

2007

Intelligent Decision Support Systems and Neurosimulators: A Promising Alliance for Financial Services Providers

Michael H. Breitner

Leibniz Universität Hannover, breitner@iwi.uni-hannover.de

F. Koller

koeller@iwi.uni-hannover.de

S. König

koenig@iwi.uni-hannover.de

Hans-Jörg von Mettenheim

Leibniz Universität Hannover, mettenheim@iwi.uni-hannover.de

Follow this and additional works at: <http://aisel.aisnet.org/ecis2007>

Recommended Citation

Breitner, Michael H.; Koller, F.; König, S.; and Mettenheim, Hans-Jörg von, "Intelligent Decision Support Systems and Neurosimulators: A Promising Alliance for Financial Services Providers" (2007). *ECIS 2007 Proceedings*. 107.
<http://aisel.aisnet.org/ecis2007/107>

This material is brought to you by the European Conference on Information Systems (ECIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ECIS 2007 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

INTELLIGENT DECISION SUPPORT SYSTEMS AND NEUROSIMULATORS: A PROMISING ALLIANCE FOR FINANCIAL SERVICES PROVIDERS

Michael H. Breitner, Frank Köller, Simon König, Hans-Jörg von Mettenheim
{breitner|koeller|koenig|mettenheim}@iwi.uni-hannover.de
Institut für Wirtschaftsinformatik, Königsworther Platz 1, 30167 Hannover
<http://www.iwi.uni-hannover.de>
Leibniz Universität Hannover
Germany

Abstract

Today self-organization and automatic usage of Artificial Neural Networks (ANN) are common in various applications for financial services providers. We analyze typical advantages and disadvantages of ANN and discuss the question: For which tasks ANN applications are most promising? We show that Intelligent Decision Support Systems (IDSS) based on ANN and Neurosimulators can support today's complex decision processes, e. g., investments or operation of a customer contact/call center. The focus is on supervised learning, here: ANN are trained with patterns from well-understood decision processes in the past. Then these ANN can benchmark a posteriori, forecast a priori or transfer knowledge to similar or analogous decision processes. Often efficient supervised learning needs advanced optimization algorithms, thin client solutions and low budget high performance computing, i. e. grid computing. Computations are realized with the neurosimulator FAUN (Fast Approximation with Universal Neural Networks), which is developed by the authors since the mid 1990's. We shortly present a long-term ANN interest rate forecasting model first. Then an ANN option/warrant market-pricing model and an ANN human-resource allocation model for contact/call centers are outlined briefly.

Keywords: Intelligent Decision Support Systems, Artificial Neural Networks, neurosimulator, forecasting model, option pricing, contact/call centers

1 MOTIVATION: WHY NEUROSIMULATION TODAY?

Intelligent decision support systems (IDSS, or shortly DSS) are interactive computer-based systems which identify problems and support decision processes to solve these problems. The IDSS are based on communications technologies, human inputs, data, documents, knowledge and models. Today many IDSS use Artificial Intelligence (AI) computer programs and autonomous robots for decision making, see (Gupta, 2006; Matsatsinis and Siskos, 2003; Turban, Aronson, and Liang, 2005) for an overview. AI programs often learn from a priori given processes, data, etc. and corresponding appropriate decisions, i. e. relations of values of input and output variables. This is called supervised learning of a priori given, structured patterns. Among today's AI approaches are Artificial Neural Networks (ANN), Fuzzy Logic based systems, Genetic Algorithms and hybrid systems, e. g. neuro-fuzzy networks and neuro-genetic programming. Here we focus on powerful ANN approaches and applications and a so-called neurosimulator implementation. Supervised learning with ANN is – mathematically – closely related to nonlinear regression. Nevertheless many neurosimulation ideas like cross-validation and generalization quality are not known in nonlinear regression theory.

At financial services providers IDSS are typically used for decisions under uncertainty, e. g. seller and buyer price decisions, investment decisions, financing decisions, working capital decisions and risk hedging decisions, for problem analyzes and for early warning concepts. Management generally must split limited resources between competing opportunities or projects. The outcome is a function of size, timing and predictability of future cash flows. For example in corporate finance short-, mid- and long-term techniques and decisions are needed to maximize return on capital minus cost of capital with limited financial risks. Techniques are, e. g., time series forecasts for market supply and demand, for interest rates and for exchange rates: Mathematically an extrapolation from known data, problems, etc. Techniques are also, e. g., market analyzes like the approximation of over the counter (OTC) option prices with available prices for standardized options: Mathematically an interpolation between known data, problems, etc. Especially today's risk management for assets, foreign currency cash flows, credits, etc. usually is based on risk estimation and risk hedging. Stochastic and AI approaches are very useful for, e. g., process analyzes and pricing of derivative instruments. In comparison stochastic approaches are mainly based on theoretical models while AI approaches try to learn from data and "observations" with easy to use black-box algorithms.

