An Adaptive Reject Option for LVQ Classifiers

L.P. Cordella, C. De Stefano, C. Sansone, M. Vento

Dipartimento di Informatica e Sistemistica, Università di Napoli
Via Claudio, 21 80125 Napoli (Italy)

Abstract. A reject rule devised for a neural classifier based on the Learning Vector Quantization (LVQ) paradigm is presented. The reject option is carried out adaptively to the specific application domain. It is assumed that a performance function P is defined which, taking into account the requirements of a given application expressed in terms of classification, misclassification and reject costs, evaluates the quality of the classification. 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. Implementation and performance of the rule are illustrated.

1 Introduction

Neural networks have been widely used in the recent past in many application areas. In Pattern Recognition, neural networks revealed to be very interesting for building classifiers with good performance and trainable in a flexible way. To this concern, various learning algorithms have been widely studied and criteria for selecting and sorting the training set have been defined [1-4]. Mostly investigated topics concern techniques for obtaining better convergence rates and criteria for stopping the learning phase when an acceptable trade-off between generalization power and specialization degree of the net has been achieved.

Anyway, in real recognition problems, the samples belonging to the real world (data set) can be affected by distortions that make them quite different from the ones belonging to the training set; therefore, even a well trained network, when attempting the classification of a distorted sample, risks to misclassify it. This problem has been addressed for other kinds of classifiers by introducing a reject option essentially based on the identification of not reliable classifications [5]; the reject decision is made by evaluating the advantage of rejecting a sample instead of running the risk to misclassify it. It appears desirable that this advantage is measured 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.

In [6] we illustrate the rationale of a method for determining the optimal reject threshold value to be used with a neural classifier in a given application domain, in order to get the best trade-off between reject and misclassification rates. In this paper, the specialization of the method in case of an LVQ classifier is discussed. We suggest to measure the performance of a classifier by means of a performance function $P=P(R_c,R_r,R_m)$ defined in terms of recognition rate (R_c), reject rate (R_r) and misclassification rate (R_m). No hypotheses are in principle necessary on the form of P, but the one that it effectively represents the quality of the classification. For several applications it can be assumed that the cost of a correct classification, of a misclassification and of a reject doesn't 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:

$$P = R_c - C_r R_r - C_m R_m$$

where C_r is the cost of a reject and C_m is the cost of a misclassification, both normalized, for the sake of simplicity, with respect to the cost (actually a gain!) of a correct classification. The values of the cost coefficients can be assigned by taking into account the specific application domain in which the classifier is used.

The proposed approach implies the introduction of a rule according to which an input sample is rejected if the value of a suitable parameter, effectively representing the classification reliability (*reliability parameter*), is lower than a given threshold. The selection of the threshold is made in such a way to maximize the performance function P in the considered context. This last aspect is especially important since, for different applications, both the form of P and the values to be assigned to the cost coefficients may be different. Moreover, the implementation of the method depends on the classifier architecture.

To evaluate the improvement of the performance function P obtained by applying the reject rule, we consider the occurrence densities of both correctly classified and misclassified patterns (in the following quoted as D_c and D_m respectively), obtained without using any reject option, as a function of the chosen reliability parameter.

It is worth noting that the distributions are computed after the training of the classifier, on a set S of labeled samples.

2 Reject Rules for LVQ Classifiers

In case of LVQ neural net based classifiers [7], the output vector is made of the distance values between the input sample and the prototypes of every class (corresponding to neurons belonging to the Kohonen layer): classification could thus be performed by selecting the class whose representative prototype has the smallest distance from the input sample (Winner-Takes-All rule). However, in this way, no samples would be rejected and consequently it would be impossible to affect the reliability of the classification.

In order to introduce the reject option, the reliability parameter can be determined by taking into account the characteristics of the output vector provided by the

network. We have experimented with three different parameters which revealed to be particularly appropriate for different situations in the feature space.

The selection of the best parameter for a given situation is carried out a-posteriori: first the threshold value maximizing P is determined for each parameter and then the parameter corresponding to the highest value of P is selected.

Let us now consider an LVQ classifier, and let C the number of classes to be recognized, N the number of neurons per class and w_{ij} the j-th neuron standing for the j-th prototype belonging to the i-th class. The distance between the input sample x and the class i is given by:

$$\delta_i=\min_j (d(w_{ij},x)) \qquad j=1..N$$

where $d(w_{ij},x)$ is the Euclidean distance between w_{ij} and x.

