An Empirical Study for the Multi-class Imbalance Problem with Neural Networks

R. Alejo[1,2,3], J.M. Sotoca[3], and G.A. Casañ[3]

[1] Centro Universitario UAEM Atlacomulco, Universidad Autónoma del Estado de México
Km. 60 Carretera Toluca - Atlacomulco (México)
[2] Lab. Reconocimiento de Patrones, Instituto Tecnológico de Toluca
Av. Tecnológico s/n, 52140 Metepec (México)
[3] Dept. Llenguatges i Sistemes Informàtics, Universitat Jaume I
Av. Sos Baynat s/n, 12071 Castelló de la Plana (Spain)

Abstract. The latest research in neural networks demonstrates that the class imbalance problem is a critical factor in the classifiers performance when working with multi-class datasets. This occurs when the number of samples of some classes is much smaller compared to other classes. In this work, four different options to reduce the influence of the class imbalance problem in the neural networks are studied. These options consist of introducing several cost functions in the learning algorithm in order to improve the generalization ability of the networks and speed up the convergence process.

Keywords: Multi-class imbalance; backpropagation; cost function.

1 Introduction

The input data for an artificial Neural Network (NN) with supervised learning consists in training samples (TS). A TS is a collection of samples previously analyzed by a human expert, that characterizes a problem. A formal definition is $\text{TS} = \text{TS}_1 \cup \text{TS}_2 \cup ... \cup \text{TS}_m$, where $\text{TS}_i = (\mathbf{x}_j, \varphi(\mathbf{x}_j))$, $j = 1, ..., n_i$, and $\mathbf{x}_j = [x_1, x_2, ..., x_d]^T$ is the characteristic vector of each sample, $\varphi(\mathbf{x}_j)$ is the class and, n_i the number of samples in the class i.

Most supervised learning methods as NN are designed to work with reasonably balanced TS [1]. However, most real world applications are not balanced [2].

A TS is said to be imbalanced, when several classes (minority classes) are underrepresented in comparison to other (majority classes) classes, i.e., if $\|\text{TS}_i\| << \|\text{TS}_j\|$, $i \neq j$; $i, j = 1, ..., m$, where m is the number classes in the TS. The class imbalance problem in NN [3] has been formulated as follows: the majority class dominates the training process, and the class elements from the less representative class can be ignored. Then the convergence process for the minority class is very slow [4].

Numerous studies have been presented to improve the classification accuracy when the NN has been trained with imbalanced TS [5], but much work has been done with only two classes [1]. Nevertheless, the class imbalance problem also exists in datasets with several classes [4].

J. Ruiz-Shulcloper and W.G. Kropatsch (Eds.): CIARP 2008, LNCS 5197, pp. 479–486, 2008.

The most popular strategies to face the class imbalance problem are the techniques of *under-sampling* (which eliminates samples in the majority class) and *over-sampling* (replicates samples in the minority class) [2].

In recent works [5], the class imbalance problem is considered as a *cost-sensitive* problem where the classification error cost should be different for each class [6]. The main disadvantage of this strategy is the need of a priori information of the problem, therefore the error cost must be quantified before the process.

In this paper, the back-propagation algorithm is analyzed and modified to deal with the multi-class imbalance problem. It is focused mainly in the evaluation of different cost functions designed to improve the NN performance.

2 Neural Networks

Three differents types of NN have been used: Radial Basis Function Neural Network (RBFNN), RBF-RVFLN (Random Vector Functional Link Net) and Multilayer Perceptron (MLP). MLP and RBFNN are two well-known NN in the pattern recognition field [7]. In spite of being very similar feedforward neural networks with nonlinear layers and the fact that there is always a RBFNN capable to equate to the MLP accuracy or vice versa, they have important differences [8]:

1. RBFNN has a single hidden layer whereas the MLP can have several.
2. Each hidden node and output node of a MLP has the same neural model. In RBFNN the hidden nodes and output nodes have different neural models.
3. While the parameters of the activation function for each hidden node in RBF are calculated using euclidean norm between the input vector and the prototype vector, the parameters for the activation function of each hidden unit in MLP are calculated as the sum of the product between the input vector and the synaptic weights of each unit.
4. MLP construct a global approximation for the nonlinear association of input-output while RBFNN construct a local approximation.

