Article

Bound the parameters of neural networks using Particle Swarm Optimization

Ioannis G. Tsoulos[1,*], Alexandros Tzallas[1], Evangelos Karvounis[1], Dimitrios Tsalikakis[2]

[1] Department of Informatics and Telecommunications, University of Ioannina, Greece
[2] University of Western Macedonia, Department of Engineering Informatics and Telecommunications, Greece
* Correspondence: itsoulos@uoi.gr;

Abstract: Artificial Neural Networks are machine learning models widely used in many sciences as well as in practical applications. The basic element of these models is a vector of parameters, the values of which should be estimated using some computational method, and this process is called training. For effective training of the network, computational methods from the field of global minimization are often used. However, for global minimization techniques to be effective, the bounds of the objective function should also be clearly defined. In this paper, a two-stage global optimization technique is presented for efficient training of artificial neural networks. In the first stage, the bounds for the neural network parameters are estimated using Particle Swarm Optimization and in the second phase the network parameters are trained within the bounds of the first phase using global optimization techniques. The proposed technique was applied to a number of well-known problems in the literature and the experimental results were more than encouraging.

Keywords: Global optimization; local optimization; stochastic methods; evolutionary techniques; termination rules.

Citation: Tsoulos, I.G.; Tzallas, A.I; Karvounis E; Tsalikakis D Bound the parameters of neural networks using Particle Swarm Optimization. *Journal Not Specified* **2022**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Artificial neural networks (ANNs) are parametric machine learning models [1,2], which have been widely used during the last decades in a variety of practical problems from many different fields such as physics problems [3–5], chemistry problems [6–8], problems related to medicine [9,10], economic problems [11–13] etc. Also, recently ANNs have been applied to models solving Differential Equations [14,15], agriculture problems [16,17], facial expression recognition [18], wind speed prediction [19], the gas consumption problem [20], intrusion detection [21] etc. Usually, neural networks are defined as a function $N(\overrightarrow{x},\overrightarrow{w})$, where the vector $\overrightarrow{x}$ is the input pattern to the networks and the vector $\overrightarrow{w}$ is called the weight vector. To estimate the weight vector, the so-called training error is minimized, which is defined as the sum:

$$E\left(N\left(\overrightarrow{x},\overrightarrow{w}\right)\right)=\sum_{i=1}^{M}\left(N\left(\overrightarrow{x}_i,\overrightarrow{w}\right)-y_i\right)^2 \quad (1)$$

In equation 1 the values t $\left(\overrightarrow{x_i},y_i\right),\ i=1,...,M$ defined the training set for the neural network. The values y_i denote the expected output for the pattern $\overrightarrow{x_i}$.

To minimize the quantity in equation 1, several techniques have been proposed in the relevant literature such as: Back Propagation method [22,23], the RPROP method [24–26], Quasi Newton methods [28,29], Simulated Annealing [30,31], Genetic Algorithms [32,33], Particle Swarm Optimization [34,35],Differential Optimization methods [36], Evolutionary Computation [37], the Whale optimization algorithm [38], the Butterfly optimization algorithm [39], etc. In addition, many researchers have focused their attention on techniques for initializing the parameters of artificial neural networks, such as the usage of decsion trees to initialize neural networks [40], a method based on Cachy's inequality [41], usage of genetic algorithms [42], initialization based on discriminant learning [43] etc. In addition, many researchers were also concerned with the construction of artificial neural network architectures, such as usage of Cross Validation to construct the architecture of neural networks [44], incorporation of the Grammatical Evolution technique [46] to construct the architecture of neural networks as well as to estimate the values of the weights [45], evolution of neural networks using a method based on cellurar automata [47] etc. Also, since in recent years there has been a leap forward in the development of parallel architectures, a number of works have been presented that take advantage of such computational techniques [48,49].

However, in many cases the training methods of artificial neural networks suffer from the problem of overfitting, i.e. although they succeed in significantly reducing the training error of equation 1 they do not perform similarly on unknown data that was not present during training. These unknown data are commonly called test set. The overfitting problem usually is tackled through a variety of methods, such as weight sharing [50,51], pruning of parameters, i.e reducing the size of the network [52–54], the dropout technique [55,56], weight decaying [57,58], the Sarporp method [59], positive correlation methods [60] etc. The overfitting problem is thoroghly discussed in the Geman et all [61] and in the artcile of Hawkins [62].

A key reason why the problem of overtraining in artificial neural networks is present, is that there is no well-defined interval of values in which the network parameters are initialized and trained by the optimization methods. This in practice means that the values of the parameters are changed indiscriminately in order to reduce the value of equation 1. In this work it is proposed to use the Particle Swarm Optimization (PSO) technique [63] for the reliable calculation of the value interval of the parameters of an artificial neural network. The PSO method was chosen since it is a fairly fast global optimization method, easily adaptable to any optimization problem, and does not require many execution parameters to be defined by the user. The PSO method was applied with success to many difficult problems such as problems that appear in physics [64,65], chemistry [66,67], medicine [68,69], economics [70] etc. In the proposed method, the PSO technique is used to minimize equation 1 to which a penalty factor has been added so as not to allow the parameters of artificial neural networks to vary uncontrollably. After the minimization of the modified function is done, the parameters of the neural network are initialized in an interval of values around the optimal value located by the PSO method, and then the original form of equation 1 is minimized without a penalty factor this time.

