A Learning Rule Safe From Local Minima for a Generalized Perceptron

Bruno Raffin and Bernard Virot

LIFO - IIIA
Université d'Orléans
4, rue Léonard De Vinci -BP 6759
F-45067 ORLÉANS Cedex 02 - FRANCE
email: {raffin|virot}@lifo.univ-orleans.fr
Phone: 38 41 71 71 poste 76 41, 38 41 73 18 - fax: 38 41 71 37

This paper is published in the Proceedings of the international ICSC Symposia on INTELLIGENT INDUTRIAL AUTOMATION and SOFT COMPUTING. P.G. Anderson and K. Warwick, editors. Publication by ICSC Academic Press. Volume B. pages 223-229. March 1996. Reading, UK.

Abstract

Known learning rules for the Perceptron are in some sense incomplete since no one ensures to detect non linear separability, to achieve linear separability when it is possible and to yield an optimal generalization error. We present in this paper **KLR**, a new learning rule for the Perceptron derived from the Kullback Distance. Our approach consists in introducing a stochastic perturbation which allows us to approximate an error function by a convex one. Hence, the minimization of this function is safe from local minima. We moreover prove that KLR asymptotically learns all linearly separable sets of patterns, and that it detects the non linearly separable ones. Simulations confirm these theoretical results. They also show that KLR provides a generalization error very close to the optimal. KLR moreover yields a sturdy learning process. Fine tuning of parameters is not needed to obtain good results.

We also show that it is possible to use KLR in order to test architecture inadequacy of classical networks as Mask Perceptron or Polynomial Classifiers [**?**, **?**]. KLR detects the non separability, hence it indicates when the considered architecture is to be upgraded. It provides an interesting information for methods of architecture development.

1 Introduction

The Single-Layer Perceptron [**?**] is the simplest neural network used for pattern classification. It is composed of one layer of input neurons and one layer of output neurons. It expresses only linear dependences between its input neurons and each output neuron. Then, it is able to learn any linearly separable learning set, but not non linearly separable ones.

Given a learning set, i.e. a set of input patterns with their expected outputs, a learning rule is used to determine the connection weights between neurons. The aim is to perform an error-free classification of these input patterns. If the learning set is separable, generally many error-free solutions are possible. The best weights are then those that minimize the generalization error, i.e. the probability that the output corresponding to a not learned pattern is wrong. The learning rule must stop, if the learning set is not separable, ensuring that error-free classification is impossible.

Known learning rules for the Perceptron are in some sense incomplete. Several learning rules are guaranteed to converge to an error-free classification when it exists, but they never stop if the learning set is not linearly separable [**?**, **?**, **?**, **?**, **?**]. Others may miss an error-free classification even if it exists [**?**, **?**, **?**, **?**]. Nabutovsky and al. present in [**?**] a learning rule that achieves linear separability when it is possible, and detects non linear separability. The obtained generalization errors are nevertheless not optimal. On the other hand, only one learning rule, Minimerror, is known to yield a generalization error close to the optimal [**?**, **?**]. Although simulations show that Minimerror minimizes the error number [**?**], no theoretical result confirms that error-free classification is obtained when it is possible. Good results are moreover dependent of fine adjustments of its parameters.

In this paper we present the **Kullback Learning Rule**, abbreviated to **KLR**, a new learning rule for the Perceptron derived from the Kullback Distance. Note that similar learning rules for the Perceptron were already proposed [**?**, **?**]. These works nevertheless do not address learning and generalization abilities. Our approach consists in introducing a stochastic perturbation which allows us to approximate an error function by a convex one. Hence, the minimization of this function is safe from local minima. We moreover prove that KLR asymptotically learns all linearly separable sets of patterns, and that it detects the non linearly separable ones. Simulations confirm these theoretical results. They also show that KLR provides a generalization error very close to the optimal given by Bayes Optimal Classification Algorithm [**?**]. KLR moreover yields a sturdy learning process. Fine tuning of parameters is not needed to obtain good results.

To increase separability abilities of the Perceptron needs to explicitly introduce non linear dependences through new interactions between input and output neurons. An approach consists in using specific computing units between input neurons and output neurons. These units allows to express and fix in the architecture desirable interactions between some set of input neurons and the intended responses to these interactions. Though these networks, such as Polynomial Classifiers, Mask Perceptrons or Clipers [**?**, **?**, **?**], lead to interesting classification abilities, their usefulness is limited by the difficulty to overcome architecture design. Recent researches lead to promising generic methods of architecture development [**?**, **?**]. We show that it is possible to use KLR in order to test inadequacy of such architectures. KLR detects the non separability, hence it indicates when the considered architecture is to be upgraded. It provides an interesting information for methods of architecture development.