The RBF-RVFLN is a variant of the RBFNN. The RVFL of Pao [9] is added to the RBFNN in order to obtain the last one, actually, it gives extra connectivity of the FLN along with any functions put into the offset hidden neurons. The addition of connections between the hidden neurons adds extra learning power [7].

2.1 The Backpropagation Algorithm and the Class Imbalance Problem

Empirical studies of the backpropagation algorithm [3], show that the class imbalance problem generates unequal contributions to the mean square error (MSE) in the training phase. Obviously the major contribution to the MSE is produced by the majority class.

Let us consider a TS with two classes such that $N = \sum_i^m n_i$ and n_i is the number of samples from class i. Suppose that the MSE by class may be expressed as

$$E_i(U) = \frac{1}{N} \sum_{n=1}^{n_i} \sum_{p=1}^{L} (y_p^n - F_p^n)^2 \,, \tag{1}$$

(L is the number outputs neurons in the NN) so that the overall MSE can be expressed as

$$E(U) = \sum_{i=1}^{m} E_i(U) = E_1(U) + E_2(U)\,. \tag{2}$$

If $n_1 << n_2$ then $E_1(U) << E_2(U)$ and $\|\nabla E_1(U)\| << \|\nabla E_2(U)\|$. Then $\nabla E(U) \approx \nabla E_2(U)$. So, $-\nabla E(U)$ it is not always the best direction to minimize the MSE in both classes.

Considering that the TS imbalance problem affects negatively the backpropagation algorithm due to the disproportionate contributions to the MSE, it is possible to consider a cost function (γ) that balances the TS class imbalance as follows

$$\begin{aligned} E(U) &= \textstyle\sum_{i=1}^{m} \gamma(i) E_i(U) = \gamma(1)E_1(U) + \gamma(2)E_2(U) \\ &= \tfrac{1}{N} \textstyle\sum_{i=1}^{m} \gamma(i) \sum_{n=1}^{n_i} \sum_{p=1}^{L} (y_p^n - F_p^n)^2\,, \end{aligned} \tag{3}$$

where $\gamma(1)\|\nabla E_1(U)\| \approx \gamma(2)\|\nabla E_2(U)\|$ avoiding that the minority class would be ignored in the learning process [10]. This work analyzes the following cost functions:

- **Option 0:** $\gamma(i) = 1$, is the sum-of-squares cost function, without modifications.
- **Option 1:** $\gamma(i) = n_{max}/n_i$; where $i = 1, ..., m$ and n_{max} is the number of samples of the majority class.
- **Option 2:** $\gamma(i) = N/n_i$; where N is the total number of samples.
- **Option 3:** $\gamma(i) = \|\nabla E_{max}(U)\|/\|\nabla E_i(U)\|$, where $\|\nabla E_{max}(U)\|$ is the majority class. This function is a simplification of [3].
- **Option 4:** when $\gamma(i)$ is included, the data probability distribution is altered [11]. To reduce the cost function impact in the data distribution probability, the cost function value is diminished gradually [10] in this way

$$\gamma(i)^t = \begin{cases} \gamma(i)^{(t-1)} \cdot (1-\varepsilon) & \text{Si } [\gamma(i)^{t-1}] > 1 \\ 1 & \text{in other case} \end{cases} \tag{4}$$

  where t is the current iteration and ε in (0,1). The imbalance effect is reduced in early iterations, later $\gamma(i)$ reduces its value to avoid modifying the data distribution probability. The $\gamma(i)$ function is initialized using Option 1, and $\varepsilon = 1/(\text{No. of iterations})$.

