IMPROVED SPIKEPROP ALGORITHM FOR NEURAL NETWORK LEARNING

FALAH.Y.H.AHMED

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Doctor of Philosophy (Computer Science)

Faculty of Computing
Universiti Teknologi Malaysia

MAY 2013

To my beloved father and mother

.

ACKNOWLEDGEMENT

In the Name of Allah, Most Gracious, Most Merciful

All praise and thanks are due to Allah, and peace and blessings be upon his messenger, Mohammed (peace be upon him).

Alhamdulillah, it is with Allah S.W.T will that I get to finish this thesis in the time given. Here, I would like to express my heartfelt gratitude to my supervisor Professor Dr. Siti Mariyam Shamsuddin and without her guidance and advice this study would not have been possible. She has been incredibly wise, helpful, understanding, and generous throughout the process. She has truly been a mentor and I owe here my deepest thanks. Also I would like to express my heartfelt to my co-supervisor Assoc. Prof.Dr.Siti Zaiton Mohd Hashim for her support and guidance during my study .And I would like to thanks to my External supervisor Prof. Dr. Nikola Kasabov for his support and guidance during my study . And finally I would like to thanks Dr.Haza Nuzly Bin Abdull Hamed for advices and guidance.

I have made many friends during my time in UTM and I thank them for their support and encouragement. Their views and tips are useful indeed. Unfortunately, it is not possible to list all of them in this limited space.

A lot of information useful to the work was found via the World-Wide Web; I thank those who made their materials available by means of this medium and those who kindly answered back to my roll-calls of help sent over the World-Wide Web.

ABSTRACT

Spiking Neural Network (SNN) utilizes individual spikes in time domain to communicate and to perform computation in a manner like what the real neurons actually do. SNN had remained unexplored for many years because it was considered too complex and too difficult to analyze. Since Sander Bothe introduced SpikeProp as a supervised learning model for SNN in 2002, many problems which were not clearly known regarding the characteristics of SNN have now been understood. Despite the success of Bohte in his pioneering work on SpikeProp, his algorithm is dictated by fixed time convergence in the iterative process to get optimum initial weights and the lengthy procedure in implementing the sequence of complete learning for classification purposes. Therefore, this thesis proposes an improvement to Bohte's algorithm by introducing acceleration factors of Particle Swarm Optimization (PSO) denoted as Model 1; SpikeProp using μ Angle driven Learning rate dependency as Model 2; SpikeProp using Radius Initial Weight as Model 3a, and SpikeProp using Differential Evolution (DE) Weights Initialization as Model 3b.The hybridization of Model 1 and Model 2 gives Model 4, and finally Model 5 is obtained from the hybridization of Model 1, Model 3a and Model 3b. With these new methods, it was observed that the errors can be reduced accordingly. Training and classification properties of the new proposed methods were investigated using datasets from Machine Learning Benchmark Repository. Performance results of the proposed Models (for which graphs of time errors with iterative timings, table of number of iterations required to reduce time error measurement to saturation level and bar charts of accuracy at saturation time error for all the datasets have been plotted and drawn up) were compared with one another and with the performance results of Standard SpikeProp and Backpropagation (BP). Results indicated that the performances of Model 4, Model5 and Model 1 are better than Model 2, Model 3a and Model 3b. The findings also reveal that all the proposed models perform better than Standard SpikeProp and BP for all datasets used.

ABSTRAK

Rangkaian Saraf Pepaku (SNN) menggunakan dedenyut tunggal dalam domain masa untuk mewujudkan komunikasi dan penghitungan seperti yang dilakukan oleh saraf tabii. SNN tidak diterokai dengan mendalam disebabkan oleh mekanisme pelaksanaannya yang amat sukar untuk dianalisa. Sejak diperkenalkan oleh Sander Bothe pada tahun 2002 sebagai model pembelajaran terpandu, banyak masalah yang tidak jelas pada masa lampau berkenaan dengan ciri-ciri SNN sudah boleh diperjelaskan pada masa kini. Walaupun SpikeProp sudah berjaya digunakan secara meluas sorotan kejayaan Bohte terhadap algoritma Spikeprop, algoritma ini masih lagi mempunyai masalah dalam konteks tetapan masa penumpuan dalam proses jujukan bagi mendapatkan pemberat awalan yang optimal dan prosedur yang lama untuk melaksanakan pembelajaran lengkap bagi tujuan pengelasan. Oleh yang demikian, tesis ini mencadangkan pembaikan terhadap algoritma Bohte dengan memperkenalkan faktor pecutan dalam *SpikeProp* menggunakan PSO dan dilabel sebagai Model 1; *SpikeProp* menggunakan Kadar Pembelajaran Sudut Terpandu Bersandarkan μ, Model 2; *SpikeProp* menggunakan Pemberat Awalan Jejari, Model 3a; dan *SpikeProp* menggunakan Pemberat Awalan Jejari Evolusi Pembahagi, Model 3b. Penhibridian Model 1 dan Model 2 memberi Model 4 dan akhirnya Model 5 dicadangkan hasil penhibridan Model 1, Model 3a dan Model 3b. Berdasarkan model baru ini, didapati bahawa ralat boleh disusutkan dengan baik dan pantas. Sifat pengelasan dan latihan bagi kaedah cadangan telah dikaji menggunakan set data daripada Storan Piawaian Pembelajaran Mesin. Prestasi keputusan bagi kaedah cadangan (iaitu dengan graf ralat masa terhadap masa lelaran, jadual bilangan lelaran yang diperlukan untuk menyusutkan pengukuran ralat masa kepada tahap tepu dan carta bar ketepatan pada ralat masa tepu bagi semua set data telah diplot) di bandingkan dengan prestasi *SpikeProp* and Rambatan Balik piawai. Hasil kajian menunjukkan bahawa prestasi Model 4, Model 5 dan Model 1 lebih baik berbanding Model 2, Model 3a dan Model 3b. Hasil dapatan juga mendapati bahawa prestasi ke semua model cadangan ini adalah lebih baik berbanding dengan prestasi *SpikeProp* and Rambatan Balik piawai bagi semua set data.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

Classification is one of the most frequently encountered processing tasks for a decision support system. Classification is necessary when a scattering of data needs to be sorted into the predefined groups or classes based on certain criteria and attributes. Classification problems are encountered in areas such as business, science, industry and medicine. They include bankruptcy prediction, credit scoring, medical diagnosis, quality control, handwritten character recognition, and speech recognition.