The paper is organized in 6 sections. In the section 2 we rapidly present the Perceptron and our notations. Then we introduce KLR and theoretical results in the section 3. The fourth one deals with KLR simulations. The KLR usefulness for architecture development of generalized Perceptrons is considered in the fifth section. The conclusion is presented in section 6.

2 The Perceptron

A Perceptron consists of a set I of **input neurons** connected to a set O of **output neurons**. The state of input neurons are real or discrete–valued. These of output neurons are binary (0 or 1 here). We denote by ξ_i the state of the input neuron i and by o_j this of the output neuron j. A connection between an input i and an output j holds a **synaptic weight** $w_{ij} \in \mathbb{R}$. The notation $|I|$ denotes the cardinal of the set I. When the input pattern $(\xi_i)_{1 \leq i \leq |I|}$ is presented in entry, the output is given by:

$$o_j = \Theta(\sum_{i \in I} w_{ij}\xi_i) \text{ where } \Theta(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

As two different output neurons do not interact, the study of a Perceptron with k output neurons can amount to the study of k Perceptrons with exactly one output neuron. We so merely consider a Perceptron with one output denoted o. The weights w_{ij} become a weight vector $w = (w_i)_{1 \leq i \leq |I|}$.

This one output Perceptron splits inputs in two classes according to its weight values.

Given a **learning set** Γ, i.e. a set of pairs $(\xi^\mu,\ \zeta^\mu)$ where $\xi^\mu = (\xi_i^\mu)_{1 \leq i \leq |I|}$ is an input pattern, and ζ^μ the corresponding desired output, the aim of the Perceptron is to find a weight vector that realises this desired classification:

$$\zeta^\mu = \Theta(w.\xi^\mu) \text{ for all } \mu \in \Gamma \quad (2)$$

where $w.\xi$ denotes the scalar product $w.\xi = \sum_{i \in I} w_i \xi_i$. Since the Perceptron expresses only linear dependences between the input neurons and the output, an **error–free** classification does not exist for all learning sets. Those which are classifiable are classically named **linearly separable sets**.

3 The Kullback Learning Rule

The **Kullback learning rule**, abbreviated to **KLR**, is a learning rule for the Perceptron, i.e a method to compute a weight vector that provides an error–free classification when it exists. KLR relies on the minimization of a convex error function derived from the Kullback distance. Its minimization leads to an error–free classification if the learning set is linearly separable, but it can also ensure the non separability of non linearly separable sets. First we present the error function, and then we deal with some theorems about its characteristics. We end by proposing a learning algorithm.

3.1 An error function based on the Kullback distance

Our approach consists in introducing a stochastic perturbation in the dynamics equation of the Perceptron. We compute the Kullback distance between the desired behavior of the Perceptron and the obtained one, for each patterns of the considered learning set $\Gamma = \{(\xi^\mu, \zeta^\mu); 1 \leq \mu \leq |\Gamma|\}$. It leads us to the error function K:

$$K(w) = \sum_{\mu \in \Gamma} p^\mu \log(1 + \exp(1 - 2\zeta^\mu) w.\xi^\mu), \quad (3)$$

where $p^\mu > 0$ is the **"natural" probability of appearance** associated to each pattern $\mu \in \Gamma$ ($\sum_{\mu \in \Gamma} p^\mu = 1$).

This error function has the important property of being convex. To prove it, it is sufficient to compute the Hessian matrix $H(K)$ of K, and to prove that $w^t H(K) w$ is positive for all weight vectors $w \neq 0$.

Theorem 1 *The function K is convex in weight space $\mathbb{R}^{|I|}$.*

Let us now show that this function K is relevant for pattern classification. We begin with some intermediate results. First of all, we give a new characterization of an error-free classification (see eq. 2). A weight vector w leads to an error-free classification of the patterns of Γ if and only if for each pattern $\mu \in \Gamma$

$$(1 - 2\zeta^\mu) w.\xi^\mu < 0 \tag{4}$$

Note that a pattern located on the hyperplan of normal vector w is considered as miss-classified. It is then possible to prove the following fundamental theorem from these results. It states that the function K allows to characterize linearly separable learning sets and non linearly separable ones:

Theorem 2 *A **finite** learning set Γ is linearly separable if and only if there exists a weight vector $w \neq 0$ such that:*

$$K(w) < \min_{\mu \in \Gamma} p^\mu \log 2$$

If we are able to compute the minimum of the function K over the weight space $\mathbb{R}^{|I|}$, we call it $min_K = \min_{w \in \mathbb{R}^{|I|}} K(w)$, then it is possible to decide if the considered learning set is linearly separable or not. Either we have $min_K < \min_{\mu \in \Gamma} p^\mu \log 2$ and the corresponding weight vector leads to an error-free classification (Th. **??**). Or it yields $min_K >= \min_{\mu \in \Gamma} p^\mu \log 2$. In this latter case, for all $w \in \mathbb{R}^{|I|}$ we have $K(w) \geq \min_{\mu \in \Gamma} p^\mu \log 2$. Therefore Γ is not linearly separable (Th. 2).

Nevertheless it is not always possible to compute min_k. As shown by the next proposition, if the learning set is linearly separable, the minimum 0 is asymptotically reached for the weight vectors for which the norm is infinite.

Proposition 1 *If Γ is a linearly separable learning set and if the weight vector $w \neq 0$ leads to an error-free classification , then we have for $\rho \in \mathbb{R}$:*

$$\lim_{\rho \to +\infty} K(\rho w) = 0$$

On the contrary, The following proposition ensures that the minimum min_K corresponds to finite weight vectors if the learning set is not linearly separable.

Proposition 2 *If Γ is a non linearly separable learning set then for all weight vectors $w \neq 0$ there exists a miss-classified pattern μ_w such that for $\rho \in \mathbb{R}$ we have:*

$$\lim_{\rho \to +\infty} p^{\mu_w} \log(1 + \exp(1 - 2\zeta^{\mu_w}) \rho w.\xi^{\mu_w}) = +\infty$$

3.2 A learning algorithm

To avoid issues bound to infinite norms and to improve minimization algorithm of the function K, we propose a learning algorithm based on the minimization of a modified function K'. Rather than minimizing K in the space $\mathbb{R}^{|I|}$, we constrain the minimization to seek the minimum among the weight vectors w for which the norm $\|w\|^2 = \sum_{i \in I} w_i^2$ is bounded. This bound is fixed by a real value of **radius** $r > 0$. The error function becomes

$$K'(w, r) = K(w) + \begin{cases} 0 & \text{if } \|w\| < r, \\ \sum_{i \in I} w_i^2 - r^2 & \text{otherwise} \end{cases} \tag{5}$$

Note that K' is still a **convex** function. The product

$$w^t H(K') w = w^t H(K) w + \begin{cases} 0 & \text{if } \|w\| < r, \\ 2 \sum_{i \in I} w_i^2 & \text{otherwise} \end{cases} \tag{6}$$

is positive for all weight vectors $w \neq 0$ (Th. 1).

The learning algorithm begins by initializing r. Then we minimize the function K'. If the found minimum provides a weight vector with a norm strictly less than r. Then, since K' is convex, it ensures that even by directly minimizing the function K it would yield to the same minimum. Therefore we can conclude that the considered learning set is not linearly separable. Otherwise the norm of the weight vector is equal to r. If it provides an error-free classification the algorithm can stop. In the opposite case, the radius is to be increased and the minimization is to be performed again. This algorithm asymptotically converges to a solution (Th. 2), i.e. it provides an error-free classification or it detects the learning set as non-linearly separable.

Remark : The presented algorithm is generic enough to be used with all possible minimization methods without problems of infinite norm vectors. But obviously, other algorithms, adapted to the chosen minimization method, may be applied.

4 Simulations

We performed learning simulations with two main purposes. The former is to validate the theoretical results about learning abilities of KLR, the latter is to study the generalization error achieved by KLR when linearly separable sets are learned.

KLR is tested on a Perceptron with N real input neurons, all connected to one binary output neuron. Each learning set is composed of P random input patterns with uniform probabilities of apparition. The expected output are either randomly generated or with a **Reference**

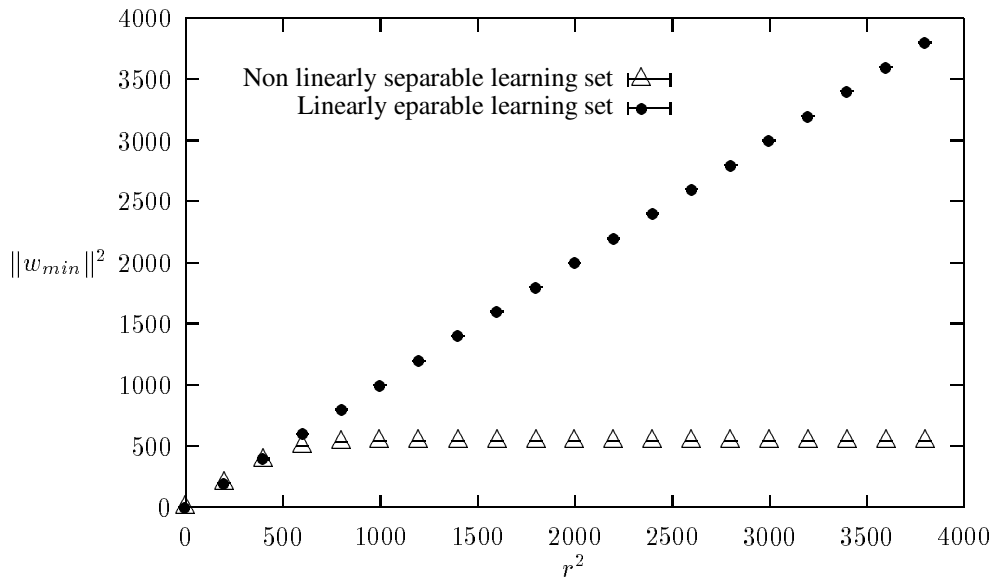

Figure 1: Averarge of $\|w_{min}\|^2$ versus r^2 for $N = 100$ and $P = 300$, for linearly separable and non linearly separable learning sets. The weight vector w_{min} corresponds to this provided by the minimization of $K'(w, r)$ in the weight space.

Perceptron [**?**]. A Reference Perceptron, with random weights, constructs linearly separable learning sets. Each learning algorithm consists of alternating between minimizing K' by conjugate gradient [**?**] and increasing r^2 by 200 with an initial radius $r^2 = 1$. Note that for sake of clarity all these steps do not appear on the plots. All simulation results correspond to mean values over 40 learning processes.

First of all, we test KLR with linearly separable learning sets. We make simulations for $N = 100$ and $P = 50, 100, 200, 300, 400, 500, 600$. For all tests, the learning process ends with an error–free classification. We remark that the smallest radius r_0 that provides an error–free classification depends on P. The greater P, the greater r_0. Each minimization yields a weight vector for which the norm is equal to the radius (Figure 1). We also learn sets with random outputs for $N = 100$ and $P = 300$. These values ensure that the probability of being separable is almost zero [**?**]. Their non–separability is effectively detected by KLR (Figure 1).

We turn now to the analyze of the generalization error ϵ_g [**?**]. It is the probability that a Perceptron gives the wrong output to a pattern not belonging to the learning set. In the thermodynamic limit, $N \to +\infty$, Opper and Haussler have computed the generalization error of Bayes algorithm [**?**] versus α ($\alpha = \frac{P}{N}$). It gives the lowest bound to the generalization error of a Perceptron learning a separable set of patterns. We estimate ϵ_g using the next formula [**?**]:

$$\epsilon_g = \frac{1}{\pi} arccos(E(R)) \tag{7}$$

where $R = \frac{w.v}{\|w\|\|v\|}$ is the overlap between the weight vector w of the Perceptron obtained from a learning process over a separable learning set and the weight vector v of the Reference Perceptron. $E(R)$ is the average of R over several learning processes. We estimate ϵ_g from the tests made above, for $\alpha = 0.5, 1, 2, 3, 4, 5, 6$ and 100. We computed it each time a value of radius provides an error–free classification for all the 40 samples considered. It appears that the best value of ϵ_g occurs close to r_0. Then, the generalization error decreases while the radius increases. This best generalization error is displayed for $N = 100$ (Figure 2). We also test the generalization error for different values of N (50, 100 and 400). It improves when N increases. Compared with the optimal generalization error, the results of KLR are excellent. The value of ϵ_g reaches the lowest bound of this generalization error. We also plot the generalization error obtained from a random choice of a Perceptron that performs an error-free classification (Boltzmann Algorithm). These values were computed by Györgyi and al. [**?**]. It shows that KLR selects a specific weight vector among the ones that provide an error–free classification, and that this selection improves significantly the generalization error.