3 Data Sets and Methodology

In the experimental phase, Cayo, Ecoli6, Feltwell and Satimage databases with multiple classes are used. Feltwell is related to an agriculture region near to Felt Ville, Feltwell (UK) and is divided in training data (5124 samples) and test data (5820 samples). Cayo which represents a particular region in the gulf of Mexico was partitioned using the holdout method(50% training and 50% test). Both are remote sensing images.

Ecoli6 is obtained from E.coli, a biological database created by the *Institute of Molecular and Cellular Biology* from Osaka University, Japan. It was, originally, distributed in eighth classes, but in this work classes 7 and 8 are eliminated, since these

Table 1. A brief summary of the some basic characteristics of the databases

Dataset	Size	Attr.	Class	Class distribution
Cayo	6019	4	11	838/293/624/322/133/369/324/722/789/833/772
Ecoli6	332	7	6	5/143/77/52/35/20
Feltwell	10944	15	5	3531/2441/896/2295/1781
Satimage	6430	36	6	1508/1531/703/1356/625/707

Table 2. Confusion matrix

	Real Classes			
Predicted Classes	1	2	k	total (n_{i+})
1	n_{11}	n_{12}	n_{1k}	n_{1+}
2	n_{21}	n_{22}	n_{2k}	n_{2+}
k	n_{k1}	n_{k2}	n_{kk}	n_{k+}
total (n_{+j})	n_{+1}	n_{+2}	n_{+k}	**n**

only have two samples and it is difficult to apply the method of cross validation. The Ecoli6 database was split using the five-fold cross-validation method (80% training and 20% test). Satimage were obtained from the *UCI Machine Learning Database Repository*, without changes. The data in Satimage is divided into: 4435 training and 200 testing samples. In Table 1, the most important characteristics of each database are summarized.

The NN were trained by the backpropagation algorithm in batch mode. This process has been repeated five times and the results correspond to the average. The learning rate (η) was set to 0.0001 for RBFNN and RBF-RVFLN and 0.9 for the MLP. In the last case only one hidden layer was used. The number of neurons for the hidden layer (in all NN) was established to 16, 15, 6 and 12 for Cayo, Ecoli6, Feltwell and Satimage respectively.

In this study, *Accuracy*, *g-mean* and *Kappa* coefficient are used as measure criteria for performance of the classifiers.

It is common to obtain measure criteria from the confusion matrix where real classes are in columns, whereas predicted ones appears in rows (Table 2). The table built in this way is a general vision assignment, the right ones (diagonal elements) like the wrong ones (elements out of the diagonal). From Table 2, the first measurement criteria is obtained $Accuracy = \frac{\sum_{i=1}^{k} n_{ii}}{n}$, where n is the total number of samples and $\frac{n_{jj}}{n_{+j}}$ is the *Accuracy by class*. The proportions of the samples p_{ij} in the cell (i, j) correspond to the number of samples n_{ij}, i.e, $p_{ij} = n_{ij}/n$. So, define p_{i+} and p_{+j} as $p_{i+} = \sum_{j=1}^{k} p_{ij}$, and $p_{+j} = \sum_{i=1}^{k} p_{ij}$.

The *Kappa* coefficient is used as a quality parameter and takes into consideration the marginal distributions for the confusion matrix. Its value gives us an idea about the right percentage obtained in the classification process, once the random part has been eliminated. It is defined as $\left(Kappa = \frac{p_o - p_c}{1 - p_c}\right)$ with $p_o = \sum_{i=1}^{k} p_{ii}$ is the well predicted percentage, and $p_c = \sum_{i=1}^{k} p_{i+}p_{+j}$ is the random coefficient. Other measure used to

quantify the classifier performance in the class imbalance problem is the geometric mean (*g-mean*) [10]. The geometric mean is defined as $g\text{-}mean = (\prod_{j=1}^{k} \frac{n_{jj}}{n_{+j}})^{\frac{1}{k}}$, this measure magnifies each class contribution in the classifier performance.

