



**Repositorio Institucional de la Universidad Autónoma de Madrid**

<https://repositorio.uam.es>

Esta es la **versión de autor** de la comunicación de congreso publicada en:  
This is an **author produced version** of a paper published in:

Artificial Neural Networks: Formal Models and Their Applications – ICANN  
2005: 15th International Conference, Warsaw, Poland, September 11-15, 2005.  
Proceedings, Part II. Lecture Notes in Computer Science, Volumen 3697.  
Springer, 2005. 13-18.

**DOI:** [http://dx.doi.org/10.1007/11550907\\_3](http://dx.doi.org/10.1007/11550907_3)

**Copyright:** © 2005 Springer-Verlag

El acceso a la versión del editor puede requerir la suscripción del recurso  
Access to the published version may require subscription

# Discriminant Parallel Perceptrons

Ana González, Iván Cantador and José R. Dorronsoro \*

Depto. de Ingeniería Informática and Instituto de Ingeniería del Conocimiento  
Universidad Autónoma de Madrid, 28049 Madrid, Spain

**Abstract.** Parallel perceptrons (PPs), a novel approach to committee machine training requiring minimal communication between outputs and hidden units, allows the construction of efficient and stable nonlinear classifiers. In this work we shall explore how to improve their performance allowing their output weights to have real values, computed by applying Fisher's linear discriminant analysis to the committee machine's perceptron outputs. We shall see that the final performance of the resulting classifiers is comparable to that of the more complex and costlier to train multilayer perceptrons.

## 1 Introduction

After their heyday in the early sixties, interest in machines made up of Rosenblat's perceptrons greatly decayed. The main reason for this was the lack of suitable training methods: even if perceptron combinations could provide complex decision boundaries, there were not efficient and robust procedures for constructing them. An example of this are the well known committee machines (CM; [4], chapter 6) for 2-class classification problems. They are made up of an odd number  $H$  of standard perceptrons, the output of the  $i$ -th perceptron  $P_i(X)$  over a  $D$ -dimensional input pattern  $X$  being given by  $P_i(X) = s(\text{act}_i(X))$  (we assume  $x_D = 1$  for bias purposes). Here  $s(\cdot)$  denotes the sign function and  $\text{act}_i(X) = W_i \cdot X$  is the  $X$  activation of  $P_i$ . The CM output is then  $h(X) = s\left(\sum_{i=1}^H P_i(X)\right) = s(\mathcal{V}(X))$ , i.e., the sign of the overall perceptron vote count  $\mathcal{V}(X)$ . Assuming that each  $X$  has a class label  $y_X = \pm 1$ ,  $X$  is correctly classified if  $y_X h(X) = 1$ . If not, CM training applies Rosenblat's rule

$$W_i := W_i + \eta y_X X \quad (1)$$

to the smallest number of incorrect perceptrons (this number is  $(1 + |\mathcal{V}(X)|)/2$ ); moreover, this is done for those incorrect perceptrons for which  $|\text{act}_i(X)|$  is smallest. Although sensible, this training is somewhat unstable and only able to build not too strong classifiers. A simple but powerful variant of classical CM training, the so-called parallel perceptrons (PPs), recently introduced by Auer et al. in [1], allows a very fast construction of more powerful classifiers, with capabilities close to the more complex (and costlier to train) multilayer

---

\* With partial support of Spain's CICYT, projects TIC 01-572, TIN2004-07676.

perceptrons (MLPs). In PP training, (1) is applied to all wrong perceptrons but the PP key training ingredient is an output stabilization procedure that tries to keep away from 0 the activation  $act_i(X)$  of a correct  $P_i$ , so that small random changes on  $X$  do not cause its being assigned to another class. More precisely, when  $X$  is correctly classified, but for a given margin  $\gamma$  and a perceptron  $P_i$  we have  $0 < y_X act_i(X) < \gamma$ , Rosenblatt's rule is essentially again applied in order to push  $y_X act_i(X)$  further away from zero. The value of the margin  $\gamma$  is also adjusted dynamically so that most of the correctly classified patterns have activation margins greater than the final  $\gamma^*$  (see section 2). In spite of their very simple structure, PPs do have a universal approximation property and, as shown in [1], provide results in classification and regression problems quite close to those offered by C4.5 decision trees or MLPs.

There is much work being done in computational learning theory to build efficient classifiers based on low complexity information processing methods. This is particularly important for high dimensionality problems, such as those arising in text mining or bioinformatics. As just mentioned, PPs combine simple processing with good performance. A natural way to try to get a richer behavior is to relax their clamping of output weights to 1, allowing these weights to have real values. In fact, usually PP performance does not depend on the number of perceptrons used, 3 being typically good enough. For classification problems, a natural option, that we shall explore in this work, is to use standard linear discriminant analysis to do so. We shall briefly describe in section 2 the training of these discriminant PPs as well as their handling of margins, while in section 3 we will numerically analyze their performance over several classification problems, comparing it to that of standard PPs and MLPs. As we shall see, discriminant PPs will give results somewhat better than those of standard PPs and essentially similar to those of MLPs.

## 2 Discriminant PPs

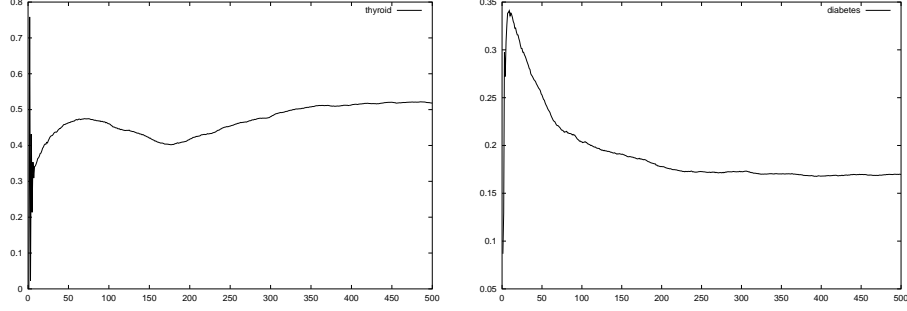
We discuss first perceptron weight and margin updates. Assume that a set  $\mathcal{W} = (W_1, \dots, W_H)$  of perceptron weights and of Fisher's weights  $A = (a_1, \dots, a_H)^t$  have been computed. The output hypothesis of the resulting discriminant PP is

$$h(X) = s\left(A \cdot (P(X) - \tilde{P})\right) = s\left(\sum_1^H a_i (P_i(X) - \tilde{P}_i)\right),$$

with  $\tilde{P} = (\bar{P}_+ + \bar{P}_-)/2$  and  $\bar{P}_\pm$  the averages of the perceptron outputs over the positive and negative classes. We assume that the sign of the  $A$  vector has been adjusted so that a pattern  $X$  is correctly classified if  $y_X h(X) = 1$ . Now,

$$|(\bar{P}_\pm)_i| \leq \frac{1}{N_\pm} \sum_{X' \in C_\pm} |P_i(X')| = 1.$$

with  $N_\pm$  the sizes of the positive and negative classes  $C_\pm$ . We can expect in fact that  $|(\bar{P}_\pm)_i| < 1$  and hence,  $|\tilde{P}_i| < 1$  too. Therefore,  $y_X a_i (P(X) - \tilde{P}) > 0$



**Fig. 1.** Margin evolution for the thyroid (left) and diabetes datasets. Values depicted are 10 times 10 fold crossvalidation averages of 500 iteration training runs.

if and only if  $y_X a_i P(X) > 0$ , and if  $X$  is not correctly classified, we should augment  $y_X a_i P_i(X)$  over those wrong perceptrons for which  $y_X a_i P_i(X) < 0$ . This is equivalent to augment  $y_X a_i act_i(X) = y_X a_i W_i \cdot X$ , which can be simply achieved by using again Rosenblatt's rule (1) adjusted in terms of  $A$ :

$$W_i := W_i + \eta s(y_X a_i) X, \quad (2)$$

for then we have

$$y_X a_i (W_i + \eta s(y_X a_i) X) \cdot X = y_X a_i W_i \cdot X + \eta |y_X a_i| |X|^2 > y_X a_i W_i \cdot X.$$

