ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/220870570

Edited Nearest Neighbor Rule for Improving Neural Networks Classifications

Conference Paper *in* Lecture Notes in Computer Science · June 2010

DOI: 10.1007/978-3-642-13278-0_39 · Source: DBLP

CITATIONS
35

READS
2,569

4 authors, including:

Roberto Alejo
Tecnológico Nacional de Mexico / IT Toluca
87 PUBLICATIONS **1,061** CITATIONS

SEE PROFILE

José Martínez Sotoca
Jaume I University
100 PUBLICATIONS **2,036** CITATIONS

SEE PROFILE

Rosa María Valdovinos
Autonomous University of Mexico State
92 PUBLICATIONS **1,063** CITATIONS

SEE PROFILE

All content following this page was uploaded by José Martínez Sotoca on 09 January 2014.

The user has requested enhancement of the downloaded file.

Edited Nearest Neighbor Rule for Improving Neural Networks Classifications

R. Alejo[1], J.M. Sotoca[1], R.M. Valdovinos[2], and P. Toribio[3]

[1] Institute of New Imaging Technologies
Dept. Llenguatges i Sistemes Informátics, Universitat Jaume I
Av. Sos Baynat s/n, 12071 Castelló de la Plana (Spain)
[2] Centro Universitario UAEM Valle de Chalco, Universidad Autónoma del Estado de México
Hermenegildo Galena No.3, Col. Ma. Isabel, 56615 Valle de Chalco (Mexico)
[3] Centro Universitario UAEM Atlacomulco, Universidad Autónoma del Estado de México
Km. 60 Carretera Toluca - Atlacomulco, 50450 Atlacomulco (Mexico)

Abstract. The quality and size of the training data sets is a critical stage on the ability of the artificial neural networks to generalize the characteristics of the training examples. Several approaches are focused to form training data sets by identification of border examples or core examples with the aim to improve the accuracy of network classification and generalization. However, a refinement of data sets by the elimination of outliers examples may increase the accuracy too. In this paper, we analyze the use of different editing schemes based on nearest neighbor rule on the most popular neural networks architectures.

Keywords: neural networks; editing techniques; reduction training set; accuracy

1 Introduction

Artificial neural networks (ANN) are computational models that have become a popular tool used in remote sensing data analysis, the computer-aided medical diagnosis and the identification of microbiological taxa. However, it is well know that in supervised classification the maximum accuracy depends on the quality of the training data set. In this sense, several approaches in the area of neural networks are towards to training data selection with the aim to improve the performance of the network in terms of speed, computational requirements and classification accuracy.

Most research on this topic has traditionally focused to obtain reduced training data sets (TS) by the identification of two kinds of samples: *i)* core training patterns and *ii)* border training patterns [1, 2]. However, the outliers or noise may introduce a false decision boundary. Consequently, the training data extracted may also be subject to refinement intelligent procedures.

An outlier has traditionally been defined as a prototype that does not follow the same model as the rest of the data [3]. In this context, some approaches to allow remove outliers from the original training set and also cleaning possibles overlapping among classes. This strategy has generality been referred as to editing [4].

The general idea behind almost any editing procedure consists of estimating the true classification of prototypes in the TS to retain only those which are correctly labeled. The first proposal to select a representative subset of prototypes for a further nearest neighbour classification corresponds to Wilson editing algorithm [5], in which a k-NN classifier is used to retain in the TS only good samples (that is, training samples that are correctly classified by the k-NN rule).

In the present work, we study the use of several editing schemes proposed in the literature and their effects on the classification performance of three supervised artificial neural networks. This study mainly tries to show how these methods used in the nearest neighbour rule improve the classification accuracy and generalization of the most popular neural networks architectures. In order to accomplish this, we experiment with two class data sets and multiclass data sets.

The structure of the paper is as follows. Section 2 presents the learning algorithms. In Section 3, we briefly describe the editing algorithms used to reduce the training data sets. Section 4 consists of experiments on real data sets and an exhaustive discussion of results. Finally, we will conclude the main remarks and outline some directions for future work in Section 5.