These simulations moreover reveal other qualities of KLR bound part to the convexity of the minimized function. KLR is robust. The same parameters are used for all the simulations and we perform only few trials to fix their

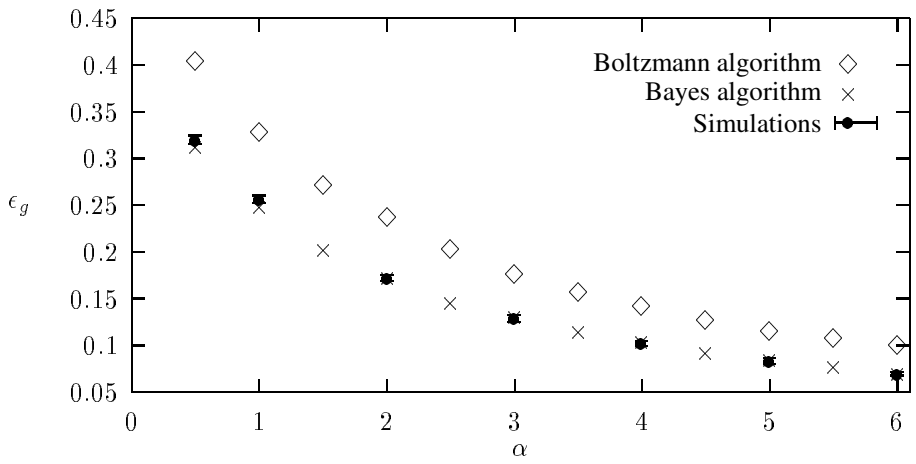

Figure 2: Generalization error ϵ_g versus α for $N = 100$. The theoretical predictions of Bayes and Boltzmann algorithms are plotted for comparison.

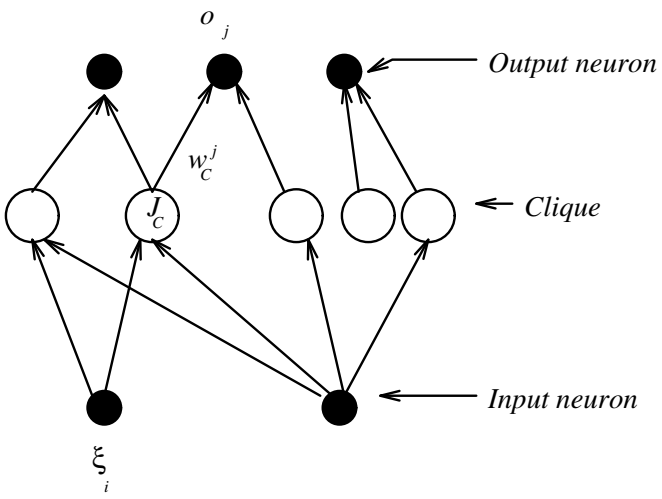

Figure 3: A Cliper

values. The learning time is not addressed in this paper, but it could be improved by a trickier way of increasing the radius. The simulations show that the scheme of radius increase does not deeply influence the result quality.

5 Application

Let us turn now towards neural networks with learning abilities more developed than those of the Perceptron. This latter can only learn linearly separable sets of patterns since its architecture allows to express only linear dependences between its input neurons and each output neuron. To increase these learning abilities needs to introduce non linear dependences through new interactions between input and output neurons. An approach consists in using possible a priori knowledge about the problem. The idea is to express and fix in the architecture desirable interactions between some set of neurons and the intended responses to these interactions. We have introduced in [**?**] a network, called a **Cliper**, able to model such interactions. It uses an approach proposed by Azencott in the framework of Boltzmann Machines [**?**]. In order to model higher order interactions he introduced new computing units, called **cliques**. A Cliper is a generalized Perceptron that owns cliques. It is composed of input neurons connected, via non–weighted connections, to cliques. These cliques are them connected, with weighted connections, to output neurons. The state of a clique is an arbitrary function of the states of the input neurons connected to this clique (Figure 3). Therefore, learning rules for the Perceptron fit to the Cliper and keep their properties. The Cliper subsumes existing networks that allow to model higher order interactions such as Polynomial Classifiers, also called Higher Order Networks, or Mask Perceptron [**?**, **?**]. The interest for such models was nevertheless limited since the choice of an adapted architecture was difficult in a general frame. But recent researches lead to promising generic methods of architecture development for some sub–classes of Clipers [**?**, **?**]. Such methods can particularly take advantage of a learning rule like KLR. Either KLR provides a generalization error close to the optimal when the architecture is sufficient to perform the expected classification or it detects architecture inadequacies. It moreover yields a sturdy learning process.

