Domains of Competence of Artificial Neural Networks Using Measures of Separability of Classes*

Julián Luengo and Francisco Herrera

Dept. of Computer Science and Artificial Intelligence, CITIC-UGR (Research Center on Information and Communications Technology), University of Granada, Granada, 18071, Spain
{julianlm,herrera}@decsai.ugr.es

Abstract. In this work we want to analyse the behaviour of two classic Artificial Neural Network models respect to a data complexity measures. In particular, we consider a Radial Basis Function Network and a Multi-Layer Perceptron. We examine the metrics of data complexity known as *Measures of Separability of Classes* over a wide range of data sets built from real data, and try to extract behaviour patterns from the results. We obtain rules that describe both good or bad behaviours of the Artificial Neural Networks mentioned.

With the obtained rules, we try to predict the behaviour of the methods from the data set complexity metrics prior to its application, and therefore establish their domains of competence.

Keywords: Artificial Neural Networks, Classification, Data complexity, Multilayer Perceptron, Radial Basis Function Networks.

1 Introduction

The use of Artificial Neural Networks (ANNs) is very common nowadays, and they have been applied in several tasks and fields. Due their excellent adjusting capabilities, they have been successfully applied in the Data Mining ambit and many others [15], becoming a referent. Particularly, there exist recent contribution in the field for all the considered models in this work [6,14].

The prediction capabilities of classifiers are strongly dependent on the problem's characteristics. Recently has arisen an emergent field that uses a set of complexity measures applied to the problem to describe its difficulty. These measures quantify particular aspects of the problem which are considered complicated to the classification task [9]. Studies of data complexity metrics applied to particular classifications algorithms can be found in [2,3,10,16].

We are interested in analysing the relationship between ANNs and the data complexity measures based on the separability of classes. We consider two models

* This work has been supported by the Spanish Ministry of Science and Technology under Project TIN2008-06681-C06-01. J. Luengo holds a FPU scholarship from Spanish Ministry of Innovation and Science.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 81–88, 2009.

of ANNs, Radial Basis Function Networks (RBFN) and Multi-Layer Preceptron (MLP).

To perform this study, we have created several binary classification data sets from real world problems, 438 ones, and computed the value of 2 metrics proposed by Ho and Basu [9]. We have analysed the intervals of the separability of classes values related to the created data sets, in which ANN methods performs well or bad, and then formulated a rule for such intervals. The rules try to describe the ranges where some information and conclusions about the behaviour of ANN methods can be stated.

This contribution is organised as follows. In Section 2 we describe the ANNs we have used. In Section 3 the considered complexity measures are described. Next, in Section 4 we include the experimental setup, the obtained results and the rules extracted, along their analysis. Finally, in Section 5 some concluding remarks are pointed out.

2 Preliminaries: Artificial Neural Networks

In this section, we will briefly describe the algorithms used. We have used the following models of ANNs:

- Multi-Layer Perceptron (MLP) with Backpropagation[12]: Classical model of a Multi-Layer Perceptron, with its weights adjusted with backpropagation. This class of networks consists of multiple layers of neurons, interconnected in a feed-forward way. Each neuron in one layer has directed connections to the neurons of the subsequent layer. The units of these networks apply a sigmoid function as an activation function. The parameters are:
  - Hidden Layers: 1 hidden layer.
  - Number of Neurons: 10 neurons.
- Radial Basis Function Network (RBFN)[5]: A RBF is a function which has built into a distance criterion with respect to a center. RBF networks have 2 layers of processing: In the first, input is mapped onto each RBF in the 'hidden' layer. RBF networks have the advantage of not suffering from local minima in the same way as multi-layer perceptrons. The parameters are:
  - Number of Neurons: 50 neurons.

3 Data Complexity Measures Based on the Separability of Classes

In this section we describe the two metrics we have used in this contribution, with their correspondent acronym.

For our study, we will examine two measures of separability of classes from [9] which offer information for the ANN methods. They are described next.

- **N1:** fraction of points on class boundary. This method constructs a class-blind minimum spanning tree over the entire data set, and counts the number of points incident to an edge going across the two classes. The fraction of such points over all points in the data set is used as a measure. For two heavily interleaved classes, a majority of points are located next to the class boundary. However, the same can be true for a sparsely sampled linearly separable problem with margins narrower than the distances between points of the same class.

