритъм на мравката [7] и метод на роящите частици [16], [17]. Пример е даден и с квантово симулирано закаляване [15]. Разглеждането на обучението на невронните мрежи като оптимизационен проблем за минимизиране на функцията на грешката, позволява въвеждането на множество локални и евристични алгоритми в областта. Въпреки това, понякога алгоритмите попадат в локален минимум, или в седлови точки, което възпира точното обучение на мрежата.

Нова обучителна стратегия

Втора част на трета глава е посветена на представянето на новия пост-обучителен процес.

В тази част е описан нов пост-обучителен процес, вдъхновен от работата в [2]. Методът имитира квантовите ефекти и по този начин е в състояние да осигури подобряване на теглото на синаптичните връзки в един пост-обучителен процес. Целта е създаването на метод, който да допринесе за нивото на точност на мрежата дори при евентуален неуспех на обучителния процес - дали поради липсващи данни, недостатъчна архитектура или поради попадане на оптимизационния алгоритъм в локален минимум или седловина. Тази нова стратегия е постигната при относително ниска изчислителна тежест. Предложението за прилагане на квантовите механични закони във всеки неврон кореспондира със състоянието на симулация на уравнението на Шрьодингер в зависимост от времето метод (или друг еквивалентен формализъм като на Файнман, Вигнър). Това е необходимо, за да бъде числено установен евентуален тунелен преход. Следователно, мрежата трябва числено да симулира следното нестационарно частно диференциално уравнение за всеки неврон:

$$i\hbar\frac{\partial\Phi}{\partial t}(\mathbf{r},t) = \left(-\frac{\hbar^2\nabla^2}{2m} + V(\mathbf{r})\right)\Phi(\mathbf{r},t), \quad (9)$$

където i е имагинерната единица, $\Phi(\mathbf{r},t)$ е вълновата функция дефинирана чрез пространство и време, \hbar е константа на Планк, r обозначава позицията на частицата, t е времето, m е масата на частицата, ∇^2 е оператор на Лаплас, и $V(\mathbf{r},t)$ представлява потенциална енергия, която влияе на частицата.

В случай на относително компактни невронни мрежи, състоящи се от един скрит слой и ограничено количество неврони, тази симулация ще бъде изчислително достъпна. В реални приложения на невронни мрежи като тези, използвани за акустичен анализ или разпознаване на реч, в която се използват стотици хиляди неврони, това представлява комплексна и изчислително непосилна задача. Ето защо, представяме изчислително удобна техника, която да имитира наличието на случайността, основна в квантова система, без алгоритъма да страда от високи изчислителни разходи, свързани с наличните и точни квантови симулации.

В този случай, моделираме неврон като хетерогенен полупроводник, който се състои от една енергийна бариера (например AlGaAs), която е разположена между две енергийно по-ниски области (например GaAs). [2] дава възможност за разглеждането на активиращата функция на неврона като една или повече частици, взаимодействащи с бариерата (виж фиг. 1), след навлизането им в хетероструктурата. Вероятността частицата в даден момент t да бъде открита в дадена точка от устройството е описана от вълновата функция $|\Phi(\mathbf{r},t)|^2$. По този начин, случайност е въведена в процеса (правило на Борн). Ако вероятността за тунелен преход е по-ниска от вероятността за обратно разсейване, функцията за активиране се счита за инхибиторна на неврона.

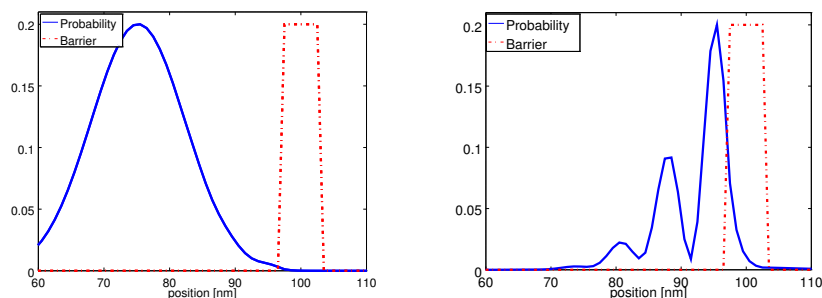
В практическо изпълнение, това може да бъде постигнато чрез добавяне към алгоритъма нова функция, описана като **addHiddenNoise**, която да прибави шум към вече изчислените тегла след процеса на обучение. Това се постига само за неврони в скрити слоеве на мрежата.

$$hiddenLayer.neurons[i].weights[j]+ = randomDouble() * weightNoise \quad (10)$$

което отговаря на

$$f(w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_nx_n). \quad (11)$$

Функцията на **RandomDouble** е да ограничи добавянето на шум в определен диапазон, зависещ от спецификата на задачата.



Фигура 1 Лева фигура: вълнова частица, описана чрез синя непрекъснатата линия, се движи към енергийна бариера, описана в червена прекъснатата линия. Дясна графика: след определен времеви период, вълнова частица взаимодейства с бариерата. Част от частицата се разсейва обратно, докато останалата част преминава през бариерата.

Функцията *backupState* е използвана за копиране на теглата и евентуалното им възстановяване при случаи, в които системата предлага решения, по-лоши от достигнатите.

Числени експерименти

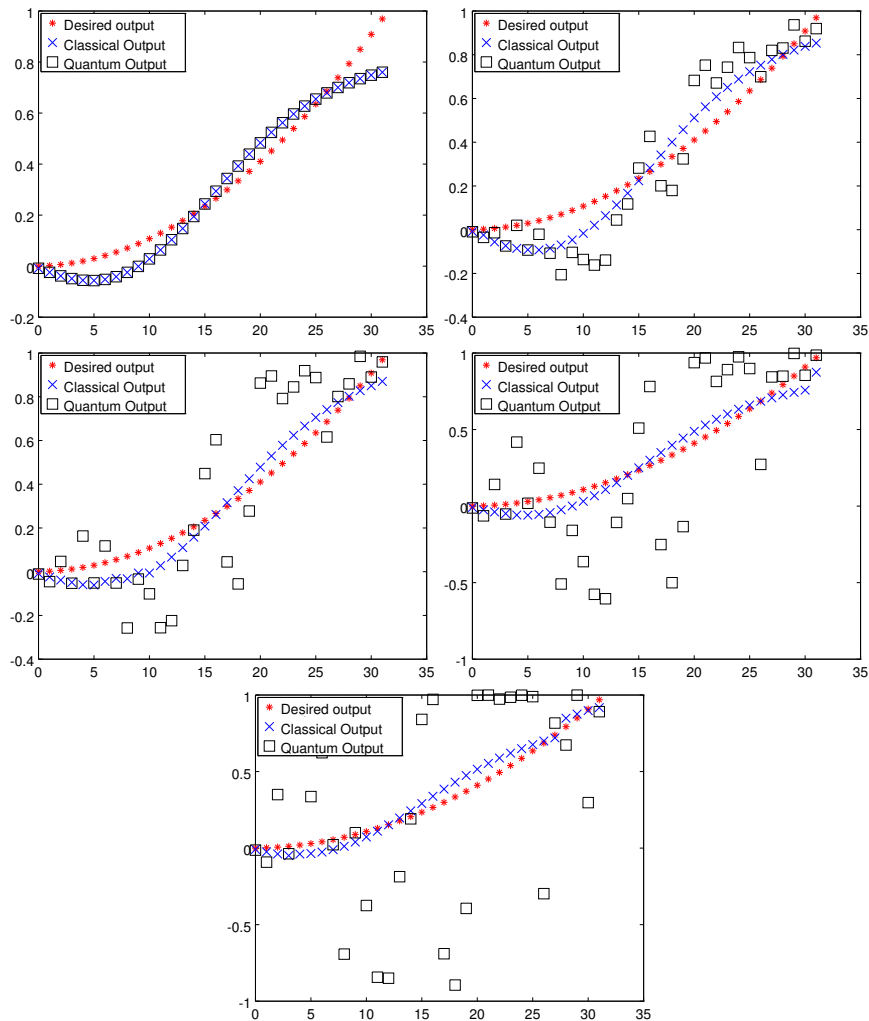
В секция числени експерименти са представени експерименти за валидация на пост-обучителната стратегия. Примери са дадени за две функции: полином от втора степен ($f(x) = x^2$) и корен квадратен на полином ($f(x) = \sqrt{x}$). Системата използва три точки за обучение. Архитектурата на система е идентична и за двете функции - три слоя - един входен с един неврон, един скрит с 4 неврона, и един изходен с един неврон. Допълнителни органичения са поставени на пространството за търсене на теглото. За първия пример пространството за търсене е ограничено в рамките $-12(min)$ и $+12(max)$, а за втората между $-1(min)$ и $+1(max)$.

