Eur. Phys. J. B **50**, 277–283 (2006)
DOI: 10.1140/epjb/e2006-00137-6

THE EUROPEAN PHYSICAL JOURNAL B

Evolving Stochastic Learning Algorithm based on Tsallis entropic index

A.D. Anastasiadis[a] and G.D. Magoulas[b]

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

Received 20 October 2005 / Received in final form 12 December 2005
Published online 12 April 2006 –

Abstract. In this paper, inspired from our previous algorithm, which was based on the theory of Tsallis statistical mechanics, we develop a new evolving stochastic learning algorithm for neural networks. The new algorithm combines deterministic and stochastic search steps by employing a different adaptive stepsize for each network weight, and applies a form of noise that is characterized by the nonextensive entropic index q, regulated by a weight decay term. The behavior of the learning algorithm can be made more stochastic or deterministic depending on the trade off between the temperature T and the q values. This is achieved by introducing a formula that defines a time-dependent relationship between these two important learning parameters. Our experimental study verifies that there are indeed improvements in the convergence speed of this new evolving stochastic learning algorithm, which makes learning faster than using the original Hybrid Learning Scheme (HLS). In addition, experiments are conducted to explore the influence of the entropic index q and temperature T on the convergence speed and stability of the proposed method.

PACS. 07.05.Mh Neural networks, fuzzy logic, artificial intelligence – 87.18.Sn Neural networks – 05.10.-a Computational methods in statistical physics and nonlinear dynamics

1 Introduction

Neural networks are widely used in many classification applications. One of the major key concept in neural networks is the interaction between microscopic and macroscopic phenomena. The goal of Feedforward Neural Network (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 (P) in a training set [1]:

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

where, net_j^L is for the jth node in the Lth layer ($j = 1,\ldots,n_L$), the sum of its weighted inputs. θ_j^L denotes the bias of the jth node ($j = 1,\ldots,N_l$) at the Lth layer ($l = 2,\ldots,L$), and w denotes the weights w in the network. This equation formulates the energy function, called *error function*, to be minimized, in which $t_{j,p}$ specifies the desired response at the jth output node for the example p and $y_{j,p}^L$ is the output of the jth node at layer L that depends on the weights w of the network, and σ is a nonlinear activation function, such as the well known logistic function $\sigma(x) = (1+e^{-x})^{-1}$. The problem of finding the global minimum of such a complex cost function, which possesses a large number of local minima, is considered very difficult task [1].

Statistical mechanics methods have been applied successfully to the study of neural network models of associative memory [2]. These models are biologically plausible and can be trained very quickly in some cases, compared with the popular neural networks such as multi-layered perceptron, which have been shown to work satisfactorily. However, this model of associative memory has still drawbacks as learning gets stuck at local minima. A variety of global optimization algorithms have also been introduced over the years to overcome the problem of local minima. One of the most popular methods is the Simulated annealing [3]. It uses Boltzmann-Gibbs (BG) statistics at two different steps, namely at the *visitation step*, which uses a Gaussian distribution, and at the *acceptance step*, that uses the Boltzmann factor [4,5].

[a] A.D. is also affiliated with London Knowledge Lab., University of London, 23–29 Emerald Street, WC1N 3QS, London, UK; e-mail: aris@dcs.bbk.ac.uk
[b] e-mail: gmagoulas@dcs.bbk.ac.uk

Another approach is based on the use of noise models. Attempts to explore the benefits of introducing noise during learning have been based on the use of Gaussian distributions [4,6,7]. One of the most famous neural model operating with noise is the Boltzmann machine [4,5], inspired by the Boltzman-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 $K > 0$. Also, a form of Langevin noise has been proved quite effective for neural learning, and has motivated the development of other methods, such as the *Simulated Annealing Rprop*-SARprop [8].