Traditional statistical classification procedures such as discriminant analysis are built on the Bayesian decision theory (Duda and Hart, 1973; Qasem and Shamsuddin, 2010) In these procedures, an underlying probability model must be assumed in order to calculate the posterior probability upon which the classification decision is made. One major limitation of the statistical models is that they work well only when the underlying assumptions are satisfied. The effectiveness of these methods depends to a large extent on the various assumptions or conditions under which the models are developed. Users must have a good knowledge of both data properties and model capabilities before the models can be successfully applied. Neural networks have emerged as an important tool for classification because they do not depend much on prior knowledge of the statistics of the data.

Several neurobiologists believe that the working process of a brain is similar to that of a huge parallel computer which has at its disposal about 10 billion simple processors, where each needs a few milliseconds to react according to input values. There are similarities between the function of parallel processing methods and conventional Artificial Neural Network (ANN) methods in many real-time applications.

For this reason ANN has been conceived with a structure which resembles closely to the human brain (DKlerfors, 2004). Because of this structure, computation property of ANN has been designated many names which include parallel distributed processing, neuron-computing, natural intelligent systems and machine learning algorithms.

1.2 Background of the Problem

The earliest models and schemes of ANN (the first generation ANNs) came nearly fifty years ago. In the earlier models which are theoretically incredibly straightforward, the primary generation of ANN consists of McCulloch-Pitts threshold neurons (Y.-J. Jin *et al.*, 2013; Maass, 1997). A strand of neurons sends a binary signal (higher value signal) only if the summation of the weighted signals which it receives rises above a certain threshold level. As the output from these neurons are digital, it is convenient to treat interconnected neurons to behave as a function with a Boolean output type which can be modeled mathematically as multi-layer perceptron (MLP) and Hopfield Nets (Rebizant, 2011).

Signal responses of second generation ANN are not calculated using threshold or step activation type functions. Instead, they are calculated using continuous activation type functions which are formulated so that responses are analog for the two cases: input and output. Generally, the activation functions take the form of the sigmoid. Sigmoidal Neural Nets, (these include Adaline Perceptrons and Radial Basis Function) have been considered under the class of second generation ANNs

Back propagation (BP) and recurrent neural networks have been applied to many applications. They are the most famous instances of neural networks consisting of neurons which make up the second generation of ANN. Obviously they outperform the first generation of ANN which are merely simple networks of binary neural units, each exhibiting either an active or an inactive, firing or non-firing states.

ANN can perform digital computations to get universal functions approximation which can also be calculated using fewer neurons in a network of the first generation whereby the corresponding circuits are resolved with Boolean threshold gates (Y.-J. Jin, *et al.*, 2013; Maass *et al.*, 1991).

Usually, the learning process for updating the weights of ANN interconnectors will require some form of optimization. One useful optimization technique uses PSO (Eberhart and Kennedy, 1995), which is inspired by how the biological swarm of animals works to achieve a desirable objective for the group. Since its introduction, PSO has been widely used to solve many real world problems. (X. Jin, 2011; Kennedy and Eberhart, 1997) also introduced the binary version of PSO. There have been a lot of developments and improvements in this area (Khanesar *et al.*, 2007) (Srinivasan and Shanmugalakshmi., 2012; Yuan *et al.*, 2009). PSO has been applied to derive universal function approximations for any analog function with random updating of weights.

For the first generations of ANN, the neurons are restricted to binary inputs and outputs. These binary impulses have a definite width, phase and time and these types of signals could be considered driven by stabilized frequencies of the neuron. However, recently it has been discovered that neurons communicate by firing short electrical pulses which operate in a mode referred to as rate coding. Higher output signal is brought about by firing at higher rate. Communication is carried out using real spikes only at the two exclusive instants known as spiking time or no spiking time. With a communication window of this type, the value of output of a neuron can be computed and the response of the network to the values of input is known or identified only after all the neurons have fired. Each neuron can be modeled since it has a basic firing-rate and a continuously constant activation function. This model which is referred to as Spiking Neural Networks (SNN) is considered to constitute the third generation of ANNs (Gr¨uning and Sporea, 2011).

For all the three generations of neural networks, the output signals can be continuously altered by variation in synaptic weights (synaptic plasticity). Synaptic plasticity is the basis for learning in all ANN. As long as there is non-variant activation function , accurate classification based on a certain vector input values can be implemented with the help of a BP learning algorithm like gradient-descent (Y.-J. Jin, *et al.*, 2013; Kempter and Hemmen, 1999).

Spiking Neural Network (SNN) utilises individual spikes in time domain to communicate and to perform computation in a manner like what the real neurons actually do (Belatreche and Paul, 2012; Ferster and Spruston, 1995). This method of sending and receiving individual pulses is called pulse coding where information which is transmitted is carried by the pulse rate. Hence, this type of coding permits multiplexing of data (Gerstner *et al.*, 1999).

For instance, analysis of visual input in humans requires less than 100ms for facial recognition. Yet, facial recognition was performed by Thorpe & Delorme (S. Thorpe *et al.*, 2001) by using SNN with a minimum of 10 synaptic steps on the retina at the temporal lobe, allowing nearly 10ms for the neurons to process. Processing time is short but it is sufficient to permit an averaging procedure which is required by pulse coding (Gerstner, *et al.*, 1999; Kasabov, 2012b; S. Thorpe, *et al.*, 2001). In fact, pulse coding technique is preferred when speed of computation is the issue (S. Thorpe, *et al.*, 2001).

1.2.1 Spiking Neural Networks (SNNs)

Neural networks which perform artificial information processing are built using processing units composed of linear or non-linear processing elements a sigmoid function is widely used (Bishop, 2000; Haykin, 1998; Kasabov, 2012a). SNN had remained unexplored for many years because it was considered too complex and too difficult to analyze. Apart from that:

1) Biological cortical neurons have long time constants. Inhibition speed can be of the order of several milliseconds while excitation speed can reach several hundreds of milliseconds. This dynamics can considerably constrain applications that need fine temporal processing (Gewaltig, 2000; Kasabov, 2009).

2) Little is known about how information is encoded in time for SNNs. Although it is known that neurons receive and emit spikes, whether neurons encode information using spike rate or precise spike time is still unclear (Thorpe and Gaustrais, 1998). For those supporting the theory of spike rate coding, it is reasonable to approximate the average number of spikes in a neuron with continuous values and consequently process them with traditional processing units (sigmoid, for instance). Therefore, it is not necessary to perform simulations with spikes, as the computation with continuous values is simpler to implement and evaluate (Kasabov, 2010).

