A Method for Improving Classification Reliability of Multilayer Perceptrons

Luigi Pietro Cordella, *Member, IEEE,* Claudio De Stefano, Francesco Tortorella, and Mario Vento, *Member, IEEE*

Abstract— **Criteria for evaluating the classification reliability of a neural classifier and for accordingly making a reject option are proposed. Such an option, implemented by means of two rules which can be applied independently of topology, size, and training algorithms of the neural classifier, allows to improve the classification reliability. It is assumed that a performance function P is defined which, taking into account the requirements of the particular application, evaluates the quality of the classification in terms of recognition, misclassification, and reject rates. Under this assumption the optimal reject threshold value, determining the best trade-off between reject rate and misclassification rate, is the one for which the function P reaches its absolute maximum. No constraints are imposed on the form of P, but the ones necessary in order that P actually measures the quality of the classification process. The reject threshold is evaluated on the basis of some statistical distributions characterizing the behavior of the classifier when operating without reject option; these distributions are computed once the training phase of the net has been completed. The method has been tested with a neural classifier devised for handprinted and multifont printed characters, by using a database of about 300 000 samples. Experimental results are discussed.**

I. Introduction

In recent years many pattern recognition systems based on the use of neural networks have been proposed [1]–[4]. To improve recognition capabilities of these systems, different aspects of the problem have been taken into account. On one side, network topologies as much as possible related to the structure of input data [1] and criteria for determining the net optimal size as a function of the given recognition problem [5] have been suggested. On the other side, learning algorithms have been widely studied and criteria for selecting and sorting the training set have been defined [6]–[9]. Namely, mostly investigated topics concern techniques for obtaining better convergence rates (e.g., by overcoming local minima of the error function or by dynamically varying the learning coefficient) and criteria for stopping the learning phase when an acceptable trade-off between generalization power and specialization degree of the net has been achieved.

In real cases, however, even a well-trained network (i.e., a net which has reached a low error value at the end of the training phase) can provide output vectors considerably different from any of the expected ideal output vectors; this is mainly due to unavoidable distortions which affect the samples belonging to the real world (data set), making them even very different from the ones belonging to the training set.

In these conditions, classification reliability attainable with simple classification rules dramatically decreases. This is the case when using the "winner takes all" rule (W-rule) which assigns the pattern to the class whose corresponding output neuron has the highest value (winner neuron) [10]. This simple rule often fails in presence of output vectors where the winner neuron has a low value or more than one neuron has a high value, so giving place to unacceptable misclassification rates.

To improve classification reliability, it is thus necessary to identify not reliable classifications and then to take a decision about the advantage of rejecting a sample instead of running the risk of misclassifying it. This advantage, however, can only be evaluated by taking into account the requirements of the specific application domain. In fact, there are applications for which the cost of a misclassification is very high, so that a high reject rate is acceptable just to keep misclassification rate as low as possible; a typical example could be the classification of medical images in the framework of a prescreening for early cancer detection. In other applications it may be desirable to assign every sample to a class even at the risk of a high misclassification rate; let us consider for instance the case of a character classifier used in applications in which a text has to be successively widely edited by man. Between these extremes, a number of applications can be characterized by intermediate requirements.

It is evident that the best trade-off between reject rate and misclassification rate can be achieved by introducing a reject rule which takes into account the costs attributed, for the specific application, to misclassifications, rejects, and correct classifications. Note that the latter cost is actually a gain, but in this paper we use the term cost when referring to it to simplify the notation.

We propose a method for determining the optimal reject threshold value to be used with a given classifier in a specific application to get the best trade-off between reject and misclassification rates. It is assumed that a performance function P is defined which, taking into account the above mentioned costs, measures, on the field, the quality of a classifier in terms of its recognition, misclassification, and reject rates. Under this assumption the optimal reject threshold value is the one for which the function P reaches its absolute maximum. No hypotheses are necessary on the form of P, but the one that it effectively represents the quality of the classification.

In Section II, the rationale of the method is discussed. In Section III, two simple classification rules are proposed to

Manuscript received April 25, 1993; revised August 21, 1994 and October 31, 1994.

The authors are with Dipartimento di Informatica e Sistemistica, Universita' di Napoli, 80125 Napoli, Italy.

IEEE Log Number 9410196.

implement the above concepts. In Section IV the results of testing the method with a character classifier are illustrated. In Appendix, the generalization of some of the results obtained in Section III is presented.

II. The Approach

The proposed method can be applied to a neural classifier of any topology and size, trained with a supervised learning algorithm. The method can still be applied if an unsupervised learning algorithm is used, provided that, after the learning phase, a correspondence between classes to be recognized and output configurations of the network is established.

As anticipated in the introduction, let us assume that a performance function P of the classification process has been specified. Let us indicate it with $P = P(R_c, R_r, R_m)$, where R_c, R_r, and R_m are recognition, reject, and misclassification rates, respectively. Note that in the expression of P the cost parameters do not explicitly appear because they can be either constant or dependent on R_c, R_r, and R_m.

For P to actually measure the quality of the classification process, it must obviously satisfy the following constraints

$$\frac{\partial P}{\partial R_c} > 0,$$
$$\frac{\partial P}{\partial R_r} < 0,$$
$$\frac{\partial P}{\partial R_m} < 0,$$
$$\left|\frac{\partial P}{\partial R_r}\right| < \left|\frac{\partial P}{\partial R_m}\right|.$$

The last condition is added to signify that it is expected that a misclassification negatively affects P more than a reject. In principle, no further hypotheses are necessary on the form of P. In the following section it will be shown that, given a function P, this can be expressed in terms of some parameters which can be experimentally evaluated. In practice, after the training phase, the classifier is applied, without reject option, to a set S of samples whose class is known and some statistical distributions, which will be discussed in the following section, are evaluated for the subsets of correctly classified and of misclassified samples. These distributions are used for determining two reject threshold values, optimal with respect to the assigned function P, to be used for implementing the classification rules.