The rest of this article is organized as follows: in section 2 the proposed method is fully analyzed and discussed, in section 3 the experimental datasets as well as the experimental results are listed and discussed and finally in section 4 some conclusions are presented.

2. The proposed method

2.1. Preliminaries

Consider a neural network wihe one processing level that uses the so - called sigmoid function as activation function. The sigmoid function is defined as:

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{2}$$

This function is graphically illustrated in Figure 1.

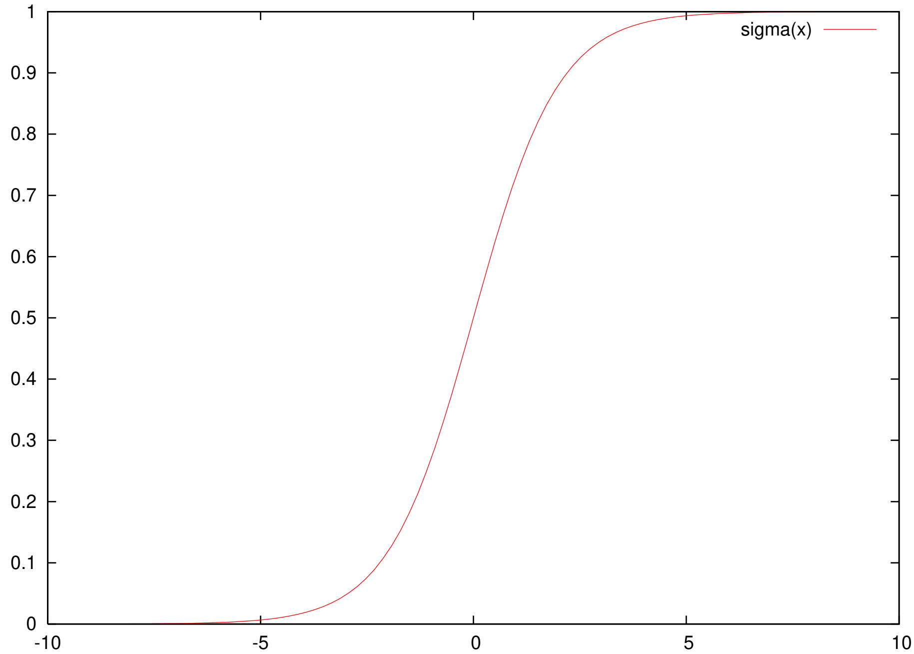

Figure 1. Plot of the sigmoid function.

The equation for every hidden node of the neural network has as:

$$o_i(x) = \sigma\left(w_i^T x + \theta_i\right), \tag{3}$$

where the vector w_i stands for the the weight vector and the value θ_i denotes the bias the node i. The equation for a neural network with H hidden is defined as:

$$N(x) = \sum_{i=1}^{H} v_i o_i(x), \tag{4}$$

The value v_i denotes the output weight for node i. Therefore writing the overall equation of the artificial neural network using equations 3 and 4 has as:

$$N(\overrightarrow{x}, \overrightarrow{w}) = \sum_{i=1}^{H} w_{(d+2)i-(d+1)} \sigma\left(\sum_{j=1}^{d} x_j w_{(d+2)i-(d+1)+j} + w_{(d+2)i}\right) \tag{5}$$

The value d stands for the dimension of input vector $\overrightarrow{x}$. Observing equation 5 but also the shape of the sigmoid function in figure 1, it is obvious that in many cases the sigmoid function is driven to 1 or 0 and therefore the training error of the neural network can get trapped in local minima and consequently the neural network it will lose its generalization abilities. Therefore, a technique should be devised by which the values of the sigmoid will be restricted to some interval of values. In the present work, the limitation of the neural network parameters to a range of values is carried out using the Particle Swarm Optimization method.

2.2. The PSO algorithm

The Particle Swarm Optimization method is based on a population of vectors that also called particles. These particles are also possible global minima of the objective function. Each particle is associated with two vectors: the current position denoted as $\overrightarrow{p}$ and the corresponding speed $\overrightarrow{u}$ at which they are moving towards to the global minimum. In addition, each particle maintains in the vector $p_{i,b}$ the best position in which it has been so far and the total population maintains in the vector p_{best} the best position that any of the particles have found in the past. The purpose of the method is to move the total population toward the global minimum through a series of iterations. In each iteration, the velocity of each particle is calculated based on its current position, its best position in the past and the optimal position of the population. The main steps of a PSO algorithm are shown in Algorithm 1.

Algorithm 1 The base PSO algorithm executed in one processing unit.