The next section briefly describes the recently proposed hybrid learning scheme [9], and then we introduce the proposed evolving stochastic learning algorithm. Next, results of an empirical evaluation are presented, demonstrating the effectiveness of the new scheme in locating acceptable solutions. The paper ends with discussion and concluding remarks.

2 The Evolving Stochastic Learning Algorithm

The recently proposed Hybrid Learning Scheme (HLS) [9] has been built on ideas from global search methods. It is worth noting that global search algorithms possess strong convergence properties. However, these methods are computationally expensive [8]. To alleviate this situation hybrid schemes for neural networks learning have been developed in an attempt to achieve improved convergence rates compared to the standard global optimization, and in some cases even maintain the guarantee of convergence to a global minimizer [6]. HLS is a hybrid training algorithm that employs a different adaptive stepsize for each weight. HLS avoids slow convergence in the flat directions and oscillations in the steep directions, and exploits the parallelism inherent in the evaluation of learning error $E(w)$ and gradient $\nabla E(w)$ by the Resilient Back-Propagation (Rprop) algorithm [10]. Inspired by [6,11], in the HLS, noise has been introduced in the training procedure according to a nonextensive schedule [9]. The HLS also applies the sign-based weight adjustment of Rprop [10], on the perturbed energy function (for a detailed description see [9]).

The new Evolving Stochastic Learning Algorithm (ESLA) introduces noise, as in HLS. The noise source is characterized by the nonextensive entropic index q. In particular, the principles of the new method are using the notion of nonextensive entropy, which has been defined as [12]:

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

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 $q = 1$, (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 [12], 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*

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

In this method, like in the HLS, noise is generated according to a schedule:

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

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* energy function:

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

where $E(w)$ is the 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. The energy landscape is modified during training so the search method is allowed to explore regions of the energy surface that were previously unavailable. Minimization of equation (6) requires calculating the gradient of the energy with respect to each weight

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

where $g_i(w^k)$ is the gradient of the energy $E(w^k)$, with respect to each weight, and $\mu' > 0$ (in our experiments a fixed value of $\mu' = 0.01$ was used). The proposed evolving stochastic hybrid scheme applies a sign-based weight adjustment, similar to HLS [9], on the perturbed energy function (6) using the gradient term of equation (7). Also the learning rates are adapted by Rprop learning procedure [10].

In our approach the weight adjustment is given by the following equation:

$$w^{k+1} = w^k - \tau^k \,\mathrm{diag}\{\eta_1^k, \ldots, \eta_i^k, \ldots, \eta_n^k\}\,\mathrm{sign}(\tilde{g}_i(w^k)), \quad k = 0,1,\ldots \tag{8}$$

where $\mathrm{sign}(\tilde{g}_i(w^k))$ denotes the column vector of the signs of the components of $\tilde{g}(w^k) = (\tilde{g}_1(w^k), \tilde{g}_2(w^k), \ldots, \tilde{g}_n(w^k))$, $\tau^k > 0$, η_m^k $(m = 1,2,\ldots,i-1,i+1,\ldots,n)$ are small positive real numbers generated by Rprop's learning rates schedule.

