Training Optimization for Artificial Neural Networks

Primitivo Toribio Luna*, Roberto Alejo Eleuterio**, Rosa María Valdovinos Rosas***,
Benjamín Gonzalo Rodríguez Méndez*

Recepción: 11 de diciembre de 2009
Aceptación: 29 de abril de 2010

* Innovación y estrategias tecnológicas
Centro Universitario Atlacomulco, Universidad Autónoma del Estado de México, Atlacomulco, México.
** Universidad Jaume I, Castelló de la Plana, España.
*** Centro Universitario Valle de Chalco, Universidad Autónoma del Estado de México, Valle de Chalco, México.
Correo electrónico:guffyprimo@hotmail.com, ralejoll@hotmail.com, li_rmvr@hotmail.com y benrome@hotmail.com .

Acknowledgements:
This work has been partially supported by the Spanish Ministry of Science and Education under project CSD2007-00018, UAEMCA-114 and SBI112 from the Mexican SEP, FE28/2009 (103.5/08/3016) of the Mexican PROMEP and the 2703/2008U from the UAEM project.

Optimizaciòn del entrenamiento para Redes Neuronales Artificiales

Resumen. Debido a la habilidad para modelar problemas complejos, actualmente las Redes Neuronales Artificiales (NN) son muy populares en Reconocimiento de Patrones, Minería de Datos y Aprendizaje Automático. No obstante, el elevado costo computacional asociado a la fase en entrenamiento, cuando grandes bases de datos son utilizados, es su principal desventaja. Con la intención de disminuir el costo computacional e incrementar la convergencia de la NN, el presente trabajo analiza la conveniencia de realizar pre-procesamiento a los conjuntos de datos. De forma específica, se evalúan los métodos de grafo de vecindad relativa (RNG), grafo de Gabriel (GG) y el método basado en los vecinos envolventes k-NCN. Los resultados experimentales muestran la factibilidad y las múltiples ventajas de esas metodologías para solventar los problemas descritos previamente.

Palabras clave: redes neuronales artificiales, perceptrón multicapa, redes de función de base radial, máquinas de vectores soporte, pre-procesado de datos.

Abstract. Nowadays, with the capacity to model complex problems, the artificial Neural Networks (NN) are very popular in the areas of Pattern Recognition, Data Mining and Machine Learning. Nevertheless, the high computational cost of the learning phase when big data bases are used is their main disadvantage. This work analyzes the advantages of using pre-processing in data sets In order to diminish the computer cost and improve the NN convergence. Specifically the Relative Neighbor Graph (RNG), Gabriel's Graph (GG) and k-NCN methods were evaluated. The experimental results prove the feasibility and the multiple advantages of these methodologies to solve the described problems.

Key words: artificial neural networks, multilayer perceptron, radial basis function neural network, support vector machine, preprocessing data.

Introduction

The artificial Neuronal Networks (NN) has become popular in many tasks of Machine Learning, Pattern Recognition and Data Mining. For example, Multilayer Perceptron (MP) has been applied in remote sensing, prediction and approach of functions and control. The Radial Basic Function NN offered an alternative to the treatment formal of the NN allows to realize constructive procedures, that is to say, to determine the optimal structure of a neuronal network for a specific application (Barandela and Gasca, 2000) and are used in applications such as function approach, interpolation with noise and classification tasks. On the other hand, the Support Vector Machines (SMV) can be defined as a static network based on kernels, which realizes a linear classification on transformed vectors in a superior dimension space. That is to say, they are separated by means of a hyperplane in the transformed space. The MVS has been applied successfully in several problems related to Pattern Recognition, recognition of the writing and the natural language processing (Vaptnik, 1995). Unfortunately, in these models, the computational cost associated to the learning phase, depends of the Training Set (TS) size.

In this study, we analyze and explore several methods in order to modify and reduce the TS size, eliminating atypical or noisy training samples and correcting possible erroneous identifications of those training samples. These proposals have the goal of decrease the computer cost and to accelerate the learning process. The proposal is taken from the experience obtained with another non-parametric rule, such as the Nearest Neighbor Rule (Lippmann, 1988).

In the experimentation, we employed 14 real-problem databases with different classes number, 6 with two-classes and 6 of multi-class problem. The paper is organized in the following way: Section 2 and 3 give the theoretical background of the neural network and the nearest neighbor rule. The Section 4 shows the preprocessing algorithms here used as solution strategy and, the Section 5 provides the experimental results. Finally, the main conclusions of this work and possible lines for future research are commented in Section 6.

