Nonextensive Entropy and Regularization for Adaptive Learning

Aristoklis D. Anastasiadis
School of Computer Science and
Information Systems
Birkbeck College, University of London
Malet Street, London WC1E 7HX, United Kingdom.
E-mail: aris@dcs.bbk.ac.uk

George D. Magoulas
School of Computer Science and
Information Systems
Birkbeck College, University of London
Malet Street, London WC1E 7HX, United Kingdom.
E-mail: gmagoulas@dcs.bbk.ac.uk

***Abstract*— There are many existing supervised learning algorithms based on the technique of gradient descent. A problem with these algorithms is that they occasionally converge to undesired local minimum. This paper builds on the theory of nonextensive statistical mechanics to develop a new adaptive gradient–based learning scheme that applies a sign–based weight adjustment, inspired from the Rprop algorithm, on a perturbed version of the original error function. The perturbations are characterized by the q entropic index of the nonextensive entropy, and their impact is controlled by means of regularization. This approach modifies the error landscape at each iteration allowing the algorithm to explore previously unavailable regions of the error surface, and possibly escape undesired local minima. The performance of the adaptive scheme is empirically evaluated using problems from the UCI Repository of Machine Learning Databases and other classic benchmarks.**

I. Introduction

It is well known that the rapid computation of a global minimum of a general error function is a difficult task because the dimensionality of the search space is high, and there are multitudes of local minima and broad flat regions adjoined with narrow steep ones [10], [26]. First–order methods are the most widely used class of algorithms for supervised learning of neural networks [5]. Among these methods, adaptive stepsize algorithms try to overcome the inherent difficulty of choosing the right stepsize for each problem [15]. 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 [24]. Another class of methods inspired from unconstrained optimization theory use second derivative related information to accelerate the learning process [5], [14], [17], [33]. However, 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 [18]; this is usually the case of neural network training problems [5]. 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 [4], [21], [22], [32].

This paper introduces an adaptive search strategy that aims to alleviate the problem of occasional convergence to local minima in supervised training. Our approach adapts the weights using only information from the sign of a gradient vector, which is calculated on a perturbed error function, and uses adaptive steps along each weight directions. The perturbations are generated from a noise sources that replaces the usual Boltzmann–Gibbs factor used in annealing schedules by the *q–exponential function* of the generalized entropy of nonextensive statistical mechanics [29], [30]. The next section briefly describes annealing schedules in neural networks learning. Next we introduce the new learning scheme. Then results of an empirical evaluation are presented demonstrating the effectiveness of the new scheme in reducing the possibility of convergence to poor local minima. The paper ends with concluding remarks.

II. Annealing Schedules in Neural Learning

Although noise plays a 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 great depth. One of the most famous neural networks model operating with noise is the Boltzmann machine, [1], [2]. 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 [1], [4], [25], have focused mainly on the use of Gaussian distributions. In particular the use of Simulated Annealing has been explored for learning of the Boltzmann machine [1], [2]. Simulated Annealing (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 [12]. In the numerical optimization framework, SA is a procedure that has the capability to move out of regions near local minima [6], [28]. 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 [34].

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 [35]. To alleviate this situation, [4] has suggested to incorporate an annealing schedule in the steepest descent algorithm:

$$w^{k+1} = w^k - \eta \nabla E(w^k) + \rho \cdot c \cdot 2^{-d \cdot k}, \tag{1}$$

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 [12], or the generalized acceptance probability in the generalized SA [31]. Instead, it implements a form of Langevin noise that has been proved quite effective in neural systems learning, [11], [25], and has motivated the development of other methods, such as the *Simulated Annealing Rprop*–SARprop and the SARprop with Restarts–ReSARprop [32].

III. The Formulation of the New Method

In this section we describe briefly the new learning algorithm.

A. Nonextensive Entropy and the Perturbed Error Function

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 [29]:

$$S_q \equiv K \, \frac{1 - \sum_{i=1}^{W} p_i^q}{q - 1} \quad (q \in \mathbb{R}), \tag{2}$$

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 $q = 1$, Equation (2) 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 (2) under appropriate constraints, [29], yields for the canonical ensemble

$$p_i \propto [1 - (1-q)\beta E_i]^{\frac{1}{(1-q)}} \equiv e_q^{-\beta E_i}, \tag{3}$$

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

$$e_q^x \equiv [1 + (1-q)x]^{\frac{1}{(1-q)}} = \frac{1}{[1 - (q-1)x]^{\frac{1}{(q-1)}}} \tag{4}$$