An important landmark study by Maass (Maass, 2001) has shown that SNN can be used as universal approximations of continuous functions. Maass proposed a three-layer SNN (consisting of the input layer, the generalization layer and the selection layer) to perform unsupervised pattern analysis. Mishra (Mishra *et al.*, 2006) applied spiking neural networks to several benchmark datasets (which include internet traffic data, EEG data, XOR problems, 3-bit parity problems, iris dataset) and performed function approximation and supervised pattern recognition (Gr¨uning and Sporea, 2011).

One of the ongoing issues in SNN research is how the networks can be trained. Much research has been done on biologically inspired local learning rules (Gerstner and Kistler, 2002a, 2002b; Kasabov, 2010), but these rules can only carry out supervised learning for which the networks cannot be trained to perform a given task. Classical neural network research became famous because of the error-backpropagation learning rule. Due to this, a neural network can be trained to solve a problem which is specified by a representative set of examples. Spiking neural networks use a learning rule called SpikeProp which operates on networks of spiking neurons and use exact spike time temporal coding (Bohte *et al.*, 2002). This means that the exact spike time of input and output spikes encode the input and output values.

1.2.2 Learning in Spike Neural Networks (SpikeProp)

Supervised learning (SpikeProp) in Spiking Neural Networks (SNNs) is usually performed by a gradient descent method which explicitly evaluates the gradient of an error function on the back-propagation algorithm (ˇS´ıma, 2009; Bohte, *et al.*, 2002; Gr¨uning and Sporea, 2011). Using this algorithm, SNN learns the desired firing times of the output neurons by adapting the weight parameters in the Spike Response Model (Gerstner and Kistler, 2002b). Several experiments have been carried out on SpikeProp to clarify several burning issues such as the role of the parameters for initialization and the significance of negative weights (Moore, 2002). SpikeProp can be further enhanced with additional learning rules for synaptic delays, thresholds, and time constants (Schrauwen and Campenhout., 2007) which will normally result in faster convergence and smaller network sizes for given learning tasks. An essential speedup was achieved by approximating the firing time function using the logistic sigmoid (Berkovec, 2005). Implementation of SpikeProp algorithm on recurrent network architectures has shown promising results (Tiˇno and Mills, 2005).

SpikeProp does not usually allow more than one spike per neuron which makes it suitable only for ‘time-to-first-spike’ coding scheme (S. Thorpe, *et al.*, 2001). Its adaptation mechanism fails for the weights of neurons that do not emit spikes. These difficulties are due to the fact that spike creation or its removal due to weight updates is very discontinuous. ASNA-Prop has been proposed (Schrauwen and Campenhout., 2007) to solve this problem by emulating the feed forward networks of spiking neurons with the discrete-time analog sigmoid networks with local feedback, which is then used for deriving the gradient learning rule (Gr¨uning and Sporea, 2011). It is possible to estimate the gradient by measuring the fluctuations in the error function in response to the dynamic neuron parameter perturbation (Fiete and Seung, 2006). Table 1 summarizes the differences among the EBP, SNN and SpikeProp algorithms.

Table 1.1 Summary of Differs of EBP, SNN and SpikeProp

EBP	They are second generation ANNs which do not use step- or threshold functions to compute their output signals. Their activation functions are continuous, making them suitable for analog in- and output operations. They are more powerful than their first generation predecessors. Software simulations associated with them are easy to implement (Y.-J. Jin, *et al.*, 2013; Zweiri *et al.*, 2003).
SNN	They are third generation ANNs. Functionally, they operate like real biological systems because the neurons facilitate the use of individual spikes. In the past, SNNs were considered too complex and difficult to analyze. Recently, many researchers have become aware of their potential to operate in a manner more powerful than the first generation and the second generation ANNs. (Kasabov, 2010; Negnevitsky, 2002).
Spikeprop	The firing time of SNNs can be shifted by adapting the incoming synaptic weights, the threshold and the membrane time constant. Bohte et al. used this idea for developing an error-backpropagation learning method based on the exact timing of spikes called SpikeProp, (Bohte, *et al.*, 2002; Gr¨uning and Sporea, 2011).

1.3 Problem statement

Performance of SNNs is dictated by its architecture algorithm. It has been important for this research work to develop a learning algorithm for the SNN so that it is able to classify data. Biologically inspired SNN is normally capable of implementing supervised learning (ˇS´ıma, 2009). However, supervised learning rule is implementable if it operates in conjunction with backpropagation (ˇS´ıma, 2009). This learning rule is called SpikeProp which utilizes spike time temporal coding while the backpropagation is treated as a network of spiking neurons . In this case, the input and output variables perform encoding according to the correct spike time of both the output and input spikes.

For the supervised learning rule of SpikeProp, the learning rate , the momentum, the bias, the minimum error for the transfer, the activation function and the initial weights are similar to that of other ANNs. The performance of SpikeProp is determined by the learning parameters of the BP network. Hence, it is necessary for this study to focus on finding ways to enhance the performance of SpikeProp by optimizing the back propagation initializing weights and the architectures of SNNs. Particle Swarm Optimization (PSO) is one technique which can be used for this purpose.

In the course of this thesis, the aim is to highlight important issues and provide solutions to a number of general and specific research questions concerning SNNs.

How will it be possible to enhance SpikeProp to enable them to efficiently process data?

This question is answered in Chapters 4 and 5, where this thesis has, among other things, suggested that accuracy (low error) and the use of less number of iterations (less time to optimize) can be achieved by using:

1. Multi decision techniques to get to the optimum (Model 1).
2. Use of Angle driven dependency Learning Rate (Model 2).
3. Radial segmentation of the weights space to aim directly on to the optimum position (Model 3a).
4. Use of chromosomes (agents) to identify optimum positions quickly (Model 3b).

What will be the optimum back propagation initializing weights and what will be the best architectures of SNNs for classification tasks?

This question is also answered in Chapters 4 and 5. This thesis stipulated that the best architecture of SNN for classification tasks is that which can guarantee so that SNN has components which get the initial weights as close as possible to the optimum positions and also has components which can search for the optimum quickly (Model 4 and Model 5).

1.4 Objectives of the Research

The goal of this study is to propose an optimum back propagation configuration for spiking neural network (using SpikeProp as the supervised learning rule) according to the following considerations:

1. Adopt new initial weights methods.

2. Propose appropriate momentum factors which are needed to speed up convergence and estimate corresponding adaptive learning average or rate
3. Suggest methods, for example, by using PSO rules to enhance architectures of SpikeProp.

The objectives of this research are focused on.

1. To enhancing Spikeprop's architecture by:

I The implementation of PSO for learning optimization (Model 1).

II Proposing the parameters for μ Angle Driven Dependency learning rate in SpikeProp (Model 2)

III Determining Radius Initial Weights (RIW) for SpikeProp to accelerate learning (Model 3a).

IV The development of DE for SpikeProp weights initialization (Model 3b).

2. To propose the hybridization for better SpikeProp performance by:

I PSO Spikeprop with learning rate μ Angle Driven Dependency (Model 4).

II PSO SpikeProp with RIW and DEA weights initialization (Model 5).

3. To evaluate and compare the proposed method with the conventional approaches SpikeProp and BackPropagation standard.

1.5 Scope of the Study

1. Using C++ program to develop several SNN Models, each having its own learning characteristic. It has been established that in order for SNN to perform classification accurately and quickly, two important issues have to be resolved. The first issue is how to get the initial weights as close to the optimum as possible. The second is how to get to the optimum with minimum error. For this reason many different novel BP SNN algorithms (SpikeProp) were tested. The development and testing of Models 1, 2, 3a, 3b, 4 and 5 are described in Chapters 4 and 5.

2. Data sets from breast cancer, Pima Indian diabetes, heart, BTX, hepatitis, liver and Iris were used for testing the different SNN models to ascertain and identify their classification properties. These datasets are known for their attributes, classes, samples, inputs and outputs (as mentioned in Table 5.1). For each of these datasets both Hold-Out Validation and K-Fold Cross Validation (training and testing procedure) were used to find-out which of the two has an advantage. XOR datasets is a test pattern to inspect the firing times of SNN. Therefore, XOR will bring about the worst case scenario in any model that is tested.

3. Matlab 8.0 was used to look at the behavior of the output generated by supervised BP (standard ANN BP). The behavior of the output generated by standard SpikeProp was studied using C++. The results from the Matlab 8.0 program (standard ANN BP) as well as the results from the C++ program (standard SpikeProp) were then compared to the models which have been developed.

1.6 Thesis Organization

Lately, SNN has captured the imagination of researchers and scientists because SNN operates in a manner similar to biological neurons which communicate with the help of electrical pulses. The advantage of SNN over standard ANN is the fact that in SNN there is no attenuation of the communicated signals, the energy used is small and SNN allows ”refresh“ in the system so that no residual signal remains which can interfere with the function of classification .

Chapter 2, *Literature Review***,** give a review on error back propagation (EBP), Spiking Neural Networks and SpikeProp algorithm and the impact of error functions and learning parameters on BP and SpikeProp. The workings of PSO and DE are explained in some length because these two systems are the basis of important modification made to SNN architecture (SpikeProp algorithm) to improve performance (training and classification ANNs, SNN network design. SNN and SpikeProp are detailed in this chapter. Broad overview about the basic concepts and traditional techniques of Artificial Neural Network (ANN), Spiking Neural Network and SpikeProp are given. Furthermore, applicability of Spiking Neural Network in ANN learning and especially BP and SpikeProp Algorithm learning were discussed.

Chapter 3, *Research Methodology*, comprises of research methodology, describing the overall solving-tools and standard techniques adopted. It also displays a general picture about each phase of the work. This chapter discussed the methodology which was used in this research and describes new techniques and parameters that are required for BP learning and Spikeprop. It discusses thoroughly the requirements, framework and phases of this research. It consists of standard algorithms of SpikeProp algorithm learning.

Chapter 4, *The Proposed Models For Spikeprop Learning Enhancements*

This chapter proposes the design and implementation technique which is improved the performance of SpikeProp. It was observed that the errors (MSE, RMSE, MAPE and MAD) of SpikeProp can be reduced by using μ Angle driven Learning rate dependency (Model 2 and used the PSO as a means of improving

SpikeProp. In other words, combining SpikeProp and PSO (to get Model 1-PSOSpikeProp) should result in an algorithm which will have many positive values. Model 1 -PSOSpikeProp. Therefore this chapter also looks at a technique of obtaining good initial weights by using RIW (Model 3a) and by using DE (Model 3b). Just as we get good strain by cross –breeding two good genes, it may be possible to get good SpikeProp algorithm by hybridizing two good techniques. Therefore, this thesis looks at the hybridization of Model 1 and Model 2 (to get Model 4) and also the combination of Model 1, Model 3a and Model 3b (to get Model 5).

Chapter 5, *Results and Discussion*, correlates different techniques by comparing different results produced by the proposed Model 1, Model 2. Model 3a, Model 3b, Model 4 and Model for classification problems. Performance of the proposed methods is compared, analyzed and benchmarked with previous results. This thesis investigates ways in which the performance of SNN can be improved. Several SNN models were developed. For each model the architecture, the learning process and the testing procedure is uniquely different. This is discussed in detail in Chapters 4 and 5. SpikeProp which is an algorithm that mimics the function of SNN is a useful tool for the investigation of SNNs.

Chapter 6, *Conclusion and Future Work*, discusses and highlights the contributions and findings of the research work, and it provides suggestions and recommendations for future study.

REFERENCES

ˇS´ıma, J. r. ı. (2009). Gradient Learning in Networks of Smoothly Spiking Neurons. (Technical report No. 1045).

Abbott, L. F. and Nelson, S. B. (2000). Synaptic plasticity: taming the beast. Nature neuroscience,. 3 Suppl, 1178–1183.

Agrawal, R., Imielinski, T. and Swami, A. (1993). Database mining: A performance perspective. *IEEE* 5: 914-925.

Altman, Marco and Varetto. (1994). Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks 505–529, 1994.

Angeline. (1998). Evolutionary optimization versus particle swarm optimization Philosophy and performance differences. In Proceedings of the 7th international conference on evolutionary programming vii *London, UK: Springer-Verlag*

Baldi and Heiligenberg. (1988). How sensory maps could enhance resolution through ordered arrangements of broadly tuned receivers. (Biological Cybernetics), 313–318,359.

Belatreche, A. and Paul, R. (2012). Dynamic cluster formation using populations of spiking neurons. *IJCNN*, 1-6.

Benuskova, Kanich and Krakovska. (2001). Piriform cortex model of EEG has random underlying dynamics. (Proc. 1st World Congress on Neuroinformatics, Vienna.), 287-292.

Bergh, den, v. and Engelbrecht. (2000). Cooperative learning in neural networks using particle swarm optimizers. *South African Computer Journal*, 26, 84-90.

Berkovec. (2005). Learning in networks of spiking neurons. ((in Czech) M.Sc. Thesis, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic).

Bishop, C. M. (2000). Neural networks for pattern recognition.

Bohte, S. M., Kok, J. N. and La Poutr´e. (2002). H.: Error-backpropagation in temporally encoded networks of spiking neurons. Neurocomputing (pp. 48 (41-44) 17–37

Bohte. (2005). The evidence for neural information processing with precise spike-timing: A survey. Natural Computing (pp. 3(2), 195–206).