If the set S is adequately representative of the data set, as it should be, this approach ensures to obtain the best trade-off between reject and misclassification rate of the classifier. It is worth noting that the set S does not necessarily coincide with the training set. This and the fact that the above computations are performed after training, make the method independent of the training phase and of the network architecture.

For several applications it can be assumed that the cost of a correct classification, of a misclassification, and of a reject does not vary with R_c, R_r, and R_m. Therefore $\partial P/\partial R_c, \partial P/\partial R_r$, and $\partial P/\partial R_m$ can be considered constant, implying that P is a linear function that can be written in the form

$$P = R_c - C_r R_r - C_m R_m \qquad (1)$$

where C_r is the cost of a reject, C_m is the cost of a misclassification, and for the sake of simplicity, these costs have been normalized with respect to the cost of a correct classification. Similar assumptions have been made in [11].

To simplify the mathematical treatment, in the following sections the discussion will be developed assuming that the function P has the form (1). In the Appendix it is shown how the discussion can be extended to the case of a function of generic form.

III. Classification Rules

The assumed ideal output coding requires that each output node O_j of the net corresponds to a single recognition class in such a way that, if the sample supplied to the net belongs to the ith class, the output node O_i must be one, while every other output node O_j (with $j \neq i$), must be zero.

In real cases, as anticipated in the introduction, the output states can differ from the ideal ones and assume values between zero and one. Thus, to take a decision about the class of a sample, suitable rules have to be defined. The simplest and most commonly used rule, the "winner takes all" rule (W-rule), does not provide the possibility of rejecting a sample as not recognizable, but obliges to assign it to a class, without taking care of the reliability of the classification. Sometimes, however, it could be convenient to reject a sample instead of having a high probability to misclassify it.

According to the W-rule and taking into account the assumed output coding of the net, a given input vector I is assigned to the class k, if k is such that

$$O_k > O_j, \forall j \neq k \qquad (2)$$

i.e., if k is the index of the winner neuron.

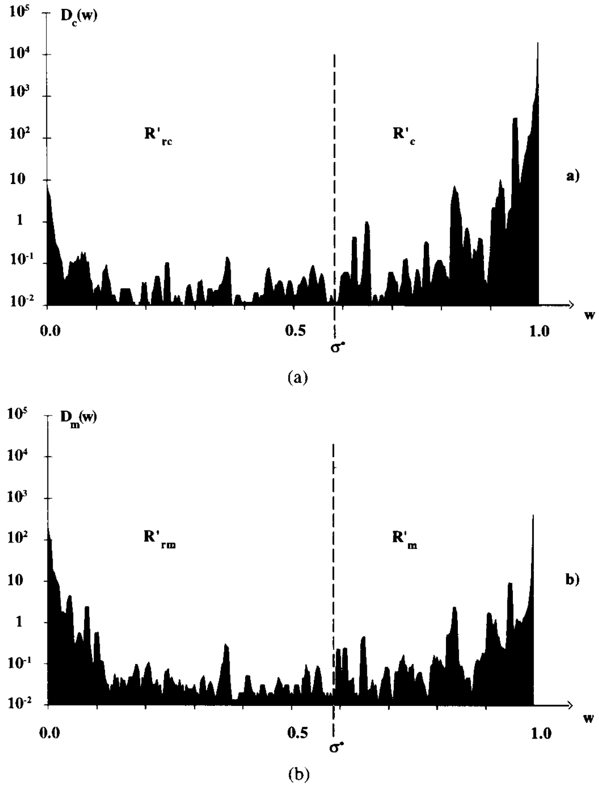
In many cases, it is possible to obtain meaningful information about the reliability of the classification performed in this way, by examining the values of all the remaining output neurons. For instance, a low value for the winner neuron may be obtained either if the sample presented to the net is quite different from the ones belonging to the training set or when it is similar to a sample which is very rare in the training set: in both cases the classification has to be retained not reliable enough.

Another critical decision case happens when more than one neuron with value close to one is present in the output vector. Generally, this is due to the presence in the training set of samples which could be assigned at the same time to several classes: even in this case the classification is unreliable.

Aim of the two classification rules illustrated in the following is to reject the input pattern if a really critical decision case occurs. This leads to improve the previously defined performance function, as it will be shown in the following.

For the W-rule, the expression of P can be easily obtained from (1) by taking into account that $R_r = 0$

$$P_W = (R_c^o - C_m R_m^o) \qquad (3)$$



(a)

(b)

Fig. 1. (a) The occurrence density $D_c(w)$ of the correctly classified samples, obtained by using the W-rule. (b) The corresponding curve $D_m(w)$ for the misclassified samples. Both diagrams refer to the experimental case discussed in Section IV. The indicated value of σ^* is the optimal one found for $C_r = 4$ and $C_m = 6$.

where R_c^o and R_m^o, respectively, represent recognition rate and misclassification rate according to the W-rule.