– **N3:** error rate of 1-NN classifier. This is simply the error rate of a nearest-neighbour classifier measured with the training set. The error rate is estimated by the leave-one-out method. The measure denotes how close the examples of different classes are. Low values of this metric indicate that there is a large gap in the class boundary.

4 Experimental Study: Analysis of the ANNs with Data Complexity Measures

In this Section we analyse the obtained results for the ANN methods. First, in Subsection 4.1 we present the experimental framework, with the data sets generation method, and the global average accuracy of the ANN methods. Next we determine several rules based on ANNs behaviour in Subsection 4.2. Finally we analyse the collective evaluation of the set of rules in Subsection 4.3.

4.1 Experimental Framework: Data Sets Generation

We evaluate the ANN methods on a set of 438 binary classification problems. These problems are generated from pairwise combinations of the classes of 21 problems from the University of California, Irvine (UCI) repository [1]. These are *iris, wine, new-thyroid, solar-flare, led7digit, zoo, yeast, tae, balanced, car, contraceptive, ecoli, hayes-roth, shuttle, australian, pima, monks, bupa, glass, haberman* and *vehicle*.

In order to do that, we construct several new data sets with the combination of the examples from two different classes. This will result in a new data set with only 2 classes and with the original examples which had two such classes as output. We perform this process for every possible pairwise combination of classes. If an obtained data set with this procedure proves to be linearly-separable, we discard it. The complexity measure L1 from [9] indicates if a problem is linearly-separable.

This method for generating binary data sets is limited by the proper combinatorics, and we can only obtain over 200 new data sets with the original 20 data sets with this first approach. In order to obtain more data sets, we group the classes two by two, that is, we create a new binary data set, and each of its two classes are the combination of two original classes each. For this second approach we have used *ecoli*, *glass* and *flare* data sets, since they have a high number of class labels. Again, those data sets with a L1 value of zero are discarded.

In order to measure the ANNs performance, we have applied a 10 fold-cross validation scheme. In Table 1 we have summarized the global average Training and Test accuracy and standard deviation obtained by the ANN methods.

Table 1. Global Average Training and Test Accuracy for RBFN and MLP

	Global % Accuracy Training	Global % Accuracy Test
RBFN	93.12% ± 7.99	90.65% ± 10.42
MLP	95.98% ± 5.25	88.39% ± 10.41

4.2 Determination of Rules Based on the ANNs Behaviour

In the following we present the results of the runs over the 438 data sets summarized in Figures 1 to 4.

For each complexity measure, the data sets are sorted by the ascending value of the corresponding complexity measure, and put altogether in a Figure. In the X axis we represent the data sets, not the complexity measure value, and the Y axis depicts the accuracy obtained both in training and test. The reason to do so is to give each data set the same space in the graphic representation. For those measures where we can find different *ad-hoc* intervals which present *good* or *bad behaviour* of the ANNs, we use a vertical line to delimit the interval of the region of interest.