1. Neural Networks

Nowadays, the Multilayer Perceptron (MP) NN with backpropagation of the error is one of the most popular models for classification purposes (Haykin, 1999), (Jain et al., 1996). This model had among other aspects: the capacity for organizing the representation of the knowledge in the hidden layers and their high power of generalization. Typical architecture has three sequential layers: input, hidden and output layer (Cristianini and Shawe-Taylor 2000), (Dasarathy, 1995), (Sherstinsky and Rosalind, 1994). Such that, a MP with one layer can build a linear hyperplane, a MP with two layers can build convex hyperplane, and a MP with three layers can build any hyperplane.

By simplicity of their architecture and the training method, the Radial Basis Function (RBF) NN, is an attractive alternative for MP. The RBF NN is designed with one hidden layer; the neurons are activated by means of non linear radial functions (Gaussian), and in the output layer use linear functions (Sánchez and Alanís, 2006). In this way, the output RBF, is influenced by a nonlinear transformation produced in the hidden layer through the radial function and a linear one in the output layer through the linear continuous function.

Differences between these NN are following:

a) The RBF has a single hidden layer and the MP could had more than one.

b) The activation function of the hidden nodes is different, in the RBF, it depends of the distance between the input vectors and the centoids in the hidden layer, whereas in the MP depends on the product of the input vector and the weights vector.

c) Generally, the hidden layer nodes and the output layer nodes of MP have the same neuronal model while in the RBF it is different.

On the other hand, the Support Vectors Machines (SVM) base their operation on the data space transformation to other of higher dimension space, through of a kernel function, thus, this function finds the hyperplane that maximizes the margin of separation in the pattern classification of different classes (Cristianini and Shawe-Taylor, 2000).

This method has been successful, due to not to suffer the local minimums such as in the MP, the model only depends of data with more information called support vectors. The main advantages of the SVM are:

a) Excellent generalization capacity, because of the structured risk which could be dimmished (Vaptnik, 1995).

b) SVM adjusts few parameters, the model only depends of data with greater information.

c) The parameters estimation is realized through the function optimization with convex cost, which avoids the existence of a local minimum.

d) The solution of the SVM is to spar itself, this is, the majority of the variables are zero in the solution. Thus, the final model can be written like a combination of a very small number of entrance vectors, called support vectors.

2. The Nearest Neighbor Rule

The Nearest Neighbor Rule (NNR), is very popular in Patter Recognition, it bases its operation in considering the nearest patterns, like which they have the greater probability of belonging to a same class (Dasarathy, 1995). NNR has the following characteristics:

a) It is a supervised method, which needs a Training set, which assumes that it was composed by patterns perfectly identified and that represent all the interest classes in the problem.

b) It is a non-parametric method, that is to say, it does not dependent of probabilistic model for the data.

c) It suffers of a considerable computer cost, due to maintain the ME in memory and by examining each training sample in order to do the classification process.

3. Preprocessing algorithms

The NN are very much sensible to any deficiency in the quality and trustworthiness of the data set. In (John, 1997) suggests the cleaning data in order improve the precision levels in the classification process. On the other hand (Guha et al., 1998) defines a procedure to clean TS by means of an algorithm hierarchic non-supervised. A similar line is defended by (Gopalakrishnan et al., 1995) in order to eliminate patterns that motivate slowness in the learning of the MP. On the other hand, (Barandela and Gasca, 2000) demonstrates the benefits to use a methodology based on the NNR to work with samples imperfectly supervised, producing a cleaning adapted of the TS and contributing to the yield of the classifier algorithm.

In this work several techniques based on the NNR and on its variant k-NNR were evaluated, for the pre-processing of the training data, such as the Gabriel's Graph and the Relative Neighbor Graph.

