DATA ANALYSIS PROGRAM BY MEANS OF NEURAL NETWORKS

A. V. Gavrilov, V. M. Kangler, S. A. Zaitsev

Novosibirsk state technical university, Novosibirsk, Russia,
e-mail: avg@vt.cs.nstu.ru, phone (3832)-46-04-92

Statement of a problem

Idea of application artificial neural networks even more often arises when they are solving various problems. Realization of this idea demands presence of special building. They should choice a model of neural network and its parameters correct, develop special software or study quite complex and software (neural networks environment for example). This fact makes using of neural networks quite difficult, especially at no technical fields. Architecture of the software, by which development attempt to simplify application artificial neural networks for analysis of databases was made, is considered in this article. The architecture allows using various models artificial neural networks for a solving of various problem classes.

When we were developing of architecture we recognized that the software should meet the following requirements:

- The program should provide realization of various kinds of the data analysis: the forecast, associative search, clustering and recognition.
- Program should carry out a role of a simulator that will help to develop and to study a various models of neural networks.
- Relational databases should be used as very convenient and universal and data source.
- The Program should contain means for a quality estimation of the carried out analysis that should allow to determine character of model parameters influence on neural network determining.
- Program should consist of a shell and dynamically linked modules. The shell should provide interaction by the user and databases, support of following data analysis stages (creation, training and use of a network). The modules should realize various neural network models and support an interface of interaction with the shell. Thus, after creation of the shell it will be possible to concentrate on work with neural network models, not distracting on the decision of questions of interaction with the user.
- A unified interface for interaction of shell and dynamic modules is necessary. The interface should support as the many neural network models as it's possible.

The architecture of the program

The program consists of a shell realizing common for various tasks functionality (access to data sources, reception of user commands, a conclusion of results of work etc.) and from neural network modules. Such division raises usability (the shell offers the unified interface for using of various neural network models) and also releases new neural network modules developers from necessity to realize the user interface.

In order to create flexibility and expansibility program interface between shell and dynamic modules the following decision was realized. . Each model is dynamic linked library (DLL - module), that should support the interface of interaction with shell. The library should be able to solve one or several tasks of the data analysis (the forecast, clustering, recognition and associative search) and support a data exchange protocol for each of them. To make the interface maximum flexible and expansible, the dynamic modules are realized as an active

part. Mutual linkage of shell and library is applied for this purpose. The library exports the set of functions realizing performance high-level commands (for example training, persistence, creating and so on). Its number does not exceed ten. The shell transfers to a network module a set of data and code pointers. They allow to obtain the exhaustive data on operation carried out by the user and also to operate data reading process.

The shell solves the following tasks:

1. Data reading from a database through SQL - inquiry. The automated creation of inquiry that does not demand a SQL knowledge presence is supported.
2. Interaction with the user (reception of commands and conclusion of result) for forecasting of time lines, recognition, classification and associative search is supported.
3. Following data types is supported: integers, floating and fixed point real values (double and currency), strings.
4. In order to increase neural networks work efficiency, the shell allows data preprocessing. At resolving forecasting tasks a trend can be used, that is an analogue of the first derivate and this allows to predict non-periodical data. It is possible to use encoding of segments in order to reach more efficient processing of fractional numbers; this makes impact from stochastic inaccuracies in input data lower. Trend is applied to the forecasting only. It is the first derivative analogue. Trend can be useful, when dependence is easier for finding out in change of value, rather than in its value. There is two kinds of trend: quantitative and qualitative in the program. Quantitative trend is a change of value, and qualitative - a direction of change.
5. Support of the dictionary. Dictionary is set of field values on which training of neural networks is made. The dictionary is necessary for data encoding by a network. For example, Hopfield - network demands for binary coding of the information (for example dictionary record number can be used for data encoding).
6. An estimation of quality of the carried out analysis. At the decision of all types of tasks the reports allowing the user to estimate result are formed. At test forecasting (when training samples are used) calculation of a forecasting mistake and formation of the mistakes report is made. The relative forecasting error distribution diagram (*fig. 1.*) also is generated.
7. Runtime connection to neural networks modules.

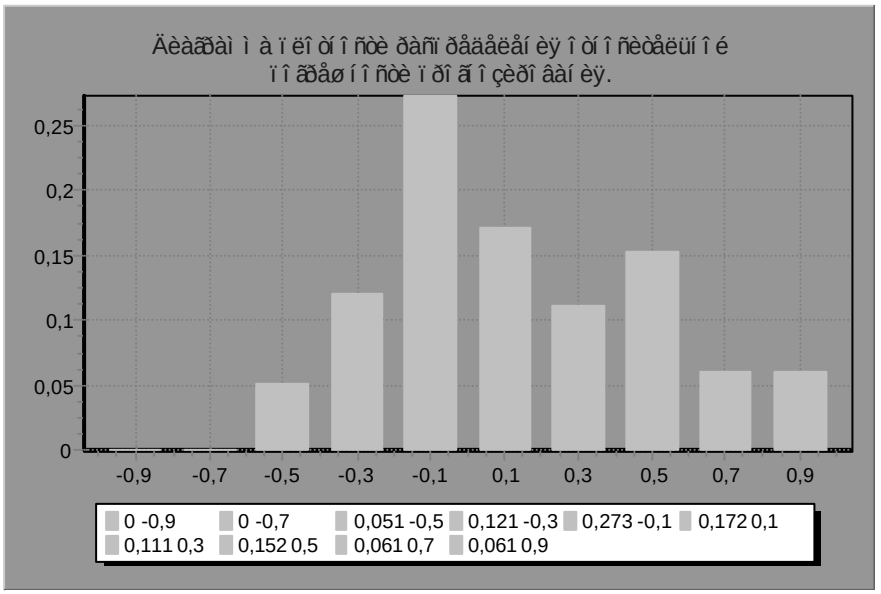

Fig. 1. The relative forecasting error distribution diagram.

The program using

Program using consists of two stages: preparatory and the basic. The preparatory stage consists of creation and training neural networks. The second stage is stage of network using and an estimation of its work results.

The preparatory stage consists of the following steps:

1. Project creation. In a basis of the project lays the SQL- inquiry. It defines training and test data sets. For project creation it is necessary to choose a data source. It can be a BDE - alias or a file system folder where DBF-tables are stored. After the data source is chosen, it is necessary to create SQL – inquiry returning necessary data.
2. Creation of the dictionary.
3. Creation of the neural network. For network creation it is necessary to choose the proper model, then you should select the task type. After that input and/or ouyput fields of neural network are selected. Having allocated an input or output field, it is possible to choose type of data preprocessing. On *fig. 2* neural network creation window is shown

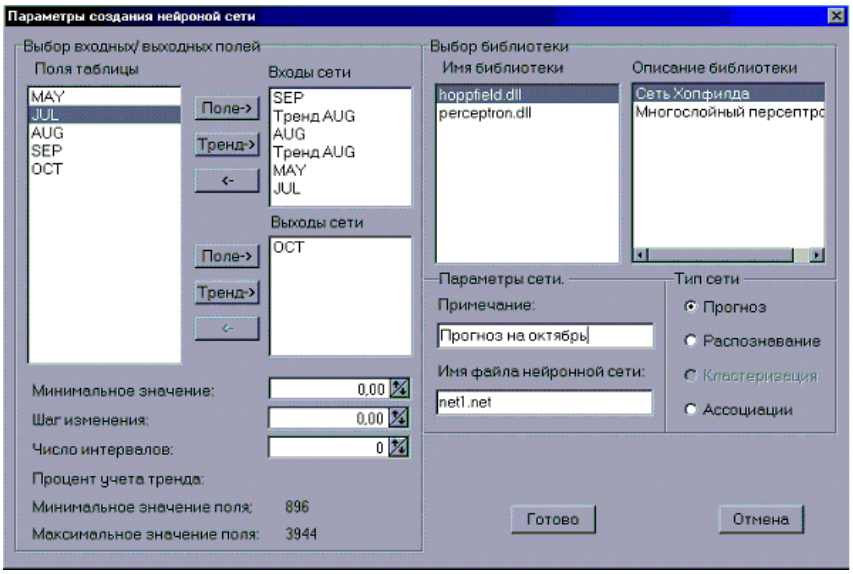

Fig. 2. Neural network creation window.

4. Network training. For training it is necessary to set a subset of records on which training will be made. After training the network is ready to use.

The basic stage consists of the trained network using. For the data analysis different dialogs are used (dialog type depends on a task type).

The conclusion

At present the described program is at the testing and condition stage. Already now with its help experiments with Hopfield network and multilayered perceptron are carried out. The shell can be used for debugging neural network models and for an estimation of models parameters influence on quality of the analysis. The further development of the project is planned in a direction of optimization of already created modules and development of the new libraries containing various neural network models. The work partly is maintained by Ministry of Education grant ÃÎ Î -4.1-48.

References

1. *Ãàâðèëîâ À.Â., Êàíãëåð Â.Ì ., Êàòîìèí Ì .Í ., Êîðîòêîâ À.È..* Îáíàðóæåíèå àññîöèàòèâíûõ âçàèìîñâÿçåé ìåæäó ïîëÿìè â áàçàõ äàííûõ ñ èñïîëüçîâàíèåì íåéðîííîé ñåòè// Òð. Ì åæäóíàð. íàó÷. – òåõí. êîíô. “Íàó÷íûå îñíîâû âû÷èñëèòåëüíîãî èíòåëëåêòà.” – Íîâîñèáèðñê, 1997. Ò.2. – Ñ. 210.
2. *Ãàâðèëîâ À.Â., Êàíãëåð Â.Ì* . Èñïîëüçîâàíèå èñêóññòâåííûõ íåéðîííûõ ñåòåé äëÿ àíàëèçà äàííûõ. - // Ñá. íàó÷í . òðóäîâ Í ÃÒÓ. - Í îâîñèáèðñê: Èçä-âî Í ÃÒÓ, 1999. - ¹ 3(16). - Ñ. 56-63.
3. *Gavrilov A.V., Kangler V.M.* The use of Artificial Neural Networks for Data Analysis // The Third Russian-Korean International Symposium on Science and Technology. - Novosibirsk: NSTU, 1999. - Proceedings/ - Vol.1. - P.257-260; Abstracts. - Vol. 1. - P. 192.
4. *Ãàâðèëîâ À.Â., Ãóáàðåâ Â.Â.* Ï ðèìåíåíèå ìîäåëè Õîïôèëäà äëÿ ðåøåíèÿ çàäà÷è ïðîãíîçèðîâàíèÿ íà ïðèìåðå àíàëèçà ïðèòîêà ðåêè Îáü. // 2-ÿ Âñåðîññèéñêàÿ íàó÷íî-òåõí . êîíô. "Íåéðîèíôîðìàòèêà-2000", Ì ., 2000. - Ñ. 33-38.
5. Ï ðîäóêòû äëÿ èíòåëëåêòóàëüíîãî àíàëèçà äàííûõ// Ðûíîê ïðîãðàììíûõ ñðåäñòâ. – 1997. ¹ 14, 15. – Ñ. 32 – 33.
6. *Êðóãëîâ Â.Â., Áîðèñîâ Â.Â.* Èñêóññòâåííûå íåéðîííûå ñåòè. Òåîðèÿ è ïðàêòèêà. – Ì .: Ãîðÿ÷àÿ ëèíèÿ – Òåëåêîì . – 382 ñ.