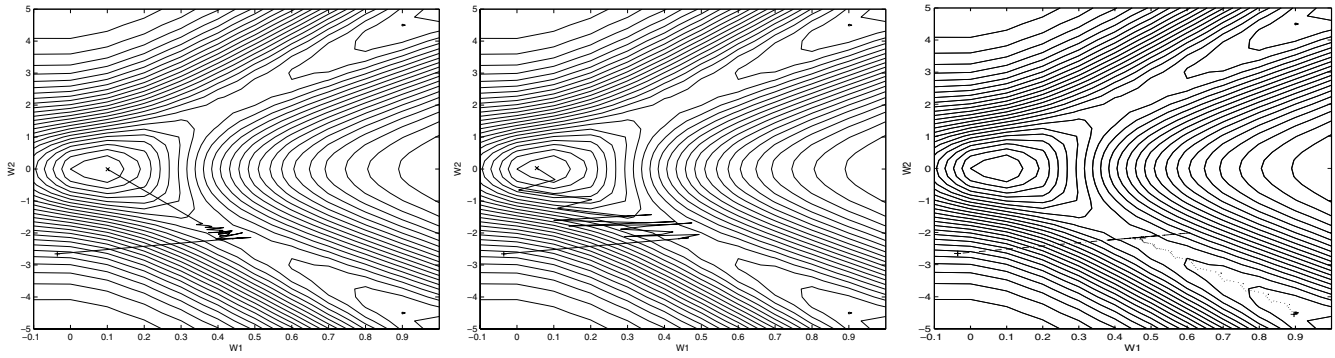

Fig. 1. Weights trajectories of the Evolving Stochastic Learning Algorithm-ESLA (left), the Hybrid Learning Scheme-HLS (center), and the Rprop (right).

Moreover, an additional condition, like in the HLS, is introduced in order to avoid using relatively small weight adjustments

$$\text{if} \quad \left(\eta_i^{k-1} < \rho \cdot Q^2(T,k)\right) \quad \text{then}$$
$$\eta_i^k = max\left(\eta_i^{k-1}\eta^- + 2c\rho \cdot Q^2(T,k), \Delta_{min}\right), \qquad (9)$$

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

Lastly, inspired from previous work [11], we apply a cooling procedure. This defines the relationship between T and q values. The application of cooling helps to regulate the training algorithm, making it more deterministic. This new *Evolving Stochastic Learning Algorithm*-ESLA behaves in a more stochastic way, during the initial stages, and then becomes more deterministic as the number of iterations increases. Thus, when we are close to the minimizer, the algorithm hopefully will avoid oscillations and converge faster. The cooling procedure is described by the next equation:

$$T = T_0 \cdot \left[\frac{2^{q-1}-1}{(1+k)^{q-1}-1}\right], q > 1 \qquad (10)$$

where T_0 is the initial temperature, T is the current temperature, k is the number of iterations, and q is the Tsallis entropic index.

The challenge is to cool the temperature the quickest we can, but still having the ability to converge to global minimum with high probability. The standard simulated annealing (SA) is one method to achieve this goal. However, the cooling procedure is computationally expensive. An efficient alternative cooling method is the fast simulated annealing (FSA) [13]. The temperature is now allowed to decrease like the inverse of time, which makes the entire cooling procedure quite more efficient. Generalised Simulated annealing (GSA) [11] is a generalization of the previous methods, which performs better than previous annealing algorithms for many problems and applications. In neural networks applications we are mainly interested in accelerating the learning speed with no affect in generalization. The cooling procedure based on GSA satisfies these two targets and contributes positively to the performance of the ESLA. This cooling procedure makes the temperature to decrease as a power-law of time, in contrast to the much slower decrease (logarithmic in time) of the $q = 1$ case.

Below, a simple problem is used to visualize the behavior of the ESLA and compare it with the HLS, and the Rprop algorithm. The energy landscape of Figure 1 has a global minimum and two local minima. Figure 1 shows that under the same initial conditions, both of the ESLA and the HLS escape the saddle point and the valley that leads to a local minimum, while the ESLA converges faster than HLS with fewer oscillations (Fig. 1, left), and the Rprop algorithm converges to the local minimizer (Fig. 1, right).