Following the above discussion and inspired by [4], [31], in our method, noise is generated according to a schedule that can be expressed as:

$$Q(T,k) = e_q^{-T(\ln 2) \cdot k} = [1 - (1-q)T(\ln 2) \cdot k]^{\frac{1}{1-q}}, \tag{5}$$

where T is the temperature; k indicates iterations. 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. Thus, noise is introduced by formulating the *perturbed* error function:

$$\tilde{E}(w^k) = E(w^k) + \mu \cdot \sum_{i=1}^{n} \frac{\left(w_i^k\right)^2}{\left[1 + \left(w_i^k\right)^2\right]} \cdot Q(T,k), \tag{6}$$

where $E(w)$ is the batch error function, $\sum_i w_i^2/(1+w_i^2)$ 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 error 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 error landscape is modified during training the search method is allowed to explore regions of the error surface that were previously unavailable. Minimization of (6) requires calculating the gradient of the error function with respect to each weight

$$\tilde{g}_i(w^k) = g_i(w^k) + \mu' \cdot \frac{w_i^k}{\left[1 + \left(w_i^k\right)^2\right]^2} \cdot Q(T,k), \tag{7}$$

where $\mu' > 0$ (in our experiments a fixed value of $\mu' = 0.01$ was used).

B. The Adaptive Learning Scheme

Our method 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 [26]. Various algorithms of this class have been suggested in the literature, such as [15], [19], [20], [24]. Among them the *Resilient Propagation* (Rprop) algorithm is one of the most popular methods [24]. 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.

The proposed adaptive scheme applies a sign–based weight adjustment, inspired by Rprop [24], on the perturbed error function (6) using the gradient term of Equation (7). The direction of the weights are changed by the following procedure:

$$w^{k+1} = w^k - \mathrm{diag}\{\eta_1^k, \ldots, \eta_i^k, \ldots, \eta_n^k\} \, sign\Big(\tilde{g}_i(w^k)\Big), \tag{8}$$

where k indicates iterations; $\mathrm{diag}\{\eta_1, \ldots, \eta_n\}$ defines the $n \times n$ diagonal matrix with elements $\eta_1, \ldots, \eta_n$, and η_i^k $(i = 1, 2, \ldots, n)$ 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 (8), $sign\Big(\tilde{g}_i(w^k)\Big)$ denotes the column vector of the signs of the components of $\tilde{g}_i(w^k)$; $\tilde{g}(w)^\top = \Big(\tilde{g}_1(w), \ldots, \tilde{g}_n(w)\Big)$ defines the transpose of the gradient $\nabla E(w)$ of the sum-of-squared-differences error function E at w;

The Adaptive Learning Scheme (ALS) works by adapting the learning rates following a rational similar to Rprop algorithm: if the sign of the gradient of the perturbed error function (7) has remained the same then η_i^k is calculated by the following equation:

$$\begin{aligned} &if \quad \Big(\tilde{g}_i(w^{k-1}) \cdot \tilde{g}_i(w^k) > 0\Big) \quad then \\ &\eta_i^k = min\Big(\eta_i^{k-1} \cdot \eta^+, \Delta_{max}\Big) \end{aligned} \tag{9}$$

where $0 < \eta^- < 1 < \eta^+$, Δ_{max} is the stepsize upper bound. When the gradient is zero then the update value is multiplied by the learning rate η_i^k which is remaining the same, i.e.

$$if \quad \Big(\tilde{g}_i(w^{k-1}) \cdot \tilde{g}_i(w^k) = 0\Big) \quad then \quad \eta_i^k = \eta_i^{k-1}, \tag{10}$$

Finally, if the gradient direction has changed we introduce an additional condition in order to avoid using relatively small weight adjustments.

$$\begin{aligned} &if \quad \Big(\tilde{g}_i(w^{k-1}) \cdot \tilde{g}_i(w^k) < 0\Big) and \Big(\eta_i^{k-1} < \rho \cdot Q^2(T,k)\Big) \quad then \\ &\eta_i^k = max\left(\eta_i^{k-1}\eta^- + 2c\rho \cdot Q^2(T,k), \Delta_{min}\right), \end{aligned} \tag{11}$$

where $0 < \rho < 1$, Δ_{min} is the stepsize lower bound, and $c \in (0, 1)$ is a random number. If only the sign of the gradient has changed, then the following rule is used:

$$\begin{aligned} &if \quad \Big(\tilde{g}_i(w^{k-1}) \cdot \tilde{g}_i(w^k) < 0\Big) \quad then \\ &\eta_i^k = max\Big(\eta_i^{k-1} \cdot \eta^-, \Delta_{min}\Big) \end{aligned} \tag{12}$$

We have followed the recommendations of [24] in setting the parameters: (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 $\Delta_{min} = 10^{-6}$.