Bonhoeffer, Staiger and Aertsen. (1989). Synaptic plasticity in rat hip pocampal slice cultures: local ’hebbian’ conjunction of pre- and postsynaptic stimulation leads to distributed synaptic enhancement. (Proceedings of National Academy of Science USA).

Booij, O. and Nguyen, H. t. (2005). A gradient descent rule for spiking neurons emitting multiple spikes. *Inf. Process. Lett.*, 95(6), 552-558. doi: 10.1016/j.ipl.2005.05.023

Bothe, Poutre, L., A, H. and Kok. (2002). Unsupervised clustering with spiking Neurons by sparse temporal coding and multi-layer RBF networks. *IEEE*, 426-435.

Bothe. (2003). *Spiking Neural Networks.* University of Leiden.

Bourlard, H. and Morgan, N. (1993). Continuous speech recognition by connectionist statistical methods. *Neural Networks, IEEE Transactions on*, 4(6), 893-909. doi: 10.1109/72.286885

Cabrera, J. L. and Milton, J. G. (2002). On-Off Intermittency in a Human Balancing Task. *Physical Review Letters*, 89(15), 158702. Canada: University of Victoria.

Chan and Fallside. (1987). An adaptive training algorithm for backpropagation networks, Computer Speech and Language

Clerc. (1999). The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In Congress on evolutionary computation. ( ), 1951–1957.

Curram and Mingers. (1994). Neural networks, decision tree induction and discriminant analysis: An empirical comparison. (Oper. Res. Soc., vol. 45, no. 4, pp. 440–450).

Cybenko. (1989). Approximation by superpositions of a sigmoidal function. *Math. Contr. Signals Syst*, 303–314.

Das, S. and Suganthan, P. N. (2011). Differential Evolution: A Survey of the State-of-the-Art. *Evolutionary Computation, IEEE Transactions on*, 15(1), 4-31. doi: 10.1109/tevc.2010.2059031

Das, S., Abraham, A., Chakraborty, U. K. and Konar, A. (2009). Differential evolution using a neighborhood-based mutation operator. *Trans. Evol. Comp,* 13(3), 526-553. doi: 10.1109/tevc.2008.2009457

DasGupta and Schnitger. (1992). The power of approximating: a comparison of activation functions, Advances in Neural Information Processing Systems. 363-374

Dayan and Abbott. (2001). Theoretical Neuroscience. *MIT Press, Cambridge, MA.*

Debanne, D., Gähwiler, B. H. and Thompson, S. M. (1998). Long-term synaptic plasticity between pairs of individual CA3 pyramidal cells in rat hippocampal slice cultures. *The Journal of Physiology*, 507(1), 237-247. doi: 10.1111/j.1469-7793.1998.237bu.x

Delorme, A. and Thorpe, S. J. (2001). Face identification using one spike per neuron: resistance to image degradations. *Neural Networks*, 14(6–7), 795-803. doi: http://dx.doi.org/10.1016/S0893-6080(01)00049-1

Delorme, Gautrais, Rullen and Thorpe. (1999). a simulator for modeling large networks of integrate and fire neurons, Neurocomputing. 989-996,926-927.

DKlerfors. (2004). Artificial Neural Network is a system loosely modeled on the human *Norsarini Salim*.

doi: 10.1186/1471-2105-7-125

Duda and Hart. (1973). Pattern classification and scene analysis. *John Wiley and Sons*.

Eberhart and Shi. (2001). Particle Swarm Optimization: Developments,Application and Resources. *IEEE*, 81-86.

Eberhart, R. and Kennedy, J. (1995). A new optimizer using particle swarm theory. Proceedings of the 1995 *Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on*. 4-6 Oct 1995. 39-43.

Eberhart. (2002). Multiobjective optimization using dynamic neighborhood particle swarm optimization.

El-Azm, F. and Vinther, M. (2002). Adaptive Regularization in Neural Network Filters.

Eurich and Wilke. (2000). Multi-dimensional encoding strategy of spiking neurons. Neural Computation. 12(17), 1519–1529.

Ferguson. (2004). Particle Swarm Canada: University of Victoria.

Ferster and Spruston. (1995). Cracking the neuronal code. *Science*, 756- 757

Fiete and Seung. (2006). Gradient learning in spiking neural networks by dynamic perturbation of conductances (pp. 97, 048104): Physical Review Letters

Fodor, J. A. and Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1–2), 3-71. doi: http://dx.doi.org/10.1016/0010-0277(88)90031-5

Fukuoka, Matsuki, Minamitani and Ishida (Eds.). (1998). *A modified back propagation method to avoid false local minima*: Japan.

Gallant. (1995). Neural network learning and expert systems.

Gerstner and Kistler. (2001). Spiking Neuron Models: Single Neurons, Populations, Plasticity. *Cambridge University Press*.

Gerstner and Kistler. (2002a). Spiking Neuron Models, Cambridge University Press.

Gerstner and Kistler. (2002b). Spiking neuron models: An introduction. *Cambridge University Press*.

Gerstner, Kempter, Hemmen and Wagner. (1999). Hebbian Learning of Pulse Timing in the Barn Owl Auditory System in Maass.

Gerstner. (1995). Time structure of the activity in neural network models. *1995*, (Phys. Rev. E, 51:738,758).

Gewaltig. (2000). Accelerated Backpropagation Learning: Extended Dynamic Parallel Tangent Learning Optimization Algorithm. *Hamilton, H. ed. Lecture Notes in Artificial Intelligence 1822. Springer- Verlag.*

Ghorbani and Bhavsar. (1993). Accelerated Backpropagation Learning: Parallel Tangent Learning Optimization Algorithm.

Gonzalez-Flesca N, V. S., Cicolella A. (2002). . BTX concentrations near a Stage II implemented petrol station. Environmental Science and Pollution Research., (9(3):169–174.).

Gr¨uning, A. e. and Sporea, I. (2011). Supervised Learning of Logical Operations in Layered Spiking Neural Networks with Spike Train Encoding," . Logical Operations in Spiking Neural Networks. 272.

Gray, onig, Engel and Singer. (1989). Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which re-ects global stimulus properties. 338, 334-337.

Gudise and Venayagamoorthy. (2003). Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks.

Gueorguieva, Valova and Georgiev. (2006). Learning and data clustering with an RBFbased spiking neuron, Journal of Experimental & Theoretical Artificial Intelligence.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*: Prentice Hall PTR.

Hebb. (1949). The organization of behaviour. *New York*.