4 Experiments and Discussions

As can be seen in Table 1, all databases have different TS imbalance levels. From a moderate imbalance up to a severe imbalance in the same data set. As an example, in Ecoli6 case, classes 1 and 2 represent a severe imbalance between them (majority class has 143 samples and minority class has only 5), while classes 5 and 6, show a reasonable imbalanced problem. A similar case is observed in Cayo, while using Feltwell and Satimage a moderate imbalance between classes can be observed.

First column in Tables 3, 4, 5 and 6, shows the used NN and second one shows the applied evaluation criteria. The following columns contain the observed values due to the applied strategies. The data in parentheses represent the standard deviation.

Cayo database shows a lower performance compared to other databases. In table 3, the zero value of *g-mean* in three network models can be observed. This is because the less representative class (class 5, see Table 1) is ignored in the training process and consequently the class accuracy is zero. Classes 2, 4 and 6 present low accuracy and they are minority classes. Notice that if the cost functions were applied in the training process, the minority class accuracy would tend to reach the balance with the other classes. In the three network models, all strategies improved the outcome and the *g-mean* values is increased in, at least 70%. Likewise, this strategies rise the accuracy no less than the 2% and as well the classifier confidence ($Kappa$ coefficient).

For Ecoli6, the imbalance data does not affect considerably the MLP performance and the strategies are not good enough to improve the outcome as can be seen in Table 4. However, RBFNN and RVLNN models are seriously affected with low *g-mean* values. Option 2 and 3 (in RBFNN and RBF-RVFLN) have better performance: no lower than 18% with respect to the *g-mean* of these classifiers. The accuracy is not affected and

Table 3. Classification performance of **Cayo** dataset

MLP		Option 0	Option 1	Option 2	Option 3	Option 4
	Acc	78.95(1.5)	85.71(0.04)	86.41(0)	85.2(0.5)	79.73(3.8)
	$gmean$	0(0)	82.86(0.3)	85.7(0.02)	80.6(0.3)	75.3(2.5)
	$Kappa$	0.76(0.02)	0.84(0)	0.85(0)	0.83(0.01)	0.77(0.04)
RBFNN		Option 0	Option 1	Option 2	Option 3	Option 4
	Acc	76.09(4)	79.3(2.5)	82.24(2.6)	81.35(0.1)	78.34(0.76)
	$gmean$	0(0)	76.27(1.9)	78.6(3.04)	71.3(4.3)	73.76(0.9)
	$Kappa$	0.73(0.04)	0.77(0.03)	0.8(0.03)	0.79(0)	0.76(0.01)
RBF-RVFLN		Option 0	Option 1	Option 2	Option 3	Option 4
	Acc	74(3.26)	77.8(0.5)	79.7(2.5)	81.91(2.2)	76.4(1.6)
	$gmean$	0.0(0.0)	73.5(2.9)	74.25(6.5)	71.6(8.32)	70.33(4.4)
	$Kappa$	0.71(0.04)	0.75(0)	0.77(0.03)	0.8(0.02)	0.74(0.02)

Table 4. Classification performance of **Ecoli6** dataset

MLP		Option 0	Option 1	Option 2	Option 3	Option 4
	Acc	87.05(2.7)	83.13(7.1)	82.23(5.9)	85.6(5)	86.43(3.9)
	$gmean$	84.5(5.67)	82.30(7.1)	81.7(5.3)	84.30(6.7)	83.7(3.6)
	$Kappa$	0.82(0.04)	0.77(0.09)	0.76(0.08)	0.8(0.07)	0.81(0.05)
RBFNN		Option 0	Option 1	Option 2	Option 3	Option 4
	Acc	84.04(5.3)	83.73(4.2)	84.63(5)	84.65(5.5)	84.05(1.9)
	$gmean$	66.21(37.45)	83.9(3.9)	84.3(5.36)	84.3(7.24)	82.81(4.2)
	$Kappa$	0.78(0.07)	0.78(0.05)	0.79(0.06)	0.79(0.07)	0.78(0.03)
RBF-RVFLN		Option 0	Option 1	Option 2	Option 3	Option 4
	Acc	84.95(4.3)	83.13(7.4)	85.55(4.3)	85.25(5.4)	84.34(5.3)
	$gmean$	62.00(35.4)	84.39(6.2)	83.94(6.2)	84.04(6.7)	85.24(4.3)
	$Kappa$	0.79(0.06)	0.77(0.1)	0.80(0.06)	0.80(0.07)	0.79(0.07)