The history of neurosimulation has known ups and downs. Neurosimulation started in the 1960's with high expectations to which it never lived up. A quiet time in the 1970's and 1980's followed with hardly significant research happening. However, in the mid 1990's the ever increasing computing power sparked a new research impetus. Mostly ANNs didn't leave the research labs as the software seemed still too complicated to set up. Computation times of days or weeks were prohibitive to real time usage and effective parallelization wasn't implemented. Now, the situation has changed. Networks of hardly used workstations at banks and insurance companies offer computing power which is only waiting to be put to purpose. High bandwidth allows fast interconnection, even world-wide. Access to financial data is easy and often cheap. On the other hand we see an increasing interest in parallel, multi-threaded and gridded programming as the hardware industry promotes dual- or even quad-core processors. It therefore seems logical to us to present an almost out of the box financial application for today's financial services providers. We will discuss three examples which stand for domains typically arising at banks or insurance companies.

Neurosimulation is usually based on complete software emulation. The FAUN (Fast Approximation with Universal Neural networks) neurosimulator enables supervised learning 3- and 4-layered perceptrons and also radial basis functions. A well trained ANN is a mathematical function which approximates the output of the patterns sufficiently accurately. This ANN reasonably approximates the patterns (generalization). Here the word "reasonable" summarizes different quality factors for the trained ANN, i. e., the

mathematical function. E. g. highly frequent oscillations which do not correspond to the patterns should be avoided. FAUN neurosimulator highlights are:

- The software development is based on today's software quality principles, i. e. suitable, correct and adequate functionality, high stability and error tolerance and good restart ability, user friendly documentation, self-learning and ergonomics, good overall performance and moderate allocation of resources, e. g. cache or RAM, good maintainability (readable, changeable and testable source code) and high portability (easy installation, high compatibility with different hardware and operating systems and high convertibility);
- The ANN types and topologies supported are very powerful for real life problems;
- Training and learning algorithms are based on approved numerical methods for constrained optimization problems and nonlinear least-squares problems, this makes FAUN one of the best available neurosimulators for *supervised* learning;
- Fully automatic prevention from over-learning (adaptable drop out rule);
- Various advanced graphics enables online or a posteriori supervision and analysis of an ANN training and learning. Special graphics helps, e. g., to detect outliers in the patterns;
- Support of optimized Java-, Fortran- and C-code Output of the best trained perceptrons;
- FAUN runs locally on most Windows, Unix and Linux computers. With the help of the web-frontend (thin client implementation for a web-browser) an ANN training on a Linux compute server can be controlled remotely and asynchronously (same look-and-feel);
- A FAUN-HPC (high performance computing) family for parallel, vector and grid computers is under development since 1999. Challenging problems needing CPU-months or even CPU-years computing time already have been solved, see (Breitner, 2003; Köller, 2007; Köller and Breitner, 2005, 2006; Mehmert, 2000; Mettenheim and Breitner, 2005).

2 NEUROSIMULATION

2.1 Supervised Learning

Supervised learning means, that for input-/output data (x_i, y_i) with input $x_i \in \mathbb{R}^{n_e}$ and target output $y_i \in \mathbb{R}^{n_a}$, $i = 1, 2, \dots, n_m$ an approximation function $f_{app}(x; W) : \mathbb{R}^{n_e} \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n_a}$ is calculated, which is as good as possible. $f_{app}(x; W)$ has derivatives of all finite orders in x and the elements of W . The elements of W are the weights of each layer of the ANN. Depending on the problem the pattern set is divided into training set, with n_t patterns, and cross-validation set with $n_m - n_t$ patterns. With W the approximation quality of a ANN can be estimated with the error functions

$$\begin{aligned}\varepsilon_t(W) &:= \frac{1}{2} \sum_{i=1}^{n_t} \sum_{k=1}^{n_a} (f_{app_k}(x_i; W) - y_{i,k})^2 \text{ and} \\ \varepsilon_v(W) &:= \frac{1}{2} \sum_{i=n_t+1}^{n_m} \sum_{k=1}^{n_a} (f_{app_k}(x_i; W) - y_{i,k})^2.\end{aligned}\tag{1}$$