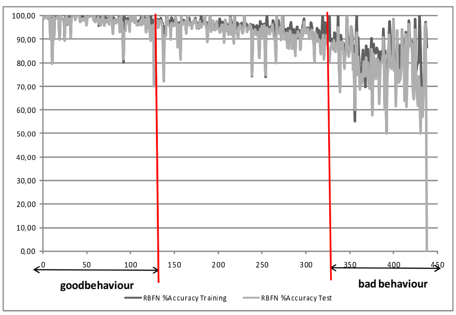

Fig. 1. RBFN accuracy in Training/Test sorted by N1

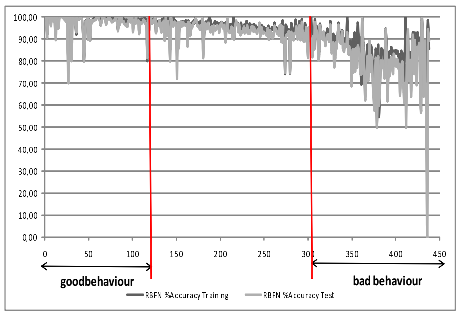

Fig. 2. RBFN accuracy in Training/Test sorted by N3

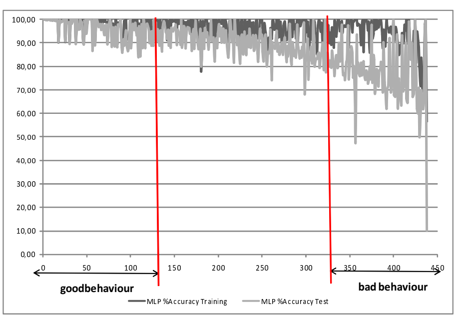

Fig. 3. MLP accuracy in Training/Test sorted by N1

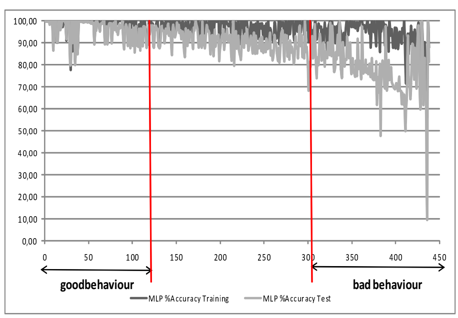

Fig. 4. MLP accuracy in Training/Test sorted by N3

Table 2. Significant intervals

Interval	ANNs Behaviour
$N1 < 0.08$	*good behaviour*
$N3 < 0.029$	*good behaviour*
$N1 > 0.25$	*bad behaviour*
$N3 > 0.108$	*bad behaviour*

– We understand for *good behaviour* an average high test accuracy in the interval, as well as the absence of over-fitting.
– By *bad behaviour* we refer to the presence of over-fitting and/or average low test accuracy in the interval.

In Table 2 we have summarized the intervals found ad-hoc from Figures 1 to 4.

From these ad-hoc intervals we construct several rules that model the performance of the ANNs we have used. In Table 3 we have summarized the rules derived from Table 2. Given a particular data set X, we get the complexity measure of X with the notation $CM[X]$. Table 3 is organised with the following columns.

– The first column corresponds to the identifier of the rule for further references.
– The "Rule'" column presents the rule itself.
– The third column "Support" presents the percentage of data sets which verifies the antecedent of the rule.
– The column "Neural Network" identifies the ANN to which this row refers to.
– The column "% Training" shows the average accuracy in training of all the examples which are covered by the rule.
– The column "Training Diff." contains the difference between the training accuracy of the rule and the training accuracy across all 438 data sets.
– The column "% Test" shows the average accuracy in test of all the examples which are covered by the rule.
– The column "Test Diff." contains the difference between the test accuracy of the rule and the test accuracy across all 438 data sets.

The positive rules (denoted with a "+" symbol in their identifier) always show a positive difference with the global average, both in training and test accuracy. The negative ones (with a "-" symbol in their identifier) verify the opposite case. The support of the rules shows us that we can characterize a wide range of data sets and obtain significant differences in accuracy.

From this set of rules we can state that a low N1 value results in a good behaviour of the ANN methods. A low N3 value obtains the same good results. In the other hand, a high value in the N1 metric produces a bad behaviour of the ANNs considered in our analysis. A high N3 value will also produce a bad behaviour of both ANN methods.

Although we have obtained some interesting rules, we can extend our study by considering the combination of these complexity metrics in order to obtain more precise and descriptive rules.

Table 3. Rules with one metric obtained from the intervals

Id.	Rule	Support	Neural Network	%Training	Training Diff.	% Test	Test Diff.
R1+	If N1[X] < 0.08 then *good behaviour*	29.22%	RBFN	98.03%	4.91%	97.20%	6.55%
			MLP	97.67%	1.69%	95.95%	7.56%
R2+	If N3[X] < 0.029 then *good behaviour*	26.71%	RBFN	98.18%	5.06%	97.28%	6.63%
			MLP	97.31%	1.33%	96.02%	7.63%
R1-	If N1[X] > 0.25 then *bad behaviour*	24.43%	RBFN	83.64%	-9.48%	78.22%	-12.43%
			MLP	93.68%	-2.30%	76.74%	-11.65%
R2-	If N3[X] > 0.108 then *bad behaviour*	31.51%	RBFN	85.33%	-7.79%	80.64%	-10.01%
			MLP	93.80%	-2.18%	78.47%	-9.92%