According to the Winner-Takes-All rule (W-rule) applied to the net output vector, the input sample x would be assigned to the class k if:

$$\delta_k=\min_i (\delta_i)=\delta_{WIN} \qquad i=1..C.$$

Relatively small values of δ_{WIN} mean that the input sample is very close to the prototype, and consequently a high classification reliability has been achieved. On the basis of this simple consideration the first reliability parameter has been defined as:

$$\rho_1=1-\delta_{WIN}/\delta_{MAX}$$

where δ_{MAX} represents the maximum value assumed by δ_{WIN} on the set S. Even if ρ_1 showed to work well in a set of cases, it fails when the input sample is close to the winner prototype, but placed in an overlapping region between two classes. In this case, comparing δ_{WIN} with the distance δ'_{WIN} of the input sample from the second winner class, can give more useful information. On this basis, we have introduced two more parameters:

$$\rho_2=(\delta'_{WIN}-\delta_{WIN})/(\max(\delta'_{WIN}-\delta_{WIN})) \quad \text{and} \quad \rho_3=1-\delta_{WIN}/\delta'_{WIN}$$

where the maximum has been evaluated on the set S and

$$\delta'_{WIN}=\min_{i,\, i\neq k} (\delta_i) \qquad i=1..C.$$

Using each of the above parameters, we can introduce a reject option by means of a classification rule (WR-rule), which rejects a sample if the corresponding value of the considered reliability parameter is lower than a given threshold s. In the following discussion, we will indicate the generic reliability parameter with the symbol ρ, since all the general considerations are independent on the specific parameter used. Note that, by definition, $0 \leq \rho \leq 1$.

Let us define the occurrence density curves $D_c(\rho)$ and $D_m(\rho)$ so that

$$R_c = \int_0^1 D_c(\rho)d\rho \qquad \text{and} \qquad R_m = \int_0^1 D_m(\rho)d\rho$$

provide recognition rate and misclassification rate respectively, according to the W-rule. The introduction of the threshold s has two opposite effects on the performance function P. On the one hand, the WR-rule classifies only that percentage R'_c of the samples correctly classified with the W-rule for which the value of P is greater than s and rejects the remaining R'_{rc} determining a decrease of P whose amount is $[R'_{rc}(1+C_r)]$. On the other hand, the WR-rule misclassifies only that subset of the samples misclassified with the W-rule for which the value of P is

greater than s (their percentage be R'_m), while the remaining R'_{rm} are rejected; this implies an increase of P whose amount is $[R'_{rm}(C_m-C_r)]$.

Thus, the use of the WR-rule gives rise to a positive variation of P if:

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

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.

The performance function P, when using the WR-rule, can be written in the form:

$$P_{WR}(s) = R'_c-C_rR'_r-C_mR'_m \qquad (1) \qquad \text{with: } R'_r = R'_{rm} + R'_{rc}$$

Note that P_{WR} depends on s through R'_{rc} and R'_{rm} which are integral functions of s. The method for determining the optimal value of s by maximizing relation (1), is not discussed here. On the analogy of (1), in the following P_W will denote the performance function when using the W-rule.

3 Experimental Results and Conclusions

The classification rule and the reliability parameters discussed above have been experimented in the framework of two tests. In Test 1 we used the same type of data chosen by Xu [8], grouped in four clusters in a bidimensional feature space, each representing a different class. The clusters, all having the same a-priori probability, were generated by two independent Gaussian distributions with variance 0.1, and are respectively centered in (0,1),(0,-1),(1,0),(-1,0). In Test 2, in order to simulate a real situation, a noisy component made of samples of the four classes uniformly distributed within the range [-2.5,2.5][-2.5,2.5] was added to the clusters. In Test 1, the training set was made of 2000 samples (500 for each cluster) while, in Test 2, 2223 samples (including about 10% of noise) were used. Two test sets, with the same number of samples, were generated according to the same criteria.

As the number of clusters per class is 1, we chose N = 1, but also carried out a test in which N was overdimensioned (N = 5). The networks were trained with the basic LVQ1 algorithm [7]. The algorithm basic version was used, instead of its improvements, e.g. [8], because our main goal was just to demonstrate that P can be improved by introducing the reject option.