A perceptron is trained iteratively, i. e., ε_t is decreased by adaption of W , until ε_v increases for two consecutive iterations (prevention of overtraining). Note that the training stops before a local minimum of ε_t is reached. Weight upgrades $W_{\text{new}} - W_{\text{old}}$ can be calculated with any minimization algorithm, e. g., with usually favored first derivative methods such as the steepest descent or with second derivative methods such as Newton method. For first derivative methods we have the iterative sequence

$$W_{\text{new}} = W_{\text{old}} + \eta(\varepsilon_t(W_{\text{old}}, \text{grad}_W \varepsilon_t(W_{\text{old}})) \Delta W(\varepsilon_t(W_{\text{old}}), \text{grad}_W \varepsilon_t(W_{\text{old}}))\tag{2}$$

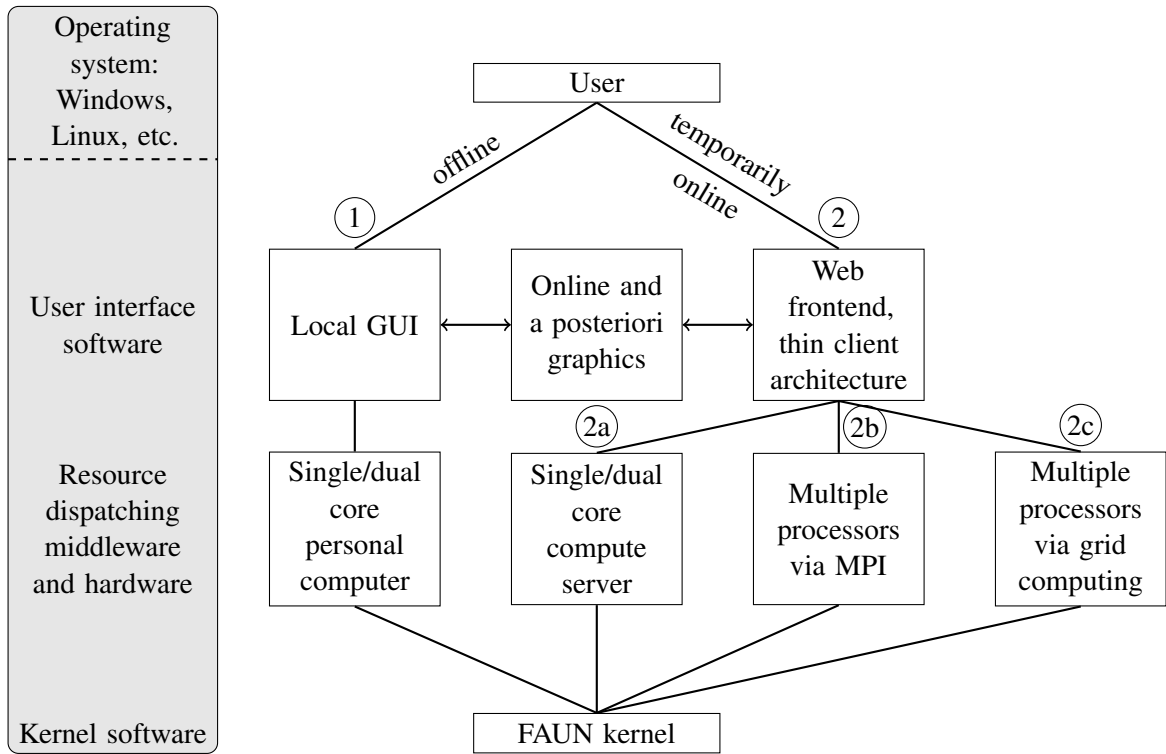


Figure 1: 3-layer architecture of the FAUN software suite. Users choose between local installation (1) or web frontend to access FAUN (2). The middleware distributes tasks user-definable to the FAUN compute kernel on one (2a) or many processors (2b and 2c). Applications of every layer are independently replaceable and available for Windows and Linux.

with search direction W and with step length η . Approved numerical methods for constrained nonlinear least-squares problems, see (Nowak and Weimann, 1998), are sequential quadratic programming (SQP) methods and generalized Gauß-Newton (GGN) methods which can exploit the special structure of the Hessian matrix of ε_t , see (Deuffhard, 2004; Fletcher, 2000; Gill, Murray, and Wright, 2004). SQP and GGN methods can automatically overcome most of the training problems of perceptrons, e. g., flat spots or steep canyons of the error function ε_t . They usually approximate the Hessian matrix of ε_t by finite-differences and update formulas, i. e., become first derivative methods, and can deal with box constraints, linear constraints and smooth nonlinear constraints.