3 Experimental study

We have evaluated the performance of the ESLA and compared it with the Rprop, and the HLS algorithms. The statistical significance of the results has been analyzed using the Wilcoxon test [14]. This is a nonparametric method that is considered an alternative to the paired t-test. All statements in the tables reported below, refer to a significance level of 0.05. Statistically significant cases are marked with $(+)$, while $(-)$ shows the cases that don't satisfy the significance level. Moreover, the following terms are used: *Epochs* is the number of iterations to converge to the error target; *Convergence* denotes the success of convergence to the error target within 2000 iterations; *Generalization* is the percentage of correctly classified test examples. Finally, for all the problems we have set the initial temperature to $T = 2$ for training using the ESLA. By keeping constant the initial temperature we found the optimal value for the Tsallis entropic index q. The parameters of the HLS were set to the same values as in the ESLA for all experiments in an attempt to test the robustness of the method in different types of problems: the temperature is equal to the initial temperature $T = 2$, and the q is set to different values depending on the problem, (i.e. in cancer $T = 2$ and $q = 1.7$, while in diabetes is $q = 1.6$). Below, we report results from 300 independent trials. These 300 random weight initializations have been the same for the three learning algorithms.

3.1 Benchmarks from the UCI repository

The data sets for the cancer1, diabetes1, thyroid1 problems were used as supplied on the PROBEN1 website. PROBEN1 provides explicit instructions for creating