In Table 1, the recognition rates achieved on the considered training and test sets by a Bayesian classifier (computed from the known probability density functions) and by the LVQ net without reject option are compared. In Table 2, the percentage increment I of P, achieved after introducing the reject option, is shown for Test 1 and Test 2. The value of $I = ((P_{WR} - P_W) / (\max(P_{WR}) - P_W)) \cdot 100$ is reported for each of the three considered reliability parameters and for different values of C_r and C_m. The quantity $(\max(P_{WR})-P_W)$ is the maximum increment of P, corresponding to the optimal case in which all the misclassified samples are rejected while the recognition rate remains unchanged. The values considered for C_r and C_m were chosen within the sets {3,4,5} and {7,9,11,13,15} respectively. This choice seemed adequate to include a bunch of possible real situations and correspond to the hypothesis that, in practical cases, C_r is at least three times greater than the cost of a correct classification, while C_m is at least twice C_r. Referring to Table 2, note that the use of

	TEST 1		TEST 2		
	Bayes	**LVQ (N=1)**	**Bayes**	**LVQ (N=1)**	**LVQ (N=5)**
Training set	97.5	97.5	89.7	89.6	88.6
Test set	97.8	97.7	90.1	90.1	89.1

Table 1. The recognition rates achieved by the Bayesian and LVQ classifiers on training set and test set, for the two considered experiments.

TEST 1		**Reject on ρ_1**				**Reject on ρ_2**				**Reject on ρ_3**			
C_m	C_r	**Clas.**	**Mis.**	**Rej.**	**I**	**Clas.**	**Mis.**	**Rej.**	**I**	**Clas.**	**Mis.**	**Rej.**	**I**
7	**3**	97.7	2.3	0	***0***	97.2	1.7	1.1	***2.2***	97.1	1.5	1.4	***4.3***
9	**3**	97.7	2.3	0	***0***	96.6	1.2	2.2	***13.8***	97.1	1.5	1.4	***13.8***
11	**3**	97.7	2.3	0	***0***	96.6	1.2	2.2	***21.7***	96.8	1.3	1.9	***20.6***
13	**3**	97.7	2.3	0	***0***	96.6	1.2	2.2	***26.5***	96.8	1.3	1.9	***24.8***
15	**3**	97.7	2.3	0	***0***	96.6	1.2	2.2	***29.7***	96.8	1.3	1.9	***27.5***
9	**4**	97.7	2.3	0	***0***	97.2	1.7	1.1	***1.7***	97.1	1.5	1.4	***4.3***
11	**4**	97.7	2.3	0	***0***	97.2	1.7	1.1	***8.7***	97.1	1.5	1.4	***12.4***
13	**4**	97.7	2.3	0	***0***	96.6	1.2	2.2	***18.8***	96.8	1.3	1.9	***18.4***
15	**4**	97.7	2.3	0	***0***	96.6	1.2	2.2	***23.7***	96.8	1.3	1.9	***22.5***
11	**5**	97.7	2.3	0	***0***	97.2	1.7	1.1	***2.2***	97.1	1.5	1.4	***4.3***
13	**5**	97.7	2.3	0	***0***	97.2	1.7	1.1	***8.1***	97.1	1.5	1.4	***11.4***
15	**5**	97.7	2.3	0	***0***	96.6	1.2	2.2	***17.0***	97.1	1.5	1.4	***15.6***

