Evaluation of Neural Networks via generic modelling using System Dynamics

Ralf Koerstein, Bernd Viehweger (April 14, 1996)
Proceedings of the System Dynamics '96 Conference (forthcoming),
MIT, Cambridge, MA, U.S.A.

Abstract

We present an architecture of a software-system for computerbased experiments with neural networks. For generating the data we use models on the basis of the theory of System Dynamics. Practical experience was received in experiments with different neural networks and various amounts of data from the well known fishing model. The results can be useful for the evaluation of neural networks.

Keywords: *System Dynamics, Neural Networks, Simulations*

Problems of the choice and assessment of neural networks

Artificial neural networks have been established as means for the solution of non-linear and multi-causal problems. There are some big advantages compared with conventional statistical techniques. Neural networks are relatively rough dealing with noisy or chaotic data. It is not necessary that data must fit to predefined distribution assumptions. Neural networks can find such categories by themselves.

On the other hand in practical application there are some difficulties. There are various types of neural networks, with very different properties. Often it makes sense to make a preprocessing on the data. There exist different learning and optimization algorithms. Types of neural networks can be combined. Generally it is not possible to understand exactly how a neural network works internally because its knowledge is stored only implicitly in the connections between the neurons.

The consequence is that there is a lot of uncertainty concerning the accuracy and efficiency in the practical use. For the user it is often difficult to make a good choice of the many parameters and to estimate the quality of his results.

In general neural networks have a complex internal organisation - there are lots of non-linearities, therefore it is not possible to analyze them completely with analytical methods. Theoretical results have to be completed with practical experiments and heuristic rules.

It is a problem that you often have not enough data for systematic experiments. In many fields there are scarce long time series because of the difficulty to collect a huge amount of data (it is expensive and takes a lot of time). Additionally empirical data have another disadvantage: Generally we do not know how a real system works. This is a disadvantage because thus it is not possible to judge with absolute certainty the accuracy of the neural network system. We do not know the rules behind the processes that generate the data. On the other hand we can generate huge sets of data by the use of mathematical functions and computer programs. Arbitrarily generated data normally have no useful relevance. Models based on the theory of System Dynamics can be an attempt to solve this problem.

Generic models of System Dynamics to generate test-data

We believe that System Dynamics is an established method to generate huge amounts of test-data in a systematic way. By the use of feedbacks and non-linear equations you can often get a good correspondence between models and parts of the reality. A set of methods for the systematic analysis of dynamic systems exists: It ranges from qualitative effect analysis, to system diagrams, computer implementations and to derivations by means of mathematical system theory. Using the existing computer languages for simulation and tools for visualization it is possible to construct System Dynamics models with little expense and waste of time. The System Dynamics concept is very flexible and you can easily construct models which show characteristic system behaviour, often seen in reality in different fields.

"Same system structures result in the same behaviour, even if the systems are very different in their physical elements."[BOSSEL94]

An example for a partial taxonomy of characteristic models is the so called "Systemzoo" by Bossel: It is a collection of 50 elementary systems, which have recurrent importance for modelling reality. Some of these systems describe important processes of daily experience. Many of these primitive systems are part of composite systems. The knowledge of the structure and behaviour of non-linear elementary systems can help to understand more complex systems.
The first approach to systematic experiments begins with applying neural networks to elementary systems.
This way we can draw general conclusions concerning the efficiency of neural networks in prototypical situations. Parts of the System Dynamics theory can be applied to the field of neural networks. Meaningful tasks for the neural networks are prediction and control of system behaviour. For example unstable systems can be stabilized by adding new elements to the structure. These elements can also be learning neural networks. Obviously a big advantage of using System Dynamics modelling is that the neural networks can learn controlling tasks in simulations, which is not possible in reality because of the risk a real experiment might have.
In a next step it can be examined if it is possible to transfer the results from the elementary system structures to composite complex systems. Similarly we can take back the effects of the reduction of complexity in models by systematically adding new elements. With the aid of such experiments we get more detailed statements about neural networks stability under changeable conditions.
In a further step the experiences gained by the use of neural networks can be tested on problems of reality. The last achievement is to have a finer categorization of different neural networks with respect to their functions in mastering prototypical situations.

