Rule Extraction from Neural Networks in Data Mining Applications

Eduardo R. Hruschka
Positivo Faculty - Computer Science Department – R. Nossa Senhora Aparecida, n° 174 CEP 80440.000, Curitiba, PR, Brazil, eduardohruschka@hotmail.com
Nelson F. F. Ebecken
COPPE/Federal University of Rio de Janeiro, Caixa Postal 68506, CEP 21945.970, Rio de Janeiro, RJ, Brazil, nelson@ntt.ufrj.br

Abstract

This work deals with the efficient discovery of valuable and nonobvious information from large collections of data, using Computacional Intelligence tools. For this purpose, a study about knowledge acquirement from supervised neural networks employed for classification problems is presented. An algorithm for rule extraction from neural networks, based on the work by Lu et al. [1] in 1996, is developed. This algorithm, named *Modified RX,* is experimentally evaluated in three different domains. The results are compared to those obtained by classification trees. In respect of the efficacy , one observes that the successful application of the algorithm mainly depends on the knowledge representation acquired by the conecctionist model, while the eficciency only depends on the neural network training time.

1 Introduction

The main challenge in using supervised neural networks in Data Mining applications means to get explicit knowledge from those models. It is difficult to understand neural networks models because [2]:

- they are represented by many real parameters (connection weights);
- they represent non-linear functions;
- the hidden units can learn distributed representations.

The knowledge acquired by the neural network is codified by the connection weights [3] and the hidden units are called feature detectors [4]. Thus, the knowledge aquisition process from supervised neural networks implies in using algorithms based on the connection weights or on the hidden units activation values. These algorithms are usually called *Algorithms for Rule Extraction from Neural Networks* [1,5,6,7,8,9,10,11,12,13,14,15,16].

Lu et al. [1] developed a rule extraction algorithm based on the hidden units activation values. The algorithm developed on the current paper is based on this methodology. Therefore, it is called *Modified RX Algorithm*. This methodology provides a way of relating domain regions to the classes, clustering hidden units activation values. Basically, the changes are based in the fact that there is an activation particular set for each class and so one can draw rules separatelly for each particular class, differently to what the RX Algorithm [1] does.

2 Neural network simulator

The neural network trainning process was accomplished by *NeuralWorks Predict* [4]. This software is produced by *NeuralWare, Inc.* at 202 Park West Drive, Pittsburgh, PA 15275. This software uses a genetic algorithm to select appropriate variable sets to each neural network model, like can be seen in [17].

3 Clustering algorithm

The clustering algorithm used in this work is similar to the *Leader Algorithm* [18], considering that the *leader* is the mean of the activation values and the tolerance is given as a function of the standard deviation. The *euclidian distance* is used. The algorithm performs the following steps:

1) Calculate the mean and the standard deviation of the C activation values set;
2) Cluster the v values whose Dv distance is smaller or equal to the standard deviation multiplied by the *MULTI* tolerance factor, where Dv is the distance between the v value and the mean;
3) Exclude the clustered values from the C set;
4) If $C = \{\phi\}$ then stop;
   Else go to 1).

The number of discovered clusters - *NC* - is variable and calculated automatically by the algorithm:

NC = 5 - D, where *D* is the number of considered standard deviations.

This algorithm was tested on three different domains: *Iris Plants Database, Pima Indians Database and Meteorology Data.*

4 Case studies

4.1 Iris plants database

This database contains three classes - each one with 50 examples - of flowers: Iris Setosa, Iris Versicolour and Iris Virginica. It is possible to do a linear separation of the first class to the other ones. There are four attributes envolved: sepal and petal lenght and width.

The training set is considered to be the complete database. Besides, the database is considered noiseless. The best result was obtained by a two hidden units neural network, considering the areas of sepals and petals as input. This network gets a average classification rate of 98%. Considering two standard deviation units as tolerance distance to the clustering algorithm, one can get the following rules:

If $\{ A_S - 3{,}98.A_P \geq 2{,}34 \}$
and $\{ 11{,}21 \leq A_S - 5{,}56.A_P \leq 21{,}87 \text{ or } A_P - 0{,}18.A_S = 1{,}47 \}$
Then SETOSA.