A. The "Winner Takes All ' Rule with Reject Option (WR-Rule)

The WR-rule rejects the sample to be classified if the value of the winner neuron is lower than a given threshold σ. To evaluate the improvement of the performance function P obtained by applying the WR-rule instead of the W-rule, let us consider the occurrence densities of correctly classified and misclassified samples according to the W-rule, as a function of the winner value w. Let us call them $D_c(w)$ and $D_m(w)$, respectively. For the sake of convenience, the density curve $D_c(w)(D_m(w))$ has been defined so that its integral extended to the interval $[w_1, w_2]$ provides the percentage of correct classifications (misclassifications) for a winner value ranging from w_1 to w_2. Examples of the density curves $D_c(w)$ and $D_m(w)$, obtained with reference to the application discussed in Section V, are shown in Fig. 1.

From the definition of $D_c(w)$ and $D_m(w)$ it follows that

$$R_c^o = \int_0^1 D_c(w)\, dw \tag{4a}$$

$$R_m^o = \int_0^1 D_m(w)\, dw. \tag{4b}$$

Although the curves shown in Fig. 1 refer to a specific experimental case, their trend appears in good accordance to theoretical considerations. In fact, if a neural classifier has been suitably trained, it is expected that the majority of correctly classified samples is found in correspondence to high values of w, while misclassified samples concentrate in correspondence to both lowest and highest values of the winner neuron, because situations discussed at the beginning of this section may happen. Take into account that the two diagrams extend over different intervals of the ordinate scales because R_c^o is usually greater than R_m^o. The introduction of the threshold σ has two opposite effects. On the one hand, the WR-rule classifies only that subset of the samples correctly classified by the W-rule for which the winner value is greater than σ and rejects the others [see Fig. 1(a)], so determining a decrease of P. As a consequence, the recognition rate relative to the use of the WR-rule is given by the relation

$$R'_c = \int_\sigma^1 D_c(w)\, dw \tag{5a}$$

while the percentage R'_{rc} of samples classified by the W-rule, but rejected by the WR-rule is

$$R'_{rc} = \int_0^\sigma D_c(w)\, dw. \tag{5b}$$

On the other hand, similar considerations can be made with reference to the number of misclassified samples [see Fig. 1(b)]: the percentage R'_{rm} of samples misclassified by the W-rule with winner values lower than σ is rejected by the WR-rule. This effect produces an increase of P because the R'_{rm} samples, which are weighted by the W-rule with the misclassification cost C_m, are weighted by the WR -rule with the reject cost C_r, which is certainly lower than C_m.

Taking into account that

$$R'_{rm} = \int_0^\sigma D_m(w)\, dw \tag{6a}$$

and calling R'_m the misclassification rate resulting from the use of WR-rule, it holds

$$R'_m = \int_\sigma^1 D_m(w)\, dw. \tag{6b}$$

Therefore, the reject rate R'_r obtained by using theWR-rule is given by the relation

$$R'_r = R'_{rc} + R'_{rm} \tag{7}$$

and then the expression of P for the WR-rule is

$$P_{WR}(\sigma) = (R'_c - C_r R'_r - C_m R'_m). \tag{8}$$

The overall improvement of P obtained by adopting the WR-rule instead of the W-rule, can be easily computed

$$P_{WR} - P_W = [R'_c - C_r R'_r - C_m R'_m - (R_c^o - C_m R_m^o)]. \tag{9}$$

Considering the expressions (4)–(6), the last equation can be written in the form

$$\begin{aligned} P_{WR} - P_W &= [R'_c - C_r(R'_{rc} + R'_{rm}) - C_m R'_m \\ &\quad - R'_{rc} - R'_c + C_m(R'_{rm} + R'_m)] \\ &= [-R'_{rc}(1 + C_r) + R'_{rm}(C_m - C_r)]. \end{aligned} \tag{10}$$

Thus, the use of the WR-rule gives place to a positive variation of the performance function if

$$R'_{rc}/R'_{rm} < (C_m - C_r)/(1 + C_r). \tag{11}$$

Note that, because of the dependence of R'_{rc} and R'_{rm} on σ, the inequality (11) also depends on this parameter.

From (11), it is clear that the greater is C_m with respect to C_r, the more convenient is the WR-rule with respect to the W-rule; if C_m has a value near to C_r, inequality (11) is likely to be not verified and the use of the WR-rule does not give advantages.

B. The "Winner Takes All" Rule with Reject-On-Difference Option (WD-Rule)

According to this rule, a sample is rejected if, denoted by k the index of the winner neuron and by r the index of the neuron with the next highest value, the difference w_d of their outputs satisfies the following condition

$$w_d = O_k - O_r < \delta \tag{12}$$

where the threshold δ is evaluated as it will be shown in the following.

The WD-rule should be applied just after the WR-rule if this latter has not produced a reject. In this way, the samples examined by the WD-rule are only those not rejected by the WR-rule, i.e., the samples for which the winner value is greater than σ; their percentage is equal to R'_c (correctly classified) plus R'_m (misclassified).

Let us now consider the occurrence density functions for both correctly classified and misclassified samples versus the difference w_d; these functions will be referred to as $D'_c(w_d)$ and $D'_m(w_d)$, respectively (see Fig. 2). For what said above, it holds

$$R'_c = \int_0^1 D'_c(w_d)\, dw_d \tag{13a}$$

$$R'_m = \int_0^1 D'_m(w_d)\, dw_d. \tag{13b}$$

Analogously to the case of σ, the introduction of the threshold δ has two opposite effects on the performance function P. In fact, all the samples correctly classified by the WR-rule with w_d values lower than δ (their percentage is indicated with R''_{rc} in Fig. 2(a)), are rejected by the WD-rule, so causing a decrease of P. On the other hand, P increases because all the samples for which w_d was lower than δ, but misclassified by the WR-rule, are rejected by the WD-rule.