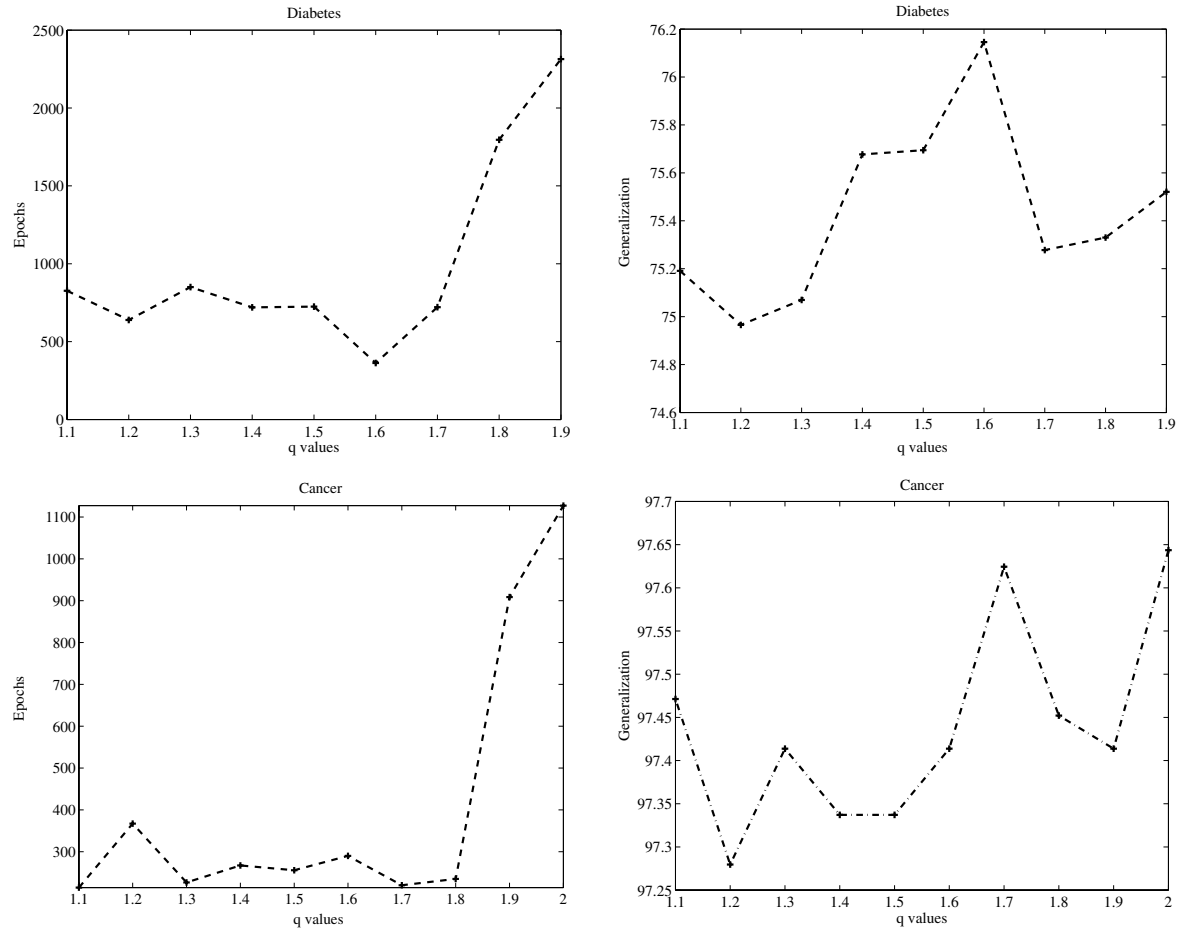

Fig. 2. Optimal q based on Epochs, and Generalization for the diabetes and cancer problems.

Table 1. Comparison of algorithms performance in the Diabetes and Cancer problems for the converged runs.

	Diabetes			Cancer		
Algorithm	*Epochs*	*Generalization*	*Convergence*	*Epochs*	*Generalization*	*Convergence*
Rprop	700 (+)	75.2 (%) (+)	86 (%) (+)	287 (+)	97.2(%) (−)	94(%) (+)
HLS	570 (+)	75.8 (%) (+)	94 (%) (−)	230 (+)	97.4(%) (−)	96(%) (+)
ESLA	480	76.2 (%)	95 (%)	195	97.4(%)	99(%)

training and testing sets and choosing network architectures for many problems [15]. The partitioning is 50% of the full data is used as training set, then the next 25% of the dataset is used as validation set, and the remaining 25% as testing set. The *diabetes1* benchmark is a real-world classification task which concerns deciding when a Pima Indian individual is diabetes positive or not [16,15]. The Proben1 collection suggests a 8–2–2–2 FNN. The termination criterion is $E \leq 0.14$ within 2000 iterations. In order to find the best value for the initial temperature and the Tsallis entropic index q, we performed 30 different runs. Figure 2 shows the ESLA's performance for an initial temperature $T = 2$ and different q values. Judging from Figure 2 the best value for $q = 1.6$, and $T = 2$. Table 1 shows that the Rprop algorithm converges many times in local minima. The new stochastic learning algorithm overcomes this problem in most of the cases. The cooling procedure seems to have a positive impact on the learning speed of the algorithm. The second benchmark is the *breast cancer diagnosis* problem which classifies a tumor as benign or malignant based on 9 features [16,15]. We have used an FNN with 9–4–2–2 nodes, as suggested in [15], and a termination criterion of $E \leq 0.02$. Figure 2 shows the best values of these two important training parameters. As we can observe from this figure, a value of the $q = 1.7$ gives the best results in terms of both learning speed and generalization. The comparative results are presented in Table 1.

The third benchmark problem is the *thyroid1*, which is not a permutation of the original data, but retains the original order instead [16,15]. The data set consists of 3600 patterns. The termination criterion is $E \leq 0.0036$. The Tsallis entropic index q in this problem is again $q = 1.7$ (see Fig. 3). The experimental results that we obtained are presented in Table 2.

Table 2. Comparison of algorithms performance in the Thyroid and Yeast problems for the converged runs.

	Thyroid			Yeast		
Algorithm	*Epochs*	*Generalization*	*Convergence*	*Epochs*	*Generalization*	*Convergence*
Rprop	780 (+)	98.2 (%) (−)	81.3 (%) (+)	930 (+)	61.6 (%) (−)	98 (%) (−)
HLS	590 (+)	98.1 (%) (−)	94.0 (%) (−)	590 (+)	61.4 (%) (−)	100 (%) (−)
ESLA	500	98.0 (%)	95.3 (%)	490	61.5 (%)	100 (%)