1. **Initialization Step** .
   (a) **Set** $k = 0$.
   (b) **Set** m as the total number of particles.
   (c) **Set** $\mathrm{k}_{\mathrm{max}}$ as the maximum number of iterations allowed for the serial algorithm.
   (d) **Initialize** randomly, the positions $p_1, p_2, ..., p_m$ for the particles. The initial positions must be within the domain of the objective function.
   (e) **Initialize** randomly the velocities $u_1, u_2, ..., u_m$.
   (f) **For** $i = 1..m$ do $p_{i,b} = p_i$. The vector $p_{i,b}$ represents the best located position of particle i.
   (g) **Set** $p_{\mathrm{best}} = \arg\min_{i \in 1..m} f(p_i)$
2. **Termination Check Step** . The termination criterion for the serial algorithm is checked. If the termination criterion is true, then the method terminates.
3. **For** $i = 1..m$ **Do**
   (a) **Compute** the velocity u_i as a combination of the vectors $u_i,\ p_{i.b}$ and p_{best}
   (b) **Set** the new position for the particle as: $p_i = p_i + u_i$
   (c) **Calculate** the $f(p_i)$ for particle p_i using the objective function $f(x)$.
   (d) **If** $f(p_i) \le f(p_{i,b})$ then $p_{i,b} = x_i$
4. **End For**
5. **Set** $p_{\mathrm{best}} = \arg\min_{i \in 1..m} f(p_i)$
6. **Set** $k = k + 1$.
7. **Goto** Step 2

3. Experiments

The proposed method is evaluated on a series of classification and regression problems from the relevant literature. The classification problems used for the experiments were found in most cases in two internet databases:

1. UCI dataset repository, https://archive.ics.uci.edu/ml/index.php
2. Keel repository, https://sci2s.ugr.es/keel/datasets.php[71].

The regression datasets are in most cases available from the Statlib URL ftp://lib.stat.cmu.edu/datasets/index.html. The proposed method is compared against a neural network trained by a genetic algorithm and the results are reported.

3.1. Experimental datasets

The following classification datasets were used:

1. **Appendictis** a medical dataset, proposed in [72].
2. **Australian** dataset [73], the dataset is related to credit card applications.
3. **Balance** dataset [74], which is used to predict psychological states.
4. **Cleveland** dataset, a dataset used to detect heart disease used in various papers[75,76].
5. **Bands** dataset, a printing problem used to identify cylinder bands.
6. **Dermatology** dataset [77], which is used for differential diagnosis of erythemato-squamous diseases.
7. **Hayes roth** dataset. This dataset[79] contains **5** numeric-valued attributes and 132 patterns.

8. **Heart** dataset [78], used to detect heart disease.
9. **HouseVotes** dataset [80], which is about votes in the U.S. House of Representatives Congressmen.
10. **Ionosphere** dataset. The ionosphere dataset contains data from the Johns Hopkins Ionosphere database and it has been studied in a bunch of papers [81,82].
11. **Liverdisorder** dataset [83], used for detect liver disorders in peoples using blood analysis.
12. **Mammographic** dataset [84]. This dataset be used to identify the severity (benign or malignant) of a mammographic mass lesion from BI-RADS attributes and the patient's age. It contains 830 patterns of 5 features each.
13. **Page Blocks** dataset [85], used to detect the page layout of a document.
14. **Parkinsons** dataset. This dataset is composed of a range of biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD)[86].
15. **Pima** dataset [87], used to detect the presence of diabetes.
16. **Popfailures** dataset [88], that is related to climate model simulation crashes of simulation crashes.
17. **Regions2** dataset. It is created from liver biopsy images of patients with hepatitis C [89]. From each region in the acquired images, 18 shape-based and color-based features were extracted, while it was also annotated form medical experts. The resulting dataset includes 600 samples belonging into 6 classes.
18. **Saheart** dataset [90], used to detect heart disease.
19. **Segment** dataset [91]. This database contains patterns from a database of 7 outdoor images (classes).
20. **Wdbc** dataset [92], which contains data for breast tumors.
21. **Wine** dataset, used to detect through chemical analysis determine the origin of wines and is been used in various research papers [93,94].
22. **Eeg** datasets. As an real word example, consider an EEG dataset described in [17] is used here. The dataset consists of five sets (denoted as Z, O, N, F and S) each containing 100 single-channel EEG segments each having 23.6 sec duration. With different combinations of these sets the produced datasets are Z_F_S, ZO_NF_S, ZONF_S.
23. **Zoo** dataset [95], where the task is classify animals in seven predefined classes.

Also, the following regression datasets were used:

1. **Abalone** dataset [97]. This data set can be used to obtain a model to predict the age of abalone from physical measurements.
2. **Airfoil** dataset, which is used by the NASA for a series of aerodynamic and acoustic tests [98].
3. **Baseball** dataset, a dataset to predict the salary of baseball players.
4. **BK** dataset. This dataset comes from Smoothing Methods in Statistics [99] and is used to estimate the points scored per minute in a basketball game.
5. **BL** dataset: This dataset can be downloaded from StatLib. It contains data from an experiment on the affects of machine adjustments on the time to count bolts.
6. **Concrete** dataset. This dataset is taken from civil engineering[100].
7. **Dee** dataset, used to predict the daily average price of the electricity energy in Spain.
8. **Diabetes** dataset, a medical dataset.
9. **Housing** dataset. This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University and it is described in [101].
10. **FA** dataset, which contains percentage of body fat and ten body circumference measurements. The goal is to fit body fat to the other measurements.
11. **MB** dataset. This dataset is available from Smoothing Methods in Statistics [102] and it includes 61 patterns.
12. **MORTGAGE** dataset, which contains the Economic data information of USA.
13. **PY** dataset, (Pyrimidines problem). The source of this dataset is the URL: https://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html and it is a problem of 27

attributes and 74 number of patterns. The task consists of Learning Quantitative Structure Activity Relationships (QSARs) and provided by [103].

14. **Quake** dataset. The objective here is to approximate the strength of a earthquake.
15. **Treasure** dataset, which contains Economic data information of USA from 01/04/1980 to 02/04/2000 on a weekly basis.
16. **Wankara** dataset, which contains weather information.

3.2. Experimental results

Table 1. Experimental results for the classification datasets.

DATASET	GENETIC	ADAM	RPROP	NEAT	PSO1	PSO2
Appendicitis	18.10%	16.50%	16.30%	17.20%	17.07%	17.90%
Australian	32.21%	35.65%	36.12%	31.98%	18.26%	17.37%
Balance	8.97%	7.87%	8.81%	23.14%	7.12%	7.43%
Bands	35.75%	36.25%	36.32%	34.30%	34.89%	34.31%
Cleveland	51.60%	67.55%	61.41%	53.44%	49.85%	49.52%
Dermatology	30.58%	26.14%	15.12%	32.43%	11.20%	10.81%
Hayes Roth	56.18%	59.70%	37.46%	50.15%	35.90%	37.03%
Heart	28.34%	38.53%	30.51%	39.27%	19.43%	20.26%
HouseVotes	6.62%	7.48%	6.04%	10.89%	6.91%	7.03%
Ionosphere	15.14%	16.64%	13.65%	19.67%	13.61%	13.98%
Liverdisorder	31.11%	41.53%	40.26%	30.67%	32.39%	31.95%
Lymography	23.26%	29.26%	24.67%	33.70%	26.91%	26.40%
Mammographic	19.88%	46.25%	18.46%	22.85%	17.47%	17.40%
PageBlocks	8.06%	7.93%	7.82%	10.22%	7.05%	6.88%
Parkinsons	18.05%	24.06%	22.28%	18.56%	15.49%	14.69%
Pima	32.19%	34.85%	34.27%	34.51%	26.26%	25.62%
Popfailures	5.94%	5.18%	4.81%	7.05%	5.26%	5.32%
Regions2	29.39%	29.85%	27.53%	33.23%	28.30%	28.86%
Saheart	34.86%	34.04%	34.90%	34.51%	32.59%	32.25%
Segment	57.72%	49.75%	52.14%	66.72%	15.95%	15.61%
Wdbc	8.56%	35.35%	21.57%	12.88%	4.79%	4.67%
Wine	19.20%	29.40%	30.73%	25.43%	8.12%	8.53%
Z_F_S	10.73%	47.81%	29.28%	38.41%	8.08%	7.97%
ZO_NF_S	8.41%	47.43%	6.43%	43.75%	5.43%	5.47%
ZONF_S	2.60%	11.99%	27.27%	5.44%	2.44%	2.51%
ZOO	16.67%	14.13%	15.47%	20.27%	5.20%	5.27%
AVERAGE	**23.47%**	**30.81%**	**25.37%**	**28.87%**	**17.54%**	**17.50%**

Table 2. Experiments for regression datasets.

DATASET	GENETIC	ADAM	RPROP	NEAT	PSO1	PSO2
ABALONE	7.17	4.30	4.55	9.88	4.39	4.33
AIRFOIL	0.003	0.005	0.002	0.067	0.003	0.003
BASEBALL	103.60	77.90	92.05	100.39	62.61	56.59
BK	0.027	0.03	1.599	0.15	0.027	0.027
BL	5.74	0.28	4.38	0.05	0.006	0.005
CONCRETE	0.0099	0.078	0.0086	0.081	0.003	0.003
DEE	1.013	0.63	0.608	1.512	0.23	0.23
DIABETES	19.86	3.03	1.11	4.25	0.59	0.62
HOUSING	43.26	80.20	74.38	56.49	22.69	15.51
FA	1.95	0.11	0.14	0.19	0.02	0.02
MB	3.39	0.06	0.055	0.061	0.05	0.05
MORTGAGE	2.41	9.24	9.19	14.11	0.21	0.16
PY	105.41	0.09	0.039	0.075	0.10	0.11
QUAKE	0.04	0.06	0.041	0.298	0.04	0.04
TREASURY	2.929	11.16	10.88	15.52	0.28	0.21
WANKARA	0.012	0.02	0.0003	0.005	0.0002	0.0002
AVERAGE	**18.55**	**11.70**	**12.44**	**12.70**	**5.70**	**4.87**

4. Conclusions

Author Contributions: All authors have conceived of the idea and methodology. A.T. and I.G.T. conducted the experiments, employing several datasets and provided the comparative experiments. D.T. and E.K. performed the statistical analysis and prepared the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Institutional Review Board Statement: Not applicable.

Acknowledgments: The experiments of this research work were performed at the high performance computing system established at Knowledge and Intelligent Computing Laboratory, Department of Informatics and Telecommunications, University of Ioannina, acquired with the project "Educational Laboratory equipment of TEI of Epirus" with MIS 5007094 funded by the Operational Programme "Epirus" 2014–2020, by ERDF and national funds.

