



Nonextensive statistical mechanics for hybrid learning of neural networks

Aristoklis D. Anastasiadis*, George D. Magoulas

Birkbeck College, School of Computer Science and Information Systems, University of London, Malet Street, London WC1E 7HX, UK

Available online 9 September 2004

Abstract

This paper introduces a new hybrid approach for learning systems that builds on the theory of nonextensive statistical mechanics. The proposed learning scheme uses only the sign of the gradient, and combines adaptive stepsize local searches with global search steps that make use of an annealing schedule inspired from nonextensive statistics, as proposed by Tsallis. The performance of the hybrid approach is empirically investigated through simulation in benchmark problems from the UCI Repository of Machine Learning Databases. Preliminary results provide evidence that the synergy of techniques from nonextensive statistics provide neural learning schemes with significant benefits in terms of learning speed and convergence success.

© 2004 Elsevier B.V. All rights reserved.

PACS: 07.05.Mh; 87.18.Sn; 05.10.–a

Keywords: Artificial neural networks; Generalized simulated annealing; Global search; Gradient descent; Nonextensive statistics; Pattern classification; Resilient propagation; Supervised learning

1. Introduction

Learning systems, such as the multilayer feedforward neural network (FNN), are non-linear systems modelled on the general features of biological systems that exhibit emergent behavior [1]. The operation of an FNN is usually based on the following equations:

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

* Corresponding author.

E-mail address: aris@dcs.bbk.ac.uk (A.D. Anastasiadis).

where net_j^l is for the j th node in the l th layer ($j = 1, \dots, n_l$), the sum of its weighted inputs. The weights from the i th node at the $(l - 1)$ layer to the j th node at the l th layer are denoted by $w_{ij}^{l-1,l}$, y_j^l is the output of the j th node that belongs to the l th layer, and $f(net_j^l)$ is the j th's node activation function.

The goal of FNN learning is to iteratively adjust the weights, in order to globally minimize a measure of the difference between the actual output of the network and the desired output, as specified by a teacher, for all examples in a training set [2]. If there is a fixed, finite set of input–output examples, the square error over the training set, which contains P representative examples, is

$$E(w) = \sum_{p=1}^P \sum_{j=1}^{n_L} (y_{j,p}^L - t_{j,p})^2 = \sum_{p=1}^P \sum_{j=1}^{n_L} [\sigma^L(net_j^L + \theta_j^L) - t_{j,p}]^2. \quad (2)$$

This equation formulates the energy function, called *error function*, to be minimized, in which $t_{j,p}$ specifies the desired response at the j th output node for the example p and $y_{j,p}^L$ is the output of the j th node at layer L that depends on the weights of the network, and σ is a nonlinear activation function, such as the well-known logistic function $\sigma(x) = (1 + e^{-x})^{-1}$. The weights in the network can be expressed using vector notation $w \in \mathbb{R}^n$, as:

$$w = (\dots, w_{ij}^{l-1,l}, w_{i+1,j}^{l-1,l}, \dots, w_{N_{l-1},j}^{l-1,l}, \theta_j^l, w_{i,j+1}^{l-1,l}, w_{i+1,j+1}^{l-1,l}, \dots)^\top, \quad (3)$$

where θ_j^l denotes the bias of the j th node ($j = 1, \dots, N_l$) at the l th layer ($l = 2, \dots, L$), and n denotes the total number of weights and biases in the network.