TEST 2		**Reject on ρ_1**				**Reject on ρ_2**				**Reject on ρ_3**			
C_m	C_r	**Clas.**	**Mis.**	**Rej.**	**I**	**Clas.**	**Mis.**	**Rej.**	**I**	**Clas.**	**Mis.**	**Rej.**	**I**
7	**3**	88.7	5.5	5.8	***30.2***	88.7	8.0	3.3	***6.8***	88.5	7.2	4.3	***12.6***
9	**3**	88.3	5.1	6.6	***36.7***	88.2	7.7	4.1	***10.7***	86.0	5.4	8.6	***18.3***
11	**3**	88.3	5.1	6.6	***39.9***	84.7	6.2	9.1	***11.3***	84.1	4.7	11.2	***22.6***
13	**3**	87.8	4.7	7.5	***42.9***	84.7	6.2	9.1	***16.7***	84.1	4.7	11.2	***28.6***
15	**3**	83.9	3.6	12.5	***43.2***	84.7	6.2	9.1	***20.2***	79.1	3.2	17.7	***31.1***
9	**4**	88.7	5.5	5.8	***30.4***	88.7	8.0	3.3	***6.8***	88.5	7.2	4.3	***12.7***
11	**4**	88.3	5.1	6.6	***35.9***	88.2	7.7	4.1	***9.8***	86.0	5.4	8.6	***16.2***
13	**4**	88.3	5.1	6.6	***38.8***	84.7	6.2	9.1	***8.4***	84.8	4.9	10.3	***22.0***
15	**4**	87.9	4.8	7.3	***42.4***	84.7	6.2	9.1	***13.8***	84.1	4.7	11.2	***25.3***
11	**5**	88.7	5.5	5.8	***30.3***	88.7	8.0	3.3	***6.9***	88.5	7.2	4.3	***12.7***
13	**5**	88.3	5.1	6.6	***35.3***	88.2	7.7	4.1	***9.2***	86.0	5.4	8.6	***14.8***
15	**5**	88.3	5.1	6.6	***38.0***	84.7	6.2	9.1	***6.0***	86.0	5.4	8.6	***21.0***

Table 2. Values of I for the three considered parameters in case of Test 1 and Test 2, for an LVQ net with N = 1.

ρ_1 with the data set of Test 1, doesn't allow to locate any threshold determining an improvement of P. This can be attributed to the symmetry of the chosen data, for which the misclassified samples have almost always a distance from the winner prototype comparable with the distance of the correctly classified samples. Conversely ρ_2 and ρ_3 perform well and, for $C_m/C_r > 3$, ρ_2 makes the increment of P greater than that generated by ρ_3. On the contrary, on Test 2, ρ_1 gives the greatest increment of P: in this case the distance of a sample from the prototype is a good indicator of the classification reliability. The above results show that, as expected, we cannot select a parameter performing better than the others independently of the distribution of the samples and/or of the cost values in the specific domain. As for N = 5, in Test 2 the recognition rate obtained with the W-rule decreases, as expected and shown in Table 1, but P still improves: the parameter ρ_1 performs better than the other parameters, with I ranging from 25.6% to 38.3%.

In conclusion, the method confirmed to be 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.

References

1. S. Becker, Y. Le Cun: Improving The Convergence of Back-Propagation Learning With Second Order Methods, in Proc. of the 1988 Connectionist Models Summer School, D. Touretzky, G. Hinton, and T. Sejnowsky Eds., San Mateo, CA: Morgan Kauffman, pp. 29-37, 1989.
2. J. Y. Han, M. R. Sayeh, J.Zhang: Convergence and Limit Points of Neural Network and its Application to Pattern Recognition, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 19, No. 5, pp. 1217-1222, 1989.
3. R. P. Brent: Fast Training Algorithms for MultiLayer Neural Nets, IEEE Transactions on Neural Networks, Vol. 2, No. 3, pp. 346-354, 1991.
4. S. E. Fahlman: Faster-Learning Variations on Back-Propagation: An Empirical Study, in Proc. of the 1988 Connectionist Models Summer School, D. Touretzky, G. Hinton, and T. Sejnowsky Eds., San Mateo, CA: Morgan Kauffman, pp. 38-51, 1989.
5. M. E. Hellman: The Nearest Neighbor Classification Rule with a Reject Option, IEEE Transactions on Systems, Science and Cybernetics, Vol. 6, No. 3, pp. 179-185, 1970.
6. L.P. Cordella, C. De Stefano, F. Tortorella, M. Vento: A Method for Improving Classification Reliability of Multi-layer Perceptrons, to appear in IEEE Transactions on Neural Networks, Vol. 6, No. 5, 1995.
7. T. Kohonen: The Self-Organizing Map, Proceedings of the IEEE, Vol. 78, No. 9, pp. 1464-1480, 1990.
8. L. Xu, A. Krzyzak, E. Oja: Rival Penalized Competitive Learning for Clustering Analysis, RBF net and Curve Detection, IEEE Transactions on Neural Networks, Vol. 4, No. 4, pp. 636-649, 1991.