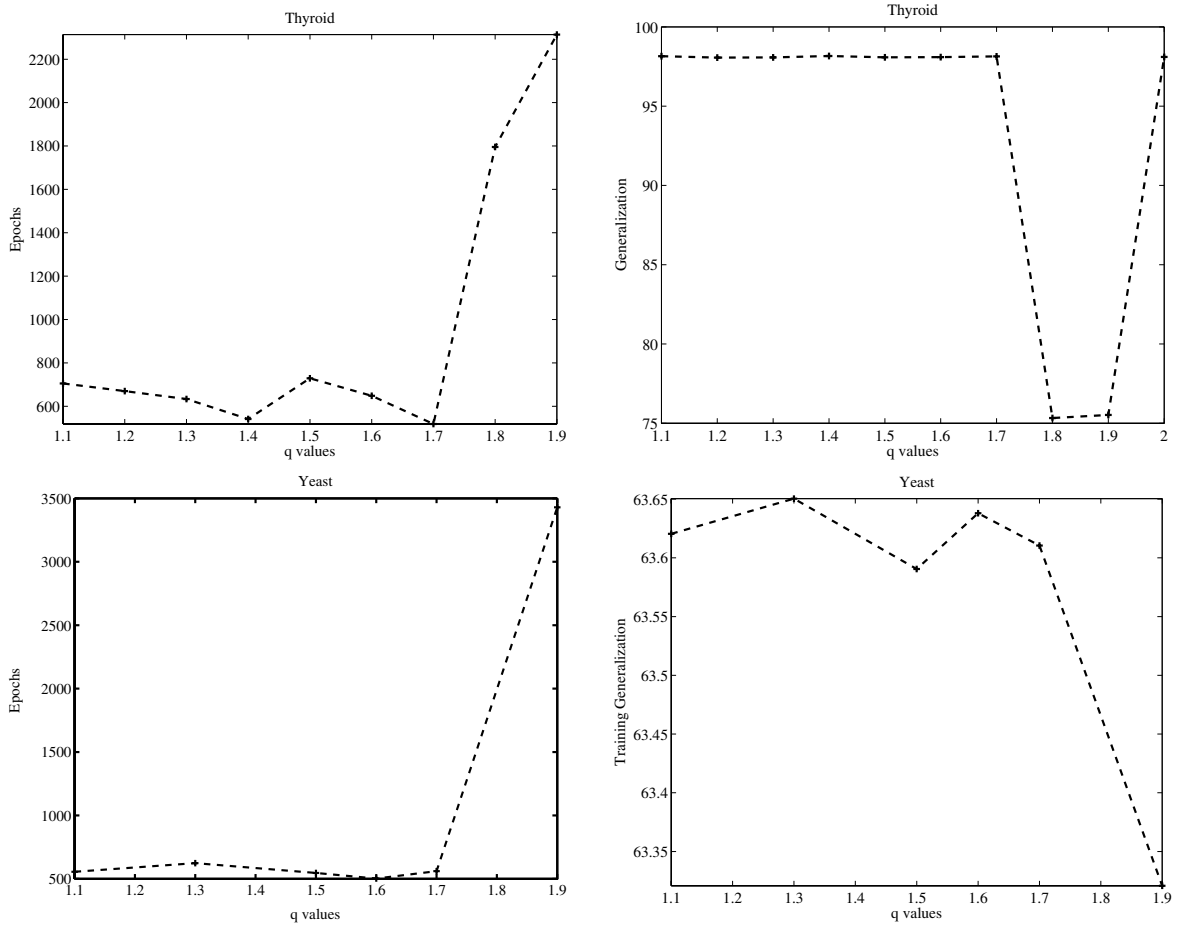

Fig. 3. Optimal q based on Epochs, and Generalization for the thyroid and Yeast problems.