It is well known in the neural networks field [1,2] that the rapid computation of such a global minimum is a difficult task because the dimensionality of the weights space is high, and the corresponding nonconvex multimodal objective function possesses multitudes of local minima and has broad flat regions adjoined with narrow steep ones. First-order methods are the most widely used class of algorithms for supervised learning of neural networks [3]. Among these methods, adaptive stepsize algorithms try to overcome the inherent difficulty of choosing the right stepsize for each problem [4]. They work by controlling the magnitude of the changes in the weight states during learning in an attempt to avoid oscillations and, at the same time, maximize the length of the minimization step [5]. A variety of approaches inspired from unconstrained optimization theory have also been applied, in order to use second derivative-related information to accelerate the learning process [3,6–8]. Nevertheless, it is not certain that the extra computational cost these methods require leads to speed ups of the minimization process for nonconvex functions when far from a minimizer [9]; this is usually the case with the neural network training problems [3]. An inherent difficulty with first-order and second-order learning schemes is convergence to local minima. While some local minima can provide acceptable solutions, they often result in poor network performance. This problem can be overcome through the use of global optimization [10–13].

This paper introduces a hybrid search strategy that aims to alleviate the problem of occasional convergence to local minima in supervised training. Our approach combines a quick and computationally cheap local search method with a global search

technique inspired from the generalized entropy of nonextensive statistical mechanics, and replaces the usual Boltzmann–Gibbs factor used in simulated annealing by the *q-exponential function* [14,15]. The next section describes the background of our approach and the new learning scheme. Then results of an empirical evaluation are presented, demonstrating the effectiveness of the new scheme in reducing the probability of convergence to poor local minima. The paper ends with concluding remarks.

2. A hybrid learning scheme

Our approach belongs to the special class of adaptive training algorithms that employ a different adaptive stepsize for each weight. Algorithms of this class avoid slow convergence in the flat directions and oscillations in the steep directions, and exploit the parallelism inherent in the evaluation of learning error $E(w)$ and gradient $\nabla E(w)$ by the Back-Propagation (BP) algorithm [1]. Various algorithms of this class have been suggested in the literature, such as Refs. [4,5,16,17]. Among them the *Resilient Propagation* (Rprop) algorithm is one of the most popular methods [5]. The basic principle of Rprop is to eliminate harmful influences of the size of the partial derivatives on the weights adjustments. As a consequence, only the sign of the derivative is considered to indicate the direction of the weights change

$$w^{k+1} = w^k - \text{diag}\{\eta_1^k, \dots, \eta_i^k, \dots, \eta_n^k\} \text{sign}(g(w^k)), \quad (4)$$

where k indicates iterations, $\text{diag}\{\eta_1, \dots, \eta_n\}$ defines the $n \times n$ diagonal matrix with elements η_1, \dots, η_n , and η_i^k ($i = 1, 2, \dots, n$) are the k th iteration stepsizes that receive small positive real values, also called *learning rates* as their role is to control the amount of weights adjustment and thus directly affect the rate of the learning process. In (4), $\text{sign}(g(w^k))$ denotes the column vector of the signs of the components of $g(w^k)$; $g(w)^\top = (g_1(w), \dots, g_n(w))$ defines the transpose of the gradient $\nabla E(w)$ of the sum-of-squared-differences error function E at w ; η_i^k ($i = 1, 2, \dots, n$) are generated by Rprop's schedule:

$$\text{if } (g_i(w^{k-1})g_i(w^k) > 0) \text{ then } \eta_i^k = \min(\eta_i^{k-1}\eta^+, \Delta_{max}), \quad (5)$$

$$\text{if } (g_i(w^{k-1})g_i(w^k) < 0) \text{ then } \eta_i^k = \max(\eta_i^{k-1}\eta^-, \Delta_{min}), \quad (6)$$

$$\text{if } (g_i(w^{k-1})g_i(w^k) = 0) \text{ then } \eta_i^k = \eta_i^{k-1}, \quad (7)$$

where $0 < \eta^- < 1 < \eta^+$, Δ_{max} is the stepsize upper bound, and Δ_{min} is the stepsize lower bound. The method requires setting the following parameters [5]: (i) the increase factor is set to $\eta^+ = 1.2$; (ii) the decrease factor is set to $\eta^- = 0.5$; (iii) the initial stepsize for all i is set to $\eta^0 = 0.1$; (iv) the maximum allowed stepsize, which is used in order to prevent the weights from becoming too large, is $\Delta_{max} = 50$; (v) the minimum allowed stepsize $\Delta_{min} = 10^{-6}$.

2.1. Annealing schedules in neural networks learning

Despite the fact noise plays an influential role in the operation of real neurons, e.g. neural cells' responses to identical stimuli have been found to be stochastic in

nature, the effect of noise on the operation of artificial neural networks has not been investigated in depth. One of the most famous neural networks model operating with noise is the Boltzmann machine [18,19]. It is inspired by the Boltzmann–Gibbs entropy $S_{BG} = -K \sum_i p_i \ln p_i$ that provides exponential laws for describing stationary states and basic time-dependent phenomena, where $\{p_i\}$ are the probabilities of the microscopic configurations, and $K > 0$. In addition, attempts to explore the benefits of introducing noise during learning, such as in Refs. [10,18,20], have been based on the use of Gaussian distributions. In particular, the use of simulated annealing (SA) has been explored for learning of the Boltzmann machine [18,19]. SA refers to the process in which random noise in a system is systematically decreased at a constant rate so as to enhance the response of the system [21]. In the numerical optimization framework, SA is a procedure that has the capability to move out of regions near local minima [22,23]. SA is based on random evaluations of the objective function, in such a way that transitions out of a local minimum are possible. First, it reaches an area in the function domain space where a global minimizer should be present, following the gross behavior irrespectively of small local minima found on the way. It then develops finer details, finding a good, near-optimal local minimizer, if not the global minimum itself [24].

In the context of neural networks, learning the performance of the classical SA is not the appropriate one: the method needs a greater number of function evaluations than that usually required for a single run of first-order learning algorithms and does not exploit derivative-related information. Notice that the problem with minimizing a neural network's error function is not the well-defined local minima but the broad regions that are nearly flat. In this case, the so-called Metropolis move is not strong enough to move the algorithm out of these regions [25]. To alleviate this situation [10] has suggested to incorporate an annealing schedule in the steepest descent algorithm:

$$w^{k+1} = w^k - \eta \nabla E(w^k) + \rho c 2^{-dk}, \quad (8)$$

where k is the iteration number, η is a common fixed stepsize for all weights, ρ is a constant controlling the initial intensity of the noise, $c \in (-0.5, +0.5)$ is a random number, and d is the noise decay constant. This approach does not use the notion of the acceptance probability, such as the Metropolis algorithm in the classic SA [21], or the generalized acceptance probability in the generalized SA [26]. Instead, it implements a form of Langevin noise that has been proved quite effective in neural systems learning, [27,20], and has motivated the development of other methods, such as the *Simulated Annealing Rprop-SARprop* and the *SARprop with Restarts-ReSARprop* [13].

2.2. The entropic index q and the derivation of a new method

In our approach noise is generated by a noise source that is characterized by the nonextensive entropic index q . In particular, Tsallis has defined the nonextensive entropy [14]:

$$S_q \equiv K \frac{1 - \sum_{i=1}^W p_i^q}{q - 1} \quad (q \in \mathbb{R}), \quad (9)$$

where W is the total number of microscopic configurations, whose probabilities are $\{p_i\}$, and K is a conventional positive constant. When the entropic index $q = 1$, Eq. (9) recovers to Boltzmann–Gibbs entropy. The entropic index works like a biasing parameter: $q < 1$ privileges rare events (values of p close to 0 are benefited), while $q > 1$ privileges common events (values of p close to 1). The optimization of the entropic form (9) under appropriate constraints, [14,28], yields for the canonical ensemble

$$p_i \propto [1 - (1 - q)\beta E_i]^{1/(1-q)} \equiv e_q^{-\beta E_i}, \quad (10)$$

where β is a Lagrange parameter, $\{E_i\}$ is the energy spectrum, and the q -exponential function is defined as

$$e_q^x \equiv [1 + (1 - q)x]^{1/(1-q)} = \frac{1}{[1 - (q - 1)x]^{1/(q-1)}}. \quad (11)$$

Following the above discussion and inspired by Burton and Miptsos [10] and Tsallis and Stariolo [26], in our method, noise is generated according to a schedule that can be expressed as

$$e_q^{-T(\ln 2)k} = [1 - (1 - q)T(\ln 2)k]^{1/(1-q)}, \quad (12)$$

where T is the temperature; k indicates iterations. In our approach, noise is not applied proportionally to the size of each weight; instead a form of weight decay is used, which is considered beneficial for achieving a robust neural network that generalizes well [29,13]. Thus, noise is introduced in neural network learning by formulating the *perturbed* energy function:

$$\tilde{E}(w^k) = E(w^k) + \mu \sum_{i=1}^n \frac{(w_i^k)^2}{[1 + (w_i^k)^2]} [1 - (1 - q)T(\ln 2)k]^{1/(1-q)}, \quad (13)$$

where k indicates iterations, $E(w)$ is given by (2), $\sum_i w_i^2/(1 + w_i^2)$ is the weight decay bias term, which can decay small weights more rapidly than large weights, and μ is a parameter that regulates the influence of the combined weight decay/noise effect. This form of weight decay modifies the energy landscape so that smaller weights are favored at the beginning of the training, but as learning progresses the magnitude of the weight decay is reduced to favor the growth of large weights. Thus, as the energy landscape is modified during training the search method is allowed to explore regions of the energy surface that were previously unavailable. Minimization of (13) requires calculating the gradient of the energy term with respect to each weight

$$\tilde{g}_i(w^k) = g_i(w^k) + \dot{\mu} \frac{w_i^k}{[1 + (w_i^k)^2]^2} [1 - (1 - q)T(\ln 2)k]^{1/(1-q)}, \quad (14)$$

where $\dot{\mu} > 0$ (in our experiments, reported in the next section, a fixed value of $\dot{\mu} = 0.01$ was used). The proposed hybrid strategy applies the sign-based weight adjustment of Rprop, defined by relation (4), on the perturbed energy function (13), using the gradient term of Eq. (14). Also, the learning rates are adapted by means of conditions (5)–(7), where $\tilde{g}_i(w^k)$ is used instead of $g_i(w^k)$. Lastly, an additional condition is introduced

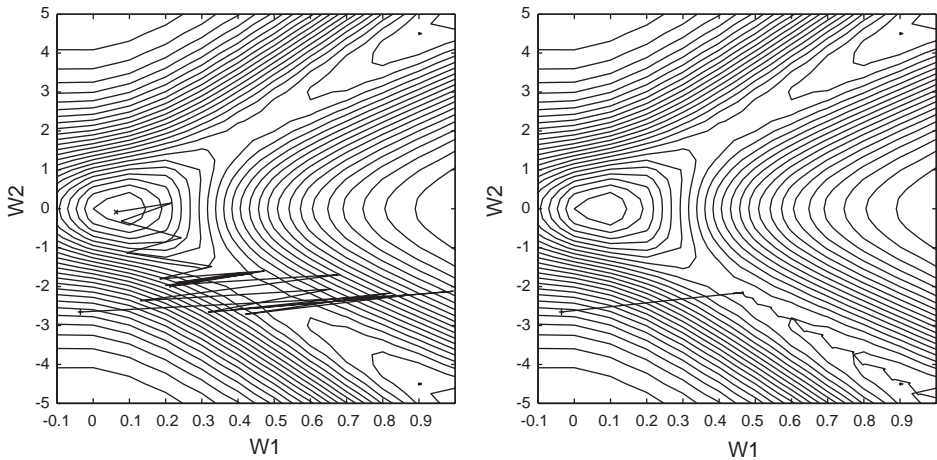


Fig. 1. The weights trajectory of the Hybrid Learning Scheme converges to the global minimum (left), whilst the trajectory of Rprop to a local minimizer (right).

in order to avoid using relatively small weight adjustments

$$\begin{aligned} \text{if } & (\eta_i^{k-1} < \rho[1 - (1 - q)T(\ln 2)k]^{2/(1-q)}) \\ \text{then } & \eta_i^k = \max(\eta_i^{k-1}\eta^- + 2c\rho[1 - (1 - q)T(\ln 2)k]^{2/(1-q)}, \Delta_{min}), \end{aligned} \quad (15)$$

where $0 < \rho < 1$ and $c \in (0, 1)$ is a random number.

Below, a simple problem is used to visualize the behavior of the *hybrid learning scheme* (HLS) and compare it with the Rprop algorithm. It is a single node with two weights and a logistic activation function. The energy landscape of Fig. 1 has a global minimum and two local minima. Fig. 1 shows that under the same initial conditions, HLS escapes a saddle point and a valley that leads to a local minimum, and converges to the global minimizer located at the center of the contour plot (Fig. 1, left), while Rprop converges to the local minimizer (Fig. 1, right).

3. Experimental study

In this section, we evaluate the performance of the HLS and compare it with the Rprop and the SARprop algorithms. We have used well-studied problems from the UCI Repository of Machine Learning Databases of the University of California [30], as well as problems studied extensively by other researchers in an attempt to reduce as much as possible biases introduced by the size of the weights space. In all cases we have used networks with classic logistic activations. The guidelines of Refs. [5,13] were adopted for setting the learning parameters of Rprop and SARprop, respectively. The parameters of the HLS were set to the same values for all experiments in an attempt to test the robustness of the method in different types of problems: the entropic index $q = 1.2$ and the temperature $T = 1$.

Table 1
Average performance in the Iris and Cancer problems

Algorithm	Iris			Cancer		
	<i>IT</i>	<i>GEN.</i>	<i>CONV.</i>	<i>IT</i>	<i>GEN.</i>	<i>CONV.</i>
Rprop	1400 (+)	98.4 (+)	96	279 (+)	97.2 (–)	94
SARprop	1430 (+)	98.9 (+)	96	282 (+)	97.6 (–)	87
HLS	1377	99.5	100	157	97.5	100

Below, we report results from 100 independent trials for four UCI problems. These 100 random weight initializations are the same for the three learning algorithms, and the training and testing sets were created according to *Proben1* [31]. The statistical significance of the results has been analyzed using the Wilcoxon test [32]. This is a nonparametric method that is considered an alternative to the paired *t*-test. It assumes there is information in the magnitudes of the differences between paired observations, as well as the signs. All statements in the tables reported below refer to a significance level of 0.05.

The first benchmark is known as the *Fisher’s Iris* problem [30,31]. The data set consists of 120 examples and the test set of 30 examples. Following Ref. [13], a 4–2–3 FNN (4 input–2 hidden–3 output nodes; 19 weights overall) was used, and the maximum number of iterations to find a “near-optimal” weight configuration (defined as a weight set w^* that results to an error function value $E(w^*) \leq 0.01$) was set to 2000. Table 1 shows the average performance in terms of: iterations to converge to the error target (*IT*), success of convergence to the target, within 2000 iterations (*CONV.*, out of the 100 runs), and generalization (*GEN.*, percentage of correctly classified test examples); a “+” indicates statistical significance of the HLS results over another method.

The second benchmark is the *breast cancer diagnosis* problem that classifies a tumor as benign or malignant based on 9 features [30,31]. We have used an FNN with 9–4–2–2 nodes (a total of 56 weights) as suggested in Ref. [31]. The results are shown in Table 1. In this case, SARprop exhibits the lowest percentage of convergence within the 2000 iterations.