6 Conclusion

We presented a theoretical study and simulations of KLR, a new learning rule for the Perceptron. It relies on the minimization of a convex error function. Our main contribution was to prove that KLR detects all the separable learning sets and all the non linearly separable ones. We also show through simulations that KLR provides a quasi–optimal generalisation error. We introduced a learning algorithm that permits to avoid issues of infinite norm vectors when the learning set is linearly separable. This rule asymptotically converges which learning set is considered, a linearly or non linearly separable one. Either the found weights yield an error–free classification and a generalization error close to the optimal. Or the weight vector corresponding to the founded minimum has a norm lower than the current radius r and then the learning set is not linearly separable. Simulations point out that the learning process is sturdy, and may be accelerated significantly.

An application domain of KLR could be the networks, like the Cliper, that are able to model higher order interactions. They allow to classify any learning set and use the learning rules conceived for the Perceptron. Such networks nevertheless require to be really useful a method of architecture development. KLR could be a good base for such methods since it detects architecture inadequacy.

References

[1] J. K. Anlauf and M. Biehl. The AdaTron: an Adaptative Perceptron Algorithm. *Europhysics letter*, 10:687–692, 1989.

[2] R. Azencott. Synchronous Boltzmann Machines and Gibbs fields: Learning Algorithms. *Neurocomputing*, F.68, 1990.

[3] G. Desrocques, D. Archambault, and J.-C. Bassano. An associative neural experts system for information retrieval. In *RIAO'91*, pages 546–566, Universitat Autònoma de Barcelona, Spain, 1991.

[4] C. Devouge. *Quelques aspects mathématiques de l'auto-organisation neuronale et des perceptrons multicouches*. PhD thesis, Université Paris 11-Orsay, 1992.

[5] M. Frean. A "thermal" Perceptron Learning Rule. *Neural Computation*, 4(6), 1992.

[6] S. I. Gallan. Optimal Linear Discriminants. In *8th Annual Conference of Cognitive Science Society*, Paris, 1986. IEEE.

[7] E. Gardner and B. Derrida. Optimal storage properties of neural network models. *Journal of Physics*, A(21):271–284, 1988.

[8] M. B. Gordon and D. Berchier. A Perceptron Learning Rule that Find Optimal Weights. In *ESANN'93*, pages 105–110, 1993.

[9] M. B. Gordon and D. Grempel. Learning with a temperature dependent algorithm. *Europhysics Letters*, 29:257–262, 1995.

[10] M. B. Gordon, P. Peretto, and D. Berchier. Learning algorithms for perceptrons from statistical physics. *Journal de Physique I*, 3:377–387, January 1993.

[11] G. Györgyi and N. Tishby. Statistical theory of learning a rule. In *Neural Networks and Spin Glasses*, pages 3–36, 1990.

[12] A. Kowalczyk and H. L. Ferrá. Developing Higher–Order Networks With Empirically Selected Units. *IEEE Transactions on Neural NETWORKS*, 5(5):698–711, 1994.

[13] W. Krauth and M. Mezard. Learning algorithms with optimal stability in neural networks. *J. Phys A*, 20:745–752, 1987.

[14] M. Minsky and S. Papert. *Perceptrons*. MIT press, Cambridge, 1969.

[15] D. Nabutovsky and E. Domany. Learning the Unlearnable. *Neural Computation*, 3:604–616, 1991.

[16] M. Opper and D. Haussler. Generalization Performance of Bayes Optimal Classification Algorithm for Learning a Perceptron. *Phys. Rev. Lett.*, 66:2677–2680, 1991.

[17] W. H. Press, B. P. Flannery, S. A. Teukosky, and W. T. Vetterling. *Numerical Recipes*. Cambridge University Press, 1986.

[18] B. Raffin and M. B. Gordon. Learning and generalisation with Minimerror, a temperature dependent learning rule. *Neural Computation*, 7(6):1206–1224, november 1995.

[19] B. Raffin and B. Virot. Learning boolean functions safe from local minima with a Generalized Perceptron. Technical Report 95-2, LIFO, Université d'Orléans, France, 1995.

[20] F. Rosenblatt. *Principles of neurodynamics*. Spartan, New York, 1962.

[21] P. Ruján. A fast method for calculating the perceptron with maximal stability. *Journal de Physique I France*, 3:277–290, 1993.

[22] S. Seewer and P. Ruján. The generalisation probabilty of perceptron using the Δ–rule. *Journal of Physics A: Math. Gen.*, 25:L505–L510, 1992.

[23] B. Widrow and M. E. Hoff. Adaptative switching circuits. In *1960 IRE WESCON Convention Record*, pages 96–104, New York, 1960.