The Gabriel's Graph GG is used for editing the TS (Sánchez et al., 1997). In its operations use a proximity condition between two vertices. In the circumference formed by those two points, there must not be one other point inside. If that condition was true, then the edge would belong to the graph. For a certain V set of n points, where V = p1, p2,…, pn, two points p_i and p_j are Gabriel's neighbor if:

$$dist^2\ (p_i; p_j) < dist^2\ (p_i; p_k) + dist^2\ (p_k; p_j)\ k \neq i; j \qquad (1)$$

Joining all the Gabriel's neighbors in pairs, by means of an edge, the Gabriel graph is obtained. In a geometric sense, both points p_i and p_j are Gabriel neighbor, if the circle with equal diameter to the distance between p_i and p_j do not contain other point $p_k Є V$. The algorithm can be defined as follows:

1. For each pair of points $(p_i; p_j)$; $i; j = 1, 2,..., n$; where $i < j$
2. If $dist^2\ (p_i; p_j) > dist^2\ (p_i; p_k) + dist^2\ (p_k; p_j)$, then, p_i, p_j are not neighbor of Gabriel, and, go to step 1.
3. Else p_i, p_j are marked as Gabriel neighbors.

The k-Nearest Centroid Neighbor rule (k-NCN) (Sánchez et al., 1997), is a modification to the Wilson's Editing. Nevertheless, this algorithm has higher computational cost than the Wilson's Editing, for that their use is limited to small TS. Is $X = x_i, x_j,..., x_n$ a training set, and is p a certain point to which we want to find its nearest centroid k-neighbor.

Now, the first neighbor of p corresponds to his nearest neighbor, whereas the successive neighbors will be chosen if they diminish the distance between p and the centroid of all neighbors selected until this moment. This rule can be formalized as follows:

$$\delta_{k\text{-}NCN}(x) = \varpi \leftrightarrow d(x,\ P_i) = min_i\ d_k\ (x,\ P_j) \qquad (2)$$

where, $min_i\ d_k\ (x,\ P_j)$ is the distance between x and their k'th neighbor chosen by the k-NCN method. In this way, the class assigned to x will be the most voted between the k neighbors of the nearest centroid.

Finally, in this work we use the Relative Neighbor Graph (RNG) (Sánchez et al., 1997) method. In the RNG, an edge belongs to graph if the end points are relative neighbors, that is to say, when there are an intersection of two circumferences, the edges of the arcs and the radio of distance between them, the geometric figure which is formed in the intersection (its physical form is a moon, that is why this drawing is also known as moon) does not contain inside any other point. That is to say, the RNG of a certain points set S has an arc between x and y if:

$$dist(x,\ y) \leq [dist(x,\ z);\ dist(y,\ z)]\ z Є S,\ z \neq x,\ y \qquad (3)$$

4. Experimental results

The experiments were carried out on 12 real data sets taken from the UCI Machine Learning Database Repository (http://archive.ics.uci.edu/ml/). A brief summary is given in the Table 1. For each database, we have estimated the overall accuracy by 5–fold cross–validation: each data set was divided into five equal parts, using four folds as the training set and the remaining block as independent test set.

The experiments have been performed using the Weka Toolkit (Witten, I. and Frank, E, 2005) with the learning algorithms described in Section 2, that is, MLP, SVM and RBF. Each classifier has been applied to the original training set and also to sets that have been preprocessed by the methods RNG, k-NCN and GG. These editing approaches has decrease the computer cost of an Nearest Neighbor Rule classifier (Sánchez et al., 1997).

Accordingly, this paper addresses the problem of selecting prototypes in order to decrease the computer cost of an ANN classifier. The Table 2 reports the percentage of size reduction yielded by the different editing.

With the elimination of atypical/noise test patterns, and the patterns overlapped, we reduce the computational cost

Table 1. Data sets.

Data set	*Number classes*	*Number features*	*Samples training*	*Samples test*
Australian	2	42	552	138
Diabetes	2	8	614	154
German	2	24	800	200
Liver	2	6	276	69
Phoneme	2	5	4323	1081
Sonar	2	60	166	42
Balance	3	4	500	125
Cayo	11	4	4815	1204
Ecoli6	6	7	265	67
Fetwell	5	15	8755	2189
Glass	6	9	171	43
Satismage	6	36	5148	1287

and the learning time in a NN. From the Table 2 we can note that the two-classes TS size was reduced approximately in a 24% and in the multi-class until a 20% in average. Also we can observe that the k-NCN technique was the one where more elements eliminate on two classes data bases, whereas in the Multi-class data sets was better the GG method.

Table 2. Training samples eliminated.