The *diabetes1* benchmark is a real-world classification task that concerns deciding when a Pima Indian individual is diabetes positive or not [30,31]. There are 8 features representing personal data and results from a medical examination. The *Proben1* collection suggests a 8–2–2–2 FNN (34 weights overall). The termination criterion is $E \leq 0.1$ within 2000 iterations. Lastly, the *thyroid1* problem, [30,31], uses a 21–4–3 nodes FNN, suggested by Treadgold and Gedeon [13], to decide whether the patient’s thyroid has over function, normal function, or under function. A data set with 3600 examples is used and the target is to find within a maximum of 2000 iterations a weight set that produces $E \leq 0.0036$. Table 2 gives the average performance of the three algorithms in the two problems. The new method outperforms the other methods in the number of iterations required to reach a suitable solution, and converges in all cases.

Table 2
Average performance in the diabetes and thyroid problems

Algorithm	Diabetes			Thyroid		
	<i>IT</i>	<i>GEN.</i>	<i>CONV.</i>	<i>IT</i>	<i>GEN.</i>	<i>CONV.</i>
Rprop	357 (+)	75.9 (+)	96	793 (+)	98.0 (–)	78
SARprop	325 (+)	75.8 (+)	96	736 (+)	98.1 (–)	92
HLS	223	76.1	100	460	98.2	100

Table 3
Average performance in the XOR and Parity-4 problems

Algorithm	XOR			Parity 4		
	<i>IT</i>	<i>GEN.</i>	<i>CONV.</i>	<i>IT</i>	<i>GEN.</i>	<i>CONV.</i>
Rprop	1110 (+)	100 (–)	23	1360 (+)	100 (–)	42
SARprop	168 (+)	100 (–)	98	1378 (+)	100 (–)	48
HLS	49	100	100	1270	100	100

Table 4
Average algorithm performance in the Parity-3 and 5 problems

Algorithm	Parity 3			Parity 5		
	<i>IT</i>	<i>GEN.</i>	<i>CONV.</i>	<i>IT</i>	<i>GEN.</i>	<i>CONV.</i>
Rprop	1105 (+)	100 (–)	22	416 (+)	100 (–)	67
SARprop	882 (+)	100 (–)	78	394 (+)	100 (–)	95
HLS	640	100	100	20	100	100

Another set of experiments has been conducted to empirically evaluate the performance of the new method in a well-studied class of boolean function approximation problems that exhibit strong local minima [33,34]. This class includes the XOR problem (whose local minima and saddle points have been analyzed in detail) and the various parity- N problems, which are considered as classic benchmarks [35,11,13,8]. The error target was set to $E \leq 10^{-7}$ within 2000 iterations in all cases (this is considered low enough to guarantee convergence to a “global” solution, especially for the XOR problem), and the adopted architectures were 2–4–1 for the XOR, 3–3–1 for the parity-3, 4–6–1 for the parity-4, 5–7–1 for the parity-5, following the recommendations of [13]. The results are presented in Tables 3 and 4. Fig. 2 gives a typical example of algorithms’ convergence. Starting from the same initial conditions, the Rprop converges to a local minimizer, whilst both SARprop and HLS escape from the local minimum. However, HLS converges to a feasible solution much faster than SARprop.

Additional experiments have been performed to explore the influence of the entropic index q on the convergence speed of the HLS. According to our experiments, large

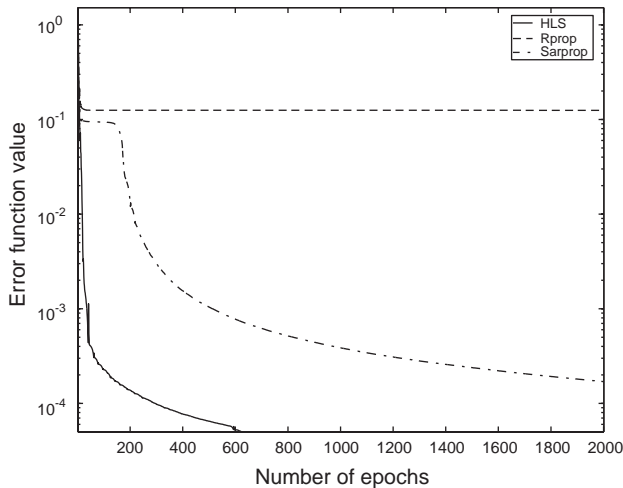


Fig. 2. Typical learning error curve for the Parity-3 problem.

values of q cause an increase to the average number of iterations required to achieve the error target but seem not to affect the convergence success. In the XOR problem, for example, the HLS requires on average 378 iterations to converge (100% success) when using $q = 1.7$ and $T = 1$ (cf. with Table 3, where 49 iterations are required with $q = 1.2$ and $T = 1$). The HLS exhibits similar behavior in the parity-5 problem, where an average of 440 iterations is required when $q = 1.7$ and $T = 1$ (cf. with Table 4 where $q = 1.2$ and $T = 1$). Nevertheless, the HLS convergence rate using $q = 1.7$ appears to be improving in both problems after fine tuning the value of T ; it then exhibits an average of 69 iterations in the XOR problem, and an average of 33 iterations in the parity-5 problem.

4. Concluding remarks

The paper introduces a new hybrid learning scheme that combines deterministic and stochastic steps with a different adaptive stepsize for each weight, and a form of noise that is characterized by the nonextensive entropic index q , regulated by a weight decay term. Preliminary experiments with the hybrid scheme, and comparisons with two other popular learning methods, namely the Rprop and the SARprop, have been very encouraging. Accelerated and reliable neural learning was achieved in all cases tested. Further testing is of course necessary to fully explore the advantages and identify possible limitations of the hybrid learning scheme.

Acknowledgements

The authors are grateful to Prof. Tsallis for insights and stimulating discussions.

References

- [1] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, in: D.E. Rumelhart, J.L. McClelland (Eds.), *Parallel Distributed Processing: explorations in the Microstructure of Cognition*, Vol. 1, MIT Press, Cambridge, 1986, pp. 318–362.
- [2] S. Haykin, *Neural Networks: a Comprehensive Foundation*, Macmillan College Publishing Company, New York, 1994.
- [3] R. Battiti, First- and second-order methods for learning: between steepest descent and Newton's method, *Neural Comput.* 4 (1992) 141–166.
- [4] G.D. Magoulas, M.N. Vrahatis, G.S. Androulakis, Improving the convergence of the backpropagation algorithm using learning rate adaptation methods, *Neural Comput.* 11 (1999) 1769–1796.
- [5] M. Riedmiller, H. Braun, A direct adaptive method for faster backpropagation learning: the Rprop algorithm. *Proceedings of the International Conference on Neural Networks*, San Francisco, CA, 1993, 586–591.
- [6] G.D. Magoulas, M.N. Vrahatis, T.N. Grapsa, G.S. Androulakis, Neural network supervised training based on a dimension reducing method, in: S.W. Ellacot, J.C. Mason, I.J. Anderson (Eds.), *Mathematics of Neural Networks: Models, Algorithms and Applications*, Kluwer, Dordrecht, 1997, pp. 245–249.
- [7] M.F. Møller, A scaled conjugate gradient algorithm for fast supervised learning, *Neural Networks* 6 (1993) 525–533.
- [8] P.P. Van der Smagt, Minimization methods for training feedforward neural networks, *Neural Networks* 7 (1994) 1–11.
- [9] J. Nocedal, Theory of algorithms for unconstrained optimization, *Acta Numerica* 1 (1992) 199–242.
- [10] R.M. Burton, G.J. Mpitsos, Event dependent control of noise enhances learning in neural networks, *Neural Networks* 5 (1992) 627–637.
- [11] V.P. Plagianakos, G.D. Magoulas, M.N. Vrahatis, Learning in multilayer perceptrons using global optimization strategies, *Nonlinear Anal.: Theory, Methods Appl.* 47 (2001) 3431–3436.
- [12] V.P. Plagianakos, G.D. Magoulas, M.N. Vrahatis, Supervised training using global search methods, in: N. Hadjisavvas, P. Pardalos (Eds.), *Advances in Convex Analysis and Global Optimization*, Vol. 54, *Nonconvex Optimization and its Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001, 421–432.
- [13] N.K. Treadgold, T.D. Gedeon, Simulated annealing and weight decay in adaptive learning: the SARPROP Algorithm, *IEEE Trans. Neural Networks* 9 (4) (1998) 662–668.
- [14] C. Tsallis, Possible generalization of Boltzmann–Gibbs statistics, *J. Statist. Phys.* 52 (1–2) (1988) 479–487.
- [15] C. Tsallis, R.S. Mendes, A.R. Plastino, *Physica A* 261 (1998) 534.
- [16] M. Pfister, R. Rojas, Speeding-up backpropagation—a comparison of orthogonal techniques, *Proceedings of the Joint Conference on Neural Networks*, Nagoya, Japan, 1993, 517–523.
- [17] M. Pfister, R. Rojas, Qrprop—a hybrid learning algorithm which adaptively includes second order information, *Proceedings of the Fourth Dortmund Fuzzy Days* (1994) 55–62.
- [18] D. Ackley, G. Hinton, T. Sejnowski, A learning algorithm for Boltzmann machines, *Cognitive Science* 9 (1985) 147–169.
- [19] E.H.L. Arts, J. Korst, *Simulated Annealing and Boltzmann Machines*, Wiley, New York, 1989.
- [20] T. Rognvaldsson, On Langevin updating in multilayer perceptrons, *Neural Comput.* 6 (1994) 916–926.
- [21] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simulated annealing, *Science* 220 (1983) 671–680.
- [22] A. Corana, M. Marchesi, C. Martini, S. Ridella, Minimizing multimodal functions of continuous variables with the Simulated Annealing algorithm, *ACM Trans. Math. Software* 13 (1987) 262–280.
- [23] H. Szu, Nonconvex optimization by fast simulated annealing, *Proc. IEEE* 75 (1987) 1538–1540.
- [24] P.J.M. Van Laarhoven, E.H.L. Arts, *Simulated Annealing: Theory and Applications*, D. Reidel, Dordrecht, The Netherlands, 1988.
- [25] S.T. Wesslstead, *Neural Network and Fuzzy Logic Applications in C/C++*, Wiley, New York, 1994.
- [26] C. Tsallis, D.A. Stariolo, Generalized simulated annealing, *Physica A* 233 (1996) 395–406.
- [27] R. Hoptroff, T. Hall, Learning by diffusion for multilayer perceptron, *Electron. Lett.* 25 (1989) 531–533.

- [28] M. Gell-Mann, C. Tsallis, (Eds.), *Nonextensive Entropy—Interdisciplinary Applications*, Oxford University Press, New York, 2004, in press.
- [29] A. Gupta, S.M. Lam, Weight decay backpropagation for noisy data, *Neural Networks* 11 (1998) 1127–1137.
- [30] P.M. Murphy, D.W. Aha, UCI Repository of machine learning databases, Department of Information and Computer Science, University of California, Irvine, CA, 1994. <http://www.ics.uci.edu/~mllearn/MLRepository.html>.
- [31] L. Prechelt, PROBEN1—A set of benchmarks and benchmarking rules for neural network training algorithms, Technical Report 21/94, Fakultt fr Informatik, Universitt Karlsruhe, 1994.
- [32] G. Snedecor, W. Cochran, *Statistical Methods*, 8th edition, Iowa State University Press, Iowa, 1989.
- [33] E.K. Blum, Approximation of Boolean functions by sigmoidal networks: Part I: XOR and other two variable functions, *Neural Comput.* 1 (1989) 532–540.
- [34] M. Gori, A. Tesi, On the problem of local minima in backpropagation, *IEEE Trans. Pattern Anal. Mach. Intell.* 14 (1992) 76–85.
- [35] G.D. Magoulas, M.N. Vrahatis, G.S. Androulakis, On the alleviation of the problem of local minima in back-propagation, *Nonlinear Anal.: Theory Methods Appl.* 30 (1997) 4545–4550.