Hervás, Silva, Gutiérrez and Serrano. (2008). Multilogistic regression by evolutionary neural network as a classification tool to discriminate highly overlapping signals: Qualitative investigation of volatile organic compounds in polluted waters by using headspacemass spectrometric analysis,. (Chemometrics Intell. Lab. Syst., vol. 92,pp. 179–185, 2008.).

Hodgkin and Huxley. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*.

Hornik, Stinchcombe and White. (1989). Multilayer feedforward networks are universal approximators. (Neural Networks, vol. 2, pp. 359–366,).

Huguenard. (2000). Reliability of axonal propagation. 9349-9350.

Ikegaya, Matsumoto, Chiou, Yuste and Aaron. (2008). Statistical significance of precisely repeated intracellular synaptic patterns. (PLoS ONE, 3(12), e3983). *In IEEE International Joint Conference on Neural Networks,*

Izhikevich and Edelman. (2008). Large-scale model of mammalian thalamocortical systems.

Izhikevich. (2003). Simple model of spiking neurons. *IEEE*

Izhikevich. (2004). Which model to use for cortical spiking neurons. *? Neural Networks, IEEE Transactions on, 15(5), 1063–1070.*

Izhikevich. (2006). Dynamical Systems in Neuroscience. *The Geometry of Excitability and Bursting (Computational Neuroscience*.

James. (1985). Classification Algorithms.

Jayalakshmi and Santhakumaran. (2009). Improving the convergence of Back Propagation Neural Network by Tuning Learning Rate Parameter. (International journal of emerging technologies and applications in engineering, technology and sciences, Vol.2. Issue 2, pp.298-301.).

Jianguo, X. and Embrechts, M. J. (2001). Supervised learning with spiking neural networks. Proceedings of the 2001 *Neural Networks, 2001. Proceedings. IJCNN '01. International Joint Conference on*. 2001. 1772-1777 vol.1773.

Jin, X. (2011). Chemical Reaction Optimization for Task Scheduling in Grid Computing. *IEEE Transactions on Parallel and Distributed Systems*, 22(10), 1624-1631.

Jin, Y.-J., Shen, B.-x., Ruo-fanRen, Yang, L., Sui, J. and Zhao, J.-g. (2013). Prediction of the Stryrene Butiend Rubber Performance by Emulsion polymerization using Backpropagation Neural Network (Hindawi Publishing corproration jornal of engineering , .http://dx.doiorg/10.1155/2013/515704.), 515704,515706.

Joarder. (2002). Arctangent activation function accelerate back propagation learning. (Monash University,Australia IEEE trans.Fundamentals .vol.E85-A.No.10).

Jones. (2005). AI Application Programming. 2nd Ed. Hingham, Massachusetts Charles River Media Inc.

Kamruzzaman and AZIZ. (2002). Anot on activation function in maultilayayer feedforward., (Proc.of international Joint Conference on network 2002.IEEE.2002.519-523).

Kasabov, N. (2009). Integrative connectionist learning systems inspired by nature: current models, future trends and challenges. 8(2), 199-218. doi: 10.1007/s11047-008-9066-z

Kasabov, N. (2010). To spike or not to spike: A probabilistic spiking neuron model. *Neural Networks*, 23(1), 16-19. doi: http://dx.doi.org/10.1016/j.neunet.2009.08.010

Kasabov, N. (2012a). Evolving, Probabilistic Spiking Neural Networks and Neurogenetic Systems for Spatio- and Spectro-Temporal Data Modelling and Pattern Recognition. (IEEE WCCI 2012, LNCS 7311, pp. 234–260.

Kasabov, N. (2012b). Evolving, Probabilistic Spiking Neural Networks and Neurogenetic Systems for Spatio- and Spectro-Temporal Data Modelling and Pattern Recognition. (IEEE WCCI 2012, LNCS 7311, pp. 234–260,

Kempter and Hemmen. (1999). Hebbian Learning and Spiking Neurons,Physical. E4, p. 4498-4514.

Kennedy and Eberhart. (1997). A discrete binary version of the particle swarm algorithm.

Khanesar, Teshnehlab and Shoorehdeli. (2007). A novel binary particle swarm optimization. (In Control automation '07. mediterranean conference on (p. 1 -6).).

Knerr, Personnaz and Dreyfus. (1992). Handwritten digit recognition by neural networks with single-layer training. (IEEE Trans. Neural Networks, vol. 3, pp. 962–968).

Koch. (1999). Biophysics of Computation: Information Processing in Single Neurons. Oxford University Press: New York.

Kojima, H. and Katsumata, S. (2009). An analysis of synaptic transmission and its plasticity by glutamate receptor channel kinetics models and 2-photon laser photolysis. Paper presented at the *Proceedings of the 15th international conference on Advances in neuro-information processing - Volume Part I*, Auckland, New Zealand.

Lacher, Coats, Sharma and Fant. (1995). A neural network for classifying the financial health of a firm. (Eur. J. Oper. Res., vol. 85, pp. 53–65).

Lampinen, Smolander and Korhonen. (1998). Wood surface inspection system based on generic visual features. (Singapore: World Scientific, 1998, pp. 35–42.).

Lazzarini and Kobata. (2008). Clinically Relevant Patch Test. Seidenari

Leshno and Spector. (1996). Neural network prediction analysis: The bankruptcy case," Neurocomput. (vol. 10, pp. 125–147.).

Lippmann. (1987). An introduction to computing with neural networks. *IEEE ASSP Mag., 4: 22.*

Loiselle, Rouat, Pressnitzer and Thorpe. (2005). Exploration of rank order coding with spiking neural networks for speech recognition.

Long, Lyle, Gupta and Ankur. (2008). Biologically-Inspired Spiking Neural Networks with Hebbian Learning for Vision Processing,. (AIAA Paper No. 2008-0885, presented at the 46th AIAA Aerospace Sciences Meeting, Reno, NV, Jan. 7- 10, 2008.).

Luger and Stubblefield. (1998). Artificial Intelligence Structures And Strategies For Complex Problem Solving (Addison Weley Longman , Inc., USA).

Maass, Schnitger and Sontag. (1991). On the computational power of sigmoid versus boolean threshold circuits. (Proc. of the 32nd Annual IEEE Symposium on Foundations of Computer Science, p. 767-776.).

Maass. (1997). The Third Generation of Neural Network Models. (Technische Universität Graz).

Maass. (2001). Computing with spiking neurons: In: Maass, W., Bishop, C., Pulsed Neural Networks, The MIT Press, Cambridge, MA. 2, 55-85.

Malsburg, v. d. (1999). The what and why of binding: The modeler's perspective. (Neuron, 24, 95-104.).

Malsburg, v. d. and Schneider. (1986). A neural cocktail-party processor. (Jybern., 54, 29-40.).