Lastly, it is important to mention that the behavior of the learning scheme can be more stochastic or deterministic depending on the trade off between T and q. Below, a simple problem is used to visualize the behavior of the *Adaptive Learning Scheme*–(ALS) for different values of T keeping q fixed. It is a single node with two weights and a logistic activation function. The error landscape of Figure 1 and Figure 2 has a global minimum and two local minima. The top rows of Figures 1 and 2 show that Rprop converges to the local minimizer from two different initial weights. We have applied the adaptive learning scheme with $q = 1.2$ for the following temperatures: $T = 0.01$ (Figures 1 and 2, second row), $T = 0.001$ (Figures 1 and 2, third row), $T = 0.0001$ (Figures 1 and 2, bottom). The ALS escapes the region around the local minimum, and converges to the global minimizer located at the center of the contour plot in all cases. The value of T influences the shape of the ALS trajectory. Small values generate more stochastic paths, while larger values lead to more deterministic behavior. Best results for this problem are achieved by setting $T = 0.01$.

In a more difficult problem, learning the XOR parity, a typical run for the Rprop method, Sarprop and the ALS is shown in Figure 3. As we can notice from Figure 3, starting with the same initial weights and learning parameters in the XOR problem, Rprop got stuck in a local minimum with higher error function value while the Adaptive Learning Scheme (ALS) successfully converges to a feasible solution ($E(w) \leqslant 10^{-16}$) after 200 iterations and the SARprop approximately in 400 iterations.

IV. Experimental Study

In this section, we evaluate the performance of the ALS 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 [16], 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 [24] and [32] were adopted for setting the learning parameters of Rprop and SARprop respectively. The parameters of the ALS 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 $T = 0.1$.

Below, we report results from 150 independent trials for four UCI problems. These 150 random weight initializations are the same for the three learning algorithms, and the training and testing sets were created according to *Proben1* [23]. The statistical significance of the results has been analyzed using the Wilcoxon test [27]. 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 [16], [23]. The data set consists of 120 examples and the test set of 30 examples. Following [32], an 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.

The second benchmark is the *breast cancer diagnosis* problem which classifies a tumor as benign or malignant based

TABLE I

AVERAGE PERFORMANCE IN THE IRIS, CANCER, DIABETES, AND THYROID PROBLEMS

	Iris		**Cancer**		**Diabetes**		**Thyroid**	
Algorithm	IT	$CONV.$	IT	$CONV.$	IT	$CONV.$	IT	$CONV.$
Rprop	1340	56	296	93	650	86	812	84
SARprop	980	96	420	97	440	97	818	90
ALS	1050	100	119	100	272	100	361	100

TABLE II

AVERAGE PERFORMANCE IN THE PARITY–3, PARITY–4 AND PARITY–5 PROBLEMS

	Parity3		**Parity4**		**Parity5**	
Algorithm	IT	$CONV.$	IT	$CONV.$	IT	$CONV.$
Rprop	866	79	1345	42	330	67
SARprop	848	94	1186	64	610	98
ALS	399	100	960	100	190	100

on 9 features [16], [23]. We have used an FNN with 9–4–2–2 nodes (a total of 56 weights) as suggested in [23]. In this case, SARprop exhibits the lowest percentage of convergence within the 2000 iterations.

The *diabetes1* benchmark is a real-world classification task which concerns deciding when a Pima Indian individual is diabetes positive or not [16], [23]. 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 $E \leq 0.14$ within 2000 iterations. Lastly, the *thyroid1* problem, [16], [23], uses a 21–4–3 nodes FNN, suggested by [32], 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 I gives the average performance of the three algorithms in the pattern recognition problems. It shows the average performance in terms of: iterations to converge to the error target (IT) and success of convergence to the target, within 2000 iterations ($CONV.$, out of the 150 runs); all results satisfy the Wilcoxon test indicating statistical significance of the ALS results over the other methods. As shown in Table I the new method outperforms the other methods in the number of iterations required to reach a suitable solution, and converges in all cases.

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 [3], [8]. This class includes the various parity–N problems, which are considered as classic benchmarks [13], [21], [32], [33]. 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). Following the recommendations of [32] we have used the architectures: 3–3–1 for the parity–3, 4–6–1 for the parity–4, 5–7–1 for the parity–5. The results are presented in Table II. Figure 4 gives a typical example of algorithms' error reduction process. Starting from the same initial conditions, the Rprop and SARprop converge to a local minimizer. Although both SARprop and ALS escape from the local minimum, ALS converges to a feasible solution much faster than SARprop.

In our experiments ALS has shown improved convergence speed and stability, which affect by no means its generalization capability.

V. CONCLUDING REMARKS

In this paper we propose a new adaptive learning scheme that combines deterministic and stochastic search steps by employing a different adaptive stepsize for each weight. ALS applies for the error function a form of noise that is characterized by the nonextensive entropic index q. This allows to modify the error surface during training so that exploration of new regions of the error landscape is achieved. Comparison with two other popular learning methods, namely the Rprop and the SARprop, show increased convergence rates and reliable neural learning in the cases tested. One direction for our future work is to conduct experiment exploring the influence of the entropic index q and temperature on the convergence speed and stability of the method.

VI. ACKNOWLEDGMENTS

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

REFERENCES

[1] D. Ackley. G. Hinton and T. Sejnowski, A learning algorithm for Boltzmann machines. *Cognitive Science*, 9, 147–169, 1985.

[2] E. H. L. Arts and J. Korst, *Simulated Annealing and Boltzmann Machines.* New York: Wiley, 1989.

[3] E.K. Blum, Approximation of Boolean functions by sigmoidal networks: Part I: XOR and other two variable functions. *Neural Computation*, 1, 532–540, 1989.

[4] R. M. Burton and G. J. Mpitsos, Event dependent control of noise enhances learning in neural networks. *Neural Networks*, 5, 627-637, 1992.

[5] R. Battiti, First– and second–order methods for learning: between steepest descent and Newton's method. *Neural Computation*, 4, 141–166, 1992.

[6] A. Corana, M. Marchesi, C. Martini, and S. Ridella, Minimizing multimodal functions of continuous variables with the Simulated Annealing algorithm. *ACM Trans. Math. Soft.*, 13, 262–280, 1987.

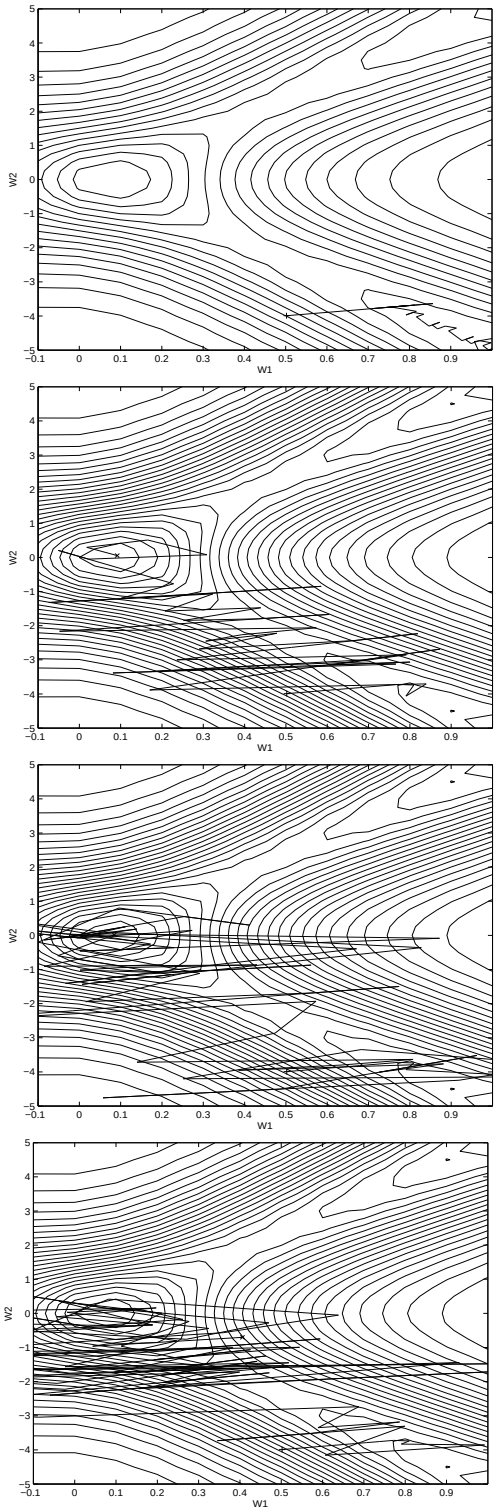

Fig. 1. Starting from the same initial weights, the trajectory of the Rprop converges to a local minimizer (top), whilst the trajectory of ALS converges to the global minimum (3 different values for the Temperature are shown – see text for details).