Table 4. Disjunction Rules from all simple rules

Id.	Rule	Support	Neural Network	%Training	Training Diff.	% Test	Test Diff.
PRD	If R1+ or R2+ then *good behaviour*	32.65%	RBFN	98.16%	5.04%	97.11%	6.46%
			MLP	97.17%	1.19%	95.52%	7.13%
NRD	If R1- or R2- then *bad behaviour*	31.96%	RBFN	85.50%	-7.62%	80.81%	-9.84%
			MLP	93.86%	-2.12%	78.57%	-9.82%
uncov-ered	If not PRD and not NRD then *good behaviour*	35.39%	RBFN	95.35%	2.23%	93.59%	2.94%
			MLP	96.80%	0.82%	90.70%	2.31%

4.3 Collective Evaluation of the Set of Rules

The objective of this section is to analyse the good and bad rules jointly. Thus we can arrive at a more general description and with wider support of the behaviour of the ANNs. We perform the disjunctive combination of all the positive rules to obtain a single rule, and all the negative ones. The new disjunctive rule will be activated if any of the component rules' antecedents are verified.

In Table 4 we summarize both disjunctions, and a third rule representing those data sets which are not charaterised by either disjunction rules.

From the collective rules we can observe that the support has been increased from the single rules both for the Positive Rule Disjunction (PRD) and Negative Rule Disjunction (NRD). In the other hand, the Test and Training Accuracy Differences are similar to the single rules from Table 3. Since there are no data sets in PRD and NRD simultaneously, we can consider three blocks of data sets with their respective support, as depicted in Figures 5 and 6(with no particular data set order within each block):

– The first block (the left-side one) represents the data sets covered by the PRD rule. They are the data sets recognized as being those in which the ANNs have good accuracy.
– The second block (the middle one) plots the data sets for the rule NRD, which are bad data sets for the ANNs methods considered.

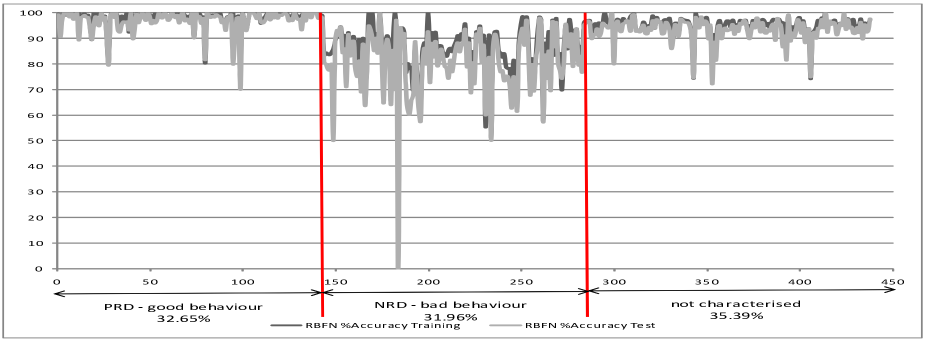

Fig. 5. Three blocks representation for PRD, NRD and not covered data sets for RBFN

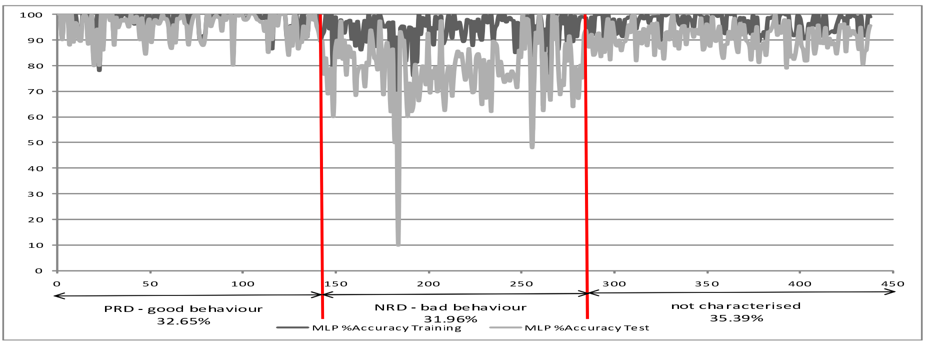

Fig. 6. Three blocks representation for PRD, NRD and not covered data sets for MLP

– The third and last block (the right-side one) contains the unclassified data sets by the previous two rules.

