MLP Networks Applied to the Problem of Prediction of Runtime of SAP BW Queries

Tatiana Escovedo[1], Tarsila Tavares[2], Rubens Melo[3], Marley M.B.R. Vellasco[4]

Abstract. The SAP BW is a BI tool used daily by about 8000 employees of a big oil company in Brazil, running monthly about 150,000 queries to assist in the analysis inherent in their professional activities. A query is created to meet a need for specific business analysis and its response time is directly affected by the use of BW server by other users. The main problem today is that there is no way to estimate the execution time in advance, for the user to decide the best time to execute the query he needs for his work. This article proposes a solution to this problem by developing a prediction classification model for the performance of BW queries at certain times of the day using Multilayer Perceptron (MLP) neural network and the Weka tool [WEKA, 2011].

1 INTRODUCTION

The constant changes in the market have prompted the need for more timely and accurate integrated information from different departments in order to accelerate the process of decision making in the organizations. An analysis of historical integrated data can provide indicators of business growth or danger. This fact also prompted the emergence of data warehouse technology.

Data Warehousing is not a product but a strategy that recognizes the need to store data in separate information systems and consolidate them in order to support various professionals of a company in making decisions quickly and effectively. A data warehouse is a subject oriented, integrated, time variant and nonvolatile collection of data, which aims to support the process of decision making [Inmon, 1992].

To meet its needs for analytical information, the company uses the SAP BW since 2004. Through so-called BW queries, users can perform various analysis of business information. Data extracted from many sources is integrated and stored in the data warehouse implemented with the BW, and then accessed through queries which give useful information for the daily work of thousands of users. However, a significant problem is that many queries take considerable time to perform, because they work with a large volume of data and also because they dispute server time with other users. Currently, users have no way to estimate the runtime of their query in advance, in order to decide the best schedule for execution of the query. This article aims to investigate the application of neural networks in the problem of prediction of the performance of BW queries along the day. Section 2 presents a summary on Neural Networks, the tools Weka and SAP BW. Section 3, in turn, explains the problem in study and section 4 describes the proposed solution. The following section 5, will present the results and section 6 will evaluate them, pointing out some possible future work. Finally, section 7 concludes this work.

[1, 2, 3] Department of Electrical Engineering - Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Brazil
[4] Department of Computer Science - Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Brazil

e-mail: [1]tatiana@inf.puc-rio.br, [2]tarsilabello@gmail.com, [3]rubens@inf.puc-rio.br, [4]marley@ele.puc-rio.br

2 BASIC CONCEPTS

The purpose of this section is to present briefly the basic concepts related to neural networks and the tools Weka and SAP BW used in this work. These concepts will provide the theoretical background to the issues raised in this study.

2.1 Neural Networks

Artificial neural networks are used in the area of intelligent systems and computational intelligence [Jang et al, 1997] [ZADECH, 1992]. A neural network is a parallel and distributed system composed of simple processing units. Its goal is to store experimental knowledge and make it available for use. Artificial neural networks are similar to the human brain, because knowledge is acquired through a learning process and the strength of connections between neurons, or synaptic weights are used to store the acquired knowledge [Haykin, 1999]. The Multilayer Perceptron (MLP) is a type of neural network widely used for its ease of implementation and for being considered a universal approximator [Hornik et al, 1989]. MLP networks have powerful computing power due to the insertion of intermediate layers that enable the solution of non-linearly separable problems. Thus, MLP networks have at least three layers: the input layer, the intermediate or hidden layer and the output layer. A network with one hidden layer can implement any continuous function, with two intermediate layers, can approximate any mathematical function [CIBENKO, 1989].

2.2 Weka

Weka [WEKA, 2011] is a free software for data mining. Its implemented features and techniques are described in detail in [Witten & Frank, 2005], the implementers of the tool. One of the uses of the tool is the extraction of classifiers in databases. A

classifier (or classification model) is used to identify the class to which a specific observation in a database belongs from its characteristics (attributes). This tool is used in this work to classify BW queries into categories according to the scheduled time of execution.

2.3 SAP BW

SAP BW is part of the solution of SAP Business Intelligence software; whose main purpose is to store in a single location (data warehouse) separate from the transactional environment, data from different sources, facilitating the provision of analytical information for users in an integrated and uniform way. Because it is integrated with SAP ERP R/3 it does not need interface files, which provides highly reliable transfer and maintenance of the quality of the information. The tool encourages the generation of queries (queries) by the users, by means of user-friendly tools on the web and / or Excel [McDonald et al, 2006]. In this environment, information is stored in a structured manner to facilitate queries and analysis, thus supporting the decision making and management. The BW software is complementary to the R/3. While the R/3 is configured to optimally run transactions from day to day business (eg buy, sell, manufacturing) BW is optimally configured to allow analysis (eg how are my indicators going, how good are the sales of a product group to a customer, how is the stock of a particular product evolving, etc). The information can be extracted from SAP BW by creating queries using the Query Designer tool, which is also part of the suite SAP BW [PALEKAR et al, 2010]. This tool has a simple interface and can be easily used by the business users to build queries to support their analysis. Figure 2.1 shows the construction of a query with the tool query designer.

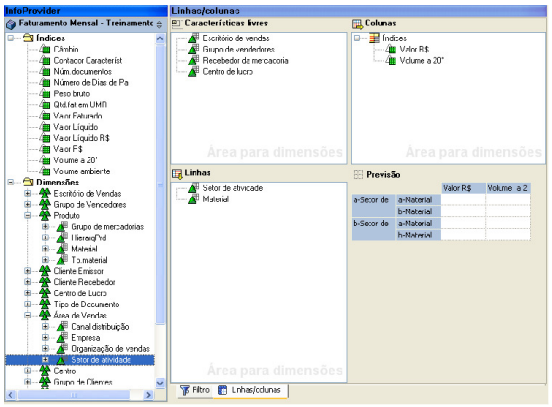

Figure 2.1. Creating a BW Query

After the creation of the queries either by business users or by IT staff of the company they are published and made available to the users so that they can extract the necessary data for their analyses. The execution of queries can be performed by the Business Explorer (BEx), a tool that is also part of the suite SAP BW. This tool is integrated with Microsoft Excel, the environment in which most business users are already familiar. Alternatively, the query may be executed through the web interface. Figure 2.2 illustrates the execution of a query using the web interface.

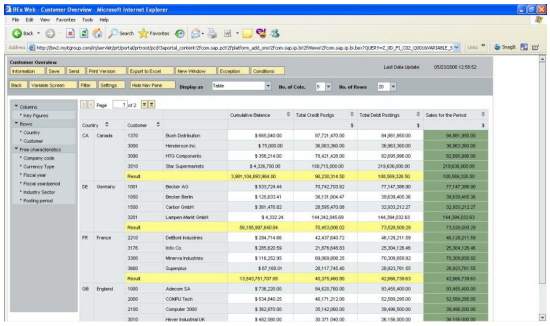

Figure 2.2. Executing a BW Query

This section presented briefly the basic concepts related to neural networks and the tools Weka and SAP BW, which are the basics necessary to follow the issues raised in this article. The next section presents the real problem addressed in this work.

3 PROBLEM DESCRITION

The SAP BW BI tool, as previously mentioned, is part of the day to day work of about 8000 users of the company, using queries to assist in the analysis inherent in their professional activities. Monthly, about 150 thousand executions of more than 8000 business user queries in the production environment are recorded.

A query is created to meet a specific need for business analysis, and can contain several input filters and several rows and columns that can be customized according to user navigation. Thus, some queries work with millions of records and take a long time to run. Besides relying on adequate filters, the query response time is also directly affected by the concurrent use of the BW server by other users. Thus, the response time of a single query using the same filters can be very different for two runs in different times.

The main problem today is that there is no way to estimate the execution time in advance, so that the user can decide whether or not to execute the query in the moment. Without this information, many users start a query and when they check that it is taking too long, they stop it. However, the query may continue to run on the server, overloading the machine unnecessarily.

If before executing the query the user had its forecast of execution time, he could decide whether to run it now or postpone it to a time that has faster response avoiding unnecessarily burden for the server with an application that will not be completed.

The execution time of all the queries from BW for each of the users is recorded by the BW Statistics BW SAP software another tool that is also part of the SAP suite. We intend to use this historical basis for predicting the execution time of queries based on previous runs. Valuable information is available from the BW Statistics, such as the user who executed the query, the total execution time, date and time of execution and number of lines returned.

There are two main benefits expected from implementing a solution to predict the execution time of queries from BW. First, users knowing in advance the predicted performance of a particular query may avoid its unproductive execution and the indefinite awaiting for the information he or she needs. Second, by possibly not executing a query classified as long for the moment, the user will relieve the server from an unproductive burden.

4 SOLUTION DESCRITPION

To execute a BW query, the user enters the desired input filters. Depending on the existing filters in the query, the same query may be specified for different periods of time. For example, for a period of one month or one complete year. Therefore, the filters play an important role in determining the size of the result set of the query. Hence, to minimize the problem of different running times of the same query caused by the difference of input filters which may imply in different sizes of the result, we have chosen for this work only queries that are always executed using the same filter. Thus, differences in execution time are caused only by the occupation of the server, not by higher or lower amount of records returned by the query. Thus, the queries chosen for this study were:

- **Query 1:** P_5_CO_P_COOM07M_00014 - Gross Administrative Expenditure
- **Query 2:** P_5_PS_P_PSCO02M_0029 – CAPEX
- **Query 3:** P_5_MM_P_MMPU14M_0002 - Total Purchasing

The following subsections will detail the process of selecting variables, handling of the historical database, setting adequate configurations of Weka and execution of simulations. The database used was extracted from BW Statistics for the years 2010 and 2011.

4.1 Variables choice and the database

The BW Statistics tool provides valuable information regarding the execution of queries. For this study, the information considered relevant in the first analysis is illustrated by Figure 4.1.

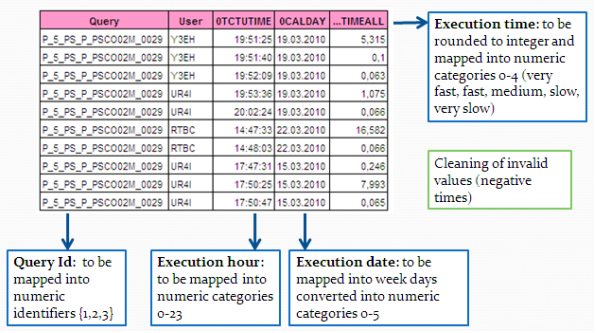

Figure 4.1. Relevant Information of BW Statistics

The column "Query" refers to the technical name of the query. As only three types of queries were used in this study, this field has been mapped to identifiers 1,2 and 3, representing the query used. The column "User" provides information on which user performed the query. As we consider that queries are executed the same way, regardless of the user, we do not consider this information relevant to this study. The information "0TCTUTIME" represents the time of execution of the query. This information is mapped into classes 0-23 representing the hour of the day when the query was performed. The column "0CALDAY" represents the date of execution of the query. As we consider the day of the week as the most relevant information instead of a specific date, this information was mapped into categories 0-5, where 0 represents Saturday or Sunday and the other values from Monday to Friday.

Finally, the column information "TIMEALL" represents the query execution time in minutes. This information is rounded to integer and then mapped into categories 0-4 numerical classification representing the query, as follows:

0: Very fast - up to 2 minutes
1: Fast - 2 to 5 minutes
2: Medium - 5 to 10 minutes
3: Slow - 10 to 20 minutes
4: Very slow - more than 20 minutes

In addition to these conversions, the database passed through a cleanup that eliminated inconsistent values, eg, negative execution time. No incomplete information was found in the database. The resulting database, consisting of 10,893 records, was divided into training set (2/3 or 7262 records) and test set (1/3 or 3631 records), both chosen randomly. Then the training and testing files for Weka were created, as detailed in the next subsection.

4.2 Weka Configurations and Execution of the Simulation

The training and testing files were generated from the database as shown in Figure 4.2. We used the Multilayer Perceptron (MLP) network topology and the BackPropagation (BP) learning algorithm.

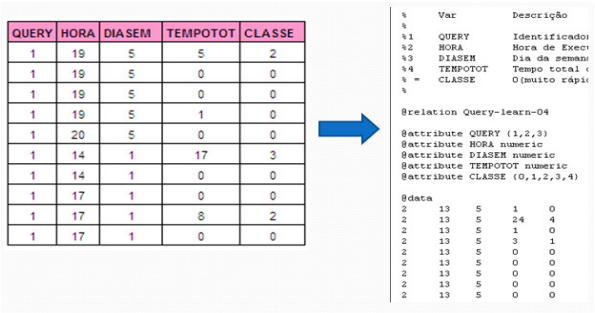

Figure 4.2. Creation of Files for Weka

Various configurations were tried in Weka in order to achieve the best classification rate in the samples. Figure 4.3 illustrates a neural network generated by Weka.

We used nine different settings, illustrated by Table 4.1:

I: Without normalization of the input attributes

II: With normalization of the input attributes and varying the neurons in the hidden layer as follows:

a) 4 neurons
b) 5 neurons
c) 6 neurons

III: With normalization of the input attributes and varying the number of training epochs as follows:

a) 1 epoch
b) 100 epochs
c) 1000 epochs

IV: With normalization of the input attributes and using a validation set. V: With normalization of input attributes and binary encoding of input attributes.

V: With normalization of input attributes and binary encoding of input attributes.

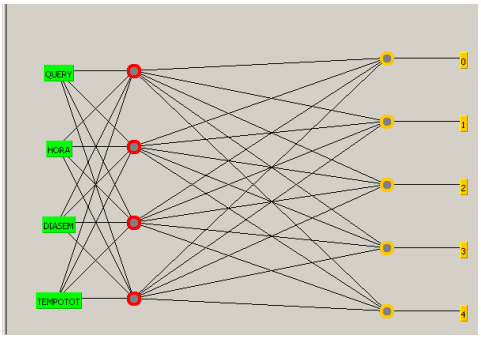

Figure 4.3. Neural Network generated by WEKA

Table 4.1. Weka Configurations

	I	II a	II b	II c	III a	III b	III c	IV	V
hiddenLayers	a	4	5	6	4				
nominalToBinary Filter	False								True
normalizeAttributes	False	True							
trainingTime	500	500	500	500	1	100	1000		
validationSetSize	0	0	0	0	0	0	0	10	0

This section described the proposed solution to the problem of predicting the execution time of BW queries. The next section presents the results.

5 RESULTS

After the execution of nine simulations as detailed in the previous section, different results were obtained. Table 5.1 summarizes these results obtained in different simulations. The results were compared using the following criteria (all in %):

Correct classification
Incorrect classification
Mean absolute error MAE
Root mean squared error RMSE
Relative absolute error RAE
Root relative squared error RRSE

As we can see in Table 5.1, the configuration that showed the best results was the configuration IIIc. This configuration used normalization of input attributes, four neurons in the hidden layer and 1000 epochs of training. It did not use the validation set and nor binary encoding of the input attributes. In contrast, the setting that showed the worst results was the setting IIIa. This configuration also used normalization of input attributes and four neurons in the hidden layer, but used only one training epoch, it also did not use neither the validation set and nor the binary encoding of the input attributes.

Table 5.1. Summary of Results

	I	II a	II b	II c	III a	III b	III c	IV	V
Epochs	500	500	500	500	1	100	1000	1000	1000
Hidden layer	a	4	5	6	4	4	4	4	4
Correct classification (%)	90,3%	90,0%	87,7%	87,9%	67,4%	81,5%	94,3%	82,6%	85,7%
Incorrect classification (%)	9,7%	10,0%	12,3%	12,1%	32,6%	18,5%	5,7%	7,4%	14,3%
Mean absolute error MAE (%)	5,5%	5,9%	6,2%	6,3%	21,0%	11,4%	4,8%	9,1%	7,4%
Root mean squared error RMSE (%)	17,0%	16,6%	19,3%	18,6%	32,1%	22,2%	13,4%	21,6%	21,4%
Relative absolute error RAE (%)	26,8%	28,8%	30,4%	30,8%	102,6%	55,7%	23,4%	44,4%	36,2%
Root relative squared error RRSE (%)	52,8%	51,7%	60,1%	57,8%	99,8%	69,1%	41,6%	67,2%	66,4%

There is a strong indication that for this particular problem and considering the data used, the number of training epochs was a decisive factor in the results, since as we reduce this number (settings IIIa and IIIb), the results become far worse. The variation in the number of processors in the hidden layer can also have some correlation, because when we start to increase this number (configurations IIb and IIb), the results begin to deteriorate, indicating that four processors in the hidden layer represent a good number for this problem.

The use of a validation set (configuration IV), which is usually a good practice, gave much worse results. In contrast, we expected a worse outcome when we do not use the normalization of the input attributes (configuration I), but the results were very similar (and even slightly better) than most of the settings that used the normalization.

6 EVALUATING THE SOLUTION

After performing the simulations and analysis of the results some suggestions of future work as well as some open questions aiming at improvements were found. These points are presented below.

6.1 Treatment of the Database

Examining the training base, after the simulations, we found that it showed some discrepancies that were not previously treated. We cite as an example, the number of occurrences of the second query (61% of training base), which is much larger than the number of occurrences of the query 1 and 3, as illustrated in Figure 6.1.

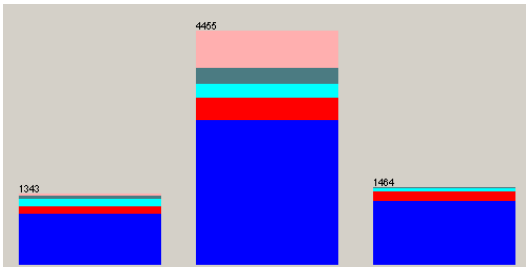

Figure 6.1. Distribution of Training Base per Query (1, 2 and 3)

We also observed by Figure 6.2 a discrepancy in the class of runtime: class 0 (very fast) represents 68% of the training base. In further work, minimization of such differences should be sought by means of homogenization techniques of the database.

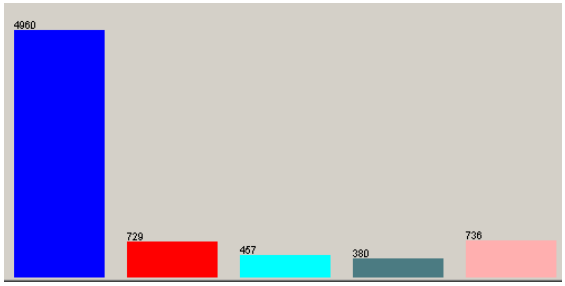

Figure 6.2. Distribution of Training Base per Class (0 to 4)

Another point observed from the analysis of the training base is the largest concentration of execution of queries between 11h30 and 0h (UTC), as illustrated in Figure 6.3. In future work, the use of hour intervals instead of the hour variable may possibly give more accurate results.

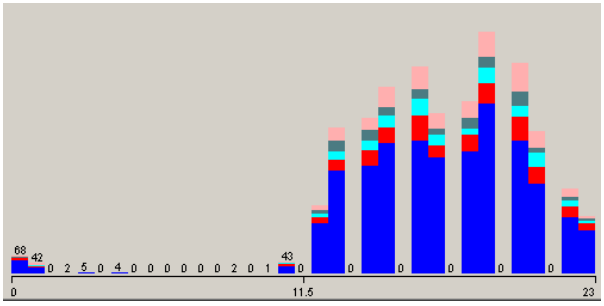

Figure 6.3. Distribution of Training Base per Execution Time (0 to 23)

6.2 Use of other queries

The aim of this paper is to investigate the application of neural networks in the problem of predicting the response time of BW queries. Only three types of queries were used in this study. New studies are required to confirm the results obtained, using a wider variety of queries. Moreover, as mentioned earlier, the parameterization of queries implies in different user input filters and, depending on the values used, the same type of query can return a very different number of records from one execution to another, influencing directly in its running time. Therefore, in future, we should not only use a larger number of queries, but also study a way to handle the customization of the filter input queries. One possible approach is to tackle the filter issue mentioned in section 4. The idea is to perform the classification of a query into different categories considering also the possibilities of filters and record for each execution, which filters were used and what were their values.

6.3 Integration of another system with Weka

Another interesting future work and extremely useful for the company would be the integration of the estimate generated by Weka with a system that the user could use to check the predicted time of execution of a query "now" versus a future time (eg, within 15 min, within 1 hour, in 4 hours). Thus, the user could decide the best time to execute the query and reduce the occupation of the server with unproductive and inconsequent queries.

6.4 Other methods

For future research, we intend to evaluate other methods for solving the problem of forecasting performance of BW queries, for example, fuzzy logic and neuro-fuzzy models. The idea is, after its implementation, to compare the results in order to raise issues not yet perceived for the problem at hand.

This section presented some suggestions for future work within the same theme explored in this article. The following section concludes this paper.

7 CONCLUSION

This paper investigated the application of neural networks in the problem of prediction of execution time of BW queries in a big organization where this represents a huge problem. Section 2 presented a summary on Neural Networks, the tool Weka and SAP BW. Section 3, in turn, detailed the problem in question and section 4 describes the proposed solution. Next, Section 5 presented the results and section 6 evaluated them, pointing out some possible future work.

The classification model utilized in this work was built using MLP networks and the BackPropagation algorithm. The setting that showed the best results used normalization of input attributes, four neurons in the hidden layer and 1000 epochs of training. It used neither validation set nor binary encoding of the input attributes. This configuration classified correctly 94.3% of the samples in the expected classes and proved to be the best configuration of this research.

Finally it must be pointed out that the implementation of such a mechanism to predict the execution time of BW queries is extremely useful to the company, not only to provide greater visibility to the users of the expected time of their queries, but also to avoid burdening the server with queries that are interrupted by the users because they are taking longer than expected. Thus, we intend to deepen this research using the suggestions for future work outlined above.

REFERENCES

[INMON, 1992] Inmon, W. H., *Building the Data Warehouse*. 1st Edition. Wiley and Sons, 1992.

[HAYKIN, 1999] Haykin, S., *Redes neurais - Princípios e prática*, 2a edição, Editora Bookman, 1999.

[JANG ET AL, 1997] Jang, J.S.R, Sun, C.T., Mizutani, E., *Neuro-fuzzy and soft computing: a computacional approach to learning and machine intelligence*. Prentice-Hall, Upper Sadler River, New Jersey, USA, 1997.

[ZADEH, 1992] Zadeh, L., *Fuzzy logic, neural networks and soft computing*. Proceedings of the 2nd International Conference on Fuzzy Logic and Neural Networks, Izuka, Japan, PP. 13-14, 1992.

[CYBENKO, 1989] Cybenko, G., *Approximations by superpositions of sigmoidal functions*. Math. Control, Signals, Systems, 2:303-314, 1989.

[HORNIK et al, 1989] Hornik, K., Stinchcombe, M., White, H., *Multilayer Feedforward Networks are Universal Approximators, Neural Networks*, Vol. 2, pp. 359-366, 1989.

[WITTEN & FRANK, 2005] WITTEN, I. H. e FRANK, E., *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Publishers, 2nd edition, 2005.

[WEKA, 2011] Site da ferramenta Weka: http://www.cs.waikato.ac.nz/ml/Weka/. Acessado em dezembro de 2011.

[McDONALD et al, 2006] McDonald; Wilmsmeier, Dixon, Inmon, *Mastering the SAP Business Information Warehouse*, Second Edition ed. Wiley, 2006.

[PALEKAR et al, 2010] Amol Palekar, Bharat Patel, and Shreekant Shiralkar, *A Practical Guide to SAP Netweaver Business Warehouse 7.0*. SAP PRESS, 2010.