Conflicts of Interest: The authors declare no conflict of interest.

Sample Availability: Not applicable.

References

1. C. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.
2. G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of Control Signals and Systems **2**, pp. 303-314, 1989.
3. P. Baldi, K. Cranmer, T. Faucett et al, Parameterized neural networks for high-energy physics, Eur. Phys. J. C **76**, 2016.
4. J. J. Valdas and G. Bonham-Carter, Time dependent neural network models for detecting changes of state in complex processes: Applications in earth sciences and astronomy, Neural Networks **19**, pp. 196-207, 2006
5. G. Carleo,M. Troyer, Solving the quantum many-body problem with artificial neural networks, Science **355**, pp. 602-606, 2017.
6. Lin Shen, Jingheng Wu, and Weitao Yang, Multiscale Quantum Mechanics/Molecular Mechanics Simulations with Neural Networks, Journal of Chemical Theory and Computation **12**, pp. 4934-4946, 2016.
7. Sergei Manzhos, Richard Dawes, Tucker Carrington, Neural network-based approaches for building high dimensional and quantum dynamics-friendly potential energy surfaces, Int. J. Quantum Chem. **115**, pp. 1012-1020, 2015.
8. Jennifer N. Wei, David Duvenaud, and Alán Aspuru-Guzik, Neural Networks for the Prediction of Organic Chemistry Reactions, ACS Central Science **2**, pp. 725-732, 2016.

9. Igor I. Baskin, David Winkler and Igor V. Tetko, A renaissance of neural networks in drug discovery, Expert Opinion on Drug Discovery **11**, pp. 785-795, 2016.
10. Ronadl Bartzatt, Prediction of Novel Anti-Ebola Virus Compounds Utilizing Artificial Neural Network (ANN), Chemistry Faculty Publications **49**, pp. 16-34, 2018.
11. Lukas Falat and Lucia Pancikova, Quantitative Modelling in Economics with Advanced Artificial Neural Networks, Procedia Economics and Finance **34**, pp. 194-201, 2015.
12. Mohammad Namazi, Ahmad Shokrolahi, Mohammad Sadeghzadeh Maharluie, Detecting and ranking cash flow risk factors via artificial neural networks technique, Journal of Business Research **69**, pp. 1801-1806, 2016.
13. G. Tkacz, Neural network forecasting of Canadian GDP growth, International Journal of Forecasting **17**, pp. 57-69, 2001.
14. Y. Shirvany, M. Hayati, R. Moradian, Multilayer perceptron neural networks with novel unsupervised training method for numerical solution of the partial differential equations, Applied Soft Computing **9**, pp. 20-29, 2009.
15. A. Malek, R. Shekari Beidokhti, Numerical solution for high order differential equations using a hybrid neural network—Optimization method, Applied Mathematics and Computation **183**, pp. 260-271, 2006.
16. A. Topuz, Predicting moisture content of agricultural products using artificial neural networks, Advances in Engineering Software **41**, pp. 464-470, 2010.
17. A. Escamilla-García, G.M. Soto-Zarazúa, M. Toledano-Ayala, E. Rivas-Araiza, A. Gastélum-Barrios, Abraham,Applications of Artificial Neural Networks in Greenhouse Technology and Overview for Smart Agriculture Development, Applied Sciences **10**, Article number 3835, 2020.
18. H. Boughrara, M. Chtourou, C. Ben Amar et al, Facial expression recognition based on a mlp neural network using constructive training algorithm. Multimed Tools Appl **75**, pp. 709–731, 2016.
19. H. Liu, H.Q Tian, Y.F. Li, L. Zhang, Comparison of four Adaboost algorithm based artificial neural networks in wind speed predictions, Energy Conversion and Management **92**, pp. 67-81, 2015.
20. J. Szoplik, Forecasting of natural gas consumption with artificial neural networks, Energy **85**, pp. 208-220, 2015.
21. H. Bahram, N.J. Navimipour, Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm, ICT Express 5, pp. 56-59, 2019.
22. D.E. Rumelhart, G.E. Hinton and R.J. Williams, Learning representations by back-propagating errors, Nature **323**, pp. 533 - 536 , 1986.
23. T. Chen and S. Zhong, Privacy-Preserving Backpropagation Neural Network Learning, IEEE Transactions on Neural Networks **20**, , pp. 1554-1564, 2009.
24. M. Riedmiller and H. Braun, A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP algorithm, Proc. of the IEEE Intl. Conf. on Neural Networks, San Francisco, CA, pp. 586–591, 1993.
25. T. Pajchrowski, K. Zawirski and K. Nowopolski, Neural Speed Controller Trained Online by Means of Modified RPROP Algorithm, IEEE Transactions on Industrial Informatics **11**, pp. 560-568, 2015.
26. Rinda Parama Satya Hermanto, Suharjito, Diana, Ariadi Nugroho, Waiting-Time Estimation in Bank Customer Queues using RPROP Neural Networks, Procedia Computer Science **135**, pp. 35-42, 2018.
27. Neural Networks, Procedia Computer Science **135**, pp. 35-42, 2018.
28. B. Robitaille and B. Marcos and M. Veillette and G. Payre, Modified quasi-Newton methods for training neural networks, Computers & Chemical Engineering **20**, pp. 1133-1140, 1996.
29. Q. Liu, J. Liu, R. Sang, J. Li, T. Zhang and Q. Zhang, Fast Neural Network Training on FPGA Using Quasi-Newton Optimization Method,IEEE Transactions on Very Large Scale Integration (VLSI) Systems **26**, pp. 1575-1579, 2018.
30. A. Yamazaki, M. C. P. de Souto,T. B. Ludermir, Optimization of neural network weights and architectures for odor recognition using simulated annealing, In: Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 **1**, pp. 547-552 , 2002.
31. Y. Da, G. Xiurun, An improved PSO-based ANN with simulated annealing technique, Neurocomputing **63**, pp. 527-533, 2005.
32. F. H. F. Leung, H. K. Lam, S. H. Ling and P. K. S. Tam, Tuning of the structure and parameters of a neural network using an improved genetic algorithm, IEEE Transactions on Neural Networks **14**, pp. 79-88, 2003
33. X. Yao, Evolving artificial neural networks, Proceedings of the IEEE, 87(9), pp. 1423-1447, 1999.
34. C. Zhang, H. Shao and Y. Li, Particle swarm optimisation for evolving artificial neural network, IEEE International Conference on Systems, Man, and Cybernetics, , pp. 2487-2490, 2000.
35. Jianbo Yu, Shijin Wang, Lifeng Xi, Evolving artificial neural networks using an improved PSO and DPSO **71**, pp. 1054-1060, 2008.
36. J. Ilonen, J.K. Kamarainen, J. Lampinen, Differential Evolution Training Algorithm for Feed-Forward Neural Networks, Neural Processing Letters **17**, pp. 93–105, 2003.
37. M. Rocha, P. Cortez, J. Neves, Evolution of neural networks for classification and regression, Neurocomputing **70**, pp. 2809-2816, 2007.
38. I. Aljarah, H. Faris, S. Mirjalili, Optimizing connection weights in neural networks using the whale optimization algorithm, Soft Comput **22**, pp. 1–15, 2018.
39. S.M.J. Jalali, S. Ahmadian, P.M. Kebria, A. Khosravi, C.P. Lim, S. Nahavandi, Evolving Artificial Neural Networks Using Butterfly Optimization Algorithm for Data Classification. In: Gedeon, T., Wong, K., Lee, M. (eds) Neural Information Processing. ICONIP 2019. Lecture Notes in Computer Science(), vol 11953. Springer, Cham, 2019.

40. I. Ivanova, M. Kubat, Initialization of neural networks by means of decision trees, Knowledge-Based Systems **8**, pp. 333-344, 1995.
41. J.Y.F Yam, T.W.S. Chow, A weight initialization method for improving training speed in feedforward neural network, Neurocomputing **30**, pp. 219-232, 2000.
42. F. Itano, M. A. de Abreu de Sousa, E. Del-Moral-Hernandez, Extending MLP ANN hyper-parameters Optimization by using Genetic Algorithm, In: 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 2018, pp. 1-8, 2018.
43. K. Chumachenko, A. Iosifidis, M. Gabbouj, Feedforward neural networks initialization based on discriminant learning, Neural Networks **146**, pp. 220-229, 2022.
44. R. Setiono, Feedforward Neural Network Construction Using Cross Validation,Neural Computation **13**, pp. 2865-2877, 2001.
45. I.G. Tsoulos, D. Gavrilis, E. Glavas, Neural network construction and training using grammatical evolution, Neurocomputing 72, pp. 269-277, 2008.
46. M. O'Neill, C. Ryan, Grammatical evolution, IEEE Trans. Evol. Comput. **5**, pp. 349–358, 2001.
47. K.J. Kim, S.B. Cho, Evolved neural networks based on cellular automata for sensory-motor controller, Neurocomputing **69**, pp. 2193-2207, 2006.
48. M. Martínez-Zarzuela, F.J. Díaz Pernas, J.F. Díez Higuera, M.A. Rodríguez, Fuzzy ART Neural Network Parallel Computing on the GPU. In: Sandoval, F., Prieto, A., Cabestany, J., Graña, M. (eds) Computational and Ambient Intelligence. IWANN 2007. Lecture Notes in Computer Science, vol 4507. Springer, Berlin, Heidelberg, 2007.
49. A.A. Huqqani, E. Schikuta, S. Ye Peng Chen, Multicore and GPU Parallelization of Neural Networks for Face Recognition, Procedia Computer Science **18**, pp. 349-358, 2013.
50. S.J. Nowlan and G.E. Hinton, Simplifying neural networks by soft weight sharing, Neural Computation 4, pp. 473-493, 1992.
51. J.K. Kim, M.Y. Lee, J.Y. Kim, B. J. Kim, J. H. Lee, An efficient pruning and weight sharing method for neural network, In: 2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Seoul, Korea (South), pp. 1-2, 2016.
52. S.J. Hanson and L.Y. Pratt, Comparing biases for minimal network construction with back propagation, In D.S. Touretzky (Ed.), Advances in Neural Information Processing Systems, Volume 1, pp. 177-185, San Mateo, CA: Morgan Kaufmann, 1989.
53. M.C. Mozer and P. Smolensky, Skeletonization: a technique for trimming the fat from a network via relevance assesment. In D.S. Touretzky (Ed.), Advances in Neural Processing Systems, Volume 1, pp. 107-115, San Mateo CA: Morgan Kaufmann, 1989.
54. M. Augasta and T. Kathirvalavakumar, Pruning algorithms of neural networks — a comparative study, Central European Journal of Computer Science, 2003.
55. Nitish Srivastava, G E Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan R Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, Journal of Machine Learning Research **15**, pp. 1929-1958, 2014.
56. A. Iosifidis, A. Tefas, I. Pitas, DropELM: Fast neural network regularization with Dropout and DropConnect, Neurocomputing **162**, pp. 57-66, 2015.
57. A. Gupta, S.M. Lam, Weight decay backpropagation for noisy data, Neural Networks **11**, pp. 1127-1138, 1998.
58. M. Carvalho and T. B. Ludermir, Particle Swarm Optimization of Feed-Forward Neural Networks with Weight Decay, 2006 Sixth International Conference on Hybrid Intelligent Systems (HIS'06), Rio de Janeiro, Brazil, 2006, pp. 5-5.
59. N.K. Treadgold, T.D. Gedeon, Simulated annealing and weight decay in adaptive learning: the SARPROP algorithm, IEEE Trans. on Neural Networks 9, pp. 662-668, 1998.
60. M.D. Shahjahan, M. Kazuyuki, Neural network training algorithm with possitive correlation, IEEE Trans. Inf & Syst. **88**, pp. 2399-2409, 2005.
61. S. Geman, E. Bienenstock and R. Doursat, Neural networks and the bias/variance dilemma, Neural Computation 4 , pp. 1 - 58, 1992.
62. Douglas M. Hawkins, The Problem of Overfitting, J. Chem. Inf. Comput. Sci. **44**, pp. 1–12, 2004.
63. F. Marini, B. Walczak, Particle swarm optimization (PSO). A tutorial, Chemometrics and Intelligent Laboratory Systems **149**, pp. 153-165, 2015.
64. Anderson Alvarenga de Moura Meneses, Marcelo Dornellas, Machado Roberto Schirru, Particle Swarm Optimization applied to the nuclear reload problem of a Pressurized Water Reactor, Progress in Nuclear Energy **51**, pp. 319-326, 2009.
65. Ranjit Shaw, Shalivahan Srivastava, Particle swarm optimization: A new tool to invert geophysical data, Geophysics **72**, 2007.
66. C. O. Ourique, E.C. Biscaia, J.C. Pinto, The use of particle swarm optimization for dynamical analysis in chemical processes, Computers & Chemical Engineering **26**, pp. 1783-1793, 2002.
67. H. Fang, J. Zhou, Z. Wang et al, Hybrid method integrating machine learning and particle swarm optimization for smart chemical process operations, Front. Chem. Sci. Eng. **16**, pp. 274–287, 2022.
68. M.P. Wachowiak, R. Smolikova, Yufeng Zheng, J.M. Zurada, A.S. Elmaghraby, An approach to multimodal biomedical image registration utilizing particle swarm optimization, IEEE Transactions on Evolutionary Computation **8**, pp. 289-301, 2004.
69. Yannis Marinakis. Magdalene Marinaki, Georgios Dounias, Particle swarm optimization for pap-smear diagnosis, Expert Systems with Applications **35**, pp. 1645-1656, 2008.
70. Jong-Bae Park, Yun-Won Jeong, Joong-Rin Shin, Kwang Y. Lee, An Improved Particle Swarm Optimization for Nonconvex Economic Dispatch Problems, IEEE Transactions on Power Systems **25**, pp. 156-16**216**6, 2010.
71. J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera. KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. Journal of Multiple-Valued Logic and Soft Computing 17, pp. 255-287, 2011.