Обучителният алгоритъм е симулирано закаляване. В настоящите примери, съзнателно спираме обучението на мрежата в произволен локален минимум посредством скоростта на температурното намаление, като това е достигнато по не-оптимален начин. Целта е да се покаже ясно, че нашата техника може да осигури начин за по-нататъшно подобряване на обучението, дори и след процеса на оптимизация.

За да се разбере степента на влияние на шума върху производството на мрежата и намаляването на обектната функция са изследвани 5 различни сценария (виж фиг. 2). Първо тестваме мрежата без прибавяне на шум, за сравняване на квантовата част от мрежата с класическата част. Останалите експерименти включват работа на мрежата при добавяне съответно на 0.5%, 1%, 2%, и 4% шум.

Добавянето на 0.5% шум има минимален ефект върху системата, влияещ на първите няколко резултата. За всяка точка, алгоритъмът сравнява нивото на грешката от класическата и квантовата част, избирайки по-качествения резултат и отхвърляйки другия. При наличие на по-добро квантово решение, мрежата го приема и продължава в следващата итерация. Умереното увеличение на шума до 1% предлага решения от мрежата, които са добри в горната и долната граница на кривата и по-неточни в средните стойности.

Точността на мрежата се влошава при прилагане на 2% шум. Въпреки това, с добавяне на шум, квантовата мрежа превъзхожда класическата в няколко случая (виж фиг. 3 за нивото на грешката). По-високите нива на шум все пак постигат малки



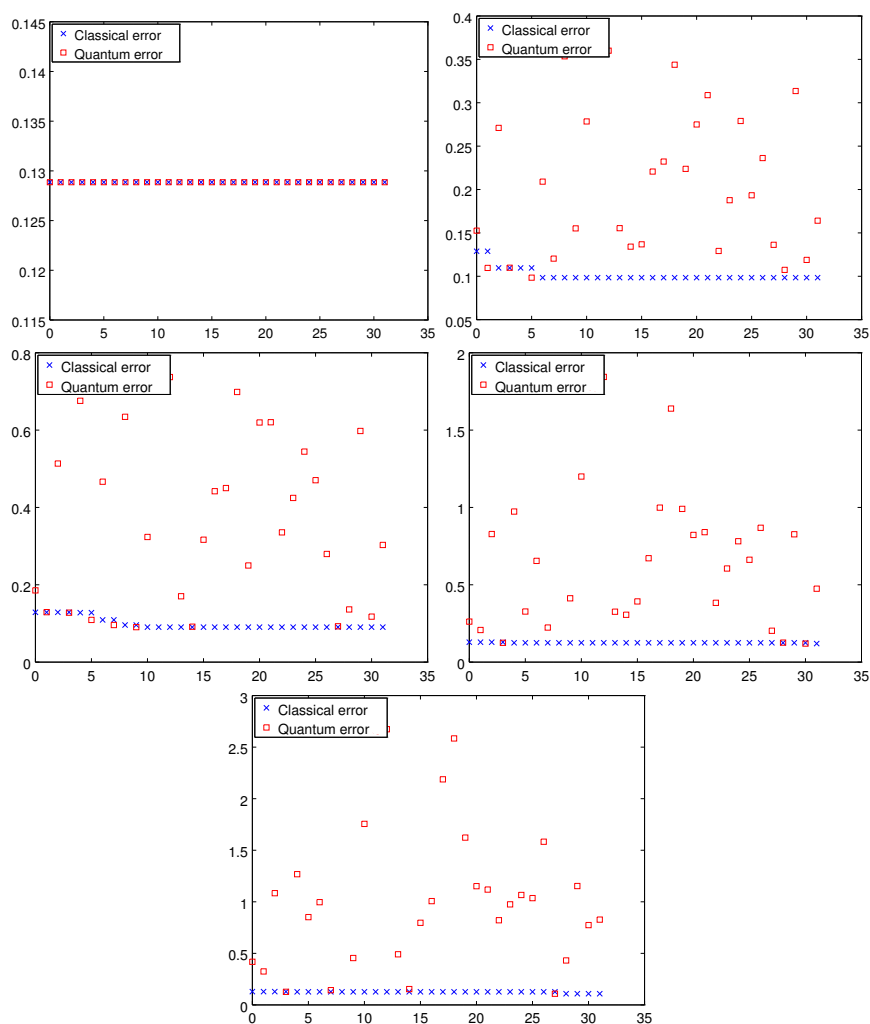
Фигура 2 Графиките се отнасят за решаване на полином от втора степен. Символът червена звезда обозначава желанния изходен сигнал на мрежата, синия x показва изходен сигнал от класическата част, а квадрата описва квантовото решение. Първата графика е валидация тест, при който няма добавяне на шум. Най-горе в дясно са резултатите от мрежата при 0.5% шум. Средните лява и дясна графика описват резултати при прибавяне на 1% и 2% шум. Последната графика е относно резултати при прибавяне на 4% шум.

подобрения в резултатите на мрежата, но при много по-малка степен и дълбочина.

При втората функция (виж фиг. 4)- за решаване на корен квадратен на полином, се забелязват значителни разлики. Тук, моделът на намаляване на грешката е стъпаловиден, което прилича на този в подобна ситуация за първата функция. При тази настройка, само няколко по-добри резултати са отбелязани, независимо от стръмното спускане на намаляване на грешката от точка до точка.

Симулацията показва, че 2% шум допринася за повишаване на ефикасността на квантовата мрежа. Мрежата осигурява два пъти повече резултати от квантовата мрежа, сравнено с прилагане на 0.5% или 1% шум, като следствие е и много по-ниските нива на грешни решения (виж фиг. 5 за нивото на грешката).

Едно от възможните обяснения за разликата на производителността на мрежата за двата числени експеримента може да бъде предоставена от промяната на границите на пространството за търсене на теглата на връзките при двете функции.