2 The Classifiers

In this section, we briefly describe the classifiers selected for the present experimental study. All these algorithms work under the assumption that there exists a set of n previously labeled examples (training set, TS), say $\mathbf{X} = \{(\mathbf{x}_1, \omega_1), (\mathbf{x}_2, \omega_2), \ldots, (\mathbf{x}_n, \omega_n)\}$, where each element has an attribute vector $\mathbf{x}_i$ and a class label ω_i.

2.1 Multilayer Perceptron

The multilayer perceptron (MLP) neural network [6] usually comprises one input layer, one or more hidden layers, and one output layer. Input nodes correspond to features, hidden layers are used for computations, and output layers are the problem classes. A neuron is the elemental unit of each layer. It computes the weighted sum of its inputs, adds a bias term and drives the result thought a generally nonlinear (commonly, sigmoid) activation function to produce a single output.

The most popular training algorithm for MLP is the backpropagation, which takes a set of training instances for the learning process. For the given feedforward network, the weights are initialized to small random numbers. Each training instance is passed through the network and the output from each unit is computed. The target output is compared with the output estimated by the network to calculate the error, which is fed back through the network. To adjust the weights, backpropagation uses gradient descent to minimize the squared error between the target output and the computed output. At each unit in the network, starting from the output unit and moving down to the hidden units, its error value is used to adjust weights of its connections so as to reduce the error. This process of adjusting the weights is repeated for a fixed number of times or until the error is small or it cannot be reduced.

2.2 Radial Basis Function

The radial basis function (RBF) [7] neural network, which has three layers, can be seen as an especial kind of multilayer feedforward networks. Each unit in the hidden layer employs a radial basis function, such as Gaussian kernel, as the activation function. The output units implement a weighted sum of hidden unit outputs. The input into an RBF network is nonlinear. The output is linear. The kernel is centered at the point specified by the weight vector associated with the unit. Both the positions and the widths of these kernels are learned from training instances. Each output unit implements a linear combination of these radial basis functions.

An RBF is trained to learn the centers and widths of the Gaussian function for hidden units, and then to adjust weights in the regression model that is used at the output unit. To learn the centers of the Gaussian functions, the k-means algorithm can be used, obtaining k Gaussian functions for each attribute in the instance. After the parameters for the Gaussian function at the hidden units have been found, the weights from these units to the output unit are adjusted using a linear regression model.

2.3 Support Vector Machine

Support vector machines (SVMs) [8] are a set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classifiers. A special property of SVMs is that they simultaneously minimize the empirical classification error and maximize the geometric margin; hence they are also known as maximum margin classifiers.

SVMs map input vectors to a higher dimensional space where a maximal separating hyperplane is constructed. Two parallel hyperplanes are constructed on each side of the hyperplane that separates the data. The separating hyperplane is the hyperplane that maximizes the distance between the two parallel hyperplanes. An assumption is made that the larger the margin or distance between these parallel hyperplanes the better the generalization error of the classifier will be.

3 Algorithms to select training samples

The present section describes the procedures for handling outliers in a TS. This alternatives are based on the employment of a surrounding neighborhood to obtain a filtered TS, that is, to detect and remove outliers from the TS.

3.1 Edited Nearest Neighbor Rule

Wilson [5] developed the Edited Nearest Neighbor (ENN) algorithm in which **S** starts out the same as TS, and then each instance in S is removed if it does not agree with the majority of its k nearest neighbors (with k=3, typically). This edits out noisy instances as well as close border cases, leaving smoother decision boundaries. It also retains all internal points, which keeps it from reducing the storage requirements as much as most other reduction algorithms. Algorithmically, the ENN scheme can be expressed as follows:

1. Let $\mathbf{S} = \mathbf{X}$.
2. For each $\mathbf{x}_i$ in $\mathbf{X}$ do:
   - Discard $\mathbf{x}_i$ from $\mathbf{S}$ if it is misclassified using the k-NN rule with prototypes in $\mathbf{X} - \{\mathbf{x}_i\}$.

3.2 Editing with the nearest centroid neighborhood

The nearest centroid neighborhood (NCN) [9] refers to a concept in which neighborhood is defined taking into account not only the proximity of prototypes to a given input sample but also their symmetrical distribution around it. From this general idea, the corresponding classification rule, the k-nearest centroid neighbours (k-NCN) [10], has been proven to overcome the traditional k-NN classifier in many practical situations. Now the NCN editing (NCNE) approach presented here corresponds to a slight modification of the original work of Wilson and basically consists of using the leaving-one-out error estimate with the k-NCN classification rule.