Advantages of these methods are:

- A much better search direction ΔW is calculated in comparison to common training methods, e. g., $\Delta W := \text{grad}_W \varepsilon_t$ for the gradient method (backpropagation);
- The step length η is optimized permanently in contrast to common training methods with fixed step length. The number of learning steps is reduced significantly (factor 10, 100, 1000, ...);
- Only ε_t , $\text{grad}_W \varepsilon_t$ and ε_v are required which mainly can be computed by very fast matrix operations. For other ANN topologies, e. g., radial basis functions, an efficient code for $\text{grad}_W \varepsilon_t$ can also be deduced by automatic differentiation;
- Maximum and minimum of each weight can be set easily (box constraints);
- The total curvature of the ANN can be constrained (prevention from ANN oscillations);
- Convexity and monotonicity constraints can be set.

2.2 Neurosimulator FAUN

There are many commercial and public domain (free- and shareware) neurosimulators, like, e. g., SNNS, MemBrain or FANN among others. But the software FAUN is standing out by special features: Based on sequential quadratic programming (SQP) FAUN efficiently trains three and four layer perceptrons with and without shortcuts and radial basis ANNs. A perceptron's error function and its gradient are computed with matrix algorithms implemented with the BLAS (Basic Linear Algebra Subprograms, see <http://www.netlib.org/blas/>). The SQP methods NPSOL (Nonlinear programming problem solver) and NLSSOL (Nonlinear least squares problem solver) are based on the BLAS, too. Optimized and fine-grained parallelised implementations of the BLAS exist for various hardware platforms and most Unix, Linux and Windows systems. The coarse-grained FAUN parallelization uses the PVM or MPI sub-routines and runs on heterogeneous and decentralized networks interconnecting many general-purpose workstations and PCs and also high-performance computers. Figure 1 shows the structure of the FAUN software suite.

FAUN synthesizes functions from high-dimensional input-/output-relations. The synthesis of functions is of importance, e. g., for mathematical modeling and the calculation of optimal feedback controls for optimal control problems. Time series-analyzes and -forecasts, are important for many economical and technical applications, e. g., for interest- or exchange rate forecasts. Exemplarily the optimization of a general queuing system's performance and an option pricing with market prices are considered.

3 SOFTWARE APPLICATION SHARING AND GRID COMPUTING

3.1 FAUN Thin Client Web-Frontend and MPI Version

The first releases of FAUN had to be controlled by editing files. As it evolved a graphical frontend was developed to ease this configuration process. Additionally this frontend supports creation of graphics to analyze the quality of the trained ANNs. Figure 2 shows a screenshot of the web-frontend. It is now possible to setup and run FAUN completely remotely. It has all the features of the graphical (local) frontend and supports remote computation. Thus a highly performant computer can be used as compute server and accelerate the training of ANNs. That way a local workstation does not need to run during the whole FAUN computation.

FAUN was refactored to isolate and encapsulate program parts that are independent from each other, see (Fowler, 1999). To ensure that FAUN still works as expected after applying changes tests have been written. These tests are written before any change is applied to the source code and therefore ensure a working program at any time during development, see (Beck, 2002). Additionally these tests document usage of program parts and serve as information source to future developers. The exchange of data has been reimplemented with MPI (Message Passing Interface), using a test first approach, too. Starting MPI-FAUN is hardly different from the non-MPI version. It integrates seamlessly into the web-frontend.

3.2 Grid Computing

Grid computing is normally used when high performance in a broad sense is needed (Abbas, 2004). This can mean high availability or HPC (High Performance Computing) in its actual sense, i. e., raw computing power is required. FAUN is ideally suited for running on a so-called Beowulf (Brown 2000) style of cluster, which consists of COTS (Commodity of the Shelf) hardware. Most institutions or companies have unused computing capacities. This leads to the wish to use these resources comfortably. A middleware is needed that interfaces FAUN with basic management functions.