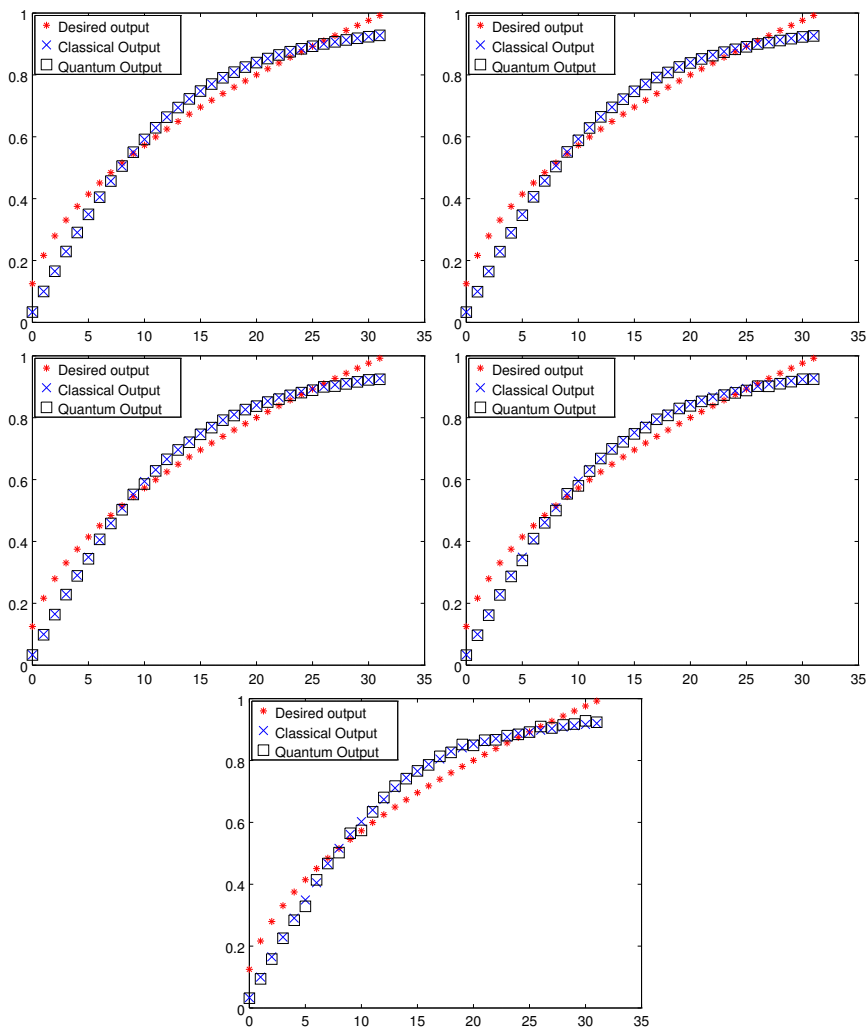


Фигура 3 Намаляване на грешката в случай на полином от втора степен. Грешката от класическата част от мрежата е изобразен чрез (син) , докато грешката от квантовата страна на мрежата е показан като (червен) квадрат. Най-горе ляво графиката се отнася за грешка при 0% шум с цел валидация. Горната дясна графика илюстрира грешка на мрежата в сценарий с 0.5% шум. Средните графики се отнасят съответно за привабяване на 1% и 2% шум. Последната графика описва грешката на мрежата при 4% шум

Глава 4. Анализ на чувствителността

В четвърта глава са представени инструменти за анализ на чувствителността на невронна мрежа, за установяване влиянието на шума в мрежата. Изследвания за влиянието на случайни или непредсказуеми колебания и смущения и потенциалните им ползи за невронни системи [8, 9] са проведени. От инженерна гледна точка, шумът се идентифицира като вреден за системата и за качеството на резултатите. В биологични нервни системи, от друга страна, шумът е естествено следствие на системата, като се смята, че предоставя определени ползи за анализ на информацията.

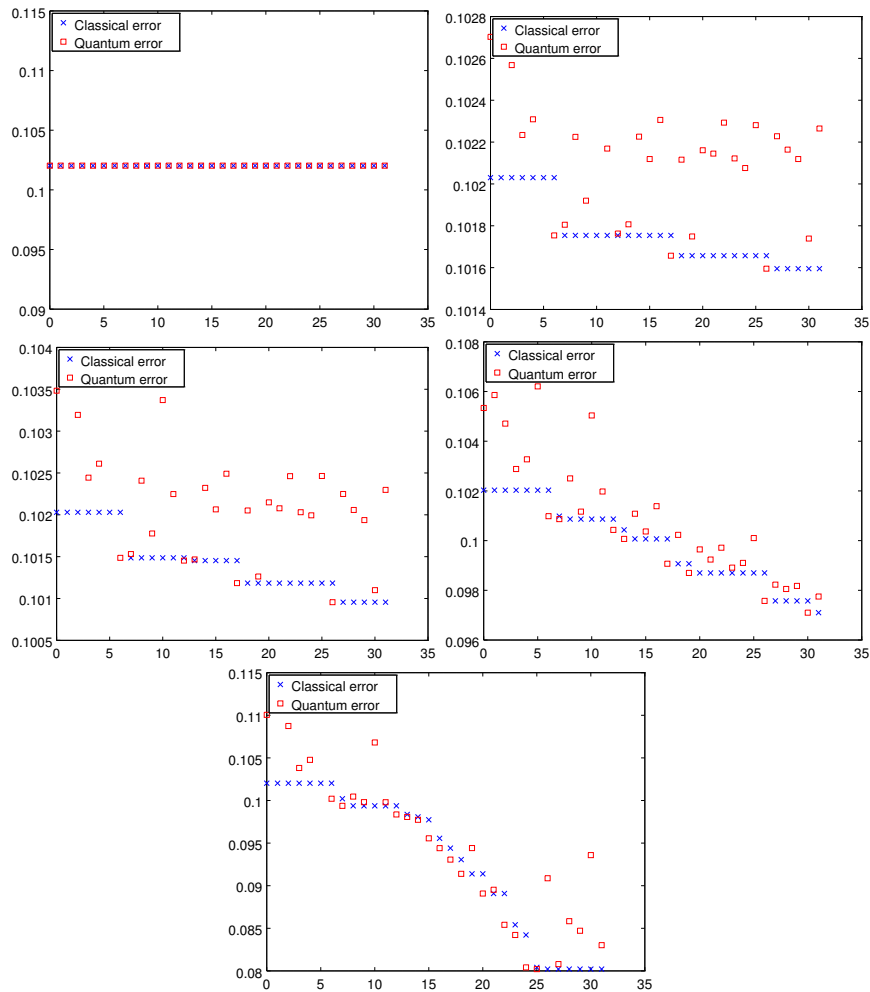
В тази глава представяме методология за анализ на чувствителността на невронна мрежа при присъствието на шум, добавен във всяко синаптично тегло от мрежата. Целта на тези техники за анализ на чувствителността е да предостави математически подход за определяне на нивата на шум в дадена мрежа. Чрез инструментите



Фигура 4 Графиките се отнасят за решаване на корен квадратен на полином. Символът червена звезда обозначава желания изходен сигнал на мрежата, синия x показва изходен сигнал от класическата част, а квадрата описва квантовото решение. Първата графика е валидиращия тест, при който няма добавяне на шум. Най-горе в дясно са резултатите от мрежата при 0.5% шум. Средните лява и дясна графика описват резултати при прибавяне на 1% и 2% шум. Последната графика е относно резултати при прибавяне на 4% шум.