If $\{-0{,}23 \leq A_P - 0{,}25.A_S \leq 3{,}45 \text{ or } A_P - 0{,}25.A_S = 3{,}89$
or $A_P - 0{,}25.A_S = 4{,}09\}$
and $\{1{,}18 \leq A_P - 0{,}18.A_S \leq 5{,}22 \text{ or } A_P - 0{,}18.A_S = 0{,}83$
or $A_P - 0{,}18.A_S = 1{,}01\}$
then VERSICOLOR.

If $\{A_P - 0{,}25.A_S \geq 3{,}85 \text{ or } A_P - 0{,}25.A_S = 3{,}21\}$
and $\{A_P - 0{,}18.A_S \geq 4{,}48\}$
then VIRGINICA.

where A_p is the petal area and A_s is the sepal area.

These rules classifies 97.33% of the database examples correctly, with a complexity measure [19] equal to 3.4. The following table shows the summary of some results:

Transactions on Information and Communications Technologies vol 15 © 1998 WIT Press, www.witpress.com, ISSN 1743-3517

Methodology – Iris Flower Data	Accuracy		Rule Complexity
	Model	Rules	
Best neural network : NeuralWorks Predict [4] – areas	98%	97%	3.4
Classification tree : *XpertRule Profiler* [20] – sepal and petal lenght and width	96%	96%	2.5
HENERY classification tree [21] - sepal and petal lenght and width	****	98%	3.0
HENERY classification tree [21] - areas	****	97%	2.2

4.2 Pima indians diabetes database

This example represents a complex classification problem [22]. The dataset contains 768 examples – 500 meaning negative conditions for diabetes – class 1 - and 268 showing positive conditions of diabetes – class 2. Therefore, this sample is not representative [23]. Each example contains 8 attributes plus the *class*. The attributes are:

A – number of times pregnant
B – plasma glucose concentration in an oral glucose tolerance test
C – diastolic blood pressure (mm hg)
D – triceps skin fold thickness (mm)
E – 2 hour serum insulin (um U/ml)
F – body mass index (kg/m^2)
G – diabetes pedigree function
H – age

The dataset was considered noiseless, and it was divided in trainning set and test set. The training set contains 75% of the examples and these were selected in according to maintain the proportion verified on the total set – 65% belonging to class 1 and 35% to class 2.

Low linear correlation factors between the attributes were verified and this is a good characteristic to the learning process [23]. However, the attributes are not highly correlated to the class values and this fact difficults the learning process [23,24]. Thus, the variable selection was performed by the genetic algorithm available in *NeuralWorks Predict* [4].

The *best* neural network model has four input units – using the linear [-1,+1] transformation – eight hidden units – using the hyperbolic

tangent function – and one output unit using the logistic function. The genetic algorithm selected four attributes: A, B, D, H.

The training set average classification rate was equal to 76.4% after 180 epochs, while the test set average classification rate was equal to 82.55%. The best result obtained to this classification problem achieved a test set average classification rate of 77.7%, using the *Logdisc* algorithm [22]. It is important to say that the training and test sets were selected randomically, what prevents the acquisition of identical sets used by the *Logdisc* algorithm.

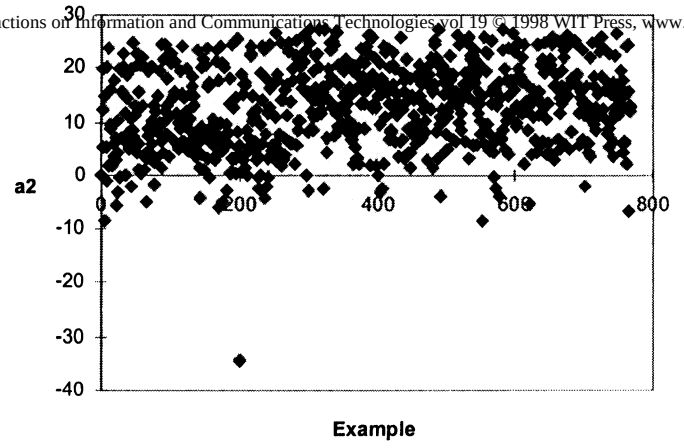
Considering the standard deviation as tolerance distance to the clustering algorithm, one can get the following rules:

If $\{(-49{,}24 \leq a_1 \leq 6{,}97)$ or $(8{,}19 \leq a_1 \leq 9{,}51)\}$
and $\{(0{,}03 \leq a_2 \leq 26{,}33)$ or $(-8{,}56 \leq a_2 \leq -0{,}057)\}$
and $\{(-0{,}25 \leq a_3 \leq 1{,}496)$ or $(-0{,}37 \leq a_3 \leq -0{,}28)\}$
and $\{(-0{,}08 \leq a_4 \leq 0{,}15)\}$
and $\{(-11{,}33 \leq a_5 \leq 35{,}12)$ or $(-28{,}52 \leq a_5 \leq -14{,}66)\}$
and $\{(-0{,}61 \leq a_6 \leq 3{,}68)$ or $(3{,}74 \leq a_6 \leq 4{,}23)\}$
and $\{(6{,}13 \leq a_7 \leq 46{,}62)$ or $(-4{,}83 \leq a_7 \leq 5{,}77)\}$
and $\{(-0{,}05 \leq a_8 \leq 0{,}45)$ or $(0{,}46 \leq a_8 \leq 0{,}63)$ ou $(-0{,}14 \leq a_8 \leq -0{,}06)\}$
then *Diabetes*;
Else, *not diabetes*.

where a_i $\{i = 1,...,8\}$ represents the hidden units activation expressions:

$a_1 = 51{,}31 - 0{,}91.A - 0{,}35.B + 0{,}07.D - 0{,}56.H$
$a_2 = 38{,}20 + 0{,}22.A - 0{,}05.B - 0{,}42.D - 0{,}35.H$
$a_3 = -0{,}17 + 0{,}02.A + 0{,}01.B - 0{,}01.D - 0{,}02.H$
$a_4 = -0{,}08 - 0{,}004.A - 0{,}001.D + 0{,}004.H$
$a_5 = -7{,}88 + 0{,}76.A + 0{,}24.B - 0{,}46.D - 0{,}26.H$
$a_6 = 2{,}13 + 0{,}06.A + 0{,}02.B - 0{,}04.D - 0{,}07.H$
$a_7 = 40{,}15 + 0{,}18.A + 0{,}11.B + 0{,}15.D - 0{,}91.H$
$a_8 = 0{,}13 + 0{,}04.A - 0{,}005.D - 0{,}003.H$

These rules classify 41% of the examples correctly. This low classification rate happens due to the distributed representation, that makes difficult the knowledge acquisition from neural network models [3,25]. This fact can be visualized by graphics like:



Graphic 1. Activation values to the hidden unit 2. The first 500 values represent examples to the class 1 while the last ones represent examples of the class 2.

4.2.1 Neural Networks and Classification Trees

In order to compare the results obtained by neural networks and classification trees, either in efficacy as in efficiency, two datasets were separeted. A training dataset, containing 500 examples, and a test dataset, containing 192 examples. Both of them were randomically selected from the original dataset. It was not possible to use all the data because the software *XpertRule Profiler* [20] does not allow to process more than 500 examples.

The classification tree was obtained considering the standard parameters of *XpertRule Profiler* [20]. The classification tree building processing time, on a PC – 486 – DX4 – 100MHz – 16Mb RAM was equal to 16 minutes and classifies 72.4% of the examples correctly.

The best neural network achieved has five input units using linear [-1,+1] transformation, three hidden units using hyperbolic tangent function, and two output units using logistic function. The genetic algorithm selected five attributes: A, B, D, F and G. The test set average classification rate was equal to 78%. The neural network training time, on a PC – 486 – DX4 – 100MHz – 16Mb RAM was equal to 36 minutes.

It is verified that the neural network was lightly better in efficacy, while the classification tree algorithm is really more efficient than the neural network.

Transactions on Information and Communications Technologies vol 19 © 1998 WIT Press, www.witpress.com, ISSN 1743-3517

It was observed that the clustering of the hidden units activation values didn't provide good results concerning to the knowledge aquisition process. Thus, the distributed representation suggests that the utilization of an algorithm wich divide the hidden units activation space would get better results. This being the case, classification tree algorithms could be appropriate to separate the activation values. The application of a classification tree to the hidden units activation values provides the following rules:

If ($a_3 < -2,13$) and ($a_1 < 0,43$) then DIABETES;
Else, NOT DIABETES.

where:
$a_1 = 4,70 - 0,13.A - 0,004.B + 0,003.D - 0,091.F - 0,82.G$
$a_3 = 15,15 - 1,25.A - 0,09.B + 0,04.D - 0,02.F - 0,47.G$

These rules classifies correctly 76.3% - against 41% to the clustering algorithm - of the examples.

4.2 Meteorology dataset

This dataset was collected at the *International Airport of Rio de Janeiro*. The original dataset contains 88,000 examples of meteorological observations. Each example is represented by 38 attribute values and the associated class.

The examples related to the *wet* and *dry fog* classes were selected to Data Mining, resulting in a 10,000 examples dataset. One has to mention that a lot of examples contain some unknown attribute values and this fact implies in establishing criterions to eliminate these examples. Therefore, all the examples whose *air relative humidity* and *precipitation* values were unknown were eliminated. The attributes related to the clouds belonging to the *third and fourth layers* (the higher ones) as well as the *blast* were not considered, because one believes that they are not important to the *wet and dry fog* phenomenom. The attributes related to the *cloud directions* were eliminated because it is only possilble to get 16 examples containing these attribute values. The *landing track level pressure*, very related to the *sea level pressure*, was also eliminated. Thus, 1509 examples were considered to be *valid* for Data Mining.

The *wet bulb temperature* was desconsidered because it has a 0.96 linear correlation to the *dew temperature* and a 0.95 linear correlation to the *dry bulb temperature*.

In brief, the neural network model was obtained considering 19 attributes: *month, hour, cloud total quantity, wind direction, wind speed, visibility, first layer cloud quantity, second layer cloud quantity, first layer cloud type, second layer cloud type, first layer cloud height, second layer cloud height, dew temperature, sea level pressure, pressure variation tendency, three hours pressure variation, dry bulb temperature, precipitation and air relative humidity.*

Test and Validation sets were not used because there is not a enough number of examples relatively to the number of attributes envolved. The genetic algorithm [4] selected the folowing attributes:

x_1 = *month*
x_2 = *wind direction*
x_3 = *visibility* - not appear on the rules because of its low weights.
x_4 = *second layer cloud height*
x_5 = *dew temperature*
x_6 = *air relative humidity*

The *best* neural network model obtained has two hidden units and classifies, after 76 epochs, 89% of the examples correctly. The neural network training time, on a PC – 486 – DX4 – 100MHz – 16Mb RAM was equal to 5 hours. Considering the standard deviation as tolerance distance to the clustering algorithm, one can get the following rules:

If { ($-7796 \leq 21.x1 + 32.x2 + x4 - 14.x5 - 84.x6 \leq -5880$) and
($-10460 \leq 27.x1 + x4 - 25.x5 - 107.x6 \leq -8060$) }
then WET FOG.

If { ($-5884 \leq 21.x1 + 32.x2 + x4 - 14.x5 - 84.x6 \leq -2614$) and
($-7920 \leq 27.x1 + x4 - 25.x5 - 107.x6 \leq -4549$) }
then DRY FOG.

These rules classifies 80% of the database examples correctly, with a complexity measure [19] equal to 2. It would be possible to get better results by means of a domain expert. It is important to mention that, even if the neural network model classifies both the classes equally (average classification rate equal to 89%), the rules classify correctly 83.5% and 60% of the *wet fog* and *dry fog* cases respectively. It is reasonable to

consider that it happens because of the distributed representation, that makes difficult to cluster the activation values.

A classification tree, obtained by means of the *Intelligent Miner for AIX* [26], provides the following rules:

If $H_r > 79,5$
then *wet fog*;
If $\{H_r \leq 79,5$ and $T_s \leq 239,5$ and $M \leq 8,5$ and $T_2 \leq 2,5$ and $H \leq 12\}$
then *wet fog*;
If $\{H_r \leq 79,5$ and $T_s \leq 239,5$ and $M \leq 8,5$ and $T_2 \leq 2,5$ and $H > 12\}$
then *dry fog*;
If $\{H_r \leq 79,5$ and $T_s \leq 239,5$ and $M \leq 8,5$ and $T_2 > 2,5\}$
then *wet fog*;
If $\{H_r \leq 79,5$ and $T_s \leq 203$ and $M > 8,5\}$
then *wet fog*;
If $\{H_r \leq 79,5$ and $(203 < T_s \leq 239,5)$ and $M > 8,5\}$
then *dry fog*;
If $\{H_r \leq 74,5$ and $T_s > 239,5\}$
then *dry fog*;
If $\{ (74,5 < H_r \leq 79,5)$ and $T_s > 239,5$ and $T_2 \leq 3,5$ and $H \leq 6\}$
then *wet fog*;
If $\{ (74,5 < H_r \leq 79,5)$ and $T_s > 239,5$ and $T_2 \leq 3,5$ and $H > 6\}$
then d*ry fog*;
If $\{ (74,5 < H_r \leq 79,5)$ and $(239,5 < T_s \leq 244,5)$ and $T_2 > 3,5\}$
then *dry fog*;
If $\{ (74,5 < H_r \leq 79,5)$ and $T_s > 244,5$ and $T_2 > 3,5\}$
then *wet fog*;

where :
H_r = air relative humidity;
T_s = dry bulb temperature;
M = month;
H = hour;
T_2 = second layer cloud type.

These rules classify 94% of the examples correctly, with a complexity measure [19] equal to 8.3 . Comparing to the neural network model, it is observed that this classification tree provides a better classification rate but a higher complexity measure.

5 Conclusions

Supervised neural networks are usually not employed in Data Mining applications because of their low computacional efficiency and because it is difficult to interpret their models. The computacional efficiency can be improved mainly by using paralel and distributed processing, by applying techniques for selecting adequate training samples, by incorporating domain knowledge and by reducing neural network model complexity.

The neural network model interpretability depends fundamentally on the application of effective rule extraction algorithms. Considering the studied cases, it is verified that the efficacy of the *Modified RX Algorithm* depends on the neural network knowledge representation.

The local representation allows the *Modified RX Algorithm* to extract non redundant rules, what does not usually happen on algorithms based on the connection weights. Besides, the number of rules obtained by the *Modified RX Algorithm* is equal to the number of classes. However, the *Modified RX Algorithm* does not provide acceptable results to the distributed representation. This representation is advantageous to the memory efficiency and adaptability, but it is not convenient for the interpreting process of the knowledge acquired by neural network models [3,25].

As well as observed in [1] it is verified that, comparing to classification trees, the *Modified RX Algorithm* provides a bigger number of premises, while classification trees generate bigger rule sets. Considering the computacional efficiency, classification trees got better results.

Reference

[1] Lu, H., Setiono, R., Liu, H. "Effective Data Mining Using Neural Networks", IEEE Transactions on Knowledge and Data Engineering, v. 8, n. 6, pp. 957-961, 1996.

[2] Craven, M. W., Shavlik, J. W., "Using Neural Networks for Data Mining", Future Generation Computer Systems, accepted to appear, 1997.

[3] Fu, L. Neural Networks in Computer Intelligence, 1ª ed., USA, McGraw-Hill Inc., 1994.

[4] NeuralWare, Inc - Technical Publications Group, documentação do programa *NeuralWorks Predict*, 1995.

[5] Gallant, S. I. "Connectionist Expert Systems", ***Communications of the ACM***, v.31, n. 2, pp. 152-169, 1988.

[6] Narazaki, H., Shigaki, I., Watanabe, T. "A Method for Extracting Approximate Rules from Neural Network", **IEEE**, 1995.

[7] Baron, R. **Knowledge Extraction from Neural Networks: A Survey**, In: Report n° 94-17, Laboratoire de l'Informatique du Parallélisme, Ecole Normale Supérieure de Lyon, 1994.

[8] Towell, G. G., Shavlik, J., "Extracting Refined Rules from Knowlege-Based Neural Networks", **Machine Learning**, v.13, pág 71-101, 1993.

[9] Oliveira, J. P., Uliana, P. B., Lima, W. C., "Algoritmos de Extração de Regras de Redes Neurais Artificiais". In: Anais do III Congresso Brasileiro de Redes Neurais, pág 173-177, Florianópolis, Julho de 1997.

[10] Thrun, S., "Extracting Provably Correct Rules from Artificial Neural Networks", *Technical Report IAI-93-5,* Institut für Informatik III, Universität Bonn, 1993.

[11] Fu, L., "Knowledge-Based Connectionism for Revising Domain Theories", **IEEE Transactions on Systems, Man. and Cybernetics**, v.23, n. 1, pag. 173-182, Feb.,1993.

[12] Fu, L. " Rule Generation from Neural Networks ", **IEEE Transactions on Systems, Man., and Cybernetics**, v.24 , n.8 , pp.1114 - 1124 , Aug, 1994.

[13] Kane, R., Milgram, M. "Financial Forecasting and Rules Extraction from Trained Networks", **IEEE**, pp. 3190-3195, 1994.

[14] Craven, M. W., Shavlik, J. W., "Using Sampling and Queries to Extract Rules from Trained Neural Networks", **Machine Learning: Proceedings of the Eleventh International Conference,** San Francisco, CA, 1994.

[15] Narazaki, H., Watanabe, T., Yamamoto, M., "Reorganizing Knowledge in Neural Networks: An Explanatory Mechanism for Neural Networks in Data Classification Problems", **IEEE Transactions on Systems, Man. and Cybernetics,** v.26, n. 1, pág. 107-117, Feb., 1996.

[16] Craven, M. W., Shavlik, J. W., "Extracting Tree-Structured Representations of Trained Networks", **Advances in Neural Information Processing Systems**, v.8, MIT Press, Cambridge, MA, 1996.

[17] Harp, S. A., Samad, T. "Genetic Synthesis of Neural Network Architecture". In: **Handbook of Genetic Algorithms**, International Thomson Computer Press, pp. 202-222, 1996.

[18] Hartigan, J. A., **Clustering Algorithms,** 1ª ed. USA, John Wiley & Sons Inc., 1975.

[19] Gaines, B. R. "Transforming Rules and Trees into Comprehensible Knowledge Structures ". In: **Advances in Knowledge Discovery and Data Mining**, MIT Press, pp. 205-229, 1996.

[20] Attar Software - *XpertRule Profiler*

[21] Henery, R, J. "Classification". In: **Machine Learning, Neural and Statistical Classification**, v.1, Ellis Horwood Series in Artificial Intelligence, Bookcraft, Midsomer Norton, 1994.

[22] Michie, D., Spiegelhalter, D. J., Taylor, C. C. "Dataset Descriptions and Results". In: **Machine Learning, Neural and Statistical Classification**, Ellis Horwood Series in Artificial intelligence, pp 157-158, 1994.

[23] Smith, M., **Neural Networks for Statistical Modeling,** 1ª ed. USA, International Thomson Computer Press, 1996.

[24] Yildiz, N., "Correlation Structure of Training Data and the Fitting Ability of Back Propagation Networks: Some Experimental Results ", **Neural Computing & Applications**, v.5, pág. 14-19, 1997.

[25] Narazaki, H., Watanabe, T., Yamamoto, M. "Reorganizing Knowledge in Neural Networks: An Explanatory Mechanism for

Neural Networks in Data Classification Problems ", **IEEE Transactions on Systems, Man., and Cybernetics - part B: Cybernetics**, v.26 , n.1 , pp.107 - 117 , Feb, 1996.

[26] IBM Intelligent Miner for AIX - User's Guide - Version 1 - Release 1, 1996.