72. Weiss, Sholom M. and Kulikowski, Casimir A., Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems, Morgan Kaufmann Publishers Inc, 1991.
73. J.R. Quinlan, Simplifying Decision Trees. International Journal of Man-Machine Studies **27**, pp. 221-234, 1987.
74. T. Shultz, D. Mareschal, W. Schmidt, Modeling Cognitive Development on Balance Scale Phenomena, Machine Learning **16**, pp. 59-88, 1994.
75. Z.H. Zhou,Y. Jiang, NeC4.5: neural ensemble based C4.5," in IEEE Transactions on Knowledge and Data Engineering **16**, pp. 770-773, 2004.
76. R. Setiono , W.K. Leow, FERNN: An Algorithm for Fast Extraction of Rules from Neural Networks, Applied Intelligence **12**, pp. 15-25, 2000.
77. G. Demiroz, H.A. Govenir, N. Ilter, Learning Differential Diagnosis of Eryhemato-Squamous Diseases using Voting Feature Intervals, Artificial Intelligence in Medicine. **13**, pp. 147–165, 1998.
78. I. Kononenko, E. Šimec, M. Robnik-Šikonja, Overcoming the Myopia of Inductive Learning Algorithms with RELIEFF, Applied Intelligence **7**, pp. 39–55, 1997
79. B. Hayes-Roth, B., F. Hayes-Roth. Concept learning and the recognition and classification of exemplars. Journal of Verbal Learning and Verbal Behavior **16**, pp. 321-338, 1977.
80. R.M. French, N. Chater, Using noise to compute error surfaces in connectionist networks: a novel means of reducing catastrophic forgetting, Neural Comput. **14**, pp. 1755-1769, 2002.
81. J.G. Dy , C.E. Brodley, Feature Selection for Unsupervised Learning, The Journal of Machine Learning Research **5**, pp 845–889, 2004.
82. S. J. Perantonis, V. Virvilis, Input Feature Extraction for Multilayered Perceptrons Using Supervised Principal Component Analysis, Neural Processing Letters **10**, pp 243–252, 1999.
83. J. Garcke, M. Griebel, Classification with sparse grids using simplicial basis functions, Intell. Data Anal. **6**, pp. 483-502, 2002.
84. M. Elter, R. Schulz-Wendtland, T. Wittenberg, The prediction of breast cancer biopsy outcomes using two CAD approaches that both emphasize an intelligible decision process, Med Phys. **34**, pp. 4164-72, 2007.
85. F. Esposito F., D. Malerba, G. Semeraro, Multistrategy Learning for Document Recognition, Applied Artificial Intelligence **8**, pp. 33-84, 1994.
86. M.A. Little, P.E. McSharry, E.J. Hunter, J. Spielman, L.O. Ramig, Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. IEEE Trans Biomed Eng. **56**, pp. 1015-1022, 2009.
87. J.W. Smith, J.E. Everhart, W.C. Dickson, W.C. Knowler, R.S. Johannes, Using the ADAP learning algorithm to forecast the onset of diabetes mellitus, In: Proceedings of the Symposium on Computer Applications and Medical Care IEEE Computer Society Press, pp.261-265, 1988.
88. D.D. Lucas, R. Klein, J. Tannahill, D. Ivanova, S. Brandon, D. Domyancic, Y. Zhang, Failure analysis of parameter-induced simulation crashes in climate models, Geoscientific Model Development **6**, pp. 1157-1171, 2013.
89. N. Giannakeas, M.G. Tsipouras, A.T. Tzallas, K. Kyriakidi, Z.E. Tsianou, P. Manousou, A. Hall, E.C. Karvounis, V. Tsianos, E. Tsianos, A clustering based method for collagen proportional area extraction in liver biopsy images (2015) Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, 2015-November, art. no. 7319047, pp. 3097-3100.
90. T. Hastie, R. Tibshirani, Non-parametric logistic and proportional odds regression, JRSS-C (Applied Statistics) **36**, pp. 260–276, 1987.
91. M. Dash, H. Liu, P. Scheuermann, K. L. Tan, Fast hierarchical clustering and its validation, Data & Knowledge Engineering **44**, pp 109–138, 2003.
92. W.H. Wolberg, O.L. Mangasarian, Multisurface method of pattern separation for medical diagnosis applied to breast cytology, Proc Natl Acad Sci U S A. **87**, pp. 9193–9196, 1990.
93. M. Raymer, T.E. Doom, L.A. Kuhn, W.F. Punch, Knowledge discovery in medical and biological datasets using a hybrid Bayes classifier/evolutionary algorithm. IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society, **33** , pp. 802-813, 2003.
94. P. Zhong, M. Fukushima, Regularized nonsmooth Newton method for multi-class support vector machines, Optimization Methods and Software **22**, pp. 225-236, 2007.
95. M. Koivisto, K. Sood, Exact Bayesian Structure Discovery in Bayesian Networks, The Journal of Machine Learning Research **5**, pp. 549–573, 2004.
96. R.G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state, Phys. Rev. E **64**, pp. 1-8, 2001.
97. W. J Nash, T.L. Sellers, S.R. Talbot, A.J. Cawthor, W.B. Ford, The Population Biology of Abalone (_Haliotis_ species) in Tasmania. I. Blacklip Abalone (_H. rubra_) from the North Coast and Islands of Bass Strait, Sea Fisheries Division, Technical Report No. 48 (ISSN 1034-3288), 1994.
98. T.F. Brooks, D.S. Pope, A.M. Marcolini, Airfoil self-noise and prediction. Technical report, NASA RP-1218, July 1989.
99. J.S. Simonoff, Smooting Methods in Statistics, Springer - Verlag, 1996.

100. I.Cheng Yeh, Modeling of strength of high performance concrete using artificial neural networks, Cement and Concrete Research. **28**, pp. 1797-1808, 1998.
101. D. Harrison and D.L. Rubinfeld, Hedonic prices and the demand for clean ai, J. Environ. Economics & Management **5**, pp. 81-102, 1978.
102. J.S. Simonoff, Smooting Methods in Statistics, Springer - Verlag, 1996.
103. R.D. King, S. Muggleton, R. Lewis, M.J.E. Sternberg, Proc. Nat. Acad. Sci. USA **89**, pp. 11322–11326, 1992.