We can see that almost the 65% of the analysed data sets are covered by these two rules, and hence the *good behaviour* and *bad behaviour* consequents represent well the accuracy of ANN methods.

5 Concluding Remarks

We have performed a study over a set of binary data sets with two ANN methods. We have computed two data complexity measures of separability of classes for the data sets in order to obtain intervals of such metrics in which the method's performance is significantly good or bad. We have constructed descriptive rules, and studied the interaction between the intervals and the proper rules.

We have obtained two rules which are simple and precise to describe both good and bad performance of the ANNs considered in this work. Furthermore, we present the possibility of determining which data sets RBFN and MLP would perform well or badly prior to their execution, using the Data Complexity measures.

We must point out that this is a particular study for two specific methods. On the other hand, this work presents a new challenge that could be extended to other ANNs models, to analyse their domains of competence, and to develop new measures which could give more information on the behaviours of ANNs for pattern recognition.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`
2. Baumgartner, R., Somorjai, R.L.: Data complexity assessment in undersampled classification. Pattern Recognition Letters 27, 1383–1389 (2006)
3. Bernadó-Mansilla, E., Ho, T.K.: Domain of Competence of XCS Classifier System in Complexity Measurement Space. IEEE Transactions on Evolutionary Computation 9(1), 82–104 (2005)
4. Bezdek, J.C., Kuncheva, L.I.: Nearest prototype classifier designs: An experimental study. International Journal of Intelligent Systems 16(12), 1445–1473 (2001)
5. Broomhead, D.S., Lowe, D.: Multivariable Functional Interpolation and Adaptive Networks. Complex Systems 11, 321–355 (1988)
6. Daqi, G., Chunxia, L., Yunfan, Y.: Task decomposition and modular single-hidden-layer perceptron classifiers for multi-class learning problems. Pattern Recognition 40(8), 2226–2236 (2007)
7. Dong, M., Kothari, R.: Feature subset selection using a new definition of classificabilty. Pattern Recognition Letters 24, 1215–1225 (2003)
8. Ho, T.K., Baird, H.S.: Pattern classification with compact distribution maps. Computer Vision and Image Understanding 70(1), 101–110 (1998)
9. Ho, T.K., Basu, M.: Complexity Measures of Supervised Classification Problems. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(3), 289–300 (2002)
10. Basu, M., Ho, T.K. (eds.): Data Complexity in Pattern Recognition. Springer, Heidelberg (2006)
11. Li, Y., Dong, M., Kothari, R.: Classifiability-Based Omnivariate Decision Trees. IEEE Transactions on Neural Networks 16(6), 1547–1560 (2005)
12. Moller, F.: A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks 6, 525–533 (1993)
13. Mollineda, R.A., Sánchez, J.S., Sotoca, J.M.: Data Characterization for Effective Prototype Selection. In: Marques, J.S., Pérez de la Blanca, N., Pina, P. (eds.) IbPRIA 2005. LNCS, vol. 3523, pp. 27–34. Springer, Heidelberg (2005)
14. Renjifo, C., Barsic, D., Carmen, C., Norman, K., Peacock, G.S.: Improving radial basis function kernel classification through incremental learning and automatic parameter selection. Neurocomputing 72(1-3), 3–14 (2008)
15. Rojas, R., Feldman, J.: Neural Networks: A Systematic Introduction. Springer, Heidelberg (1996)
16. Sánchez, J.S., Mollineda, R.A., Sotoca, J.M.: An analysis of how training data complexity affects the nearest neighbor classifiers. Pattern Analysis and Applications 10(3), 189–201 (2007)
17. Singh, S.: Multiresolution Estimates of Classification Complexity. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(12), 1534–1539 (2003)