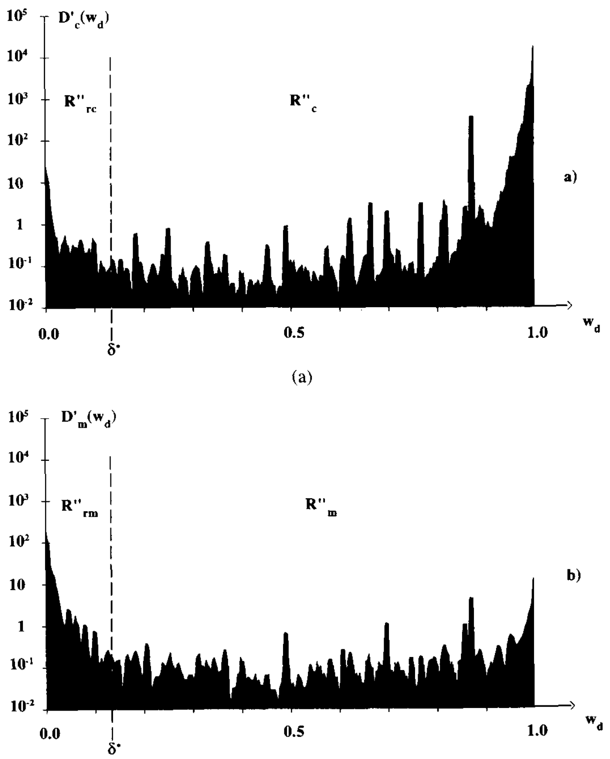
The performance function P_{WD} for the WD-rule can be easily computed

$$P_{WD}(\delta) = (R''_c - C_r R''_r - C_m R''_m) \tag{14}$$

where R''_c and R''_m indicate the recognition rate and the misclassification rate, respectively

$$R''_c = \int_\delta^1 D'_c(w_d)\, dw_d \tag{15a}$$

$$R''_m = \int_\delta^1 D'_m(w_d)\, dw_d. \tag{15b}$$



(a)

(b)

Fig. 2. (a) The occurrence density $D'_c(w_d)$ of the correctly classified samples obtained by using the WR-rule. (b) The corresponding curve $D'_m(w_d)$ for the misclassified samples. The diagrams refer to the same experimental case of Fig. 1. $\delta^*$is the optimal threshold value found for the WD-rule with $C_r = 4$ and $C_m = 6$.

The reject rate obtained as a consequence of the sequential application of the WR-rule and of the WD-rule, is equal to

$$\begin{aligned} R''_r &= R''_{rc} + R''_{rm} + R'_{rc} + R'_{rm} \\ &= \int_0^\delta [D'_c(w_d) + D'_m(w_d)]\, dw_d + R'_{rc} + R'_{rm}. \end{aligned} \tag{16}$$

Consequently, P improves of the quantity

$$\begin{aligned} P_{WD} - P_{WR} = &(R''_c - C_r R''_r - C_m R''_m - R'_c \\ &+ C_r R'_r + C_m R'_m). \end{aligned}$$

With reference to Fig. 2 and taking into account (13a) and (13b), it is easy to see that

$$\begin{aligned} R_c &= \int_\delta^1 D'_c(w_d)\, dw_d \\ &= \int_0^1 D'_c(w_d)\, dw_d - \int_0^\delta D'_c(w_d)\, dw_d \\ &= R'_c - R''_{rc} \end{aligned} \tag{17}$$

and similarly

$$R''_m = R'_m - R''_{rm}. \tag{18}$$

Eventually it results

$$P_{WD} - P_{WR} = [-R''_{rc}(1 + C_r) + R''_{rm}(C_m - C_r)]. \tag{19}$$

Because of the dependence of R_{rc} and R_{rm} on δ, (19) also depends on this parameter.

In the part A of the Appendix it is shown how the results derived in Section III-A and III-B with reference to a linear function P, can be obtained in the case of a function P of any form.

C. Evaluation of the Thresholds

As already shown, the WR-rule and the WD-rule imply the operative definition of the thresholds σ and δ, respectively. The computed values of the thresholds are the ones which maximize the performance functions P_{WR} and P_{WD} on the set S.

By calculating and equating to zero the derivative of expressions (8) and (14) with respect to σ and δ, we obtain

$$(1 + C_r)D_c(\sigma) - (C_m - C_r)D_m(\sigma) = 0 \qquad (20a)$$

$$(1 + C_r)D'_c(\delta) - (C_m - C_r)D'_m(\delta) = 0. \qquad (20b)$$

The functions $D_c(w), D_m(w), D'_c(w_d), D'_m(w_d)$ are not available in their analytical form, but they can be experimentally obtained in tabular form; in particular, if the set S is submitted to the neural classifier without reject option, it splits in a subset S_m of misclassified samples and in a subset S_c of correctly classified samples. For these sets the distributions D_m and D_c are, respectively, computed as a function of the winner value (examples of such distributions are given in Fig. 1), and then the value σ^* maximizing P_{WR} can be obtained by numerically solving (20a). By using as a threshold in the WR-rule the value σ^*, the set of not rejected samples will result composed by two subsets, S'_m and S'_c, respectively, containing the samples of S_m and S_c having a winner value greater than σ^*. The distributions D'_m and D'_c are simply obtained by grouping the samples of S'_m and S'_c as a function of w_d (examples of these distributions are given in Fig. 2). The value of δ^* is obtained by numerically solving (20b).