установяваме възможно ли е определено количество шум да се използва за подобряване на производителността на мрежата, а когато се влияе негативно - как действа шума като източник на вариации. Чрез инструментите на анализ на чувствителността може да бъде установено влиянието на променливия шум на поведението на мрежата и каква е чувствителността на резултатите от прилагането на съответното количество шум. Използваните индикатори за анализ на чувствителността показват какво е възможното отклонение на резултатите от системата поради размествания на параметрите на синаптичните тегла. Функционалността на мрежата може да бъде значително променена с изменение на шума в теглата и следователно ще има значителна чувствителност към системата. И обратно, с незначителна чувствителност, производителността на мрежата не трябва да бъдат променени значително.

Използваните индикатори за анализ на чувствителността предоставят статистичес-



Фигура 5 Намалване на грешката в случай на корен квадратен на полином. Грешката от класическата част от мрежата е изобразен чрез (син) \times , докато грешката от квантовата страна на мрежата е показан като (червен) квадрат. Най-горе ляво графиката се отнася за грешка при 0% шум с цел валидация. Горната дясна графика илюстрира грешка на мрежата в сценарий с 0.5% шум. Средните графики се отнасят съответно за привабяване на 1% и 2% шум. Последната графика описва грешката на мрежата при 4% шум

ки данни, че добавянето на умерено количество шум в действителност съдейства за подобрение на производителността на мрежата. Това се доказва от няколко използвани показатели, които да установят обхвата на възможните решения. Използвани са Евклидовото разстояние (L_2), Чебешева норма (L_∞), и косинусова близост (L_{cos}). При наличие на два вектора на синаптичното тегло, избрани от пространството на възможни решения, косинусова близост предоставя информация за тяхната ориентация, а Евклидово разстояние произвежда възможни посоки и свойства на избрани тегла. L_∞ индикаторът представлява параметър на максимално разминаване. Всеки от изчислителните експерименти е проведен с еднакви начални условия, за да може да се осигури сравнимост на резултатите.

Чрез анализа на чувствителността се цели постигането на по-качествено разбиране относно функционирането на мрежата при въведен шум в пространството на синаптичните тегла. Също така е установена каква е чувствителността на изходните

результати при въвеждането на шум в теглата на връзките. Важно е отбелязването, че шумът е разглеждан като шумова вариация и следователно в някои настройки нивото на шум не води до подобрение на резултатите.

Първият показател, на базата на L_{cos} , се въвежда като

$$l_{cos}(w'') = l_{cos}[u(w''), v] = \frac{u \cdot v}{|u||v|}, \quad (12)$$

където \cdot е продукта и $|u|$ е абсолютната стойност на вектор u . L_{cos} също така може да бъде негативен за вектори, които са в противоположни посоки на N_x пространство на изходните вектори. Това е добър показател за сходство между два вектора. В действителност, когато два вектора сочат в една и съща посока и имат сравними абсолютни стойности, L_{cos} е близо до 1, когато са ортогонални тогава $L_{cos} = 0$, а при противоположност $L_{cos} = -1$. Може да се отбележи, че ако смущаването на тежестите не е твърде чувствителен (даден определен диапазон), тогава L_{cos} трябва да се приближава към 1.

Вторият индикатор, който е използван се основава на евклидовото разстояние и може да бъде описан като

$$I_2(w'') = I_2[u(w''), v] = |u(w''), v|_2 = \sqrt{\sum_{i=1}^{N_x} (u_i - v_i)^2}, \quad (13)$$

където u_i и v_i са i компоненти на вектори u and v . Този показател е метричен и може да се тълкува като разстоянието между началните тегла (u) и изменените след шума (v). Това е ясен показател за качеството на изчисленото решение. По-ниската стойност на L_2 в този случай означава по-добро решение на задачата.

Последният индикатор L_∞ може да бъде приложен за да се определи допълнително метричен показател

$$I_\infty(w'') = I_\infty[u(w''), v] = |u(w''), v|_\infty = \max |u_i - v_i| \quad (14)$$

В този специфичен случай, е използвана стойността L_∞ като индикатор за максималното разстояние, което може да има между изменените тегла след прибавяне на шум в дадена точка от пространственото решение. Като допълнителни единици са представени числовите производни на показателите, въпреки достъпността до тяхното аналитично производно. Целта е постигането на по-добра универсалност.

4.1. Числени експерименти

За анализа на чувствителността, са представени два числени експеримента за изпъкнали и вдлъбнати не-намаляващи функции, с цел избягване изместването на резултатите. Двете функции са полином от втора степен ($f(x) = x^2$) и корен квадрата на полином. И в двата случая архитектурата на невронната мрежа е идентична - с един входящ неврон, 4 скрити неврона и един изходен неврон.

Резултатите от изследването на анализа на чувствителността на невронните мрежи и трите индикатора са изобразени на fig. 6-7. Фигура 6 показва резултатите от тестове за полином от втора степен, където горните графики представляват Евклидовото разстояние, средните графики илюстрират L_∞ и последните графики за показател на косинусовата близост.

Изобразяването на пика на всяка графика представлява оптимално решение на функцията, която отговаря на функция на приложения шум. Важно е да се отбележи,

че независимо от вида на индикатора, който е използван, се наблюдава че върховете на възможните решения са представени в една и съща позиция.

В горната лява графика на фиг. 6 евклидовото разстояние показва, че при наличие на 45 – 47% шум, решението на задачата е значително подобро. В допълнение, изходите от L_2 и L_∞ са качествено (тяхното поведение) еднакви, въпреки разлика в количествената част. За този конкретен проблем, се очаква тези параметри да предоставят една и съща информация. Може да се отбележи, че малко количество шум представлява достатъчно добро решение на проблема с оптимизацията, въпреки че същото не може да се наблюдава при L_{cos} , където има колебание на решението.

Като се има предвид, че за този конкретен показател, колкото по-близо до 1 е решението, толкова по-добре, графиката на долния ляв участък на фиг. 6 показва влошаване на резултата (движещ се по-далеч от 1). Показателят за косинусовата близост в този вариант е представен с абсолютната стойност $L_{cos} = \left| \frac{u \cdot v}{|u||v|} \right|$. По този начин, независимо от подобни върхове от трите показателя, е възможно да се наблюдава различна информация. Докато първите два показателя демонстрират задоволителни резултати за нива на шум и поведението на мрежата, третият разкрива обратното заключение.

При евклидовото разстояние (фиг. 6) горна лява графика сочи, че при 40% шум има един връх, който ясно осигурява индикация за подобряване производителността на мрежата. Нивото на шум в този случай драстично подобрява резултатите на невронната мрежа. Третият индикатор осигурява моментна снимка на влошаване на поведението, когато повече шум се прилага. Също така, при намаляващи стойности на L_2 и L_∞ , те показват добро решение на задачата, а при L_{cos} се наблюдават трептения близо до 1, където мрежата произвежда добри резултати, а по-далеч от 1 няма наличие на добри решения.

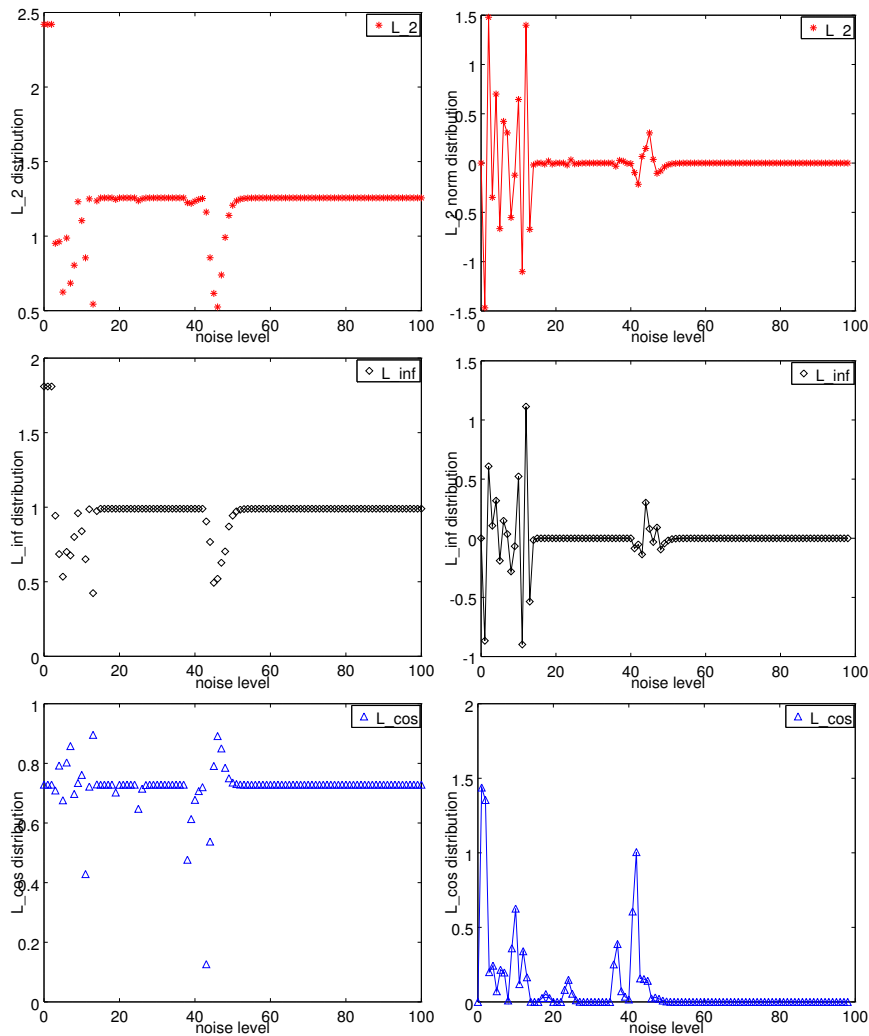
Ситуацията е по-малко ясна в началните нива на шум, когато е налице колебание на оптималните решения. Специално за косинусовата близост, е изчислена втората производна, за да се разбере по-добре свързаните с тях процеси. Долна дясна графика на фиг. 6 на деривата за L_{cos} параметъра, сочи за ситуация, в която по-добри решения могат да бъдат намерени близо до нивото 50% шум, както и в по-малките суми, където шумът е само за 15%.

При втория числен експеримент (виж фиг. 7) за корен квадратен на полином ($f(x) = \sqrt{x}$), трите параметъра и техните деривативни стойности предават подобни наблюдения. Първите два параметъра отново са качествено подобни, въпреки че количествено различни. Наблюдават се пикове за оптималните решения за L_2 и L_∞ , които отново предоставят подобна информация от анализа на чувствителността. Макар че има няколко по-добри решения, получени при по-малко от 10% шум, се наблюдава и появяването на няколко по-добри решения около 70%. Последователно има по-голям шанс за оптимално решение, когато се прибавят малки количества от шум в системата, който подобрява производителността на мрежата (виж фиг. 7, горни и средни фигури). Разликата от предишния случай е, че тук може да се илюстрира, че има няколко малки хълмове сочещи към решение с по-малко колебание от точка до точка. Отново индикатора за косинусовата близост осигурява разнопосочни резултати в сравнение с информацията от предишните показатели. Фигура 7 долу ляво илюстрира този L_{cos} параметър в абсолютната му стойност.

За по-добро разбиране на показателите, отново са използвани производните на показателите, както е обяснено по-горе. Горен лява и средна фигура от фиг. 7 отбелязват че индикаторите гравитират към нулата независимо от количеството шум,

въпреки че добри решения са намерени при 5 – 10% шум, както и при 40% и малко над 60% прибавяне на шум.

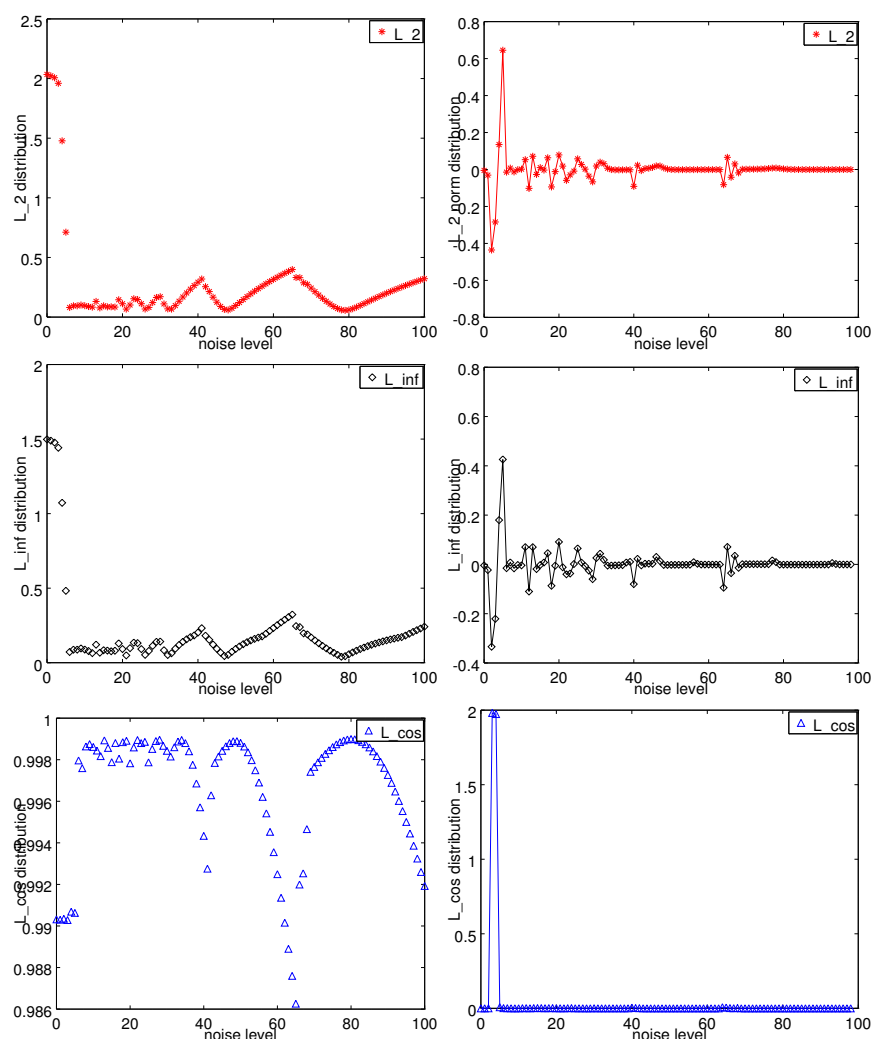
Всеки един от използваните индикатори допринася важни знания за оптималното решение относно приемливия шум приложен към мрежата. Освен това, с тези инструменти се допринася за установяването на количество шум, което няма да повлияе отрицателно на поведението на мрежата или да има отрицателен ефект върху изходните данни. Нещо повече, тези показатели дават възможност точно да се определи нивото на шума, който предвижда подобрене за процеса на оптимизация, или вместо това действа като източник на колебание.



Фигура 6 Фигурите от лявата страна представляват трите показателя за анализ на чувствителността, приложени за полином от втора степен ($f(x) = x^2$). Графиките от дясната страна описват деривативните стойности на индикаторите

Глава 5. Нов хибриден еволюционен алгоритъм при дизайн на топологии на невронни мрежи

В глава пета са разгледани основните параметри, които влияят на производителността на невронната мрежа[10]- като брой слоеве, брой неврони в цялата мрежа,



Фигура 7 Фигурите от лявата страна представляват трите показателя за анализ на чувствителността, приложени за корен квадрат на полином ($f(x) = \sqrt{x}$). Графиките от дясната страна описват деривативните стойности на индикаторите.

както и във всеки слой, видове синаптични връзки между невроните, видове активационни функции, стратегии за инициализиране на теглана на връзките, обективни функции, обучаващи алгоритми, както и влиянието на предварителната обработка на входни данни към мрежата. Тези компоненти са от съществено значение за изчислителна производителност, ефективност и точност на всяка невронна мрежа[19], [1]. Установено е, че топологията на невронна мрежа влияе върху ефективността на мрежата с примери в областта на компютърното зрение[21]. Броят параметри може да се избира зависимост от проблема и този процес представлява сложен комбинаторен оптимизационен проблем. Различни самоорганизиращи топологии са били предложени и приложени [21], [26], [4], [5].

В настоящата глава е въведен нов подход към развитието на невронна мрежова архитектура, за осигуряването на автоматично и изчислително осъществима самоорганизация на невронни мрежи за решаване на даден проблем. Осъществен е хибриден стохастичен генетичен алгоритъм, който предоставя на мрежата да избере всяка възможна структура от даден интервал от възможни архитектури в зависимост от специфичен проблем. Чрез тази еволюционна стратегия е изграден лесно паралели-

зиращ се метод в състояние да избяга от проблема с обобщението на изходните данни когато минималното количество на данни за обучение е достъпно.

Описание на алгоритъма Осъщественият алгоритъм предоставя избор на брой неврони, брой скрити слоеве на мрежата, видове синаптична връзка, активационните функции, вид алгоритъм за обучение при избора на топология. Конфигурирането на всички тези компоненти установява многомерно пространство от възможни структури, като същевременно е в състояние да постигне достатъчна производителност с достъпни изчислителни ресурси. По време на числените експерименти, хибридният процес избира между горе описаните елементи.

В настоящата разработка единственото ограничение към броя неврони, слоеве и брой връзки е определен в даден диапазон от минимални и максимални стойности. Няма специфично изпълнение на кода за добавяне или отстраняване на невроните по време на процеса.

Мрежата не се ограничава само до вид връзки между неврони. Алгоритъмът няма ограничение при избор на синаптични връзки между невроните - те могат да бъдат както последователни, така и непоследователни, независимо от позицията на невроните в слоевете. Алгоритъмът предоставя възможност за ограничаване броя на входящите връзки към всеки неврон, зависимост от конкретната задача, която трябва да бъде решена (в зависимост контролирането размера на пространството на решението).

Използваният в тази работа обучителен алгоритъм е симулираното закаляване [18].

Еволюционна стратегия В тази работа, е въведен нов хибрид генетичен подход чрез прилагането на допълнителен стохастичен слой, функциониращ едновременно с генетичния слой. Алгоритъмът включва две синхронни стъпки. Генетичният алгоритъм се инициализира и създава група от популации (50) на отделните архитектури $C_1, C_2, C_3, \dots, C_n$ от пространството на възможните конфигурации. Паралелно с това, стохастичен слой генерира друга група популации (50) на отделните архитектури $S_1, S_2, S_3, \dots, S_n$. Генетичният алгоритъм съчетава вероятност за кръстосване и вероятност за мутация, чрез които се развиват потенциалните топологични комбинации. Следва избор на елементите с най-добри стойности.

Стартирането на алгоритъма определя стойностите на обективната функция за всеки отделен елемент в популацията, което показва, качеството на специфична мрежова топология. В тази работа, е използвана $L_2 - norm$ функция на грешката (въпреки че не е необходимо ограничаването само до тази функция). На всяка времева стъпка, алгоритъмът приема за по-нататъшна еволюция елемента с най-ниската грешка. Процесът е подобен на Tournament метода [25].

На всяка стъпка, алгоритъмът оценява нивото на грешката на всеки индивид от двете популации. След това избира един индивид от генетичния и стохастичен слой. От там, индивидът с най-ниската грешка е селектиран. Той бива еволюиран стохастично, като се прилагат контролирани настройките описани по-горе.

Допълнителният стохастичен слой води до няколко предимства. Първо осигурява начин алгоритъма да избяга от локален минимум, особено ако цялостното решение има разнообразни популации. Това е осъществено чрез адаптация на минималните стандарти за проектиране в променящата се среда. Стохастичната част позволява на програмата да разработи нова архитектура специфична за даден проблем в достъпно изчислително време дори в многомерно пространство.

За да бъде прекратен алгоритъма трябва да отговорят на едно от следните две

условия. То трябва да достигне максималния позволен брой на поколенията или обектната функция да достигне под определена прагова стойност след r последователни стъпки.

Числени експерименти За да демонстрира възможностите на предложения метод, е извършен числен експеримент за решение на функция $f(x) = x^2$. Предоставени са няколко експеримента в четири сценария с контролиране максималния брой на скрити слоеве и максималния брой неврони. За настоящите експерименти, броя на синаптичните тегла към всеки неврон е ограничен до 2. Тези ограничения се прилагат само за тези конкретни експерименти и не са присъщи за ефективната работа на предложената еволюционен метод.

Според числените експерименти, извършени и дискутирани в този раздел, може да се направи извод, че не всеки път, последователно свързана мрежа осигурява най-доброто решение за определена задача. Когато хибридният алгоритъм използва повече от един скрит слой, често автоматично биват отстранени неврони, което съществено намалява излишни изчислителни ресурси. В тези ситуации, мрежата предоставя достатъчно добри резултати (фигури 8, 9, 10). Интересно наблюдение е, че най-често, невроните в изходния слой използват линейна активираща функция.

В тази глава е въведен нов метод за автоматично търсене на оптимална топология на невронна мрежа за дадена конкретна задача. Чрез реализирания подход са предложени няколко степени на свобода на параметрите, които хибридният алгоритъм може да използва. Следователно алгоритъма произвежда надеждна мрежова топология не въз основа на опита на изследователя, но чрез автоматичен метод.

Заклучение

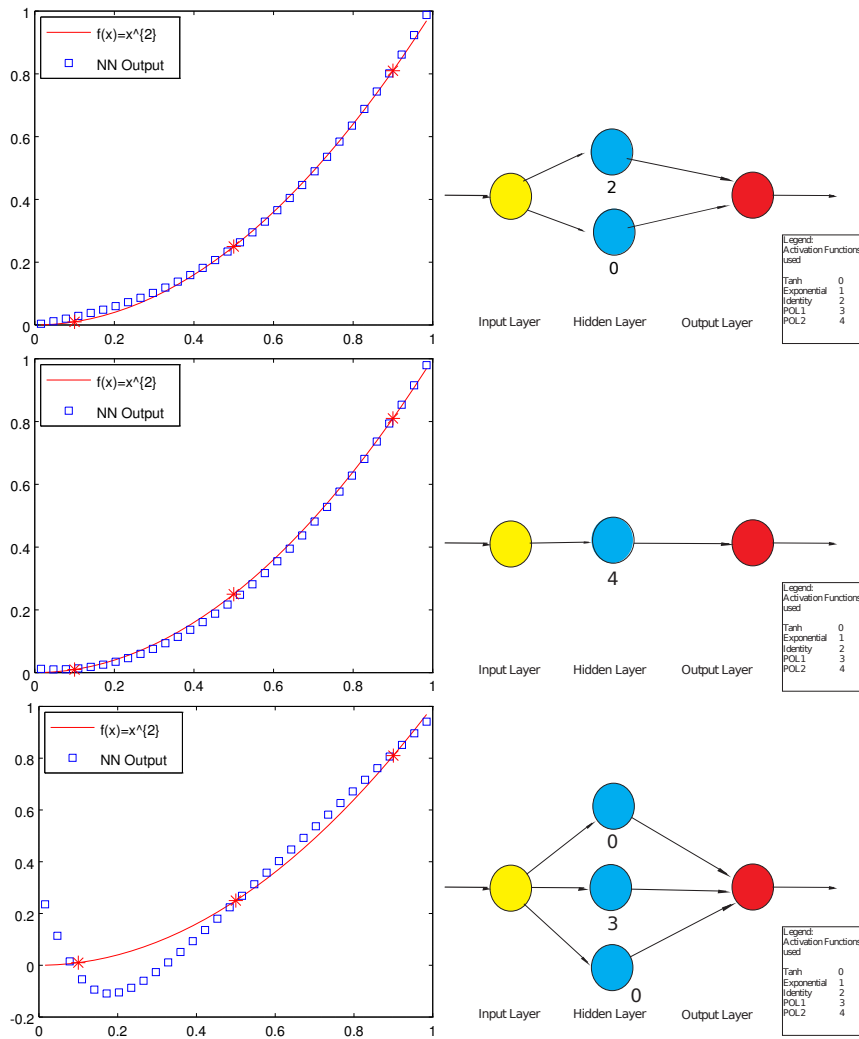
В заключителната част на дисертацията са обобщени методологията и резултатите от изследванията. Новият пост-обучителен модел се въвежда с цел подпомагане на невронната мрежа да излезе от попадане в локален минимум или седловина по време на оптимизационни процес. Въведохме математически инструмент, за да се измери количественото влияние на шума в синаптичните тегла върху работата на мрежата. Въведохме хибриден генетичен алгоритъм за автоматичната еволюцията на невронни мрежови архитектури. Оставени са няколко степени на свобода за проектирането на мрежата, включително: брой на неврони, брой на скрити слоеве и видове синаптични връзки, използване на функции за трансфер и вид обучаващ алгоритъм.

Авторът се надява, че това изследване ще предостави полезни инструменти за по-нататъшното развитие в областта на невронните мрежи. Описаните резултати са повлияни от квантовата механика. Тези изследвания са само една малка част от бъдещото прилагане на такъв интердисциплинарен подход, както и за практическите приложения на тези методи в практически задачи.

Авторска справка

Изследванията, представени в настоящия дисертационен труд са със следните научни и научно-приложни резултати.

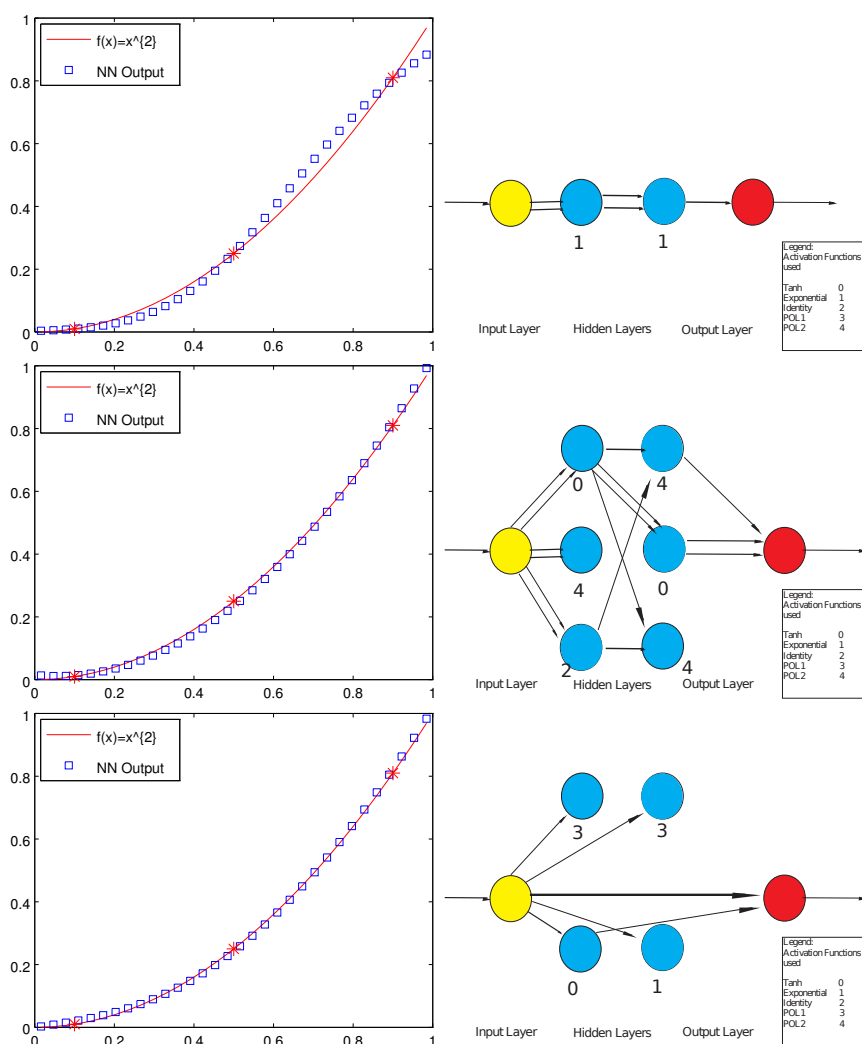
Научно-приложните резултати са формирани предимно от разработването на алгоритми, приложими към обучителния, както и архитектурния процес на изкуствените невронни мрежи:



Фигура 8 Графиките в лявата колона представляват изходните резултати на мрежата, докато дясната част описва различните архитектурни решения при ситуация, в която броя на слоевете е 3.

1. Разработен е и е приложен нов пост-обучителен алгоритъм, осигуряващ системата да не попада в локални минимуми или седловини.
2. Въведена е метрика за оценка на влиянието на шума при стойностите на теглата на връзките в невронна мрежа и влиянието на шума за устойчивостта на системата и качеството на резултатите. Направените числени експерименти показват какво е влиянието на шума в мрежата върху ефекта на процеса като цяло.
3. Предложен е нов хибриден генетичен алгоритъм при проблема за автоматичната еволюция на архитектури на невронни мрежи. Методът осигурява търсене в мултимодално пространство за всички възможни решения на проблема даже и при подаване на минимални обучителни данни, намалявайки влиянието на преобучителния процес.

Основните **приложни** приноси на дисертацията са свързани с разработването на софтуер, който прилага новият пост-обучителен алгоритъм, хибридният генетичен алгоритъм и индикаторите за анализ на чувствителността. Софтуерът е разработен на

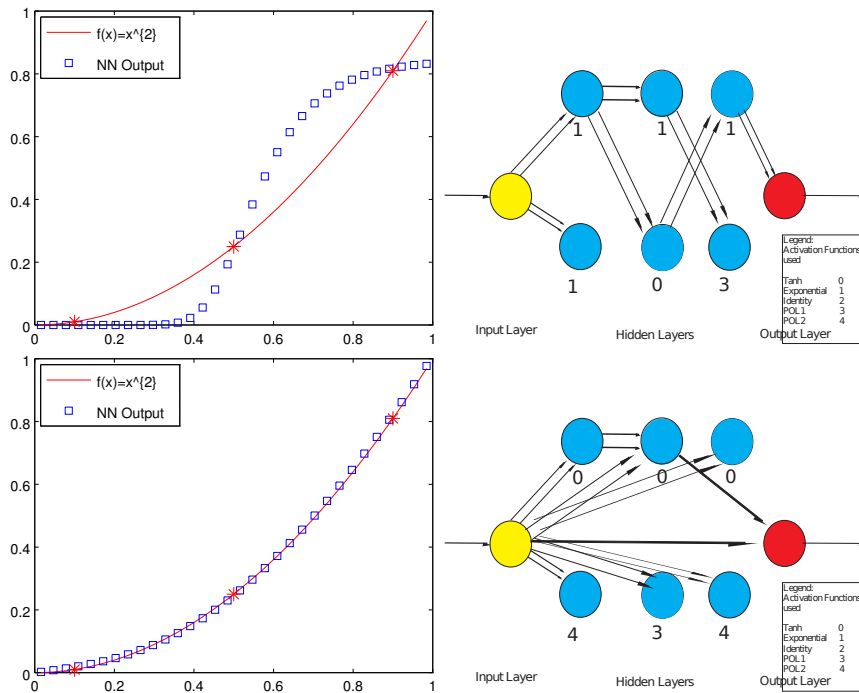


Фигура 9 Графиките в лявата колона представляват изходните резултати на мрежата, докато дясната част описва различните архитектурни решения при ситуация, в която броя на слоевете е 4.

С и е пригоден за лесна дефиниция на специфични характеристики при изграждане на топологията на мрежите:

1. минимален и максимален брой на невроните в мрежата,
2. минимален и максимален брой на слоевете в мрежата,
3. вид синаптична свързаност между невроните и слоевете в мрежата,
4. спецификация на различни активационни функции,
5. спецификация на три разновидности обучителни алгоритми - обучение с обратно разпространение на грешката, генетичен алгоритъм, симулирано закаляване.

Част от кода е описан в приложението на дисертацията.



Фигура 10 Графиките в лявата колона представляват изходните резултати на мрежата, докато дясната част описва различните архитектурни решения при ситуация, в която броя на слоевете е 5.

Благодарности

Авторът изказва най-искрени благодарности към своите научни консултанти проф. Иван Димов и доц. Жан Мишел Селие за търпението, разбирането и безкрайната подкрепа, които осигуриха по време на изработването на настоящия дисертационен труд. Авторът изказва и благодарности на екипа от секция "Паралелни алгоритми-проф. Стефка Фиданова, доц. Пенчо Маринов, доц. Цветан Остромски, доц. Михаил Недялков и гл.ас. д-р Райна Георгиева за приобщаването на автора към колектива. Благодарности следват и за целия екип на Института по Информационни и Комуникационни Технологии и към неговия Директор за осигурените качествени условия за работа и обучение.

Авторът благодари и на своето семейство за търпението и разбирането по време на изграждането на дисертационния труд, както и на Райна Седова и Николета Георгиева за цялостната подкрепа в БАН.

Литература

- [1] Leandro M Almeida and Teresa B Ludermir. A multi-objective memetic and hybrid methodology for optimizing the parameters and performance of artificial neural networks. *Neurocomputing*, 73(7):1438–1450, 2010.
- [2] Friedrich Beck and John C Eccles. Quantum aspects of brain activity and the role of consciousness. In *How the SELF Controls Its BRAIN*, pages 145–165. Springer, 1994.
- [3] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

- [4] Jürgen Branke. Evolutionary algorithms for neural network design and training. In *In Proceedings of the First Nordic Workshop on Genetic Algorithms and its Applications*. Citeseer, 1995.
- [5] Adenilson R Carvalho, Fernando M Ramos, and Antonio A Chaves. Metaheuristics for the feedforward artificial neural network (ann) architecture optimization problem. *Neural Computing and Applications*, 20(8):1273–1284, 2011.
- [6] Gary Chartrand, Linda Lesniak, and Ping Zhang. *Graphs & digraphs*. CRC Press, 2010.
- [7] Marco Dorigo and Mauro Birattari. Ant colony optimization. In *Encyclopedia of machine learning*, pages 36–39. Springer, 2010.
- [8] G Bard Ermentrout, Roberto F Galán, and Nathaniel N Urban. Reliability, synchrony and noise. *Trends in neurosciences*, 31(8):428–434, 2008.
- [9] A Aldo Faisal, Luc PJ Selen, and Daniel M Wolpert. Noise in the nervous system. *Nature Reviews Neuroscience*, 9(4):292–303, 2008.
- [10] Martin T Hagan, Howard B Demuth, Mark H Beale, et al. *Neural network design*. Pws Pub. Boston, 1996.
- [11] Barbara Hammer. *Learning with recurrent neural networks*, volume 254. Springer, 2007.
- [12] Frank Harary. *Graph theory*. 1969.
- [13] Simon S Haykin, Simon S Haykin, Simon S Haykin, and Simon S Haykin. *Neural networks and learning machines*, volume 3. Pearson Education Upper Saddle River, 2009.
- [14] John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, 1975.
- [15] Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse ising model. *Physical Review E*, 58(5):5355, 1998.
- [16] James Kennedy. Particle swarm optimization. In *Encyclopedia of Machine Learning*, pages 760–766. Springer, 2010.
- [17] James Kennedy, James F Kennedy, Russell C Eberhart, and Yuhui Shi. *Swarm intelligence*. Morgan Kaufmann, 2001.
- [18] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.
- [19] Pavel Kordík, Jan Koutník, Jan Drchal, Oleg Kovářík, Miroslav Čepek, and Miroslav Šnorek. Meta-learning approach to neural network optimization. *Neural Networks*, 23(4):568–582, 2010.
- [20] Vera Kuurkova. Kolmogorov’s theorem and multilayer neural networks. *Neural networks*, 5(3):501–506, 1992.
- [21] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [22] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- [23] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.
- [24] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

- [25] Dirk Thierens and David Goldberg. *Convergence models of genetic algorithm selection schemes*, pages 119–129. Springer Berlin Heidelberg, Berlin, Heidelberg, 1994.
- [26] Xin Yao and Yong Liu. A new evolutionary system for evolving artificial neural networks. *Neural Networks, IEEE Transactions on*, 8(3):694–713, 1997.

Abstracts of Dissertations

Number 8, 2017

**INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGIES
BULGARIAN ACADEMY OF SCIENCES**

**БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ
ИНСТИТУТ ПО ИНФОРМАЦИОННИ И КОМУНИКАЦИОННИ ТЕХНОЛОГИИ**

Брой 8, 2017

Автореферати на дисертации