Table 5. Classification performance of **Feltwell** dataset

MLP		Option 0	Option 1	Option 2	Option 3	Option 4
	Acc	88.59(1.32)	88.82(0.67)	89.27(1.19)	88.86(0.39)	83.84(1.8)
	$gmean$	84.74(3.2)	87.03(0.86)	87.41(1.3)	86.64(0.4)	79.85(2.3)
	$Kappa$	0.85(0.02)	0.85(0.01)	0.86(0.02)	0.85(0.0)	0.78(0.02)
RBFNN		Option 0	Option 1	Option 2	Option 3	Option 4
	Acc	87.00(0.79)	86.54(1.76)	87.77(1.8)	85.87(2.6)	83.58(1.2)
	$gmean$	81.11(2.5)	84.91(2.16)	86.34(1.73)	82.81(4.7)	80.67(2.7)
	$Kappa$	0.83(0.01)	0.83(0.02)	0.84(0.02)	0.81(0.04)	0.78(0.02)
RBF-RVFLN		Option 0	Option 1	Option 2	Option 3	Option 4
	Acc	86.18(1.75)	88.02(0.81)	87.04(2.9)	88.1(1.84)	85.12(0.93)
	$gmean$	74.04(15.6)	85,88(0.74)	84.49(4.1)	84.71(2.36)	81.92(1,5)
	$Kappa$	0.82(0.02)	0.84(0.01)	0.83(0.04)	0.84(0.02)	0,80(0,01)

the $Kappa$ coefficient is incremented. Option 2 and 3 show the best results in models RBFNN and RBF-RVFLN.

The results with Feltwell database provide interesting conclusions. The *g-mean* values are not significantly affected as regards the TS imbalance problem (see Table 5). Better results in accuracy, *g-mean* values and $Kappa$ coefficient are noticed (except for the Option 4 in MLP and RBFNN, and Option 3 in model RBF). Option 2 will presents the best outcome in models MLP and RBFNN.

Regarding to Satimage database (Table 6), better results in accuracy, *g-mean* values and $Kappa$ coefficient are observed as long as strategies Option 1, 2, and 3 are applied. The best results are obtained with Option 2 strategy and the worst with Option 4 (with the exception of RBF-RVFLN which increases the *g-mean* values while accuracy and the classifier confidence remain equal).

Table 6. Classification performance of **Satimage** dataset

MLP		Option 0	Option 1	Option 2	Option 3	Option 4
	Acc	82.19(0.16)	85.64(0.59)	87.41(0.55)	86.31(0.49)	83.05(0.62)
	$gmean$	50.81(2.76)	84.23(0.60)	86.31(0.41)	83.64(1.13)	77.20(4.06)
	$Kappa$	0.78(0.0)	0.82(0.01)	0.85(0.01)	0.83(0.01)	0.79(0.01)
RBFNN		Option 0	Option 1	Option 2	Option 3	Option 4
	Acc	83.21(0.74)	83.33(0.65)	85.63(0.54)	84.27(1.36)	81.36(1.32)
	$gmean$	76.82(1.84)	81.63(1.17)	84.57(0.51)	81.35(2.56)	78.92(2.18)
	$Kappa$	0.79(0.01)	0.80(0.01)	0.82(0.01)	0.81(0.02)	0.77(0.02)
RBF-RVFLN		Option 0	Option 1	Option 2	Option 3	Option 4
	Acc	82.48(1.58)	83.30(1.32)	84.60(0.31)	85.33(0.31)	82.02(1.96)
	$gmean$	73.56(2.48)	81.24(1.25)	82.83(0.48)	82.31(0.63)	79.54(1.62)
	$Kappa$	0.78(0.02)	0.80(0.02)	0.81(0.0)	0.82(0.0)	0.78(0.02)

We can see that in Cayo, Ecoli6 and Feltwell databases (Tables 3, 4 y 5) the MLP model reports a better performance that models RBFNN and RBF-RVFLN. Futhermore, a better tolerance is reported with respect to the TS imbalance. Is also remarkable that the strategy which produces worse results is Option 4 since Option 2 presents better results in both models (MLP and RBFNN) with Cayo, Feltwell and Satimage databases.