3.2 Prediction of localisation sites of the Yeast proteins

The study of protein localization is considered very useful in the post-genomics and proteomics era, as it provides information about each protein that is complementary to the protein sequence and structure data [17]. One of the most thoroughly studied single-cell organisms is the eukaryote Saccharomyces cerevisiae, also called Yeast. It has rapid growth rate and very simple nutritional requirements [18]. The Yeast dataset is 1484 proteins labeled according to 10 sites [19]. Yeast proteins are organized as in [16]. The most suitable architecture for this problem, as suggested by [20], is an 8–16–10 FNN architecture. A termination criterion of $E \leq 0.05$ within 2000 iterations (*Epochs*) is used. The evaluation method that we have employed to estimate the accuracy of the methods was a 10-fold cross validation following the guidelines of [19, 20]. The proportion of the number of the patterns for all the classes is equal in each partition, as this procedure provides more accurate results than a plain cross validation does [21]. Figure 3 gives an overview of the experiments conducted in order to choose the best value of q for this problem. A value of $q = 1.6$ was applied as this gave the best results in terms of learning speed and generalization. Table 2 shows the experimental results for this difficult problem.

3.3 Boolean function approximation problems

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 [22]. This class includes the XOR problem, and the parity-3 problem, which

Table 3. Comparison of algorithms performance in the XOR and Parity 3 problems for the converged runs.

	XOR			Parity 3		
Algorithm	*Epochs*	*Generalization*	*Convergence*	*Epochs*	*Generalization*	*Convergence*
Rprop	120 (+)	100 (%) (−)	59 (%) (+)	877 (+)	100 (%) (−)	74 (%) (+)
HLS	80 (+)	100 (%) (−)	68 (%) (−)	430 (+)	100 (%) (−)	78 (%) (+)
ESLA	70	100 (%)	64 (%)	390	100 (%)	81 (%)

is considered as classic benchmarks [9,8]. The adopted architectures for the XOR problem is a 2–2–1, and the error target was set to $E \leq 10^{-5}$. A 3–3–1 architecture was used for the parity-3 problem. The error target for parity-3 problem was set to $E \leq 5 \times 10^{-5}$. The activation function for this problem is the tansig function. These target values are considered low enough to guarantee convergence to a "global" solution.

By applying the same procedure as before, the best q entropic index value for the XOR problem is $q = 2.1$, and for the parity 3 problem is $q = 1.1$ with initial temperature $T = 2$. Table 3 shows that the ESLA outperforms in convergence speed. The HLS achieves the best Convergence success on XOR problem. However, the ESLA has better convergence performance compared to Rprop.

4 Discussion and concluding remarks

A recently introduced training algorithm, the hybrid learning scheme-HLS achieves generally very good and reliable performance, and improved learning speed compared to the Rprop algorithm. In this paper, we proposed a new evolving stochastic learning scheme, which constitutes an efficient improvement of the HLS algorithm that is built on a theoretical basis. The ESLA combines deterministic and stochastic search by employing a different adaptive stepsize for each weight, and a form of noise that is characterized by the nonextensive entropic index q. An adaptive formula that introduces a relationship between the T and q was applied. Our experimental study showed that there is a range of q values ($1.1 < q < 2.3$) that gives good performance for the new learning scheme.

In the previous tables, results are based only on the converged runs. Therefore, we don't have the actual performance description of the tested algorithms (i.e. in thyroid problem the Rprop algorithm achieves the best mean generalization success. However, its convergence success is the worst within the tested algorithms. Therefore, the convergence results present the Rprop's generalization for the $0.813 \cdot 300 = 244$ runs out of 300, while the mean generalization success of ESLA is based on $0.953 \cdot 300 = 286$ runs out of 300). In this case it is better to have results for more runs (i.e. patients) although the generalization success is slightly worse. In order to have better view of the overall performance of the tested algorithms, we introduce the parameter *Performance*, which is defined as follows: $Performance = \frac{(Convergence) \times (Generalisation)}{100}$. Thus,

Table 4. Summary of the results in terms of the algorithms' performance.