On the other hand, the margin stabilization of discriminant PPs is essentially that of standard PPs. More precisely, if  $X$  is correctly classified,  $y_X a_i P_i(X) > 0$  and thus  $s(y_X a_i) act_i(X) > 0$ , which we want to remain  $> 0$  after small  $X$  perturbations. For this we may again apply (2) now in the form  $W_i := W_i + \lambda \eta s(y_X a_i) X$  to those correct perceptrons with a too small margin, i.e., those for which  $0 < s(y_X a_i) act_i(X) < \gamma$ , so that we push  $s(y_X a_i) act_i(X)$  further away from zero. The new parameter  $\lambda$  measures the importance we give to wide margins. The value of the margin  $\gamma$  is also adjusted dynamically from a starting value  $\gamma_0$ . More precisely, at the beginning of the  $t$ -th batch pass, we set  $\gamma_t = \gamma_{t-1}$ ; then, if a pattern  $X$  is processed correctly, we set  $\gamma_t := \gamma_t + 0.25\eta$  if all perceptrons  $P_i$  that process  $X$  correctly also verify  $s(y_X a_i) act_i(X) \geq \gamma_{t-1}$ , while we set  $\gamma_t := \gamma_t - 0.75\eta$  if for at least one  $P_i$  we have  $0 < s(y_X a_i) act_i(X) < \gamma_{t-1}$ . These  $\gamma_t$  usually have a stable converge to a limit margin  $\gamma^*$  (see figure 1). We normalize the  $W_i$  weights after each batch pass so that the margin is meaningful. We also adjust the learning rate as  $\eta_t = \eta_0 / \sqrt{t}$  after each batch pass, as suggested in [1].

We recall that for 2-class problems, Fisher's discriminants are very simple to construct. In fact, the vector  $A = S_T^{-1}(\bar{P}_+ - \bar{P}_-)$  minimizes [2] the ratio  $J = s_T/s_B = s_T(A)/s_B(A)$  of the total variance  $s_T$  of discriminant PP outputs to their between class variance  $s_B$ . However, the total covariance matrix  $S_T$  of

			discr. PPs		PPs		MLPs
Problem set	size pos. %	input dim.	num. hid.	lr. rate	num. hid.	lr. rate	num. hid.
breast cancer	34.5	9	5	0.001	3	0.001	5
diabetes	34.9	7	5	0.01	3	0.001	5
glass	13.6	9	5	0.01	3	0.01	5
heart dis.	46.1	13	5	0.001	5	0.001	5
ionosphere	35.9	33	5	0.001	5	0.0001	7
thyroid	7.4	8	5	0.0005	5	0.001	5
vehicle	25.7	18	10	0.01	5	0.001	5

**Table 1.** Input dimensions and training parameters used for the 7 comparison datasets. MLPs were trained by conjugate gradient minimization.

the perceptrons’ outputs is quite likely to be singular (notice that the output space for  $H$  perceptrons has just  $2^H$  distinct values). To avoid this, we will take as the output of the perceptron  $i$  the value  $P'_i(X) = \sigma_\gamma(W_i \cdot X)$ , with the ramp function  $\sigma_\gamma$  taking the values  $\sigma_\gamma(t) = s(t)$  if  $|t| > \lambda = \min(1, 2\gamma)$  and  $\sigma_\gamma(t) = t/\lambda$  when  $|t| \leq \lambda$ . This makes quite unlikely that  $S_T$  will be singular and together with the  $\eta$  and  $\gamma$  updates allows for a fast and quite stable learning convergence. We finally comment on the complexity of this procedure. For  $D$ -dimensional inputs and  $H$  perceptrons, Rosenblat’s rule has an  $O(NDH)$  cost. For its part, the  $S_T$  covariance matrix computation has an  $O(NH^2)$  cost, that dominates the  $O(H^3)$  cost of its inversion. While formally similar to the complexity estimates of MLPs, computing times are much smaller for discriminant PPs (and more so for standard PPs), as their weight updates are much simpler.

### 3 Numerical results

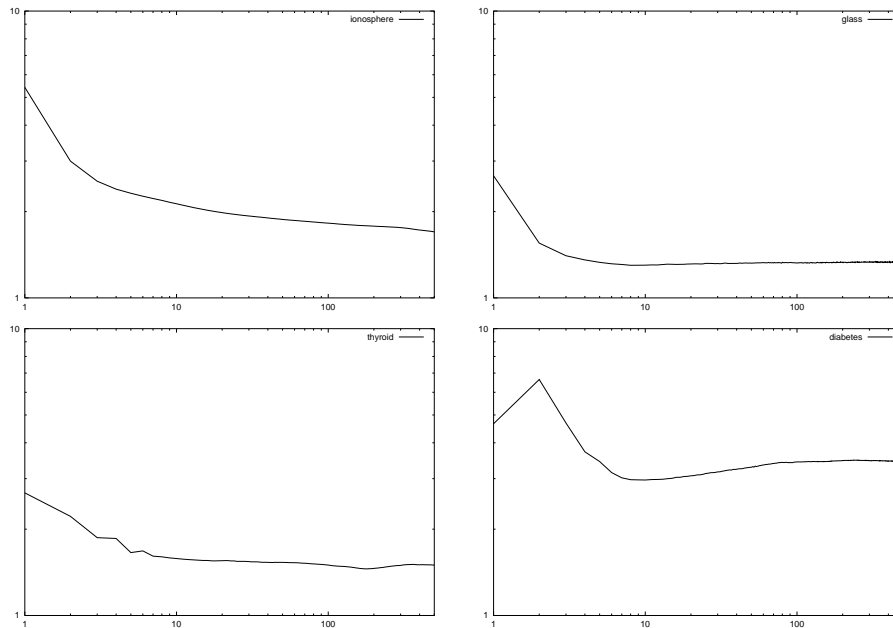
We shall compare the performance of discriminant PPs with that of standard PPs and also of multilayer perceptrons (MLPs) over 7 classification problems sets from the well known UCI database; they are listed in table 1, together with the positive class size, their input dimensions and the training parameters used. Some of them (glass, vehicle, thyroid) are multi-class problems; to reduce them to 2-class problems, we are taking as the minority classes the class 1 in the vehicle dataset and the class 7 in the glass problem, and merge in a single class both sick thyroid classes. We refer to the UCI database documentation [3] for more details. In what follows we shall compare the performance of standard and discriminant PPs and also that of standard multilayer perceptrons first in terms of accuracy, that is, the percentage of correctly classified patterns, but also in terms of the value  $g = \sqrt{a^+a^-}$ , where  $a^\pm$  are the accuracies of the positive and negative classes (see [5]). Notice that for sample imbalanced data sets a high accuracy could be achieved simply by assigning all patterns to the (possibly much larger) negative classes;  $g$  gives a more balanced classification performance measure. In all cases, training has been carried out as a batch procedure using 10-times