3.3 Proximity graph based editing

Proximity graph based editing scheme are based on the concepts of Gabriel Graph (GG) and Relative Neighborhood Graph (RNG) [11]. The method consists of applying the general idea of Wilson's editing algorithm [5] but using the graph (GG or RNG) neighbors of each sample, instead of the Euclidean distance-based neighborhood, in order to estimate whether a sample is mislabeled or not. In a few words, the simplest GG and RNG prototypes based editing can be summarized as follows: after computing the graph neighborhood of every sample in the original training set, discard those samples that are misclassified by their graph neighbors (instead of their k nearest neighbors).

This editing technique provides some advantages as compared to conventional methods. Firstly, it considers the neighborhood size as a characteristic which depends on each one of the prototypes in the training set. Secondly, GGE and RNGE provides some kind of information about prototypes close enough but homogeneously distributed around a given sample, which can be specially interesting to detect outliers close to the inter-class or decision boundaries. A more detailed description of GGE and RNGE can be found in [12].

4 Experimental results

The experiments were carried out on ten 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 Overall Accuracy were calculated from the equation:

$$Acc = 1 - \frac{n_e}{n_t} \tag{1}$$

where n_e is the number of misclassified examples and n_t is the total number of testing examples.

The experiments have been performed using the Weka Toolkit [13] 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 NNE, RNGE, NCNE and GGE. These editing approaches has been improved the classification accuracy of an Nearest Neighbor Rule classifier [14]. Accordingly, this paper addresses the problem of selecting prototypes in order to improve the classification accuracy of an ANN classifier.

The Table 2 reports the percentage of size reduction yielded by the different editing algorithms over ten databases. The results show that in the case of the two-classes data sets, the average reduction rate was approximately 26.1% , meanwhile in the situation of multi-class data sets was 8% to 17% (approximately). From this, the NCNE and the GGE achieve the highest rates in the two-class and multiclass data sets, respectively. Consequently, this implies a important decrease in computational cost in the learning phase. This issue is important in applications with high-size data sets.

On other hand, in Feltwell the number of discards was minimal (except for the GGE method). This is due to the measuring criteria used by NNE, NCNE, and RNGE suggest that this data set is not overlapped, i.e., the decision boundary is well defined. However, it is doubtful whether this is really so or if necessary use other measures to locate the prototypes in the overlap region.

Examining the Table 3, the results shown that the editing algorithms obtain similar classification accuracy (Acc) to that of original data sets (i.e., without performing editing). This indicate that, the shrink of training set give a lower computational loads, but also not involve a loss performance. In this sense, although in the case of NCNE no outperform the original data sets (for three classifiers with two-class data set), the differences in such cases are not statistically significant. A final remark from the experiments and perhaps important, refers to that in certain cases the classification accuracy is improved.

In summary, the observations reported in Table 3 suggest that, on multiclass problems the performance editing techniques is better than that of two class problem, and illustrate that these techniques have a tendency to improve the classification accuracy.

5 Conclusion and further extensions

When using an ANN, the presence of mislabelled prototypes can degrade the corresponding classification accuracy. Many models for identifying and removing outliers have been proposed. This paper has reviewed some works in the frame of editing the nearest neighbor rule. A number of experiments over ten real data sets have been carried out in order to evaluate the accuracy of those editing methods. The experiments illustrate that editing techniques have a tendency improve the classification accuracy.

In the other hand, the editing techniques here studied, we can diminish the size of the data bases, and with this the diminution of the computational cost and the learning time in the ANN (RBFNN, MLP and SVM). Especially, in two class problem (was reported more of the 21% of reduction, approximately).

Table 1. Data sets used in the experiments.

Data set	Classes	Features	Samples training	Samples test
Diabetes	2	8	614	154
German	2	24	800	200
Heart	2	25	217	55
Liver	2	6	276	69
Phoneme	2	5	4323	1081
Sonar	2	60	166	42
Cayo	11	4	4815	1204
Ecoli6[1]	6	7	265	67
Fetwell	5	15	8755	2189
Satimage	6	36	5148	1287

[1] Ecoli6 is obtained from Ecoli. In this work classes 7 and 8 have been eliminated since these only have two samples.

Table 2. Size reduction rate using different editing techniques.

Data set	NNE	RNGE	NCNE	GGE
Diabetes	29.75	22.60	32.03	23.08
German	31.71	23.47	33.70	26.46
Heart	35.48	29.03	41.47	31.33
Liver	35.74	29.96	38.98	31.04
Phoneme	11.40	7.83	9.85	12.18
Sonar	18.56	17.96	18.56	34.13
Average	27.11	21.81	29.1	26.37
Cayo	8.20	6.41	9.01	8.20
Ecoli6	14.66	14.28	17.66	18.42
Fetwell	1.77	1.49	1.31	7.38
Satimage	9.53	7.20	9.53	17.98
Average	8.54	7.35	9.38	12.99

Future work is primarily addressed to investigate the potential of these editing methods applied to 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 feature space, such as commonly happens with the Wilson editing and its variants.

Acknowledgment

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 (103.5/09/4195) of the Mexican PROMEP and the 2703/2008U from the UAEM project.

Table 3. Experimental results (*Acc*) for three different artificial neural network architectures.

Neural Network	Data set	Original	NNE	RNGE	NCNE	GGE
MLP	Diabetes	75.01(4.88)	74.74(3.56)	75.66(4.31)	75.65(3.05)	75.39(1.69)
	German	70.30(2.20)	72.50(3.81)	73.90(2.38)	72.50(2.57)	71.20(1.48)
	Heart	82.59(4.83)	81.11(7.57)	78.52(4.83)	78.52(4.65)	80.74(4.46)
	Liver	70.14(4.65)	69.28(5.46)	64.06(4.74)	68.12(8.51)	73.04(3.01)
	Phoneme	80.96(1.64)	81.48(1.19)	81.24(1.23)	80.92(1.34)	81.74(0.89)
	Sonar	86.52(5.03)	82.69(1.99)	86.53(2.78)	85.57(5.64)	76.91(2.86)
	Average	77.59(6.8)	76.97(5.55)	76.65(7.61)	76.88(6.20)	76.50(4.17)
	Cayo	86.59(0.29)	86.29(0.34)	86.18(0.33)	86.48(0.59)	86.48(0.71)
	Ecoli6	86.73(2.97)	87.65(3.27)	85.24(2.47)	87.35(1.67)	86.75(1.23)
	Fetwell	95.11(0.77)	95.53(0.48)	94.80(0.37)	95.88(0.44)	94.90(0.27)
	Satimage	88.78(0.67)	89.62(0.49)	89.49(0.66)	89.68(0.58)	87.29(1.56)
	Average	89.30(4.00)	89.77(4.07)	88.93(4.32)	89.85(4.24)	88.86(4.04)
Neural Network	Data set	Original	NNE	RNGE	NCNE	GGE
RBF	Diabetes	72.91(4.93)	75.13(4.76)	74.35(6.67)	72.78(5.55)	75.26(4.65)
	German	74.50(4.11)	70.80(2.46)	72.60(3.27)	72.30(3.83)	71.70(1.96)
	Heart	80.00(6.06)	81.11(5.14)	80.37(4.26)	82.59(4.65)	79.63(3.46)
	Liver	65.22(2.29)	61.45(3.64)	65.80(4.42)	61.45(2.20)	65.22(3.55)
	Phoneme	78.57(1.29)	77.81(1.20)	77.54(0.78)	77.98(1.71)	78.18(0.66)
	Sonar	74.52(8.18)	72.57(6.61)	74.97(5.68)	76.42(4.06)	77.89(6.25)
	Average	74.29(5.20)	73.15(6.81)	74.27(4.96)	73.92(7.17)	74.65(5.40)
	Cayo	87.26(0.58)	87.81(0.38)	88.14(0.54)	87.86(0.55)	87.71(0.47)
	Ecoli6	85.83(2.37)	84.94(2.09)	86.76(2.82)	87.35(2.75)	84.64(1.23)
	Fetwell	90.63(0.61)	89.99(1.72)	89.99(0.95)	89.99(1.25)	91.08(0.76)
	Satimage	84.01(0.48)	85.86(0.83)	86.05(0.99)	85.84(0.86)	85.70(1.12)
	Average	86.93(2.80)	87.15(2.24)	87.74(1.74)	87.76(1.72)	87.28(2.83)
Neural Network	Data set	Original	NNE	RNGE	NCNE	GGE
SVM	Diabetes	76.95(1.97)	75.65(4.15)	76.82(2.53)	76.56(3.01)	76.56(3.17)
	German	75.40(3.31)	72.30(1.89)	73.60(2.13)	74.00(1.90)	71.00(0.71)
	Heart	84.07(4.46)	81.48(6.93)	81.85(4.22)	81.11(5.14)	82.96(3.56)
	Liver	58.55(1.30)	57.97(0.00)	57.97(0.00)	58.26(0.65)	57.97(0.00)
	Phoneme	77.37(1.08)	77.00(1.14)	77.48(1.13)	77.15(1.23)	77.59(1.04)
	Sonar	79.28(7.07)	82.21(4.01)	81.72(2.81)	79.31(4.06)	78.37(5.57)
	Average	75.27(8.72)	74.42(8.90)	74.90(8.90)	74.40(8.27)	74.06(8.80)
	Cayo	66.94(0.48)	66.37(0.13)	66.62(0.48)	65.36(0.39)	69.13(1.00)
	Ecoli6	83.43(1.51)	83.13(1.37)	85.55(0.75)	84.32(3.03)	84.94(1.08)
	Fetwell	91.01(0.14)	91.25(0.23)	91.22(0.26)	91.15(0.28)	91.07(0.33)
	Satimage	86.53(0.99)	86.54(0.84)	86.40(0.86)	86.65(0.85)	85.45(0.99)
	Average	81.98(10.50)	81.82(10.83)	82.45(10.84)	81.87(11.37)	82.65(9.43)

References

1. Foody, G.: The significance of border training patterns in classification by a feedforward neural network using back propagation learning. JRS **20**(18) (December 1999) 3549
2. Kavzoglu, T.: Increasing the accuracy of neural network classification using refined training data. Environmental Modelling & Software **24**(7) (2009) 850 – 858
3. Sánchez, J., Barandela, R., Marqués, A., Alejo, R., Badenas, J.: Analysis of new techniques to obtain quality training sets. Pattern Recognition Letters **24**(7) (2003) 1015–1022
4. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. Machine Learning **38**(3) (2000) 257–286
5. Wilson, D.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man and Cybernetics **2**(4) (1972) 408–420
6. Bishop, C.M.: Neural Networks for Pattern Recognition. 1 edn. Oxford University Press, USA (January 1996)
7. Buhmann, M.D.: Radial basis functions: theory and implementations. Cambridge University Press, United Kingdom (2003)
8. Vapnik, V.: Estimation of Dependences Based on Empirical Data: Springer Series in Statistics (Springer Series in Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA (1982)
9. Chaudhuri, B.B.: A new definition of neighborhood of a point in multi-dimensional space. Pattern Recogn. Lett. **17**(1) (1996) 11–17
10. Sánchez, J.S., Pla, F., Ferri, F.J.: On the use of neighbourhood-based non-parametric classifiers. Pattern Recogn. Lett. **18**(11-13) (1997) 1179–1186
11. Jaromczyk, J., Toussaint, G.: Relative neighborhood graphs and their relatives. Proceedings of the IEEE **80**(9) (1992) 1502–1517
12. Sánchez, J.S., Pla, F., Ferri, F.J.: Prototype selection for the nearest neighbour rule through proximity graphs. Pattern Recogn. Lett. **18**(6) (1997) 507–513
13. Witten, I., Frank, E.: 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 (2005)
14. Sánchez, J., Barandela, R., Marqués, A., Alejo, R.: Performance evaluation of prototype selection algorithms for nearest neighbor classification. In: SIBGRAPI '01: Proceedings of the XIV Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'01), Washington, DC, USA, IEEE Computer Society (2001) 44–50

View publication stats