

Chapter 4

The Role of Artificial Neural Networks in Evolutionary Optimisation: A Review

M. Maarouf, A. Sosa, B. Galván, D. Greiner, G. Winter, M. Mendez and R. Agasca

Abstract This paper reviews the combination of Artificial Neural Networks (ANN) and Evolutionary Optimisation (EO) to solve challenging problems for the academia and the industry. Both methodologies has been mixed in several ways in the last decade with more or less degree of success, but most of the contributions can be classified into the two following groups: the use of EO techniques for optimizing the learning of ANN (EOANN) and the developing of ANNs to increase the efficiency of EO processes (ANNEO). The number of contributions shows that the combination of both methodologies is nowadays a mature field but some new trends and the advances in computer science permits to affirm that there is still room for noticeable improvements.

Keywords Artificial neural networks · Evolutionary optimisation · Evolutionary algorithm

4.1 Introduction

Artificial Neural Network (ANN) and Evolutionary Algorithm (EA) are relatively young research areas that were subject to a steadily growing interest nowadays; represent two evolving technologies that are inspired by biological information science.

ANN is derived from brain theory to simulate learning behavior of an individual, which is, used for approximation and generalization, while EA is developed from the evolutionary theory raised by Darwin to evolve the whole population for better fitness. Evolutionary Algorithm is actually used as an optimisation algorithm, and not a learning algorithm.

In particular, EO has been used to search for the design and structure of the network and to select the most relevant features of the training data. It is well known that to solve nonlinearly separable problems, the network must have at least one

M. Maarouf (✉) · A. Sosa · B. Galván · D. Greiner · G. Winter · M. Mendez · R. Agasca
División de Computación Evolutiva y Aplicaciones (CEANI), Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (SIANI), Universidad de Las Palmas de Gran Canaria, Islas Canarias, Spain
e-mail: maaroufmusta@gmail.com

hidden layer; but determining the number and the size of the hidden layers is mostly a matter of trial and error. EOs has been used to search for these parameters, to generate the appropriate network to solve specific problems.

On the other hand, ANN has yield many benefits solving a lot of problems in fields as diverse as biology, physics, computer science and engineering. In many applications, the real-time solutions of optimisation problems are widely required. However, traditional algorithms may not be efficient since the computing time required for a solution is greatly dependent on the structure of the problems and there dimension.

A promising approach to solving such problems in real time is to employ artificial neural networks based on circuit implementation [133]. ANNs possess many desirable properties such as real-time information processing. Therefore, neural networks for optimisation, control, and signal processing received tremendous interests. The theory, methodology, and applications of ANNs have been widely investigated.

The present work focuses on Evolutionary Algorithms (EO), Artificial Neural Networks (ANN) and their joint applications as a powerful tool to solve challenging problems for the academia and the industry. In this section the essentials and importance of both methodologies are presented.

4.1.1 Evolutionary Algorithms

Since 1960s, there has been an increasing interest in emulating evolutionary process of living beings to solve hard optimisation problems [34, 101]. Simulating these features of living beings, yields stochastic optimisation procedures called Evolutionary Optimisation (EO). EO belongs to global search meta-heuristics methods since, by its own nature, explores the whole decisional space for global optima.

Evolutionary algorithms (EAs) are a class of stochastic and probabilistic optimisation methods that are inspired by some presumed principles of evolution; attempt to emulate the biological process of evolution, incorporating concepts of selection, reproduction, and mutation. These techniques, inspired in Darwinian evolution postulates, consider a population of individuals on which selection and diversity generation procedures are performed, guaranteeing better fitted individuals to survive through successive iterations [12, 46, 47]. Each individual is a potential solution of the optimisation problem, so it belongs to decision space. Every iteration (generation), individual features are combined by means of recombination operators such selection, crossover and mutation, driving solutions to global optima. By mimicking this process, EAs are able to evolve solutions to real world problems, if they have been suitably encoded.

Evolutionary Optimisation has demonstrated to be effective in engineering and science optimisation problems in several fields such as: Aérospatiale applications [15], energy [9], transport planning, RAMS [87, 96, 97, 111], task scheduling and so on.

The applications mentioned above usually results in high dimensional search spaces, highly non-linear optimisation problems, non-convex optimisation, highly constrained problems, uncertainty effects and/or multicriteria paradigm. These numerical difficulties are commonly tackled by these meta-heuristics, which usually

outperforms traditional optimisation strategies with lower probability of being stacked in local optima, and being able to yield a Pareto set in a single run of the algorithm (multi-objective approach). Because of this, Evolutionary Optimisation has been an important R&D matter in the last decade.

Among the variety of EO meta-heuristics, the most relevant nowadays are: Genetic Algorithm (GA), Evolutionary Strategy (ES), Evolutionary Programming (EP), Collective Intelligence, Memetic Algorithms, and Differential Evolution.

In recent years, there has been an increase in the use of evolutionary approaches in the training and optimisation of artificial neural networks(ANNs). Different works are presented in Sect. 4.2.1.

4.1.2 Artificial Neural Networks ANN

Artificial neural networks (ANNs) are biologically inspired computer programs, inspired from the morphological and biophysical properties of neurons in the brain. ANNs are designed to simulate the way the human brain processes information.

Neural networks are similar to the human brain in the following two ways:

1. A neural network acquires knowledge by learning.
2. The knowledge of a neural network is stored in the connections between neurons known as synaptic weights.

[98] were the first patterned biological neurons from the binary automata. A second generation of neurons integrates a nonlinear activation function that has allowed growing up the interest in ANNs [69], allowing to solve nonlinear problems.

The power and utility of artificial neural networks have been shown in several applications including speech synthesis, diagnostic problems, medicine, business and finance, control, robotics, signal processing, computer vision and many other industrial problems that are included in the category of pattern recognition. But knowing that there exists a suitable network for a specific problem is interesting, finding it proved to be difficult. Although there exist some algorithms to set the weights by learning from data training given a fixed topology, even if get stuck in local minima. To lead to good results, they strongly depend on problem specific parameter settings and on the topology of the network.

Training procedures of neural networks are optimisation algorithms aim to minimize the global error output respect to connection weights under conditions of noise, qualitative uncertainty, nonlinearity, etc. Therefore training procedures of neural networks provide a common approach to the optimisation in process optimisation and control applications [149]. However, this is an important issue because there are strong biological and engineering evidences to support that an ANN is determined by its architecture. Using EOs as a procedure to assist neural network design and training seems to be a straightforward idea. Branke in [23] explain how EO improve design and training ANNs.

Today, significant progress has been made in the field of neural networks, enough to attract a lot of attention and research funding. Research in the field is advancing on many fronts. New neural concepts are emerging and applications to complex problems developing. Clearly, today is a period of transition for neural network technology.

ANNs are inherently parallel architectures which can be implemented in software and hardware. One important implementation issue is the size of the neural network and its weight adaptation. This makes the hardware implementation complex and software learning slower.

ANNs has two distinct steps [172];

1. Choosing proper network architecture.
2. Adjusting the parameters of a network so as to minimize certain fit criterion.

Even if, the most of problems treated in real study are complex, so the unknown architecture of the ANN is set arbitrarily or by trial and error [121], and small networks cannot achieve the solution in much iteration, but if the network is too large, it leads to overfitting and a bad generalization, and the majority of neural networks suffers of premature convergence and low global convergence speed etc. In order to overcome these limitations, some improvements were made for EOs in the last decade on ANNs.

4.2 Different Use of ANNEO and EOANN

In the last decade, there has been a great interest combining learning and evolutionary techniques in computing science to solve complex problems for different fields. Different works presented how ANNs are an optimisation tools [33, 120, 131, 136, 173, 175]. In this paper we just limit this review to several ways in which EOs and ANNs may be combined.

The next section presents different approaches of combining EOs and ANNs.

4.2.1 The Use of EOs in ANNs: EOANN

In recent years, evolutionary algorithms (EAs) have been applied to the ANN's optimisation. The first applications of EAs to ANN parameter learning date back to the late 80s in the fields of Genetic Algorithms (GAs) [45, 47] and Evolutionary Programming (EP) [47, 103]. EAs are powerful search algorithms based on the mechanism of natural selection. Since the 90s, EAs have been successfully used for optimizing the design and the parameters of ANNs [132, 150]. This special class of ANNs in which evolution is another fundamental form of adaptation in addition to learning creating an EOANN [40, 167]. EOANN are used to find the best data subset which optimizes the ANN training for a specific problem.

The result of EOANN is an ANN with optimal performance to estimate the value of one or more variables and the estimation error is strongly related to the quality of the

training set in terms of size and treatment of the possible outputs. Another approach is the evolutionary weight training process, to optimizing connection weights, learning rules, and optimizing the network architecture by identifying how many inputs neurons, hidden neurons, hidden layers we have to use, to get a good performance [18, 39, 88, 92, 128]. Considerable research has been conducted on the evolution of topological structures of networks using evolutionary algorithms [3, 6, 13, 14, 20, 73, 84].

In the other hand, an essential issue is to improve generalization of the neural network training models. Early stopping, weight decay and curvature-driven smoothing and other techniques are been used to resolve this problem, another approach is including an additional term in the cost function of learning algorithms, which penalizes overly high model complexity. Regularization of neural training was treated with Eos in [1, 75].

In general, constructing neural network consists of two major steps, design and training component networks, combining of the component networks predictions to produce the neural networks solutions. ANN training method has some limitations associated with overfitting, local optimum problems and slow convergence rate. In order to overcome the limitations, some scientist proposed particle evolutionary algorithm to train ANN.

The research use EOs to evolve and design the structure architecture or the selection of the training algorithms and optimisation of its synaptic weight initialization, thresholds, training ratio, momentum factor, etc., of neural network roundly. The scientist object is to accelerate the convergence speed of network and optimize the result in case of trapping into local optimal value, and a better searching space is found out in the solution space.

The initial set of weights to be used in learning of ANN has a strong influence in the learning speed and in the quality of the solution obtained after training. An inadequate initial choice of the weight values may cause the training process to get stuck in a poor local minimum or more time to converge. Inappropriate topology selection and learning algorithm are frequently bad; there is little reason to expect that one can find a uniformly best algorithm for selecting the weights in an ANN [94]. This kind of topology was chosen for the following main reasons:

1. Additional links that skip adjacent layers allow the genetic algorithm to remove a whole layer of nodes while still keeping the network functional.
2. Some additional shortcut connections, e.g., those between the network input and output, may ease training and therefore the whole genetic topology optimisation may become faster.

There are different ways to evaluate weights of the component networks. For example Jimenez [74] use weights determined by confidence of the component networks. Zhou [184] utilize the genetic algorithm to find proper weights for each member of an ensemble. In [143], present and define many operators and crossover applied to weights of an ANN. The importance of a good choice for the initial set of weights is stressed by Kolen and Pollak [83].

ANNs strongly depend on the network topology, the neurons activation function, the learning rule, etc. optimisation for these factors are usually unknown a priori

because they depend mainly on the particular training set to be considered and on the nature of the solution [137]. So, for practical purposes, the learning rule should be based on optimisation techniques that employ local search to find the optimal solution [124, 126].

Evolutionary approaches have been shown to be very effective as an optimisation technique, their efficiency could be exploited in training and constructing neural networks, their architecture/design and learning, they can evolve towards the optimal architecture without outside interference, thus eliminating the tedious trial and error work of manually finding an optimal network, adapting the connection weights and learning algorithms according to the problem environment. Many considerable efforts at obtaining optimal ANNs based on EAs have been reported in the literature [4–7, 10, 13, 24–28, 31, 52, 53, 58, 63–67, 82, 99, 117, 122, 132, 139, 144, 166, 168–171].

The EAs are using diverse methods to encode ANNs for the purpose of training and design. The common approach is to encode the ANN weights into genes that are then concatenated to build the genotype. Encoding methods can be divided into three main groups according to the process of creating the network from the encoded genome: direct, parametric and indirect encoding. A real coded genetic algorithm is used to optimize the mean square of the error produced by training a neural network established by Aljahdali in [11]. Benaddy et al. [17] present a real coded genetic algorithm that uses the appropriate operators type to train feed-forward neural network. Larranaga et al. [86] describes various methods used to encode artificial neural networks to chromosomes to be used in evolutionary computation.

Another important point to note is the use of EOs to extract rules from neural networks trained. Rule extraction from neural networks is attracting wide attention because of its computational simplicity and ability to generalize [49–51, 141].

4.2.2 The Use of ANNs in EO: ANNEO

In real world applications, sometimes it is not easy to obtain the objective value. Therefore we need complicated analysis or time consuming simulation to evaluate the performance of design variables [147]. As ANNs represent a nonlinear robust modeling technique which are developed, or trained, based on analytical or simulated results of a subset of possible solutions [35], give to ANNs an important role in solving problems with extremely difficult or unknown analytical solution. ANNs can be used with a huge reduction of cost in terms of objective function evaluations [22]. One of the first pioneers in ANNEO were Hopfield and Tank who presented an ANN for solving combinatorial problems, that was mapped into a closed-loop circuit [68, 133], named Hopfield Neural Network (HNN). HNN is a continuously operating model very close to analog circuit implementation. Since 1985 a wide variety of ANNs have been designed for improving the performance of HNN.

In ANNEO fast objective functions evaluations are performed using pre-trained ANN. Normally in iterative process subsets of objective function values obtained

using exact procedures are used in an embedded EOANN algorithm, and some of the new objective function evaluations are performed using the ANN. The result is an EO which evolves faster than conventional ones but special care must be paid to the selection of appropriate training subsets and the number of objective functions evaluated using ANN in order to avoid convergence problems. Acceleration of the convergence speed is done in [176] as an ANN model trained to approximate the fitness function according to an adaptive scheme for increasing the number of network fitness calculation.

EOs usually needs a large number of fitness evaluations before a satisfying result can be obtained. And as an explicit fitness function does not exist, or his evaluation is computationally very expensive. It is necessary to estimate the fitness function by constructing an approximate model or presenting an interpolation of the true fitness function via some interpolation technique as ANNs [22, 55] employ a Feedforward neural networks for fitness estimation, reduce the number of expensive fitness function evaluations in evolutionary optimisation. The idea of the implementation of an ANN, that approximates the fitness function, comes from the universal approximation capability of multi-layer neural networks [69]. An artificial neural networks model is used in order to reduce the number of time-consuming fitness evaluations [55].