It has to be taken into account, however, that several values satisfying (20a) or (20b) may exist, because the density curves not necessarily have a monotonic trend. Thus, to be sure of evaluating the value of $\sigma^*(\delta^*)$ corresponding to the absolute maximum of $P_{WR}(P_{WD})$, it is necessary first to determine the values corresponding to all the relative maxima and then to select among them the value corresponding to the absolute maximum.

In the part B of the Appendix it is shown how to compute the optimal reject threshold values in the case of a function P of any form.

IV. Experimental Results

The proposed approach has been tested on a neural classifier made of a three-level feedforward fully connected network with 30 hidden neurons, using a sigmoidal output function (see Fig. 3). The net has been trained by using the backpropagation algorithm [10]. The chosen test bed was the recognition of unconstrained handprinted and multifont printed characters. In this field, because of the enormous variability of character shapes, it is desirable that a classifier is able to evaluate the recognition reliability and to reject a sample in case of unreliable decision. In principle, depending on the specific applicative context in which character recognition is performed, the form of P as well as the costs introduced in the previous sections could vary. It can be assumed, however, that, in several cases, the linear form, implying constant costs, specifies an adequately representative quality function. This means, in fact, that the cost of a misclassification or of a reject does not vary with the number of misclassified or rejected characters. Thus the experimental results presented in the following will refer to the case that a linear form is assumed for P.

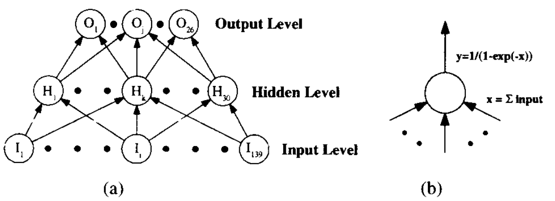


Fig. 3. (a) The considered neural network model; (b) a generic neuron whose output is the value of the sigmoidal function of the sum of all its inputs.

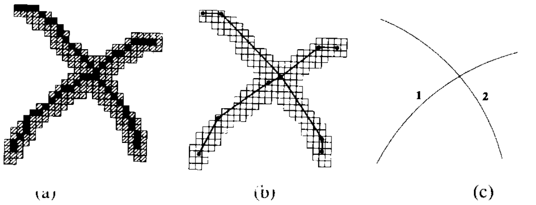


Fig. 4. (a) The bit-map of a character and its skeleton (superimposed black pixels); (b) polygonal approximation of the skeleton after corrections of distinctions; (c) the character representation in terms of circular arcs approximating pieces of polygonal.

The characters to be classified are preliminarily submitted to a process leading to describe them in terms of structural features [12]. The main steps of the process are briefly summarized in the following. Since the thickness of character strokes is generally not a significant feature for recognition purposes, character bit maps are first thinned [see Fig. 4(a)]. Unluckily, skeletonizing algorithms typically give place to distorted representations of character shapes; the most significant shape distortions occur at joins and crossings of strokes. To better preserve the original shape information, a skeleton correction technique [13] is used: after this correction a character is represented by a set of polygonal lines [see Fig. 4(b)]. A further step consists of approximating pieces of polygonal lines with circular arcs [see Fig. 4(c)] according to a method illustrated in [14]. The aim of the above processing is to absorb most of the large variability among different samples of the same character class and to single out the features most characteristic and invariant for a recognition class. The obtained character representations are then described in terms of attributes of each circular arc and of geometrical relations between pairs of arcs. All the possible values of the parameters used for the description, quantized and normalized so as to range from zero to one, are coded in a vector of 139 real elements together with the information about their actual occurrence in the sample at hand.

Experiments were performed by using a test set made of about 300 000 among multifont printed and handprinted characters; about 15 000 of them were used for training the net

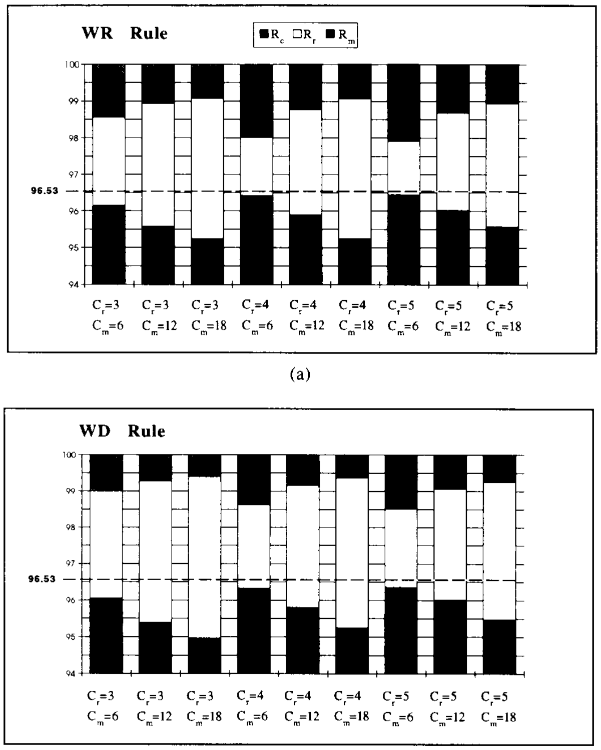
TABLE I
VALUES OF σ^* AND δ^*, FOR DIFFERENT PAIRS (C_r, C_m)

σ^*

C_r \ C_m	6	12	18
3	0.82	0.91	0.94
4	0.58	0.87	0.94
5	0.51	0.85	0.91

δ^*

C_r \ C_m	6	12	18
3	0.276	0.291	0.293
4	0.133	0.281	0.295
5	0.080	0.279	0.288



(a)

(b)

Fig. 5. Values of R_c, R_r, and R_m for different pairs (C_r, C_m), obtained by using the WR-rule (a) and the WD-rule (b) (the dashed horizontal line indicates the recognition rate obtained by using the W-rule).

and the same number of samples was used to build the set S. The values considered for C_r and C_m, were arbitrarily chosen within the sets $\{3, 4, 5\}$ and $\{6, 12, 18\}$, respectively. This choice seemed adequate to include a bunch of possible real situations and corresponds to the hypothesis that, in practical cases, C_r is at least three times greater than the cost of correct classification, while C_m is at least slightly greater than C_r; it allows to verify the behavior of the system for a set of situations ranging from the case in which C_r assumes values near to those of C_m, to the case in which C_r is much smaller than C_m.