5 Conclusions

In this work, the class imbalance problem is analyzed by means of NN trained with the backpropagation algorithm in batch mode using databases with multiple classes. Four strategies have been studied in order to balance the MSE for each class contribution, every strategy consists of using different kind of cost function in the training algorithm.

The proposed strategies help to balance the class accuracy, and increment the overall accuracy and classification confidence as the empirical results show. Briefly, best performance is obtained using Option 2 and 3 whereas worst is generated by Option 4. Notice that MLP performance is also better than RBFNN and RBF-RVFLN models when imbalanced TS is used during the training process without any cost function.

In conclusion, the proposed strategies improve the MLP, RBFNN and RBF-RVFLN models performance over the less representative classes contained in the TS, reducing the class imbalance problem in the training process and so improving the performance during the test phase.

Future research must consider the relationship between TS imbalance and data complexity (overlapping, noise or decision frontiers). Also severe TS imbalance (for example in remote perception images) must be further considered. Eventually, using classifier ensembles with the aim of exploiting the main characteristics of each individual classifier seems an interesting field to be investigated.

Acknowledgment

This work has been partially supported by grants DPI2006-15542-C04-03 from the Spanish CICYT , SEP-2003-C02-44225 from the Mexican CONACyT, and Generalitat Valenciana under the project GV/2007/105.

References

1. Kotsiantis, S., Pintelas, P.: Mixture of expert agents for handling imbalanced data sets. Annals of Mathematics and Computing & TeleInformatics 1(1), 46–55 (2003)
2. Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. Intelligent Data Analysis 6, 429–449 (2002)
3. Anand, R., Mehrotra, K.G., Mohan, C.K., Ranka, S.: An improved algorithm for neural network classification of imbalanced training sets. IEEE Transactions on Neural Networks 4, 962–969 (1993)
4. Bruzzone, L., Serpico, S.B.: Classification of imbalanced remote-sensing data by neural networks. Pattern Recognition Letters 18, 1323–1328 (1997)
5. Zhou, Z.-H., Liu, X.-Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge and Data Engineering 18, 63–77 (2006)
6. Kukar, M., Kononenko, I.: Cost-sensitive learning with neural networks. In: 13th European Conference on Artificial Intelligence, pp. 445–449 (1998)
7. Looney, C.: Pattern Recognition Using Neuronal Networks - theory and algorithms for engineers and scientists, 1st edn. Oxford University Press, New York (1997)
8. Ding, C., Xiang, S.Q.: From multilayer perceptrons to radial basis function networks: a comparative study. In: IEEE Conference on Cybernetics and Intelligent Systems, vol. 1, pp. 69–74 (2004)
9. Pao, Y.-H., Park, G.H., Sobajic, D.J.: Learning and generalization characteristics of the random vector functional-link net. Neurocomputing 6(2), 163–180 (1994)
10. Alejo, R., García, V., Sotoca, J.M., Mollineda, R.A., Sánchez, J.S.: Improving the performance of the rbf neural networks with imbalanced samples. In: Sandoval, F., Gonzalez Prieto, A., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 162–169. Springer, Heidelberg (2007)
11. Lawrence, S., Burns, I., Back, A.D., Tsoi, A.C., Lee Giles, C.: Neural network classification and unequal prior class probabilities. In: Orr, G.B., Müller, K.-R. (eds.) NIPS-WS 1996. LNCS, vol. 1524, pp. 299–314. Springer, Heidelberg (1998)