Problem set	discr. PPs		PPs		MLPs	
	acc.	<i>g</i>	acc.	<i>g</i>	acc.	<i>g</i>
cancer	<i>96.50</i> (2.16)	<i>96.10</i> (2.22)	<b>96.57</b> (2.15)	<b>96.15</b> (2.22)	95.84 (1.67)	95.53 (1.72)
diabetes	<i>74.97</i> (2.45)	<b>71.87</b> (3.98)	74.25 (3.21)	68.63 (5.34)	<b>76.00</b> (3.09)	<i>70.45</i> (4.33)
glass	<b>96.91</b> (2.09)	<b>92.12</b> (11.01)	<i>94.26</i> (2.09)	84.29 (11.09)	94.05 (2.87)	<i>85.27</i> (8.52)
heart dis.	<b>79.97</b> (3.80)	<b>78.95</b> (3.88)	73.90 (3.80)	73.80 (3.88)	<i>75.22</i> (4.05)	<i>74.68</i> (4.24)
ionosphere	<i>84.06</i> (4.58)	<b>82.16</b> (4.54)	76.97 (3.91)	74.32 (4.11)	<b>84.83</b> (4.19)	<i>81.36</i> (4.54)
thyroid	<b>97.89</b> (0.40)	<i>92.06</i> (1.84)	96.86 (0.91)	82.55 (9.46)	<i>97.62</i> (1.63)	<b>94.01</b> (4.41)
vehicle	<i>76.18</i> (0.41)	<i>70.57</i> (3.46)	74.82 (2.50)	65.03 (5.00)	<b>81.51</b> (4.26)	<b>74.48</b> (5.77)

**Table 2.** Accuracy,  $g$  test values and their standard deviations for 7 datasets and different classifier construction procedures. It can be seen that discriminant PPs results are comparable to those of MLPs and both are better than those of standard PPs.

10-fold cross-validation. Updates (1) and (2) have been applied in standard and discriminant PP training, while conjugate gradient has been used for MLPs. The number of perceptrons in all cases and the initial learning rates for PPs and discriminant PPs for each dataset are described in table 1. Table 2 presents average values of the cross-validation procedure just described (best values in bold face, second best in cursive) for accuracies and  $g$  values, together with their standard deviation. As it can be seen, discriminant PPs give the best accuracy in 3 problems and second best in the other 4, MLPs give the best accuracy in 3 problems and second best in another 2, while standard PPs's accuracy is best only in the cancer problem and is second best over the glass data set. If we consider  $g$  values, discriminant PPs'  $g$  is highest in 4 problems and second best in the other 3, MLPs'  $g$  is highest in 2 problems and second best in another 4, while standard PPs' is highest only in the cancer problem. The performance of discriminant PPs and MLPs is thus quite close and better than that of standard PPs.

We finish this section by noticing that the weight update (2) only aims to reduce the classification error and to achieve a clear margin, but there is no reason that it should minimize the Fisher criterion  $J$ . However, as seen in figure 2, this also happens. The figure depicts in logarithmic X and Y scales the evolution of  $J$  for the ionosphere, glass, diabetes and thyroid datasets (clockwise from top left). Values depicted are 10 times 10 fold cross-validation averages of 500 iteration training runs. Although not always monotonic (as in the glass, thyroid and diabetes problems), the overall  $J$  behavior is clearly decreasing and it converges.



**Fig. 2.** From top left, clockwise: evolution of Fisher's criterion for the ionosphere, glass, diabetes and thyroid datasets. Values depicted are 10 times 10 fold crossvalidation averages of 500 iteration training runs (all figures in log X and Y scale).

## 4 Conclusions

Parallel perceptron training offers a very fast procedure to build good and stable committee machine-like classifiers. In this work we have seen that their classification performance can be improved by allowing their output weights to have real values, obtained by applying Fisher's analysis over the perceptron outputs. The final performance of these discriminant PPs is essentially that of the powerful but costlier to build standard MLPs.

## References

1. P. Auer, H. Burgsteiner, W. Maass, *Reducing Communication for Distributed Learning in Neural Networks*, Proceedings of ICANN'2002, Lecture Notes in Computer Science 2415 (2002), 123–128.
2. R. Duda, P. Hart, D. Stork, **Pattern classification** (second edition), Wiley, 2000.
3. P. Murphy, D. Aha, *UCI Repository of Machine Learning Databases*, Tech. Report, University of California, Irvine, 1994.
4. N. Nilsson, **The Mathematical Foundations of Learning Machines**, Morgan Kaufmann, 1990.
5. J.A. Swets, *Measuring the accuracy of diagnostic systems*, Science 240 (1998), 1285–1293.