In Table I the values of the optimal thresholds $\sigma^*$and δ^* are shown as a function of the cost coefficients C_m and C_r. It can be noted that, for a given C_r, the value of σ^* increases with C_m; in fact, when the ratio C_m/C_r becomes larger, the maximum of P is obtained by rejecting a larger number of samples, and then increasing σ^*. Similar considerations hold for δ^*.

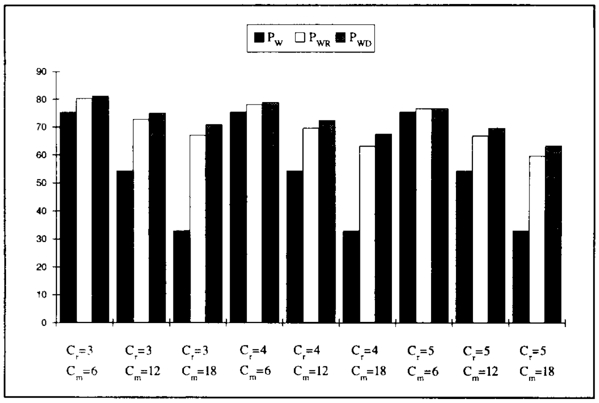


Fig. 6. Values of P_W, P_{WR}, and P_{WD} for different pairs (C_r, C_m).

In Fig. 5, classification, reject, and misclassification rates obtained with the two considered rules are plotted for different values of the pair (C_r, C_m). By employing the WR-rule instead of the W-rule, a considerable reduction of the misclassification rate was obtained, at the expense of only a slight decrease of the recognition rate. In fact, the recognition rate obtained with the W-rule was 96.53% [dashed line in Fig. 5(a)] while the misclassification rate was 3.47%; after applying the WR-rule, for values of the pair (C_r, C_m) varying in the considered range, the recognition rate decreased to values from 95.25% to 96.4% and the misclassification rate to values from 0.9% to 2.1 %.

Fig. 6 shows the values of P_W, P_{WR}, and P_{WD} obtained for different pairs (C_r, C_m); for instance, from this figure it is clear that, for a given C_r, the larger the ratio C_m/C_r, the greater the increase of P obtained by using the WR-rule.

These experimental results are in agreement with the theoretical considerations made in Section III-A with reference to (10).

All the qualitative considerations made for the WR-rule hold for the WD-rule also (see Fig. 5(b) and Fig. 6). From a quantitative point of view, however, since the WD-rule is a refinement of the WR-rule, the improvement obtained with respect to the case that the only WR-rule is used, is smaller than the enhancement achieved when passing from the W-rule to the WR-rule.

V. CONCLUSIONS

In this paper two classification rules, which apply to a neural classifier with any topology and size, have been discussed.

Suitable criteria have been defined for determining the optimal values of the thresholds to be used in the rules to reject samples unreliably classified. This allows to optimize the classification reliability and to achieve the best trade-off between reject rate and misclassification rate, once a performance function measuring the quality of the classification process has been assigned. This is true regardless of how good the classifier performs at the end of the training phase: in any case its classification reliability improves as a consequence of turning some misclassifications into rejects.

The proposed method is especially useful in recognition problems characterized by high variability among the samples belonging to a same class and by partial overlaps between the regions pertaining to different classes. Results of testing the method with a real application demonstrated its effectiveness.

Appendix

In this appendix, it is assumed that the performance function P has a generic form and the mathematical treatment developed in Section III is generalized accordingly. With respect to the discussion presented in Section III, changes are necessary only with reference to the evaluation of the variation of P when passing from a reject rule to another and as regards the way of deriving the equations from which the values of σ^* and δ^* are computed.

A. Evaluating the Variations of P

To evaluate how P varies when passing from the W-rule to the WR-rule, it is worth noting that the values assumed by the performance function in the two cases are

$$\begin{aligned} P_W &= P(R_c, R_r, R_m)|_{(R_c^o, 0, R_m^o)} \\ P_{WR} &= P(R_c, R_r, R_m)|_{(R'_c, R'_r, R'_m)} \end{aligned}$$

where, for the parameters R_c, R_m, R'_c, R'_r, and R'_m, the definitions and relations stated in Section III-A hold. Approximating the function P by means of the Taylor series up to the first order, with starting point $R_0 = (R_c^o, 0, R_m^o)$, where R_c^o and R_m^o are the recognition rate and the misclassification rate obtained when the W-rule is used, the difference $P_{WR} - P_W$ is given by