Data set	RNG	k-NCN	GG
Australian	24.49	34.31	27.43
Diabetes	22.69	32.13	23.18
German	23.47	33.72	26.42
Liver	19.86	39.84	31.09
Phoneme	7.83	9.84	12.18
Sonar	17.79	18.51	34.02
Balance	6.43	8.67	8.22
Cayo	14.23	7.77	8.52
Ecoli6	1.46	1.30	7.37
Fetwell	25.35	28.50	39.60
Glass	7.22	9.49	18.12
Satismage	61.27	60.81	71.90

Table 3. MP classification results.

Data set	Original TS	RNG	k-NCN	GG
Australian	82.46	82.75	84.05	83.04
Diabetes	75.00	75.65	75.65	75.39
German	70.30	73.90	72.50	71.20
Liver	70.13	64.05	68.11	73.04
Phoneme	80.95	81.23	80.92	81.73
Sonar	86.51	86.52	85.57	76.91
Balance	90.72	89.12	89.60	90.24
Cayo	86.67	86.74	86.49	86.42
Ecoli6	87.04	88.79	87.37	87.67
Fetwell	95.27	95.64	95.53	94.84
Glass	68.26	64.50	67.82	62.63
Satismage	89.55	89.85	89.38	87.19

Table 4. RBF clasification results.

Data set	Original	RNG	k-NCN	GG
Australian	82.17	83.33	83.47	81.01
Diabetes	72.91	74.35	72.78	75.26
German	74.50	72.60	72.30	71.70
Liver	65.21	69.79	61.44	65.21
Phoneme	78.57	77.53	77.97	78.18
Sonar	74.51	74.97	76.42	77.88
Balance	85.92	81.60	86.56	84.48
Cayo	87.22	87.88	87.08	87.97
Ecoli6	85.53	89.07	86.43	86.76
Fetwell	90.33	91.63	90.19	90.46
Glass	67.30	66.38	68.25	66.36
Satismage	84.21	86.01	85.96	85.57

Table 5. SVM clasification results.

Data set	Original	RNG	k-NCN	GG
Australian	84.63	84.34	84.92	84.63
Diabetes	76.95	76.82	74.95	76.49
German	75.40	73.60	74.00	71.00
Liver	58.55	57.97	58.26	57.97
Phoneme	77.36	77.47	77.15	77.59
Sonar	77.27	81.77	79.31	78.37
Balance	88.00	87.84	87.68	86.56
Cayo	66.98	66.58	65.72	68.96
Ecoli6	84.93	87.02	84.64	86.25
Fetwell	91.02	91.20	91.25	90.99
Glass	59.76	55.11	49.56	55.59
Satismage	86.71	86.68	87.77	84.97

5.1 Classification results

In the experimental results showing here, the classification index for the NN trained on TS without preprocessing has also been included as the reference values. On the other hand, the preprocessing methods were firstly applied on the data sets and after that, each NN was trained with these preprocessed data sets.

To evaluate the performance of learning NN, we use the standard evaluation measure in pattern recognition, the Overall Accuracy is:

$$Ac = 1 - \frac{Ne}{Nx} \tag{4}$$

where *Ne* is the mistakes number and *Nx* is the number of training samples.

Table 3, Table 4 and Table 5, show the results obtained with several preprocessing methods (RNG, k-NCN and GG), using MP, RBF and SVM. The results for each original training set (i.e., without preprocessing) has also been included as a baseline. The values marked with bold typeface indicate the best results.

From these results, some initial comments can be drawn. Firstly, for the majority data sets there exist al least one preprocessing method whose classification accuracy is higher than that obtained when using the original TS (without preprocessing). Also, we can observe that, after applying the preprocessing algorithms with methods described previously, the RNG method was the best one maintaining the overall accuracy as much with the MLP as well with the RBN models.

On the other hand, when comparing the overall predictive accuracies of the NN models, we found that the MLP generally has a favorable behavior on the 85% of the data set used and, in opposite way, the accuracy obtained with the SVM can differ from one problem to another (depending of each particular data set), outperforming de original TS only in 67% of the cases. Nevertheless, when the classifier behavior decrease, it is not significant, for example: Liver with MLP (using RNG), Glass with the SVM or k-NCN (using RGN and GG respectively). On the other hand, is possible to observe that with the edition process, the SVM behavior is affected more than the MLP and RBF models, especially when GG strategy is applied.