Real-time solutions to resolve problems are often needed in engineering applications. Solve many problem of optimisation in real time usually contain time-varying parameters, such in signal processing, robotic, time series, etc., and we have to reduce and optimize the performance. The numbers of decision variables and constraints are usually very large and large-scale optimisation problems are even more challenging when they have to be solved in real time to optimize the performance of dynamical systems. For such applications, Conventional numerical methods may not be effective at all due to the problem dimensionality and stringent requirement on computational time [16]. The employment of ANNs techniques as Recurrent Neural Networks (RNN) [21, 33, 48, 79, 89, 91, 113, 115, 127, 133, 145, 146, 160, 165, 177] (Papers below proposed neural networks guaranteed to be globally convergent in finite time to the optimal solutions), Fuzzy Neural Network (FNN) are a promising approach to resolve this inefficiency [123].

Application of ANNs algorithms receive increase interests for optimisation as we see in [21, 32, 33, 37, 79, 93, 105–107, 118, 133, 134, 151, 154, 156, 157, 182], using gradient and projection methods [91, 104], Bouzerdoum and Pattison [21] presented a neural network for solving quadratic optimisation problems with bounded variables only, which constitutes a generalization of the network described by Sudharsanan [127]. Rodrguez-Vzquez et al. [118] proposed a class of neural networks for solving optimisation problems, in which their design does not require the calculation of a penalty parameter. To avoid using finite penalty parameters, many other studies have been done in [21, 56, 70, 78, 118, 134, 148, 153, 156, 157, 159, 182, 183]. In [151–154], the authors presented several neural networks for solving linear and quadric programming problems with no unique solutions, which are proved to be globally convergent to exact solutions, and in which there is no variable parameter to tune.

Romero [119] approached optimisation problems with a multilayer neural network. Da Silva [36] coupled fuzzy logic with Hopfield Neural Networks to solve linear and non-linear optimisation problems. Case studies with convex and non-convex optimisation problems are studied in illustrate the approach with a trained Multilayer neural networks [143]. Xia et al. [155, 158, 160] proposed a general projection neural network, that includes projection neural network, the primal-dual neural network, and/or the dual neural network, as special cases, for solving a wider class of variational inequalities and related optimisation problems.

A reliability network reflect a network optimized whose topology is optimist, at minimal cost, under the constraint that every pair of neurons can communicate with each other. Abo El ftooh et al. [2] presents an ANN for solving this problem o reliability, by constructing an energy function whose minimization process drives the neural network into one of its stable states.

An ANN aided with other algorithms as simulated-annealing (SA) algorithm, can be usefully used to resolve optimisation problems as they do in [85]. Chaotic artificial neural are studied and established as optimisation model in [8, 85]. Another point to be mentioned is the use of ANNs to re-optimisation online as it presented in [138].

4.3 Some Applications Using ANNEO and EOANN

Many academic papers show the applicability of EOANN to optimize different parameters of ANN, to improve their training and their stability [13, 20, 43, 71–73, 76, 84] and the papers cited below in Sect. 4.2.2. Other EOANN applications were performed in several fields such as Financial Engineering [41, 59], grammatical inference [19], Chemical Reaction [178], Hydrochemistry [142], Time series prediction [77, 81, 180], Classification Process [29, 30, 44, 80, 90, 95, 109, 112, 135], Medicine [54], Diagnosis problems [17, 72, 129, 130], Diverse Engineering Applications [57, 100, 125, 135, 174], Robotic [161], Monitoring [38], Traffic Prediction [114], Control Systems [181], Neutron spectrometry and Dosimetry research areas [108], Multi-agent systems [81], Regression problems [61], Chaos dynamics Problem [179, 180], Reliability [62], etc.

Also, the role of ANNs in EO is presented in many academic papers as in [22, 147, 176], and in many fields like Reliability Systems [35], Electromagnetics optimisation [22], pressure vessel design problem [147], aerodynamic design optimisation [73], Electric-Field optimisation [85, 116, 164], Cybernetics [162], Design optimisation [102, 163], Diagnosis [60], Power Technology [110, 140], Resource Management optimisation [42], etc.

4.4 Conclusions

We can deduce that if the purpose of using ANNs is to find the best network configuration for solving particular problems, it has been possible employing EOs, as it has been mentioned above in several works. EOs provides good approximation to get success and speed of training of neural network based on the initial parameter settings, such as architecture, initial weights, learning rates, and others.

In the other hand, ANNs based objective function allows the fitness to be evaluated in a small fraction of the time taken to perform first principles analysis and permits the EAs to complete in a reasonably small amount of time.

Due to the last advances in both methodologies, there are several chances for future develops of joint procedures especially when complex industrial applications are addressed. Anyway the use of big computing facilities will continue being still necessary for that applications.

References

1. Abbass HA (2003) Speeding up back-propagation using multiobjective evolutionary algorithms. *Neural Comput* 5:2705–2726
2. Abo El Fotoh HMF, Al-Sumait LS (2001) A neural approach to topological optimization of communication networks, with reliability constraints. *IEEE Trans Reliab* 50(4):397–408
3. Abraham A, Nath B (2001) ALEC—an adaptive learning framework for optimizing artificial neural networks. In: Alexandrov VN et al (eds) *Computational science*, Springer, Germany, San Francisco, 171–180
4. Abraham A (2002) Optimization of evolutionary neural networks using hybrid learning algorithms. *Int Symp Neural Netw* 3:2797–2802. doi:[10.1109/IJCNN.2002.1007591](https://doi.org/10.1109/IJCNN.2002.1007591)
5. Abraham A (2004) Meta learning evolutionary artificial neural networks. *Neurocomputing* 56:138
6. Aguilar J, Colmenares A (1997) Recognition algorithm using evolutionary learning on the random neural networks. *International symposium on neural networks*. doi:[10.1109/ICNN.1997.616168](https://doi.org/10.1109/ICNN.1997.616168)
7. Ajith A (2004) Meta-learning evolutionary artificial neural networks. *Neurocomputing* 56:1–38. doi:[10.1016/S0925-2312\(03\)00369-2](https://doi.org/10.1016/S0925-2312(03)00369-2)
8. Akkar HAR (2010) Optimization of artificial neural networks by using swarm intelligent
9. Alarcon-Rodriguez A, Ault G, Galloway S (2010) Multi-objective planning of distributed energy resources: a review of the state-of-the-art. *Renew Sustain Energy Rev* 14(5):1353–1366
10. Alexandridis A (2012) An evolutionary-based approach in RBF neural network training. *IEEE workshop on evolving and adaptive intelligent systems*. doi:[10.1109/EAIS.2012.6232817](https://doi.org/10.1109/EAIS.2012.6232817)
11. Aljhadali S, Buragga KA (2007) Evolutionary neural network prediction for software reliability modeling. *The 16th international conference on software engineering and data engineering*
12. Alonso S (2006) *Propuesta de un Algoritmo Flexible de Optimización Global*. Ph.D. thesis, Departamento de Informática y Sistemas. Universidad de Las Palmas de Gran Canaria
13. Angeline P, Saunders G, Pollack J (1994) An evolutionary algorithm that constructs recurrent neural networks. *IEEE Trans Neural Netw* 5(1):54–65
14. Arena P, Caponetto R, Fortuna L, Xibilia MG (1992) Genetic algorithms to select optimal neural network topology. In: *Proceedings of the 35th midwest symposium on circuits and systems*, vol 2, pp 1381–1383. doi:[10.1109/MWSCAS.1992.271082](https://doi.org/10.1109/MWSCAS.1992.271082)

15. Arias-Montano A, Coello CA, Mezura-Montes E (2012) Multiobjective evolutionary algorithms in aeronautical and aerospace engineering. *IEEE Trans Evol Comput* 16(5):662–694
16. Bazaraa MS, Sherali HD, Shetty CM (1993) *Nonlinear programming: theory and algorithms*, 2nd edn. Wiley, New York
17. Benaddy M, Wakrim M, Aljahdali S (2009) Evolutionary neural network prediction for cumulative failure modeling. *ACS/IEEE international conference on computer systems and applications*. doi:[10.1109/AICCSA.2009.5069322](https://doi.org/10.1109/AICCSA.2009.5069322)
18. Bevilacqua V, Mastronardi G, Menolascina F, Pannarale P, Pedone A (2006) A novel multi-objective genetic algorithm approach to artificial neural network topology optimisation: the breast cancer classification problem. *IJCNN '06. International joint conference on neural networks*, pp 1958–1965
19. Blanco A, Delgado M, Pegalajar M (2000) A genetic algorithm to obtain the optimal recurrent neural network. *Int J Approx Reason* 23(1):67–83
20. Bornholdt S, Graudenz D (1993) General asymmetric neural networks and structure design by genetic algorithms: a learning rule for temporal patterns. In: *International conference on systems, man and cybernetics. Systems engineering in the service of humans, conference proceedings, vol 2*, pp 595–600
21. Bouzerdoum A, Pattison TR (1993) Neural network for quadratic optimization with bound constraints. *IEEE Trans Neural Netw* 4:293–303
22. Bramanti A, Di Barba MF, Savini A (2001) Combining response surfaces and evolutionary strategies for multiobjective pareto-optimization in electromagnetics. *Stud Appl Electromag Mech JSAEM* 9:231–236
23. Branke J (1995) Evolutionary algorithms for neural network design and training. In: *Proceedings of the first nordic workshop on genetic algorithms and its applications*, pp 145–163
24. Branke J, Kohlmorgen U, Schmeck H (1995) A distributed genetic algorithm improving the generalization behavior of neural networks. In: Lavrac N et al *Proceedings of the European conference on machine learning*, pp 107–112
25. Braun H, Weisbrod J (1993) Evolving neural networks for application oriented problems. In: Fogel DB (ed) *Proceedings of the second conference on evolutionary programming*
26. Braun H (1995) On optimizing large neural networks (multilayer perceptrons) by learning and evolution. In: *Proceedings of the third international congress on industrial and applied mathematics, ICIAM*
27. Bukhtoyarov VV, Semenkin OE (2010) Comprehensive evolutionary approach for neural network ensemble automatic design. *IEEE congress on evolutionary computation* p 1–6. doi:[10.1109/CEC.2010.5586516](https://doi.org/10.1109/CEC.2010.5586516)
28. Bundzel M, Sincak P (2006) Combining gradient and evolutionary approaches to the artificial neural networks training according to principles of support vector machines. In: *International symposium on neural networks*, pp 2068–2074. doi:[10.1109/IJCNN.2006.246976](https://doi.org/10.1109/IJCNN.2006.246976)
29. Cantu-Paz E, Kamath C (2002) Evolving neural networks for the classification of galaxies. *Genetic and evolutionary computation conference*, pp 1019–1026
30. Castellani M (2006) ANNE—a new algorithm for evolution of artificial neural network classifier systems. doi:[10.1109/CEC.2006.1688728](https://doi.org/10.1109/CEC.2006.1688728)
31. Chi-Keong G, Eu-Jin T, Kay CT (2008) Hybrid multiobjective evolutionary design for artificial neural networks. *IEEE Trans Neural Netw* 19(9):1531–1548. doi:[10.1109/TNN.2008.2000444](https://doi.org/10.1109/TNN.2008.2000444)
32. Cichocki A, Unbehauen R (1991) Switched-capacitor artificial neural networks for differential optimization. *J Circuit Theory Appl* 19:61187
33. Cichocki A, Unbehauen R (1993) *Neural networks for optimization and signal processing*. Wiley, New York
34. Coello CAC, Lamont GB, Veldhuizen DAV (2006) *Evolutionary algorithms for solving multi-objective problems (genetic and evolutionary computation)*. Springer, New York
35. Coit DW, Smith AE (1996) Solving the redundancy allocation problem using a combined neural network/genetic algorithm approach. *Comput Oper Res* 23(6):515–526

36. Da Silva IN (1997) A neuro-fuzzy approach to systems optimization and robust estimation. PhD thesis, School of electrical engineering and computer science, State University of Campinas, Brazil
37. Dempsey GL, McVey SE (1993) Circuit implementation of a peak detector neural network. *IEEE Trans Circ Syst II* 40:585–591
38. Desai CK, Shaikh AA, (2006) Drill wear monitoring using artificial neural network with differential evolution learning. In: *IEEE International conference on industrial technology ICIT 2006*, pp 2019–2022. doi:[10.1109/ICIT.2006.372500](https://doi.org/10.1109/ICIT.2006.372500)
39. Di Muro G, Ferrari S (2008) A constrained-optimization approach to training neural networks for smooth function approximation and system identification. *IJCNN 2008*. (IEEE world congress on computational intelligence). *IEEE international joint conference on neural networks*, vol 2353–2359
40. Ding S, Li H, Su C, Yu J, Jin F (2013) Evolutionary artificial neural networks: a review. *Artif Intell Rev* 39(3):251–260
41. Edwards D, Brown K, Taylor N (2002) An evolutionary method for the design of generic neural networks. In: *Proceedings of the 2002 congress on evolutionary computation. CEC '02*. vol 2, pp 1769–1774
42. Efstratios FG, Sotiris MG (2007) Solving resource management optimization problems in contact centers with artificial neural networks. *Int Conf Tools Artif Intell* 2:405–412
43. Farzad F, Hemati S (2003) An algorithm based on evolutionary programming for training artificial neural networks with nonconventional neurons. In: *Canadian conference on electrical and computer engineering*. vol 3. doi:[10.1109/CCECE.2003.1226270](https://doi.org/10.1109/CCECE.2003.1226270)
44. Fiszlelew A, Britos P, Ochoa A, Merlino H, Fernandez E, Garcia-Martinez R (2007) Finding optimal neural network architecture using genetic algorithms. *Advances in computer science and engineering research in computing science* 27:15–24
45. Fogel DB, Fogel LJ, Porto VW (1990) Evolutionary programming for training neural networks. In: *Proceedings of the international joint conference on NNs*, San Diego, CA, pp 601–605
46. Fogel D (1997) The advantages of evolutionary computation, bio-computing and emergent computation. In: *Lundh D, Olsson B, and Narayanan A (eds) Skve*, Sweden, World scientific press, Singapore, pp 1–11
47. Fogel D (1999) *Evolutionary computation: towards a new philosophy of machine intelligence*. 2nd edn, IEEE press
48. Forti M, Nistri P, Quincampoix M (2004) Generalized neural network for nonsmooth nonlinear programming problems. *IEEE Trans Circ Syst I* 51(9):1741–1754
49. Fukumi M, Akamatsu N (1996) A method to design a neural pattern recognition system by using a genetic algorithm with partial fitness and a deterministic mutation. In: *Proceedings of IEEE international conference on SMC*, vol 1 3, pp 1989–1993
50. Fukumi M, Akamatsu N (1998) Rule extraction form neural networks trained using evolutionary algorithms with deterministic mutation. In: *International symposium on neural networks*. vol 1. doi:[10.1109/IJCNN.1998.682363](https://doi.org/10.1109/IJCNN.1998.682363)
51. Fukumi M, Akamatsu N (1999) An evolutionary approach to rule generation from neural networks. *IEEE Int Fuzzy Syst Conf Proc* 3:1388–1393
52. Funabiki N, Kitamichi J, Nishikawa S (1998) An evolutionary neural network approach for module orientation problems. *IEEE Trans Syst Man Cybern—Part B Cybern* 28(6):849–855
53. Garcia-Pedrajas N, Hervás-Martínez C, Muñoz-Pérez J (2003) COVNET: a cooperative coevolutionary model for evolving artificial neural networks. *IEEE Trans Neural Netw* 14(3):575–596
54. Gorunescu F, Gorunescu M, Gorunescu S (2005) An evolutionary computational approach to probabilistic neural network with application to hepatic cancer diagnosis. In: *Proceedings of the 18th IEEE symposium on computer-based medical systems (CBMS05)*
55. Grning L, Jin Y, Sendhoff B (2005) Efficient evolutionary optimization using individual-based evolution control and neural networks: a comparative study. *The European symposium on artificial neural networks*, pp 273–278