$$\begin{aligned} P_{WR} - P_W &= P(R'_c, R'_r, R'_m) - P(R_c^o, 0, R_m^o) \\ &= P(R_c - R'_{rc}, R'_{rc} + R'_{rm}, \\ &\quad R_m - R'_{rm}) - P(R_c^o, 0, R_m^o) \\ &\cong \left.\frac{\partial P}{\partial R_c}\right|_{R_0} \cdot (-R'_{rc}) + \left.\frac{\partial P}{\partial R_r}\right|_{R_0} \\ &\quad \cdot (R'_{rc} + R'_{rm}) + \left.\frac{\partial P}{\partial R_m}\right|_{R_0} \cdot (-R'_{rm}) \\ &= \left.\left(\frac{\partial P}{\partial R_r} - \frac{\partial P}{\partial R_m}\right)\right|_{R_0} \\ &\quad \cdot R'_{rm} - \left.\left(\frac{\partial P}{\partial R_c} - \frac{\partial P}{\partial R_r}\right)\right|_{R_0} \cdot R'_{rc} \end{aligned}$$

where the derivatives are computed in R_0. The difference is positive if the following condition holds

$$\frac{R'_{rc}}{R'_{rm}} < \frac{\left.\left(\frac{\partial P}{\partial R_r} - \frac{\partial P}{\partial R_m}\right)\right|_{R_0}}{\left.\left(\frac{\partial P}{\partial R_c} - \frac{\partial P}{\partial R_r}\right)\right|_{R_0}}$$

which is completely equivalent to inequality (11), provided that

$$\frac{\partial P}{\partial R_c} = 1, \qquad \frac{\partial P}{\partial R_r} = -C_r$$

and

$$\frac{\partial P}{\partial R_m} = -C_m.$$

Analogously, the variation of the performance function when the WD-rule is used instead of the WR-rule can be expressed as

$$\begin{aligned} P_{WD} - P_{WR} \cong {} & \left.\left(\frac{\partial P}{\partial R_r} - \frac{\partial P}{\partial R_m}\right)\right|_{R_1} \\ & \cdot R''_{rm} - \left.\left(\frac{\partial P}{\partial R_c} - \frac{\partial P}{\partial R_r}\right)\right|_{R_1} \cdot R''_{rc} \end{aligned}$$

which is positive if

$$\frac{R''_{rc}}{R''_{rm}} < \frac{\left.\left(\frac{\partial P}{\partial R_r} - \frac{\partial P}{\partial R_m}\right)\right|_{R_1}}{\left.\left(\frac{\partial P}{\partial R_c} - \frac{\partial P}{\partial R_r}\right)\right|_{R_1}}.$$

In the last expression, all the derivatives are evaluated in the point $R_1 = (\underline{R}'_c, \underline{R}'_r, \underline{R}'_m)$, corresponding to the rates obtained when using the WR-rule with $\sigma = \sigma^*$.

B. Evaluating the Optimal Reject Thresholds

In the general case, the optimal values σ^* and δ^* of the reject thresholds are determined by equating to zero the total derivatives of P with respect to σ and δ and by solving the resulting equations. The total derivatives of P can be written in the form

$$\begin{aligned} \frac{dP}{d\sigma} &= \frac{\partial P}{\partial R'_c} \cdot \frac{dR'_c}{d\sigma} + \frac{\partial P}{\partial R'_r} \cdot \frac{dR'_r}{d\sigma} + \frac{\partial P}{\partial R'_m} \cdot \frac{dR'_m}{d\sigma} \\ \frac{dP}{d\delta} &= \frac{\partial P}{\partial R''_c} \cdot \frac{dR''_c}{d\delta} + \frac{\partial P}{\partial R''_r} \cdot \frac{dR''_r}{d\delta} + \frac{\partial P}{\partial R''_m} \cdot \frac{dR''_m}{d\delta}. \end{aligned}$$

Taking into account (5a), (5b), (6a), (6b), (15a), (15b), and (16), the above expressions become

$$\begin{aligned} \frac{dP}{d\sigma} &= \left(\left.\frac{\partial P}{\partial R'_r}\right|_{R_\sigma} - \left.\frac{\partial P}{\partial R'_c}\right|_{R_\sigma}\right) \cdot D_c(\sigma) \\ &\quad + \left(\left.\frac{\partial P}{\partial R'_r}\right|_{R_\sigma} - \left.\frac{\partial P}{\partial R'_m}\right|_{R_\sigma}\right) \cdot D_m(\sigma) \\ &= \varphi_1(\sigma) \cdot D_c(\sigma) + \varphi_2(\sigma) \cdot D_m(\sigma) \\ \frac{dP}{d\delta} &= \left(\left.\frac{\partial P}{\partial R''_r}\right|_{R_\delta} - \left.\frac{\partial P}{\partial R''_c}\right|_{R_\delta}\right) \cdot D'_c(\delta) \\ &\quad + \left(\left.\frac{\partial P}{\partial R''_r}\right|_{R_\delta} - \left.\frac{\partial P}{\partial R''_m}\right|_{R_\delta}\right) \cdot D'_m(\delta) \\ &= \omega_1(\delta) \cdot D'_c(\delta) + \omega_2(\delta) \cdot D'_m(\delta) \end{aligned}$$

where the partial derivatives are evaluated in $R_\sigma = [R'_c(\sigma), R'_r(\sigma), R'_m(\sigma)]$ and in $R_\delta = [R''_c(\delta), R''_r(\delta), R''_m(\delta)]$, respectively. Equating to zero the expressions of the two total derivatives, and solving the obtained equations with the same technique illustrated in Section III-C, the wanted values σ^* and δ^* can be computed.

References

[1] K. Fukushima, S. Miyake, and T. Ito, "Neocognitron: A neural network model for a mechanism of visual pattern recognition," *IEEE Trans. Syst., Man, Cybern.*, vol. 13, no. 5, pp. 826–834, 1983.