Marom and Abbott. (1994). Modeling state-dependent inactivation of membrane currents. (Biophysical Journal. 67 2 , 515-520).

Martin and Pitman. (1991). Recognizing hand-printed letter and digits using backpropagation learning,. *Neural Comput*, 258–267.

McLachlan and Ambroise. (2004). Analyzing microarray gene expression data. Wiley.

Mehrtash, N., Jung, D., Hellmich, H. H., Schoenauer, T., Lu, V. T. and Klar, H. (2003). Synaptic plasticity in spiking neural networks (SP²INN): a system approach. *Neural Networks, IEEE Transactions on*, 14(5), 980-992. doi: 10.1109/tnn.2003.816060

Meissner, M., Schmuker, M. and Schneider, G. (2006). Optimized Particle Swarm Optimization ({OPSO}) and its application to artificial neural network training. *BMC Bioinformatics, 2006, 7, 125*, 7(1), 125-136. doi: citeulike-article-id:6592206

Mendes, Cortez, Rocha and Neves. Particle swarms for feedforward neural network training. (In Neural networks, 2002. ijcnn ’02. proceedings of the 2002 international joint conference on (Vol. 2, p. 1895 -1899).).

Mergenthaler and Engbert (Eds.). (2007). *Modeling the control of fixational eye movements with neurophysiological delays.*: Phys. Rev. Lett. 98, 138104.

Michie, Spiegelhalter, Taylor and Eds. (1994). Machine Learning, Neural, and Statistical Classification. (London, U.K.: Ellis Horwood).

Mishra, D., Yadav, A., Dwivedi, A. and Kalra, P. K. (2006). A Neural Network Using Single Multiplicative Spiking Neuron for Function Approximation and Classification. Proceedings of the 2006 *Neural Networks, 2006. IJCNN '06. International Joint Conference on*. 0-0 0. 396-403.

Mohemmed, A., Johnston, M. and Zhang, M. (2011). Particle swarm optimisation based AdaBoost for object detection. *Soft Computing*, 15(9), 1793-1805. doi: 10.1007/s00500-010-0615-x

Moore. (2002). Back-propagation in spiking neural networks. *M.Sc. Thesis, Department of Computer Science, University of Bath, UK*

Naganathan, R.Venkatesh and Maheswari, U. (2008). Intelligent tutoring system: predicting students results using neural networks. *Journal of Convergence Informarion Technology*, Vol. 3 No. 3, .

Nakazawa, Quirk, Chitwood, Watanabe, Yeckel, Sun, et al. (2002). Requirement for hippocampal CA3 NMDA receptors in associative memory recall. (Science 297, 211–218).

NatschlÄager and Ruf. (1998). Spatial and temporal pattern analysis via spiking neurons. 9(3):319,332.

Natschlager and Ruf. (1998). Spatial and temporal pattern analysis via spiking neurons. (Network: Computation in Neural Systems. 9 3 , 319-338).

Negnevitsky. (2002). Artificial intelligence. A guide to intelligent systems, Addison Wesley.

Nelson and Rinzel. (1995). The Hodgkin-Huxley model, In: Bower, J. M., Beeman. (Springer-Verlag, New York. 4 , 27-51).

Neuron, G. a. (2007). *New neuron models have been developed using the simulation tools of Genesis and Neuron*.

Pan, Q.-K., Suganthan, P. N., Wang, L., Gao, L. and Mallipeddi, R. (2011). A differential evolution algorithm with self-adapting strategy and control parameters. *Computers & Operations Research*, 38(1), 394-408. doi: http://dx.doi.org/10.1016/j.cor.2010.06.007

Peng-Yeng. (2004). A discrete particle swarm algorithm for optimal polygonal approximation of digital curves. Journal of Visual Communication and Image Representation, . (15(2), 241 - 260.).

Pouget, Deneve, Ducom and Latham. (1999). Narrow vs. wide tuning curves. (What's best for a population code? Neural Computation, 11, 85–90.).

Price and Storn. (1997). Differential evolution: A simple evolution strategy for fast optimization. (Dr. Dobb's J., vol. 22, no. 4, pp. 18–24).

Price. (1997). Differential evolution vs. the functions of the 2nd ICEO. (in Proc. IEEE Int. Conf. Evol. Comput., Apr. pp. 153–157.).

Qasem and Shamsuddin. (2011). Radial Basis Function Network Based on Time Variant Multi-Objective Particle Swarm Optimization for medical diseases diagnosis. Applied Soft Computing. *IEEE, San Jose, CA*, 11(11): 1427–1438.

Qasem, S. N. and Shamsuddin, S. M. (2010). Memetic Elitist Pareto Differential Evolution Algorithm based Radial Basis Function Neural Networks for Classification Problems. (Applied Soft Computing, Elsevier).

Qasem, S. N. and Shamsuddin, S. M. (2011). Radial basis function network based on time variant multi-objective particle swarm optimization for medical diseases diagnosis. *Appl. Soft Comput.*, 11(1), 1427-1438. doi: 10.1016/j.asoc.2010.04.014

Qin, A. K., Huang, V. L. and Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. *Trans. Evol. Comp*, 13(2), 398-417. doi: 10.1109/tevc.2008.927706

Quinlan. (1993). Programs for Machine Learning. Morgan Kaufman.

Rahnamayan, S., Tizhoosh, H. R. and Salama, M. M. A. (2008). Opposition-Based Differential Evolution. *Evolutionary Computation, IEEE Transactions on*, 12(1), 64-79. doi: 10.1109/tevc.2007.894200

Rebizant. (2011). Fundamentals of System Analysis and Synthesis. Digital Signal Processing in Power System Protection and Control Signals and Communication Technology. (29-52.).

Roman, N. (1996). Teoretické otázky neuronových sítí, Matfyzpress, Praha.

Rosenblatt. (1961). Principles of neurodynamics: Perception and the theory of brain mechanisms. (Washington, CD: Spartan Books).

Roy, S. (1994). Factors influencing the choice of a learning rate for a backpropagation neural network. Proceedings of the 1994 *Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on*. 27 Jun-2 Jul 1994. 503-507 vol.501.

Rumelhart and Hinton. (1986). Learning internal representations by error propagation. (In D. E. Rumelhart and J. L.).

S´ıma. (2009). Gradient Learning in Networks of Smoothly Spiking Neurons *Advances in Neuro-Information Processing* (pp. 179-186): Springer-Verlag.

Sander, B., Joost, K. and Han, L. P. (2000). Spike-prop: error- Backpropagation in multi-layer networks of spiking neurons. ((Tech.Rep.) CWI Technical Report SEN-R0036.).