56. Guo-Cheng L, Zhi-Ling D (2008) Sub-gradient based projection neural networks for non-differentiable optimization problems. In: Proceedings of the seventh international conference on machine learning and cybernetics, Kunming
57. Gwo-Ching L (2012) Application a novel evolutionary computation algorithm for load forecasting of air conditioning. In: Power and energy engineering conference (APPEEC), 2012 Asia-pacific, pp 1–4. doi:[10.1109/APPEEC.2012.6307573](https://doi.org/10.1109/APPEEC.2012.6307573)
58. Haykin S (1999) Neural network, 2nd edn., A comprehensive foundation Prentice Hall, New Jersey
59. Hayward S (2004) Evolutionary artificial neural network optimisation in financial engineering. In: Fourth international conference on hybrid intelligent systems. HIS '04, pp 210–215
60. He Y, Sun Y (2001) Neural network-based L1-norm optimisation approach for fault diagnosis of nonlinear circuits with tolerance. In: Lee proceedings-circuits devices and systems, pp 223–228. doi:[10.1049/ip-cds:20010418](https://doi.org/10.1049/ip-cds:20010418)
61. Hieu TH, Yonggwang W (2008) Evolutionary algorithm for training compact single hidden layer feedforward neural networks. In: International symposium on neural networks, pp 3028–3033. doi:[10.1109/IJCNN.2008.4634225](https://doi.org/10.1109/IJCNN.2008.4634225)
62. Hochman R, Khoshgoftaar TM, Allen EB, Hudepohl JP (1997) Evolutionary neural networks: a robust approach to software reliability problems. In: International symposium on software reliability engineering. doi:[10.1109/ISSRE.1997.630844](https://doi.org/10.1109/ISSRE.1997.630844)
63. Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor
64. Holland JH (1980) Adaptive algorithms for discovering and using general patterns in growing knowledge-based. Intl J Policy Anal Inf Syst 4(3):245–268
65. Holland JH (1986) Escaping brittleness: the possibilities of general purpose learning algorithms applied in parallel rule-based systems. In: Michalski RS, Carbonell JG, Mitchell TM (eds) Machine learning II, pp 593–623
66. Holland JH, Holyoak KJ, Nisbett RE, Thagard PR (1987) Classifier systems, Q-morphisms, and induction. In: Davis L (ed) Genetic algorithms and simulated annealing, pp 116–128
67. Honavar V, Uhr L (1993) Generative learning structures and processes for generalized connectionist networks. Inf Sci 70:75–108
68. Hopfield JJ, Tank DW (1985) Neural Y computation on decisions optimization problem. Biol Cybern 52:141–152
69. Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Netw 2(5):359–366
70. Hu X, Jun W (2006) Solving pseudomonotone variational inequalities and pseudoconvex optimization problems using the projection neural network. IEEE Trans Neural Netw 17(6):1487–1499
71. Huning H (2010) Convergence analysis of a segmentation algorithm for the evolutionary training of neural networks. In: IEEE symposium on combinations of evolutionary computation and neural networks. doi:[10.1109/ECNN.2000.886222](https://doi.org/10.1109/ECNN.2000.886222)
72. Husken M, Gayko JE, Sendhoff B (2000) Optimization for problem classes—neural networks that learn to learn 98–109
73. Husken M, Jin Y, Sendhoff B (2002) Structure optimization of neural networks for evolutionary design optimization. In: Proceedings of the 2002 GECCO workshop on approximation and learning in evolutionary computation, pp 13–16
74. Jimenez D (1998) Dynamically weighted ensemble neural networks for classification. In: Proceedings of the IJCNN-98, vol 1, Anchorage, AK, IEEE Computer Society Press, Los Alamitos, pp 753–756
75. Jin Y, Okabe T, Sendhoff B (2004) Neural network regularization and ensembling using multi-objective evolutionary algorithms. Congr Evol Comput CEC2004 1:1–8
76. Jinn-Moon Y, Jong-Tzong H, Cheng-Yen K (1999) Incorporation family competition into gaussian and cauchy mutations to training neural networks using an evolutionary algorithm. IEEE Congress on evolutionary computation. doi:[10.1109/CEC.1999.785519](https://doi.org/10.1109/CEC.1999.785519)

77. Katagiri H, Nishizaki I, Hayashida T, Kadoma T (2010) Multiobjective evolutionary optimization of training and topology of recurrent neural networks for time-series prediction. In: Proceedings of the international conference on information science and applications (ICISA), p 1–8
78. Kazuyuki M (2001) A new algorithm to design compact two-hidden-layer artificial neural networks. *Neural Netw* 14(9):1265–1278. doi:[10.1016/S0893-6080\(01\)00075-2](https://doi.org/10.1016/S0893-6080(01)00075-2)
79. Kennedy MP, Chua LO (1988) Neural networks for nonlinear programming. *IEEE Trans Circ Syst* 35(5):554562
80. Khorani V, Forouzideh N, Nasrabadi AM (2010) Artificial neural network weights optimization using ICA. Comparing performances. In: Proceedings of the IEEE workshop on hybrid intelligent models and applications, GA, ICA-GA and R-ICA-GA. doi:[10.1109/HIMA.2011.5953956](https://doi.org/10.1109/HIMA.2011.5953956)
81. Kisiel-Dorohinicki M, Klapper-Rybicka M (2000) Evolution of neural networks in a multi-agent world
82. Kok JN, Marchiori E, Marchiori M, Rossi C (1996) Evolutionary training of CLP—constrained neural networks. In: Proceedings of the 2nd international conference on practical application of constraint technology, pp 129–142
83. Kolen JF, Pollack JB (1990) Back propagation is sensitive to initial conditions. Technical report TR 90-JK-BPSIC
84. Koza JR, Rice JP (1991) Genetic generation of both the weights and architecture for a neural network. In: Proceedings of the international joint conference on neural networks., IJCNN-91-Seattle, vol 2, pp 397–404
85. Lahiri A, Chakravorti S (2005) A novel approach based on simulated annealing coupled to artificial neural network for 3-D electric-field optimization. *IEEE Trans Pow Deliv* 20(3):2144–2152
86. Larranaga P, Karshenas A, Bielza C, Santana R (2013) A review on evolutionary algorithms in bayesian network learning and inference tasks. *Inform Sci* (2013), <http://dx.doi.org/10.1016/j.ins.2012.12.051>
87. Levitin G (2007) Computational intelligence in reliability engineering: evolutionary techniques in reliability analysis and optimization. Volume 39 of studies in computational intelligence. Springer
88. Lin T, Ping HC, Hsu TH, Wang LC, Chen C, Chen CF, Wu CS, Liu TC, Lin CL, Lin YR, Chang FC (2011) A systematic approach to the optimization of artificial neural networks. In: IEEE 3rd International conference on communication software and networks (ICCSN), pp 76–79
89. Liu Q, Wang J (2008) A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming. *IEEE Trans Neural Netw* 19(4):558–570
90. Liyanage SR, Xu J-X, Guan C, Ang KK, Zhang CS, Lee TH (2009) Classification of self-paced finger movements with EEG signals using neural network and evolutionary approaches. In: Proceedings of the international conference on control and automation. doi:[10.1109/ICCA.2009.5410152](https://doi.org/10.1109/ICCA.2009.5410152)
91. Long C, Zeng-Guang H, Yingzi L, Min T, Wenjun CZ, Fang-Xiang W (2011) Recurrent neural network for non-smooth convex optimization problems with application to the identification of genetic regulatory networks. *IEEE Trans Neural Netw* 22(5):714–726
92. Lu M, Shimizu K (1995) An epsi-approximation approach for global optimization with an application to neural networks. In: Proceedings of the IEEE international conference on neural networks, vol 2, pp 783–788
93. Maa CY, Shanblatt MA (1992) Linear and quadratic programming neural network analysis. *IEEE Trans Neural Netw* 3:580594
94. Macready WG, Wolpert DH (1997) The no free lunch theorems. *IEEE Trans Evol Comput* 1(1):67–82
95. Malinak P, Paksa R (2007) Simultaneous gradient and evolutionary neural network weights adaptation methods. IEEE congress on evolutionary computation, pp 2665–2671. doi:[10.1109/CEC.2007.4424807](https://doi.org/10.1109/CEC.2007.4424807)

96. Marseguerra M, Zio E, Martorell S (2006) Basics of genetic algorithms optimization for RAMS applications. *Reliab Eng Syst Saf* 91(9):977–991
97. Martorell S, Sánchez A, Carlos S, Serradell V (2004) Alternatives and challenges in optimizing industrial safety using genetic algorithms. *Reliab Eng Syst Saf* 86(1):25–38
98. McCulloch WS, and Pitts WH (1943) A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 5:115–133
99. Miller GF, Todd PM, Hedge SU (1989) Designing neural networks using genetic algorithms. In: Schaffer JD (ed) *Proceedings of the third international conference on genetic algorithms*, pp 379–384
100. Misgana KM, John WN (2004) Joint application of artificial neural networks and evolutionary algorithms to watershed management. *Water Resour Manage* 18(5):459–482. doi:[10.1023/B:WARM.0000049140.64059.d1](https://doi.org/10.1023/B:WARM.0000049140.64059.d1)
101. Mitsuo-Gen RC (1997) *Genetic algorithms and engineering design*. Wiley, New York
102. Mohankumar N, Bhuvan B, Nirmala Devi M, Arumugam S (2008) A modified genetic algorithm for evolution of neural network in designing an evolutionary neuro-hardware. In: *Proceedings of the international conference on genetic and evolutionary methods*, pp 108–111
103. Montana D, Davis L (1989) Training feedforward neural networks using genetic algorithms. In: *Proceedings of the 11th international joint conference on AI, Detroit*, pp 762–767
104. More JJ, Toraldo G (1991) On the solution of large quadratic programming problems with bound constraints. *SIAM J Optim* 1(1):93–113
105. Niknam A, Hoseini P, Mashoufi B, Khoei A (2013) A novel evolutionary algorithm for block-based neural network training. In: *Proceedings of the first iranian conference on pattern recognition and image analysis (PRIA)*, pp 1–6
106. Nobuo F, Junji K, Seishi N (1998) An evolutionary neural network approach for module orientation problems. *IEEE Trans Syst Man Cybern* 28(6):849–855. doi:[10.1109/3477.735394](https://doi.org/10.1109/3477.735394)
107. Nunes De Castro L, Iyoda EM, Von Zuben FF, Gudwin RR (1998) Feedforward neural network initialization: an evolutionary approach. *Brazilian symposium on neural networks*, pp 43–48. doi:[10.1109/SBRN.1998.730992](https://doi.org/10.1109/SBRN.1998.730992)
108. Ortiz-Rodriguez JM, del Rosario BM, Gallego E, Vega-Carrillo HR (2008) Artificial neural networks modeling evolved genetically, a new approach applied in neutron spectrometry and dosimetry research areas. In: *Proceedings of the electronics, robotics and automotive mechanics conference*
109. Pal S, Vipsita S, Patra PK (2010) Evolutionary approach for approximation of artificial neural network. In: *Proceedings of the IEEE international advance computing conference*. doi:[10.1109/IADCC.2010.5423015](https://doi.org/10.1109/IADCC.2010.5423015)
110. Pantic D, Trajkovic T, Milenkovic S, Stojadinovic N (1995) Optimization of power VDMOS-FET's process parameters by neural networks. In: *Proceedings of the 25th European solid state device research conference, ESSDERC '95*. 793, 796, pp 25–27
111. Pattison RJA (1999) Genetic algorithms in optimal safety system design. *Proc Inst Mech Eng Part E: J Process Mech Eng* 213:187–197
112. Pavlidis NG, Tasoulis DK, Plagianakos VP, Nikiforidis G, Vrahatis MN (2004) Spiking neural network training using evolutionary algorithms. In: *Proceedings of the international joint conference on neural networks*, pp 2190–2194
113. Prez-Ilzarb MJ (1998) Convergence analysis of a discrete-time recurrent neural network to perform quadratic real optimization with bound constraints. *IEEE Trans Neural Netw* 9:1344–1351
114. Qiang Z, Pan-chi L (2012) Training and application of process neural network based on quantum-behaved evolutionary algorithm. In: *Proceedings of the 2nd international conference on computer science and network technology (ICCSNT)*, pp 929–934
115. Qingshan L, Jun W (2011) Finite-time convergent recurrent neural network with a hard-limiting activation function for constrained optimization with piecewise-linear objective functions. *IEEE Trans Neural Netw* 22(4):601–614
116. Rayas-Sánchez JE (2004) EM-based optimization of microwave circuits using artificial neural networks: the state-of-the-art. *IEEE Trans Microw Theory Tech* 52(1):420–435

117. Reddipogu A, Maxwell G, MacLeod C, Simpson M (2002) A novel artificial neural network trained using evolutionary algorithms for reinforcement learning. In: Proceeding of the international conference on neural information processing. doi:[10.1109/ICONIP.2002.1199013](https://doi.org/10.1109/ICONIP.2002.1199013)
118. Rodriguez-Vzquez A, Dominguez-Castro R, Rueda A, Huertas JL, Sanchez-Sinencio E (1990) Nonlinear switched-capacitor neural networks for optimization problems. *IEEE Trans Circ Syst II* 37:384397
119. Romero RAF (1996) Otimizao de Sistemas atravs de Redes Neurais Multi-camadas. XI Congresso Brasileiro de Automtica, pp vol 2. So Paulo, Brasil pp 1585–1590
120. Rossana MS, Cruz HMP, Magalhaes RM (2011) Artificial neural networks and efficient optimization techniques for applications in engineering. *Artificial neural networks—methodological advances and biomedical applications*, pp 45–68
121. Schaffer JD, Whitley D, Eshelman LJ (1992) Combinations of genetic algorithms and neural networks: a survey of the state of the art. International workshop on combinations of genetic algorithms and neural networks (1992) COGANN-92. 1–37. doi:[10.1109/COGANN.1992.273950](https://doi.org/10.1109/COGANN.1992.273950)
122. Sebald AV, Chellapilla K (1998) On making problems evolutionarily friendly part 2: evolving the most convenient representations. The seventh international conference on evolutionary programming, EP98, San Diego, pp 281–290
123. Sheng-Fuu L, Jyun-Wei C (2013) Adaptive group organization cooperative evolutionary algorithm for tsk-type neural fuzzy networks design. *Int J Adv Res Artif Intell (IJARAI)* 2(3):1–9
124. Shepherd AJ (1997) Second-order methods for neural networks fast and reliable methods for multi-layer perceptrons. Springer
125. Shumeet B (1996) Evolution of an artificial neural network based autonomous land vehicle controller. *IEEE Trans Syst Man Cybern* 26(3):450–463. doi:[10.1109/3477.499795](https://doi.org/10.1109/3477.499795)
126. Stepniewski SW, Keane AJ (1996) Topology design of feedforward neural networks by genetic algorithms. Parallel problem solving from nature, pp 771–780. doi:[10.1007/3-540-61723-1040](https://doi.org/10.1007/3-540-61723-1040)
127. Sudharsanan SI, Sundareshan MK (1991) Exponential stability and a systematic synthesis of a neural network for quadratic minimization. *Neural Netw* 4:599–613
128. Suraweera NP, Ranasinghe DN (2008) A natural algorithmic approach to the structural optimisation of neural networks. In: Proceedings of the 4th international conference on information and automation for sustainability, ICIAFS 2008, pp 150–156
129. Taishan Y, Duwu C, Yongqing T (2007) A new evolutionary neural network algorithm based on improved genetic algorithm and its application in power transformer fault diagnosis. doi:[10.1109/BICTA.2007.4806406](https://doi.org/10.1109/BICTA.2007.4806406)
130. Tai-shan Y (2010) An improved evolutionary neural network algorithm and its application in fault diagnosis for hydropower units. In: Proceedings of the international conference on intelligent computation technology and automation. doi:[10.1109/ICICTA.2010.589](https://doi.org/10.1109/ICICTA.2010.589)
131. Talbot C, Massara R (1993) An application oriented comparison of optimization and neural network based design techniques. In: Proceedings of the 36th midwest symposium on circuits and systems. vol 1, pp 261–264
132. Tan ZH (2004) Hybrid evolutionary approach for designing neural networks for classification. *Electr Lett* 40(15). doi:[10.1049/el:20045250](https://doi.org/10.1049/el:20045250)
133. Tank DW, Hopfield JJ (1986) Simple neural optimization network: an AD onverter. Signal decision circuit and a linear programming circuit *IEEE trans circuits and systems*, CAS-33, 533–541
134. Tao Q, Cao JD, Xue MS, Qiao H (2001) A high performance neural network for solving nonlinear programming problems with hybrid constraints. *Phys Lett A* 288(2):8894
135. Taylor CM (1997) Selecting neural network topologies: a hybrid approach combining genetic algorithms and neural network. B.S. Computer Science Southwest Missouri State University
136. Tenne Y, Goh CKE (2010) Computational intelligence in optimization, volume adaptation, learning, and optimization, vol 7. Springer
137. Thimm G, Fiesler E (1997) High-order and multilayer perceptron initialization. *IEEE Trans Neural Netw* 8(2):349–359

138. Tian Y, Zhang J, Morris J (2001) On-line re-optimisation control of a batch polymerisation reactor based on a hybrid recurrent neural network model. In: Proceedings of the american control conference. Arlington, pp 350–355
139. Topchy AP, Lebedko OA (1997) Neural network training by means of cooperative evolutionary search. Nucl Instrum Methods Phys Res Sect A Accel Spectrom Detect Assoc Equip 389(1–2):240–241
140. Trajkovic T, Pantic D (1995) Inverse modeling and optimization of low-voltage power VDMOSFET's technology by neural networks. Semiconductor, international conference. doi:[10.1109/SMICND.1995.494868](https://doi.org/10.1109/SMICND.1995.494868)
141. Ueda H, Ishikawa M (1997) Rule extraction from data with continuous valued inputs and discrete valued outputs using neural networks. Technical report IEICE Japan, NC96-121, pp 63–70
142. Valdes J, Barton A (2007) Multi-objective evolutionary optimization of neural networks for virtual reality visual data mining: application to hydrochemistry. In: Proceedings of the international joint conference on neural networks, 2007. IJCNN 2007, pp 2233–2238
143. Velazco MI, Lyra C (2002) Optimization with neural networks trained by evolutionary algorithms. Int Symp Neural Netw 2:1516–1521. doi:[10.1109/IJCNN.2002.1007742](https://doi.org/10.1109/IJCNN.2002.1007742)
144. Vonk E, Lakhmi CJ, Veelenturf LPJ, Johnson R (1995) Automatic generation of a neural network architecture using evolutionary computation. Electron Technol Dir Year 2000:144–149. doi:[10.1109/ETD.1995.403479](https://doi.org/10.1109/ETD.1995.403479)
145. Wai ST, Jun W (2000) Two recurrent neural networks for local joint torque optimization of kinematically redundant manipulators. IEEE Trans Syst Man Cybern Part B: Cybern 30(1):120–128
146. Wang J (1996) Recurrent neural networks for optimization. In: Chen CH (ed) Fuzzy logic and neural network handbook. New York: McGraw-Hill, 4.14.35
147. Wang L (2005) A hybrid genetic algorithmneural network strategy for simulation optimization. Appl Math Comput 170(2):1329–1343
148. Wei B, Xiaoping X (2009) Subgradient-based neural networks for nonsmooth nonconvex optimization problems. IEEE Trans Neural Netw 20(6):1024–1038
149. Werbos P (1991) An overview of neural networks for control. Control Syst IEEE 11(1):40–41
150. Whitley D (2001) An overview of evolutionary algorithms: practical issues and common pitfalls. Inf Softw Technol 43:817–831
151. Wu X, Xia Y, Li J, Chen WK (1996) A high performance neuralnetwork for solving linear and quadratic programming problems. IEEE Trans Neural Netw 7:643–651
152. Xia SY, Wang J (1995) Neural network for solving linear programming problems with bounded variables. IEEE Trans Neural Netw 6:515–519
153. Xia SY (1996) A new neural network for solving linear and quadratic programming problems. IEEE Trans Neural Netw 7:1544–1547
154. Xia SY (1996) A new neural network for solving linear programming problems and its applications. IEEE Trans Neural Netw 7:525–529
155. Xia SY, Wang J (1998) A general methodology for designing globally convergent optimization neural networks. IEEE Trans Neural Netw 9(6):1331–1343. doi:[10.1109/72.728383](https://doi.org/10.1109/72.728383)
156. Xia SY, Leung H, Wang J (2001) A dual neural network for kinematic control of redundant robot manipulators. IEEE Trans Syst Man Cybern B 31:147–154
157. Xia SY, Leung H, Wang J (2002) A projection neural network and its application to constrained optimization problems. IEEE Trans Circ Syst II 49:447–458
158. Xia Y, Jun W (2003) A general projection neural network for solving optimization and related problems. In: Proceedings of the international joint conference on neural networks, vol 3, pp 2334–2339. doi:[10.1109/IJCNN.2003.1223776](https://doi.org/10.1109/IJCNN.2003.1223776)
159. Xia Y, Jun W (2004) A general projection neural network for solving monotone variational inequalities and related optimization problems. IEEE Trans Neural Netw 15(2):318–328. doi:[10.1109/TNN.2004.824252](https://doi.org/10.1109/TNN.2004.824252)
160. Xia Y, Wang J (2004) A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints. IEEE Trans Circ Syst I 51(7):1385–1394

161. Xiao S, Yo D, Li Y (2006) Parallel learning evolutionary algorithm based on neural network ensemble. In: Proceedings of the international conference on information acquisition. doi:[10.1109/ICIA.2006.305824](https://doi.org/10.1109/ICIA.2006.305824)
162. Xiaolin H, Bo Z (2009) An alternative recurrent neural network for solving variational inequalities and related optimization problems. *IEEE Trans Syst Man Cybern Part B: Cybern*, 39 (6)
163. Xiyu L, Huichuan D, Mingxi T (2005) Design optimization by functional neural networks. *Computer supported cooperative work in design*. 824–829. doi:[10.1109/CSCWD.2005.194292](https://doi.org/10.1109/CSCWD.2005.194292)
164. Xue W (2010) Chaotic artificial neural network in reactive power optimization of distribution network. *China international conference on electricity distribution*
165. Xue-Bin L, Jun W (2000) A recurrent neural network for nonlinear optimization with a continuously differentiable objective function and bound constraints. *IEEE Trans Neural Netw* 11 (6)
166. Xuejun C, Jianzhou W, Donghui S, Jinzhao L (2008) A novel hybrid evolutionary algorithm based on PSO and AFSA for feedforward neural network training. In: Proceedings of the international conference on wireless communications, networking and mobile computing. doi:[10.1109/WiCom.2518](https://doi.org/10.1109/WiCom.2518)
167. Yao X (1993) A review of evolutionary artificial neural networks. *Int J Intell Syst* 4:539–567
168. Yao X (1995) Designing artificial neural networks using co-evolution. In: Proceedings of IEEE singapore international conference on intelligent control and instrumentation, pp 149–154
169. Yao X, Liu Y (1997) A new evolutionary system for evolving artificial neural networks. *IEEE Trans Neural Netw* 8(3):694–713
170. Yao X, Liu Y (1998) Making use of population information in evolutionary artificial neural networks. *IEEE Trans Syst Man Cybern Part B: Cybern* 28(3):417–425
171. Yao X, Liu Y (1998) Towards designing artificial neural networks by evolution. *Appl Math Comput* 91(1):83–90
172. Yao X (1999) Evolving artificial neural networks. *Proc IEEE* 87(9):1423–1447
173. Yaochu J, Tatsuya O, Bernhard S (2004) Neural network regularization and ensembling using multi-objective evolutionary algorithms. *IEEE congress on evolutionary computation*. doi:[10.1109/CEC.2004.1330830](https://doi.org/10.1109/CEC.2004.1330830)
174. Yasin ZM, Rahman TKA, Zakaria Z (2013) Quantum-inspired evolutionary programming-artificial neural network for prediction of undervoltage load shedding. In: Proceedings of the 8th IEEE conference on industrial electronics and applications (ICIEA), pp 583–588. doi:[10.1109/ICIEA.2013.6566436](https://doi.org/10.1109/ICIEA.2013.6566436)
175. Yong C, Xia L, Qi H, Chang-hua Z (2008) An artificial neural network based on CIEA. In: Proceedings of the international conference on computational intelligence and security, pp 35–40. doi:[10.1109/CIS.2008.178](https://doi.org/10.1109/CIS.2008.178)
176. Young-Seok H, Hungu L, MJT (2010) Acceleration of the convergence speed of evolutionary algorithms using multilayer neural networks. *Eng Optim* 35:91–102
177. Youshen X, Henry L, Jun W (2002) A projection neural network and its application to constrained optimization problems. *IEEE Trans Circuits Syst-I: Fundam Theory Appl* 49(4):447–458
178. Yu J, Lam A, Li VK (2011) Evolutionary artificial neural network based on chemical reaction optimization. *IEEE Congress on evolutionary computation (CEC)*, pp 2083–2090
179. Yuji S, Shigei N (1996) Evolutionary algorithms that generate recurrent neural networks for learning chaos dynamics. In: Proceedings of the international conference on evolutionary computation. 144–149. doi:[10.1109/ICEC.1996.542350](https://doi.org/10.1109/ICEC.1996.542350)
180. Young-Chin L, Yung-Chien L, Kuo-Lan S, Wen-Cheng C (2009) Mixed-integer evolutionary optimization of artificial neural networks. In: Proceedings of the international conference on innovative computing, information and control. doi:[10.1109/ICICIC.2009.260](https://doi.org/10.1109/ICICIC.2009.260)

181. Yun L, Alexander H (1996) Artificial evolution of neural networks and its application to feedback control. *Artif Intell Eng* 10(2):143–152. doi:[10.1016/0954-1810\(95\)00024-0](https://doi.org/10.1016/0954-1810(95)00024-0)
182. Zhang S, Zhu X, Zou LH (1992) Second-order neural networks for constrained optimization. *IEEE Trans Neural Netw* 3:1021–1024
183. Zhang BT, Mhlenbein H (1993) Evolving optimal neural networks using genetic algorithms with Occam's razor. *Complex Syst* 7(3):199–220
184. Zhou ZH, Wu J, Tang W (2002) Ensembling neural networks: many could be better than all. *Artif Intell* 137(1–2):239–263