[2] D. J. Burr, "A neural network digit recognizer," in *Proc. Int. Conf. Syst., Man, Cybern.*, IEEE, 1986.

[3] I. Sethi and A. K. Jain, Eds., *Neural Networks and Statistical Pattern Recognition*. Amsterdam: North Holland, 1991.

[4] R. P. Lippmann, "Pattern classification using neural networks," *IEEE Commun. Mag.*, pp. 47–64, Nov. 1989.

[5] Y. Hirose, K. Yamashita, and Y. Hijiya, "Backpropagation algorithm which varies the number of hidden units," *Neural Networks,* vol. 4, pp. 61–66, 1991.

[6] S. Becker and Y. Le Cun, "Improving the convergence of backpropagation learning with second-order methods," in *Proc. 1988 Connectionist Models Summer School,* D. Touretzky, G. Hinton, and T. Sejnowsky, Eds. San Mateo, CA: Morgan Kauffman, 1989, pp. 29–37.

[7] J. Y. Han, M. R. Sayeh, and J. Zhang, "Convergence and limit points of neural network and its application to pattern recognition," *IEEE Trans. Syst., Man, Cybern.,* vol. 19, no. 5, pp. 1217–1222, 1989.

[8] R. P. Brent, "Fast training algorithms for multilayer neural nets," *IEEE Trans. Neural Networks,* vol. 2, no. 3, pp. 346–354, 1991.

[9] S. E. Fahlman, "Faster-learning variations on backpropagation: An empirical study," in *Proc. 1988 Connectionist Models Summer School,* D. Touretzky, G. Hinton, and T. Sejnowsky, Eds. San Mateo, CA: Morgan Kauffman, 1989, pp. 38–51.

[10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by backpropagating errors," *Nature,* vol. 323, no. 9, pp. 533–536, 1986.

[11] C. K. Chow, "On optimum recognition error and reject trade-off," *IEEE Trans. Inform. Theory,* vol. IT-16, no. 1, pp. 41–46, 1970.

[12] A. Chianese, L. P. Cordella, M. De Santo, A. Marcelli, and M. Vento, "A structural method for handprinted character recognition," in *Recent Issues in Pattern Analysis and Recognition, Lecture Notes in Computer Science,* V. Cantoni *et al.,* Eds. New York: Springer-Verlag, vol. 399, pp. 289–202, 1989.

[13] G. Boccignone, A. Chianese, L. P. Cordella, and A. Marcelli, "Using skeletons for OCR," in *Progress in Image Analysis and Recognition,* V. Cantoni *et al.,* Eds. Singapore: World Scientific., 1990.

[14] A. Chianese, L. P. Cordella, M. De Santo, and M. Vento, "Decomposition of ribbon-like shapes," in *Proc. 6th Scandinavian Conf. Image Anal.,* Oulu, Finland, 1989, pp. 416–423.

Luigi Pietro Cordella (M'78) received the Ph.D. degree in physics from the University of Rome, Italy, in 1964.

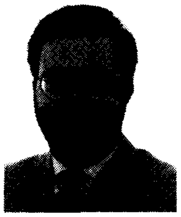
From 1965 to 1969 he was Assistant Professor of Experimental Physics at the University of L'Aquila, Italy. In 1969 he joined the Institute of Cybernetics of the CNR (Italian National Research Council) in Naples, where afterwards he became Head of the Image Analysis Department and Member of the Board of Directors. In 1983 he joined the University of Naples "Federico II," where he is presently Full Professor of Computer Science at the Faculty of Engineering. From 1989–1992 he was Chairman of the Department of Computer Science and Systems of the above University. He has been active in the fields of control systems, electronic circuits, image processing, pattern recognition, computer applications in biomedicine, mathematical models of biological systems, and parallel computing. He has published more than 90 papers and is co-editor of three books. His current research interests include structural pattern recognition, shape analysis, neural networks, document recognition, and parallel computer architectures for vision.

Dr. Cordella has served on the scientific and program committees of several international conferences and has been Chairman of the International Workshop on Visual Form (IWVF), held in 1991, and co-Chairman of IWVF2 in May 1994. He is a member of the IEEE Computer Society and of the IAPR.

Claudio De Stefano was born in Naples, Italy, on October 4, 1961. He received the Laurea degree with honors in electronic engineering in 1989 and the Ph.D. degree in electronic and computer Engineering in 1994, both from the University of Naples "Federico II," Italy.

He is currently a Lecturer in Computer Science at the Department of Computer Science and Systems of the above University. His research interests are in the fields of optical character recognition, neural networks, and parallel computing.

Dr. De Stefano is a member of the IAPR.



Francesco Tortorella was born in Salerno, Italy, on August 10, 1963. He received the Laurea degree with honors in electronic engineering from the University of Naples "Federico II" in 1991. He is currently a Ph.D. student at the Department of Computer Science and Systems of the University of Naples "Federico II."

His research interests are in the fields of optical character recognition, map and document processing, and neural networks.

Mr. Tortorella is a member of the IAPR.

Mario Vento (M'90) was born in Naples, Italy, on January 5, 1960. In 1984 he received the Laurea degree with honors in electronic engineering, and the Ph.D. degree in 1988 in electronic and computer engineering, both from University of Naples "Federico II," Italy.

Since 1989, he has been a Researcher associated with the Department of Computer Science and Systems at the above University. Currently he is Assistant Professor of Artificial Intelligence and Computer Science at the Faculty of Engineering of the University of Naples. His present research interests are in the field of document analysis and recognition, parallel computing, and artificial intelligence.

Dr. Vento is a member of the IEEE Computer Society and IAPR.