Performance		Algorithms	
Problems	Rprop (%)	HLS (%)	ESLA (%)
Diabetes	64.7	71.2	72.4
Cancer	91.4	93.5	96.4
Thyroid	79.8	92.3	93.6
Yeast	60.3	61.4	61.5
XOR	59.0	68.0	64.0
Parity-3	74.0	78.0	81.0

Table 4 gives a summary of our results from this perspective for all the tested algorithms.

Further testing is of course necessary to fully explore the advantages and identify possible limitations of this cooling evolving scheme. Moreover, exhaustive testing of the new method in other classes of problems will be done. We will also investigate the performance of ESLA in a restarting mode. Finally, we are going to explore further the properties of Tsallis entropy into Optimization methods in Artificial Intelligence applications.

Aristoklis Anastasiadis would like to thank Dr. G. Kaniadakis and would also like to address special thanks to Prof. Constantino Tsallis for very helpful discussions related to this work, during his stay as research visitor at the Santa Fe Institute.

References

1. S. Haykin, *Neural Networks: A Comprehensive Foundation* (Macmillan College Publishing Company, 1994)
2. G. Gyorgyi, Physics Reports **342**, issue 4–5, 263 (2001)
3. S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Science **220**, 671 (1983)
4. D. Ackley, G. Hinton, T. Sejnowski, Cogn. Sci. **9**, 147 (1985)
5. E.H.L. Arts, J. Korst, *Simulated Annealing and Boltzmann Machines* (Wiley, New York, 1989)
6. R.M. Burton, G.J. Mpitsos, Neural Networks **5**, 627 (1992)
7. T. Rögnvaldsson, Neural Computation **6**, 916 (1994)
8. N.K. Treadgold, T.D. Gedeon, IEEE Tr. Neural Networks **9**, 4, 662 (1998)

9. A.D. Anastasiadis, G.D. Magoulas, Physica A **344**, 372 (2004)
10. M. Riedmiller, H. Braun, *A direct adaptive method for faster backpropagation learning: The Rprop algorithm*, *Proc. Int. Conf. Neur. Net.* (San Francisco, CA, 1993), pp. 586–591
11. C. Tsallis, D.A. Stariolo, Physica A **233**, 395 (1996)
12. C. Tsallis, J. Stat. Phys. **52**, 479 (1988)
13. H. Szu, *Nonconvex optimization by fast simulated annealing*, *Proceedings of IEEE*, Vol. **75** (1987), pp. 1538–1540
14. G. Snedecor, W. Cochran, *Statistical Methods*, 8th edn. (Iowa State University Press, 1989)
15. L. Prechelt, *PROBEN1–A set of benchmarks and benchmarking rules for neural network training algorithms*, Technical report 21/94, Fakultät für Informatik, Universität Karlsruhe, 1994
16. P.M. Murphy, D.W. Aha, UCI Repository of machine learning databases, `http://www.ics.uci.edu/mlearn/MLRepository.html`, 1994
17. M.V. Boland, R.F. Murphy, IEEE Engineering in Medicine and Biology, Sept./Oct. (1999), pp. 115–119
18. H. Lodish, A. Berk, S.L. Zipursky, P. Matsudaira, D. Baltimore, J. James Darnell, *Molecular Cell Biology*, 5th edn. (Freeman, 2003)
19. P. Horton, K. Nakai, *Better Prediction of Protein Cellular Localization Sites with the k Nearest Neighbors Classifier*, Proc. of *Intelligent Systems in Molecular Biology* (1997), pp. 368–383
20. A.D. Anastasiadis, G.D. Magoulas, X. Liu, *Classification of protein localisation patterns via supervised neural network learning, Proc. of the Fifth Symposium on Intelligent Data Analysis*, *Lecture Notes in Computer Science* **2810**, (Springer-Verlag, 2003), pp. 430–439
21. R. Kohavi, *A study of cross-validation and bootstrap for accuracy estimation and model selection, International Joint Conference on Artificial Intelligence* (1995), pp. 223–228
22. E.K. Blum, Neural Computation **1**, 532 (1989)