The Glass data set constitutes a very particular case, in which NN behavior is strongly affected. In all cases, the results obtained without preprocessing always is higher than the overall accuracies obtained with any NN model. A similar situation is observed with German data set when a RBF and SVM are used. This situation could be produced for another complexity data inside of the data set, such as, imbalance problem or if some data cover up other data.

6. Concluding Remarks

In the NN paradigm, the high computer cost associated to the learning process is a serious problem. This cost is related directly with the training data set size. In this work, we proposed to reduce the computational cost, by reducing the training data size when a NN is used. Specifically, we use the MLP, RBF and SVM models. For that, three strategies well known in the nearest neighbor rule context for reducing the training set improving the classifier behavior were used.

The experimental results shown that, in general, the methods used reduce the data bases size at least 20%. This reduction could be translated in an important computational cost diminution associated to the learning process of the network. On the other hand, was possible to observe that in the majority of the data bases the classifier behavior was stayed or increase, few cases shown lost in the classifier effectiveness.

Finally, the strategies proposed in this work can be useful when the computational cost associated to the NN learning is expected (without losing effectiveness in the classification). Nevertheless, the main impact of this proposal is not only the reduction of the computational cost, but the incorporation of a criterion based on the space neighborhood of the data, for identify those samples which give few information to the learning process of the NN.

Future work is primarily addressed to deepening in this study, not only by the importance of the subject, but by its relation with other areas such as the medicine, astronomy or the economy. At the moment, these disciplines are helped by the pattern recognition techniques, especially by the artificial neuronal networks.

Specifically, we have contemplated to deepen in the analysis of data complexity subject (dimensionality, overlapping, representativeness, and probabilistic density), due to the observed results; it suggests that some of these aspects are responsible of the low behavior of the classifier. As second line of investigation, is the idea to generate methods based on the Wilson editing (Wilson, 1972), can work in the hidden space of the neural network. This idea could generate great expectations. In order to obtain this one would be to use a dissimilarity measurement in the transformation space of the training sample and not in the entrance space, such as commonly happens with the Wilson editing and its variants.

References

Barandela, R. & E. Gasca (2000). "Decontamination of training samples for supervised pattern recognition methods", *Lecture Notes in Computer Science*. Springer. 1876.

Cristianini, N. & J. Shawe-Taylor (2000). *An Introduction to Support Vector Machines, Cambridge University Press*, Cambridge, UK.

Dasarathy, B. V. (1994). "Minimal consistent set (MCS) identification for optimal nearest neighbor decision system design", *Transactions on Systems Man and Cybernetics*. 24(3).

Gopalakrishnan, M.; V. Sridhar & H. KrishNamurthy (1995). *Some application of clustering in the desing of neural networks*. Patter Recognition Letters, 16.

Haykin, S. (1999). *Neural Networks, A Comprehensive Foundation*. 2nd ed. Pretince Hall, New Jersey.

Guha, S; R. Rastogi & K. Shim (1998). *CURE: An efficient clustering algorithm for large databases*. ACM SIGMOD International Conference on Management of Data, Seattle, Washington.

Jain, A.; J. Mao & K. Mohiuddin (1996). "Artificial neural networks: A tutorial". *Computer* 29(3).

John, G. H. (1997). *Enhancements to the Data Mining Process*. Ph. D. Thesis, Stanford University.

Lippmann, R. (1988). "An introduction to computing with neural nets", in: *Artificial neural networks: theoretical concepts*. IEEE Computer Society Press, Los Alamitos, CA, USA.

Sánchez, E. N. & Alanís A. Y. (2006). *Redes Neuronales. Conceptos Fundamentales y aplicaciones a control automático*. Pearson-PrenticeHall.

Sánchez J. S.; F. Pla & F. J. Ferri (1997). "Using the nearest centroid neighbourhood concept for editing purpose", in *Proceedings of VII Simposium Nacional de Reconocimiento de formas y Análisis de Imágenes 1*.

Sherstinsky A.; Rosalind P. (1996). "On The Efficiency Of The Orthogonal Least Squares Training Method For Radial Function Networks", *IEEE Transactions on Neural Networks*. 7(1).

Vaptnik V.N. (1995). *The nature of Statical Learning Theory*. Wiley. New York.

Wilson D. L. (1972). *Asymptotic properties of nearest neighbor rules using edited data sets*. IEEE Transaction on Systems, Man and Cybernetics. 2.

Witten, I. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*, Second Edition (Morgan Kaufmann Series in Data Management Systems). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA