CLASSIFICATION OF DATA TO EXTRACT KNOWLEDGE FROM NEURAL NETWORKS

Ana Martinez, Angel Castellanos, Rafael Gonzalo

Abstract: *A major drawback of artificial neural networks is their black-box character. Therefore, the rule extraction algorithm is becoming more and more important in explaining the extracted rules from the neural networks. In this paper, we use a method that can be used for symbolic knowledge extraction from neural networks, once they have been trained with desired function. The basis of this method is the weights of the neural network trained. This method allows knowledge extraction from neural networks with continuous inputs and output as well as rule extraction. An example of the application is showed. This example is based on the extraction of average load demand of a power plant.*

Keywords: *Neural Network, Backpropagation, Control Feedback Methods.*

ACM Classification Keywords: *F.1.1 Models of Computation: Self-modifying machines (neural networks); F.1.2 Modes of Computation: Alternation and nondeterminism.*

Conference*: The paper is selected from Seventh International Conference on Information Research and Applications – i.Tech 2009, Varna, Bulgaria, June-July 2009*

Introduction

The ability of artificial neural network to learn and generalize from examples makes them very suitable for use in numerous applications, where exact algorithmic approaches are unknown or too difficult to implement. The knowledge learned during the training process is distributed in the weights of the different neurons; it is very difficult to comprehend exactly what the neural network is computing. The problem of representing the knowledge learned by the network in a comprehensible form received a great deal of attention in the actual literature [Andrews, R., Diederich, J., Tickle,A. 1995], [Andrews, R., Diederich, J., Golea, M. 1998], [Cloete, I., Zurada, J.M. 2000].

Although both expert systems and neural networks are typical systems in the domain of artificial intelligence, the basic components of these two kinds of systems are different. The knowledge base of expert systems is a set of rules which are stored in symbolic form, while neural networks encode learned knowledge within an established structure with adjustable weights in numerical form. Hence, it is difficult to transfer the training results of a neural network to the knowledge base of an expert system.

In contrast, neural networks have excellent abilities for classifying data and learning inputs [Freeman J.A., Skapura D.M. 1992], but it is difficult to describe the decision process of a neural network or to merge more than one trained neural network [Krishnan R., Sivakumar G., Bhattacharya P. 1999].

This paper shows the importance of the knowledge stored in the weights of a neural network. A trained neural network stores the acquired knowledge in numeric values that weights define [Apolloni, B. et al 2004], [Garcez d'Avila, A. S., Broda, K. and Gabbay D. M. 2001], [Chang, B.L., Hirsch, M. 1991]. The interpretation and extraction of such knowledge is a difficult task due to the special configuration of neural network and to the wide domain of patterns.

Method to Extract Knowledge

Tasks to follow in order to perform a study of the importance of input, variables over output variables are the following ones:

1. Normalization of the input and output variables into the interval $[-1,1]$.
2. A neural network with *n* inputs and one output. The training algorithm considered is the backpropagation. Defining the activation function as sigmoid function.
3. Division of the values associated to the variable to forecast into two intervals, the positive one with a positive output $[0,1]$ and the negative interval with a negative output $[-1,0)$. These way two independent neural networks are defined in order to be trained.
4. Established an error threshold for the forecasting process, each one of the two output classes of the variable to forecast (positive output values in the interval $[0,1]$ and negative output values in the interval $[-1,0)$ are divided into two new classes. For each one of the obtained classes (four classes), neural networks are trained and the value of the weights is observed. If in these new classes obtained, the values of weights that are fixed after the training process, is the same that the one obtained in the previous division, or is proportional, then go back to the previous division. If the value is not the same then this division is valid, therefore they will exist four neural networks associated to the output intervals. This iterative division must go ahead until the weights of a new division will be the same of the previous division. When the weights are similar, then the successive divisions end. This process achieves a better error ratio, getting more powerful classification properties than classical nets, and this way a set of neural networks with their corresponding weights the following information:
    a. The variable with the most influence over the variable to forecast will be the one with the highest absolute weight after the training process. These data must verify that the sign of the input variable multiplied by the sign of the weight must be equal to the sign of the variable to forecast.
    b. And if the relationship between the forecasting variable and the variable to forecast is a direct or inverse function, that is, if the sign of both variables are the same or not. If the output interval of the variable to forecast, is a subinterval of interval $[0,1]$ or a subinterval of interval $[-1,0]$ and, if the domain of the forecasting variable multiplied by the corresponding weight is positive for a subinterval of the variable to forecast of interval $[0,1]$, we will say that the relationship is a direct one, other way it will be an inverse one, taking into account that the absolute value of the highest weight shows the importance of the forecasting variable over the variable forecast. That is, the higher absolute value of the variable over the variable to forecast.

       That is, the higher absolute value of the variable, the deeper influence in the output. Different divisions of initial set of training data, obtained from study of weights in the training subset, make that each one of the obtained training subset defines a different neural network to train the whole subset. Each network, with is corresponding set of weights denotes the importance of the forecasting variables over the variable to forecast.
    c. Besides extracting the importance of each variable in each output interval, for each one of the input variables it exits a network and a weight set that define the forecasting equation.

Therefore, the method is divided into two steps in order to better understand the two main processes on it.

- The first step is used to classify using the bisection method the patterns of the initial set into several subsets, taking into account that this division is performed iteratively, studying the variation of the weights. When in a new division the weights do not change, then go back to the initial division.

- The second step is used once the initial pattern set is classify into several subsets and therefore into several neural networks. The importance of each input variable must be studied for each different network, taking into account the weight values, the variation domain of the input variable and the variation of the output; to study the influence over the variable to forecast.

  It must be considered:

  1. The variables with the highest absolute weight.
  2. Which of them verify that their variation domain for the input variable multiplied by its corresponding weight has the same sign of the variable to forecast according to the positive or negative interval $[0,1]$ or $[-1,0]$.

Experimental data

The previous theoretical results described have been used in the construction of a rule-oriented knowledge base, applied on a system to predict the load demand for the next day in a power plant [1].

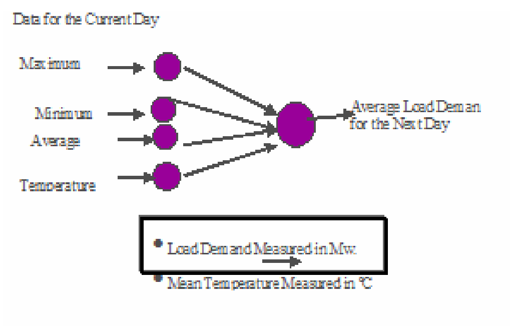
1. Obtaining the best classification: The proposed model takes into account the characteristics of forecasting variables could change from a different class to another, and that is the way it is necessary to use a division method, bisection method, studying the weights. This can be employed when dealing with a high number of patterns or to improve the error ratio.
2. Extract and study of the influence inputs variables: studying the weights decides which is the variable with more influence in the output using standardized weights and the bigger is the most important for the output.

Example of application

The data used to design the training and test sets has been supplied by one of the most important spanish load suppliers on a specific format. That is featured by providing for each day the load demand data sampled for each hour measured in Mw., and the mean temperature of the day measured in C° for two years. The input variables considered for the network were the maximum, minimum, average load demand and temperature for the current day. The output variable was the average load demand for the next day. The data was standardized in the range [-1,1]

There is a demand for making electrical charge per hour, taking data of 660 consecutive days. It also provides the average temperature each day. Taking 480 patterns for training and the rest for testing.

We have a number of input variables, which are defining the load curve for the next day. As we had shots of 24 hours a day and the average temperature of the day the variables used for forecasting were the maximum, minimum, average and the temperature of the previous day


Data for the Current Day
Maximum
Minimum
Average
Temperature
Average Load Deman for the Next Day
Load Demand Measured in Mw.
Mean Temperature Measured in °C

Obtaining the optimal classification

First level the output is ordered from lowest to highest. After the output range is standardized in [-1, 1], the output is divided in intervals by the middle of the range.

When the set is divided into different classes of patterns out of training improves, reaching a satisfactory ratio. At first you try to train the network with the entire set of patterns, to see what kind of predictions, and that knowledge was reflected in the weights, the data obtained were in table 1 and the ratio of Learning 0.2 is not good.

Table 1. Data weights

Patterns	Error	Bias	Max	Min	Average	Temp	Output
All	0.23969	0.5328	0.53414	-0.2008	1.1656	0.1260	[-1, 1]

We try to train networks with different configurations, working with a hidden layer in which it was increasing the number of hidden neurons. But in any case learning improved, initially tested the whole set, with the values that are the table 1. The ratio of error should not be acceptable; the knowledge learned by the network is not good. The error is too large.

The bisection process begins by deciding the range of patterns that is obtained in each subclass and the values obtained for the weights associated with input variables after each division.

If the weights indicate the same importance for the variables, is no longer necessary to continue with the class divide. The network has found homogeneity in the patterns.

At this stage, the method of heuristic features, and was drawn to the rules. It is assumed that knowledge of the neural network must be stored in the weights.

The best classes were obtained testing with different division for classifications of the outputs.

The first branch was divided into two-out, or a class for the output, one class for positives outputs and other for negatives outputs. Obtaining two classes and then again divided in two new classes. For each one of the obtained classes (four classes) neural network is trained and the value of the weights is observed. If in this new obtained classes, the values of the weights are fixed after the training process is the same that the one obtained in the previous division, or is proportional then go back to the previous division.

If the values of the weights are similar or proportional we stop the division in classes, in this case, we obtained eight intervals or classes. It reached a suitable learning rate (average error 0.003) and is considered good to denormalize output.

Final classification of all patterns

Follows the evolution of weights in different classifications for all patterns.

The first division in positives and negatives outputs:

Table 2. Data weights with and without temperature

Nº patterns	Bias	Max	Min	Average	Temp	Output
All	0.3271	0.0958	-0.6177	2.193	0.2442	[-1, 1)
All	0.4733	0.2993	-0.2206	1.5576		[-1, 1]
positives	0.2774	0.2255	-0.5021	2.1254	0.2034	[0, 1]
positives	0.3888	0.3042	-0.1897	1.6928		[0, 1]
negatives	-0.2542	-0.5023	-0.3518	0.7689	0.1996	[-1, 0)
negatives	-0.1865	-0.3413	-0.1020	0.3613		[-1, 0)

Study of the weight with different classes

Five networks trained for 459 patterns with the usual configuration

Table 3. Data weights for division in five output classes

Patterns	Bias	Max	Min	Average	Temp	Output
90	-0.61717	-0.4824	-0.3528	0.4055	0.3523	[-1, -0.13)
87	-0.1490	-0.3317	-0.1123	0.5451	0.0134	[-0.13, 0)
93	0.1726	-0.1849	-0.0986	0.4709	0.0365	[0 , 0.2)
114	0.4687	0.3752	-0.1565	0.8179	0.1631	[0.2, 0.5)
75	0.7894	1.8697	0.0296	-0.0226	0.1625	[0.5 , 1]

The error is less when you divide the total pattern set in subsets and is trained one RNA for each subset of pattern. In this example, finally we need construct 8 RNA: one for each Set of patterns S1, S3, … , S8 obtained, which outputs are I1,I3, … , I8, the subsets are obtained from de output division. One neural network is trained for each interval and different rule with the most important variable are obtained for each output interval, and one collection of rules R1, R3, …, R8 in the last step of the algorithm is obtained.

Table 4. Data weights of neural network training

	Nº pattern	Interval	Bias	Max	Min	Average	Temp
(-) I1	19	[-1 , - 0.5]	-2.0023	-1.1458	0.1405	0.3903	0.7733
(-) I2	55	[-0.48 ,-0.31]	-0.7072	0.4279	0.1467	-0.5488	0.014
(-) I3	84	[-0.30 , -0.2]	-0.4491	-0.1087	0.0111	0.2206	-0.0224
(-) I4	116	[-0.19 , 0]	-0.0967	0.1333	-0.0437	0.1915	0.0135
(+) I1	90	[0 , 0.19]	0.2021	0.3023	-0.052	0.3942	0.0914
(+) I2	58	[0.2 , 0.35]	0.5569	0.2408	-0.0452	0.0681	0.0437
(+) I3	40	[0.35 , 0.59]	0.6616	1.7795	0.0926	-0.6939	0.1453
(+) I4	18	[0.62 , 1]	0.0274	-3.8252	-1.7025	7.8928	0.403

The error is better than the first time with all patterns.

Table 5. Mean squared error of the trained ANN

RNA i	Mean squared error
1	0.006
3	0.042
4	0.049
5	0.038
6	0.043
7	0.042
8	0.04

We were looking for two things: a good learning and extracting a good knowledge in each class, it is, extract the most important input variables for each output interval.

- That knowledge stored by the network is reflected in the weights.
- Find the most important variables from the values of the weights.
- The rules that are obtained reflect what the network learned.

Extracting rules for each intervals

Once patterns have been divided in classes, we get eight subnets that give the best possible rating. We study the weights obtained from the network is trained and a characterization of the weights.

What is attempted is the order of importance of the variables and the degree of importance. The order and degree of importance of the variables is given by the value of the defined set of weights associated with the network.

The higher of the normalized weight, give the greater importance of the primary or principal input variable.

Table 6. Variables that can take part in the rules for each class of output

Interval	Max	Min	Average	Temp	Output
I1 (-)	-1.145	0.14	0.39	0.773	[-1 , -0.5]
I2 (-)	0.4279	0.1467	-0.5488	0.014	[-0.48 , -0.31]
I3 (-)	-0.1087	0.0111	0.2206	-0.0224	[-0.3 ,-0.2]
I4 (-)	0.1333	-0.0437	0.1915	0.0135	[-0.19 , 0]
I1 (+)	0.3023	-0.052	0.3942	0.0914	[0 , 0.19]
I2 (+)	0.2408	-0.0452	0.0681	0.0437	[0.2 , 0.35]
I3 (+)	1.7795	0.0926	-0.6939	0.1453	[0.35 , 0.59]
I4 (+)	-3.8252	-1.7025	7.8928	0.403	[0.62 , 1]

As shown in the table that follows, with the values obtained from the different networks once trained.

The values of the averages are almost identical and the standard deviations are not significant. What we succeeded in demonstrating that learning is good for every class.

Table 7. Pattern output and learning output

	Nº pattern	Pattern output	Learning output	Average learning	Average output
I1 (-)	19	[-1 , -0.5]	[-0.79,-0.53]	-0.65	-0.66
I2 (-)	55	[-0.49,0.3]	[-0.38, 0.34]	-0.36	-0.38
I3 (-)	84	[-0.3 ,0.2]	[-0.27,-0.21]	-0.23	-0.25
I4 (-)	117	[-0.19, 0]	[-0.13,-0.01]	-0.07	-0.11
I1 (+)	90	[0 , 0.19]	[0.02 , 0.18]	0.1	0.08
I2 (+)	58	[0.2 ,0.35]	[0.27 , 0.33]	0.29	0.27
I3 (+)	40	[0.35,0.59]	[0.32 , 0.57]	0.43	0.43
I4 (+)	18	[0.62, 1]	[0.58 , 0.91]	0.77	0.77

Conclusion

In the algorithm proposed to extract knowledge from a neural network that has been trained, it is improved the learning of the RNA with a division of the output range while the weights are changing. In this way, we obtained the best division for getting the most important variables in the possible rule. It allows both antecedent (the most important variable in each interval together with the domain values for this variable) and consequent (the interval for the output obtained with iterative Method previously described in this article) obtain rules to take continuous values, and make them able to be applied to a greater number of cases.

In this way, the rules obtained will allow to complete the knowledge that could be extracted from an expert when building the knowledge base for an expert system. In the proposed method, the first task is to divide the problem in output ranges; then the most important variables are extracted from each interval, and finally the solution (set of rules) is globalize with all the output intervals. The proposed method also computes the forecasting value from the equation of weights.

The proposed model takes into account the fact that the characteristics of forecasting variables could change from a different class to another, and because of that it is necessary to use a division method or a bisection method. This can be used when dealing with a high number of patterns or to improve the error ratio.

The main advantage of this method is the simplicity of itself. The matrix of weight defines the most important forecasting variables as well as the equation to return a value. The only thing to do is to apply the bisection method to the data set and to train a neural network for each class identified by the algorithm.

Bibliography

[Andrews, R., Diederich, J., Tickle,A. 1995] Survey and critique of techniques for extracting rules from trained artificial neural networks. Knowsledge-Based Systems (1995)

[Andrews, R., Diederich, J., Golea, M. 1998] The truth will come to light directions and challenges in extracting the knowledge embedded within trained artificial neural networks. IEEE Trans. Neural Networks(1998).

[Apolloni, B. et al 2004] A general framework for learning rules from data," IEEE Trans. Neural Networks., vol. 15, no. 6, pp. 1333–1349, Nov. 2004.

[Chang, B.L., Hirsch, M. 1991] Knowledge Acquisition and Knowledge Representation in a Rule-Based Expert Systems. Computers in Nuring. Volume 9, Number5 Pp 174-178 (1991)

[Cloete, I., Zurada, J.M. 2000] Knowledge- Based Neurocomputing. MIT Press (2000).

[Freeman J.A., Skapura D.M. 1992] Neural Networks. Addison-Wesley, Reading.

[Garcez d'Avila, A. S., Broda, K. and Gabbay D. M. 2001] Symbolic knowledge from trained neural networks: A sound approach, Artif. Intell., vol. 125, no. 1, pp. 155–207, 2001.

[Krishnan R., Sivakumar G., Bhattacharya P. 1999] A search technique for rule extraction from trained neural networks. Patern Recognit Lett 20:273-280 (1999).

Authors' Information

Castellanos Angel *– Departamento de Ciencias Basicas aplicadas a la Ingeniería Forestal. Escuela de Ingeniería Técnica Forestal. Universidad Politécnica de Madrid, Avda. de Ramiro de Maeztu s/n 28040 Madrid, Spain. e-mail: angel.castellanos@upm.es*

Gonzalo Rafael *– Natural Computing Group. Universidad Politécnica de Madrid, Spain.*
e-mail: rgonzalo@fi.upm.es

Martinez Ana *– Natural Computing Group. Universidad Politécnica de Madrid, Spain.*
e-mail: ana.martinez@upm.es