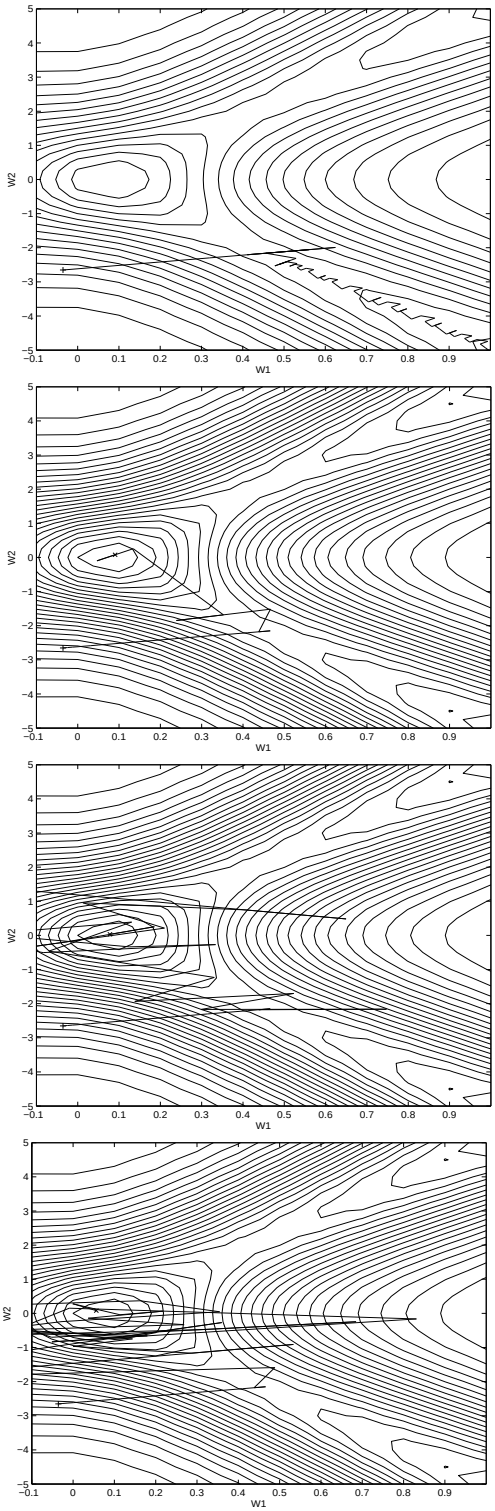

Fig. 2. Starting from the same initial weights, the trajectory of the Rprop converges to a local minimizer (top) , whilst the trajectory of ALS converges to the global minimum (3 different values for the Temperature are shown – see text for details).

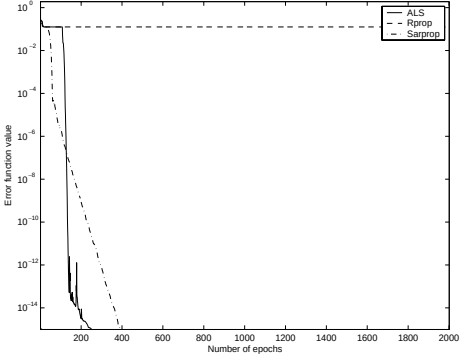

Fig. 3. Typical learning error curve for the XOR function

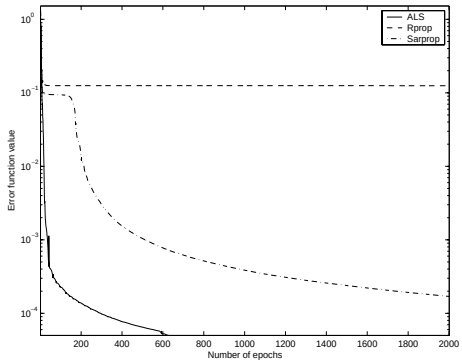

Fig. 4. Typical learning error curve for the Parity–3 problem

[7] M. Gell-Mann and C. Tsallis, eds., *Nonextensive Entropy – Interdisciplinary Applications*, Oxford University Press, New York, 2004, in press.

[8] M. Gori and A. Tesi, On the problem of local minima in backpropagation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14, 76–85, 1992.