System architecture of an integrated environment for neural network experiments

For the systematic making and recording of experiments we use the following system architecture as an infra-structure:

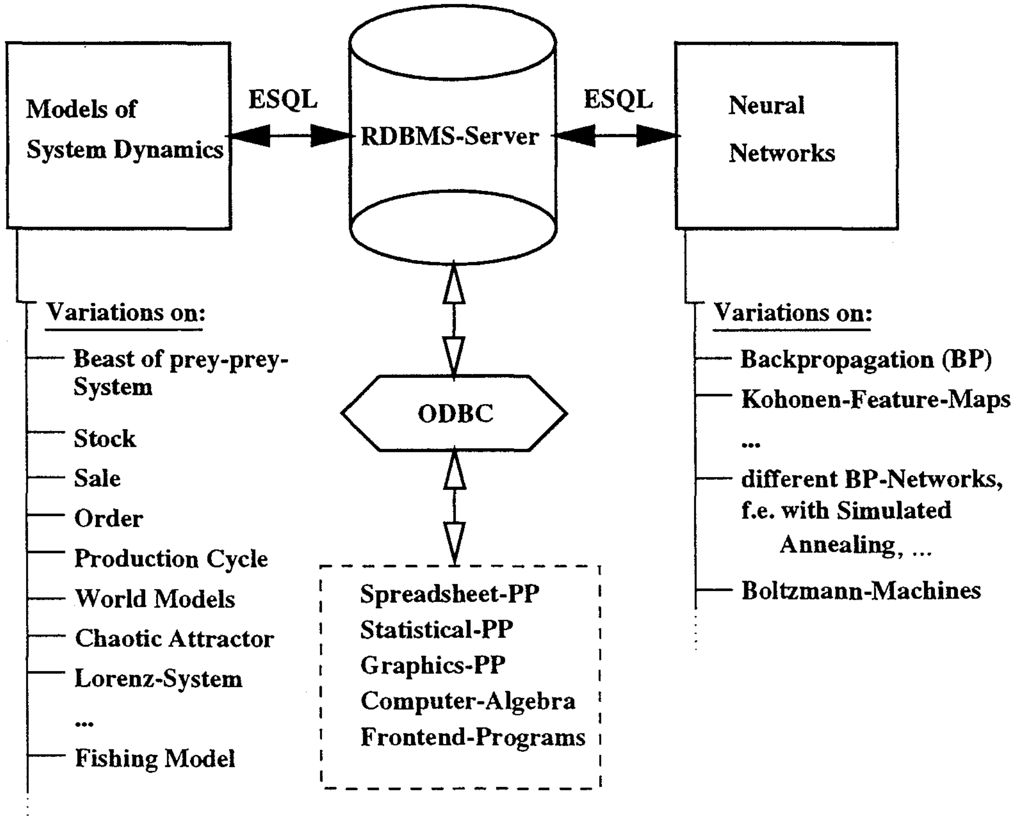

Fig. 1: **Architecture of an environment for neural network experiments**

So far the equations of System Dynamics models are implemented in a 3GL language (Pascal or C++). Input and output of the data are done by embedded SQL-statements according to the ISO Norm. Neural network programs get an SQL interface as well [MASTERS93] [MASTERS95]. All the experimental data are stored in the data base management system. In the future we will add meta-informations and additional knowledge, for example relevant theoretical results from the mathematical research. We offer tables and forms to experimentators with predefined parameters, which will be refined and improved in the course of time.

Using the standard ODBC (Open DataBase Connectivity) interface it is possible to connect application programs directly to the RDBMS (Relational DataBase Management System), for example statistical packages or computer algebra systems. With spreadsheet programs we can do preprocessing of the data in an integrated way. So designers of neural networks receive means to classify, validate and evaluate their creations. A collection of standard data generated by System Dynamics models

with known mathematical rules serves as a basis for that.
For additional experiments we can store empirical data in the data base system and use them to build System Dynamics models or to use neural networks directly on these data.

Results and future projects

The most complex application we implemented was the well known fishing model. We tested the capacity of different feedforward-networks in approximation of the whole model and in prediction. For example: We introduced random parameters for the price of fish. So the model was repeatedly treated with exogenous shocks and new balance points have been found. The different variations of neural networks had the task of estimating the amount of catched fish or respectively of making a prediction of the development in the future. We were surprised to see how the choice for some parameters of the neural networks was very critical for the success of the experiments, while several other options were of minor importance.
As we expected, many neural networks were very good function approximators provided that the choice of the parameters was correct. The possibility of making good predictions was naturally limited by the amount of random influences. On the other hand we could realize evident differences in the variations of the neural networks.
In the future the number of datasets and neural networks in the database system shall significantly be extended. We believe that it is necessary to strengthen the technical connection between neural networks and System Dynamics modelling. This will lead to a better automatization of the experiments.

Literature:

[BOSSEL94] H. Bossel: Modellbildung und Simulation, 2nd edition, Vieweg; Braunschweig, Germany, 1994.
[MASTERS93] T. Master: Practical neural network recipes in C++, Academic Press Inc., San Diego CA, U.S.A., 1993.
[MASTERS95] T. Masters: Advanced algorithms for neural networks, John Wiley & Sons Inc., New York, U.S.A., 1995.

Address:
Ralf Koerstein, Dr. Bernd Viehweger; Humboldt University, Department of Economics,
Institute of Wirtschaftsinformatik, Spandauer Str. 1, D-10178 Berlin, Germany.
E-Mail: koerstein@wiwi.hu-berlin.de, bv@wiwi.hu-berlin.de