Schliebs, S., Defoin-Platel, M., Worner, S. and Kasabov, N. (2009). Integrated feature and parameter optimization for an evolving spiking neural network: Exploring heterogeneous probabilistic models. *Neural Networks*, 22(5–6), 623-632. doi: http://dx.doi.org/10.1016/j.neunet.2009.06.038

Schraudolph and Graepel. (2002). Towards stochastic conjugate gradient methods “,Neural information Processing. *Singapore*.

Schrauwen, B. and Campenhout., J. V. (2007). Parallel hardware implementation of a broad class of spiking neurons using serial arithmetic. In M. Verleysen, editor, Proceedings of the 14th European Symposium on Artificial Neural Networks. 623–628, Evere, 624

Schrauwen, B. and Van Campenhout, J. (2006). Backpropagation for Population-Temporal Coded Spiking Neural Networks. Proceedings of the 2006 *Neural Networks, 2006. IJCNN '06. International Joint Conference on*. 0-0 0. 1797-1804.

Shadlen and Movshon. (1999). Synchrony unbound: A critical evaluation of the temporal binding hypothesis. (Neuron), 24, 67-77.

Shamsuddin, Sallehuddin, R. and Norfadzila. (2008). Artificial Neural Network Time Series Modeling forRevenue Forecasting.

Shee, B. K., Vipsita, S. and Rath, S. K. (2011). Protein feature classification using particle swarm optimization and artificial neural networks. Paper presented at the *Proceedings of the 2011 International Conference on Communication, Computing & Security*, Rourkela, Odisha, India.

Shi. (2004). Particle swarm optimization. (IEEE neural network society: 8-13).

Snippe and Koenderink. (1992). Discrimination thresholds for channelcoded systems. Biological Cybernetics. 66, 543–551.

Srinivasan and Shanmugalakshmi. (2012). Neural Approach for Resource Selection with PSO for Grid Scheduling. (. International Journal of Computer Applications 53(11):37-41, September Published by Foundation of Computer Science, New York, USA. BibTeX).

Storn and Price. (1997). Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. (J. Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.).

Storn. (1996). On the usage of differential evolution for function optimization. (Proc. North Am. Fuzzy Inform. Process. Soc. 519–523.).

Suel, Garcia, Liberman and Elowitz. (2006). An excitable gene regulatory circuit induces transient cellular differentiation.

the 7th international conference on evolutionary programming vii *London, UK: Springer-Verlag.*

Thorpe and Gaustrais. (1998). Rank Order Coding. (In: Bower, J. (ed), Computational Neuroscience: Trends in Research. Plenum Press, New York.).

Thorpe, S., Delorme, A. and Van Rullen, R. (2001). Spike-based strategies for rapid processing. *Neural Networks*, 14(6–7), 715-725. doi: http://dx.doi.org/10.1016/S0893-6080(01)00083-1

Tiˇno and Mills. (2005). Learning beyond finite memory in recurrent networks of spiking neurons. (Neural Computation 18 , 591–613).

Valle, Venayagamoorthy, Mohagheghi, Hernandez and Harley. (2008). Particle swarm optimization. (Basic concepts variants and applications in power systems. Evolutionary Computation, IEEE Transactions on, 12(2), 171 -195.).

Vardakos, S., Gutierrez, M. and Xia, C. (2012). Parameter identification in numerical modeling of tunneling using the Differential Evolution Genetic Algorithm (DEGA). *Tunnelling and Underground Space Technology*, 28(0), 109-123. doi: http://dx.doi.org/10.1016/j.tust.2011.10.003

Wachowiak, M. P., Smolikova, R., Yufeng, Z., Zurada, J. M. and Elmaghraby, A. S. (2004). An approach to multimodal biomedical image registration utilizing particle swarm optimization. *Evolutionary Computation, IEEE Transactions on*, 8(3), 289-301. doi: 10.1109/tevc.2004.826068

Wang, Tang, Tamura and Sun. (2003). An Improved Back propagation Algorithm To Avoid The Local Minima Problem. (Neurocomputing 56:455-460.).

Wasserman. (2004). Neural Computing Theory And Practice (Van Nostrand Reinhold, New York).

Werbos, P. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.* Harvard University.

Widrow, B., Rumelhart, D. E. and Lehr, M. A. (1994). Neural networks: applications in industry, business and science. *Commun. ACM*, 37(3), 93-105. doi: 10.1145/175247.175257

Wilson and Sharda. (1994). Bankruptcy prediction using neural networks. (Decision Support Syst., vol. 11, pp. 545–557.).

Wysoski, Benuskova and Kasabov. (2006). On-line learning with structural adaptation in a network of spiking neurons for visual pattern recognition. (In Artificial Neural Networks ICANN Berlin / Heidelberg: Springer.), 61-70.

Wysoski, S. G., Benuskova, L. and Kasabov, N. (2007). Text-independent speaker authentication with spiking neural networks. Paper presented at the *Proceedings of the 17th international conference on Artificial neural networks*, Porto, Portugal.

Yoshida, H., Kawata, K., Fukuyama, Y., Takayama, S. and Nakanishi, Y. (2000). A particle swarm optimization for reactive power and voltage control considering voltage security assessment. *Power Systems, IEEE Transactions on*, 15(4), 1232-1239. doi: 10.1109/59.898095

Yu, J., Xi, L. and Wang, S. (2007). An Improved Particle Swarm Optimization for Evolving Feedforward Artificial Neural Networks. *Neural Processing Letters*, 26(3), 217-231. doi: 10.1007/s11063-007-9053-x

Yuan, X., Nie, H., Su, A., Wang, L. and Yuan, Y. (2009). An improved binary particle swarm optimization for unit commitment problem. *Expert Syst. Appl.*, 36(4), 8049-8055. doi: 10.1016/j.eswa.2008.10.047

Zainud-Deen, S. H., Hassen, W. M., Ali, E. M., Awadalla, K. H. and Sharshar, H. A. (2008). Breast cancer detection using a hybrid Finite difference frequency domain and particle swarm optimization techniques. Proceedings of the 2008 *Radio Science Conference, 2008. NRSC 2008. National*. 18-20 March 2008. 1-8.

Zhang, Patuwo and Indro. (1999). Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis. *Eur. J. Oper*, 16–32.

Zurada. (1994). Introduction To Artificial Neural Systems “, Jaico Publishing House , Mumbai *Introduction To Artificial Neural Systems*

Zweiri, Y. H., Whidborne, J. F. and Seneviratne, L. D. (2003). A three-term backpropagation algorithm. *Neurocomputing*, 50, 305-318. doi: 10.1016/s0925-2312(02)00569-6