[9] A. Gupta and S. M. Lam, Weight decay backpropagation for noisy data. *Neural Networks*, 11, 1127–1137.

[10] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing Company, 1994.

[11] R. Hoptroff and T. Hall, Learning by diffusion for multilayer perceptron. *Electronic Letters*, 25, 531–533, 1989.

[12] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi, Optimization by simulated annealing. *Science*, 220, 671–680, 1983.

[13] G.D. Magoulas, M.N. Vrahatis, and G.S. Androulakis, On the alleviation of the problem of local minima in back–propagation. *Nonlinear Analysis: Theory, Methods and Applications*, 30, 4545–4550, 1997.

[14] G. D. Magoulas, M. N. Vrahatis, T. N. Grapsa, and G. S. Androulakis, Neural network supervised training based on a dimension reducing method. In: S.W. Ellacot, J.C. Mason, and I.J. Anderson (eds.), *Mathematics of Neural Networks: Models, Algorithms and Applications*, 245–249, Kluwer, 1997.

[15] G. D. Magoulas, M. N. Vrahatis, and G.S. Androulakis, Improving the Convergence of the Backpropagation Algorithm Using Learning Rate Adaptation Methods. *Neural Computation*, 11, 1769–1796, 1999.

[16] P.M. Murphy and D.W. Aha, UCI Repository of machine learning databases, Irvine, CA: University of California, Department of Information and Computer Science, 1994. http://www.ics.uci.edu/ mlearn/MLRepository.html.

[17] Møller, M.F., A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6, 525–533, 1993.

[18] Nocedal J., Theory of algorithms for unconstrained optimization. *Acta Numerica*, 1, 199–242, 1992.

[19] M. Pfister and R. Rojas, Speeding–up backpropagation– A comparison of orthogonal techniques. *Proc. of the Joint Conference on Neural Networks*, Nagoya, Japan, 517–523, 1993.

[20] M. Pfister and R. Rojas, Qrprop–a hybrid learning algorithm which adaptively includes second order information. *Proc. of the 4th Dortmund Fuzzy Days*, 55–62, 1994.

[21] V.P. Plagianakos, G.D. Magoulas and M.N. Vrahatis, Learning in multilayer perceptrons using global optimization strategies. *Nonlinear Analysis: Theory, Methods and Applications*, 47, 3431–3436, 2001.

[22] V.P. Plagianakos, G.D. Magoulas and M.N. Vrahatis, Supervised training using global search methods. N. Hadjisavvas and P. Pardalos (eds.), *Advances in Convex Analysis and Global Optimization, vol. 54, Noncovex Optimization and its Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 421-432, 2001.

[23] 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.

[24] M. Riedmiller and H. Braun, A direct adaptive method for faster backpropagation learning: The Rprop algorithm. *Proc. International Conference on Neural Networks*, San Francisco, CA, 586-591, 1993.

[25] T. Rögnvaldsson, On Langevin updating in multilayer perceptrons. *Neural Computation*, 6, 916–926, 1994.

[26] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, Learning internal representations by error propagation. D.E. Rumelhart, J.L. McClelland (eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition 1*, MIT Press, 318–362, 1986.

[27] G. Snedecor and W. Cochran, *Statistical Methods*, Iowa State University Press, 8th edition, 1989.

[28] H. Szu, Nonconvex optimization by fast simulated annealing. *Proceedings of IEEE*, 75, 1538–1540, 1987.

[29] C.Tsallis, Possible Generalization of Boltzmann-Gibbs Statistics. *J. Statistical Physics*, 52(1–2),479–487, 1988.

[30] C. Tsallis, R.S. Mendes and A.R. Plastino, *Physica A*, 261, 534, 1998.

[31] C. Tsallis and D. A. Stariolo, Generalized Simulated Annealing. *Physica A*, 233, 395–406, 1996.

[32] N. K. Treadgold and T. D. Gedeon, Simulated Annealing and Weight Decay in Adaptive Learning: The SARPROP Algorithm. *IEEE Tr. Neural Networks*, 9, 4, 662–668, 1998.

[33] Van der Smagt P.P., Minimization Methods for training feedforward neural networks. *Neural Networks*, 7, 1–11, 1994.

[34] P. J. M. Van Laarhoven and E. H. L. Arts, *Simulated Annealing: Theory and Applications*. Dordrecht, The Netherlands: D. Reidel, 1988.

[35] S.T. Weslstead, *Neural network and fuzzy logic applications in C/C++*, Wiley, 1994.