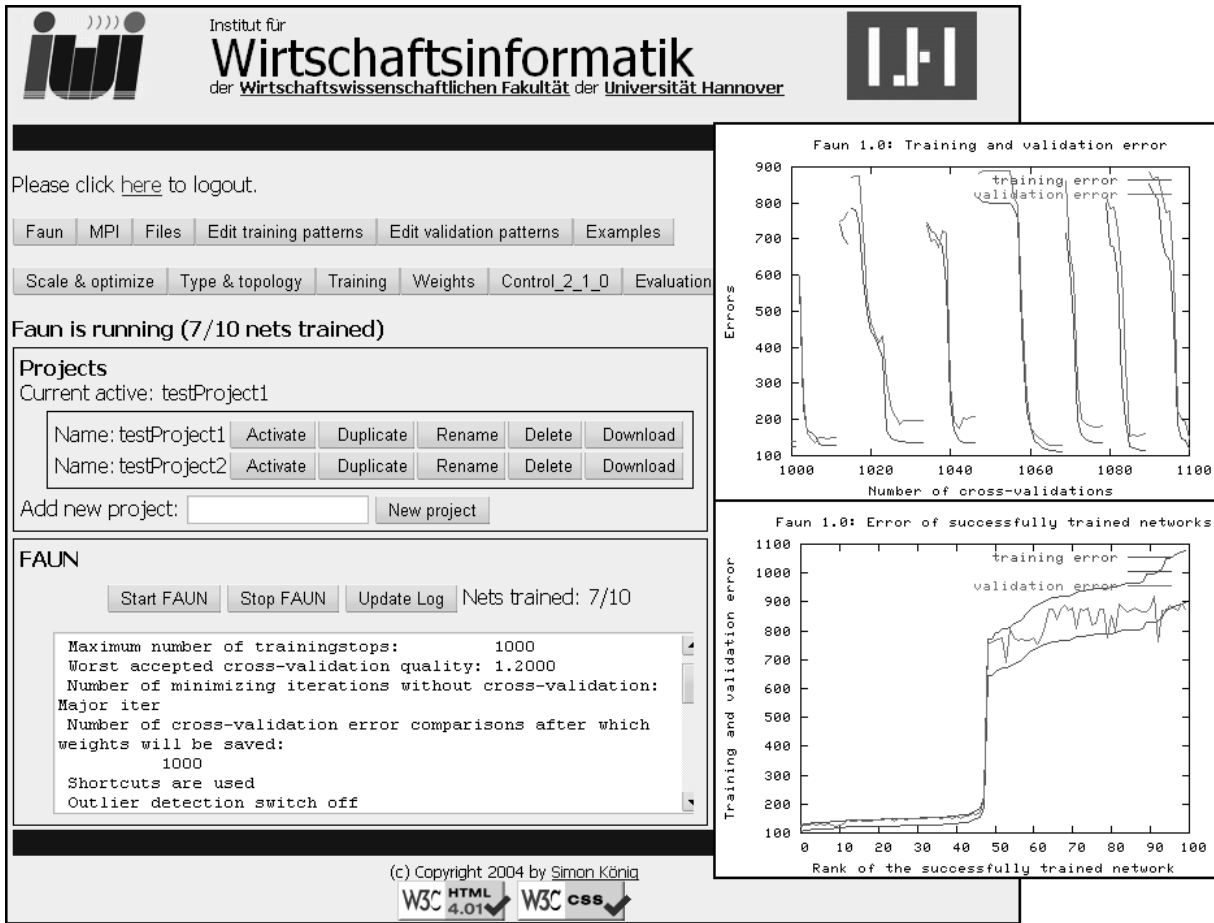


Figure 2: FAUN web-frontend page for a typical project. In the “FAUN” sub-window the last FAUN kernel messages are displayed. The two graphics right are examples for on the fly graphics: Top both training and validation error histories can be seen for several ANN, below all successfully trained ANN are ranked by their training error.

The FAUN grid computing client is designed to be a simple generic wrapper for distributed computation (Mettenheim and Breitner, 2005). It is written in Ruby, a modern fully object oriented and platform independent language. A speed-up factor of more than 0.9 can be achieved. The slight performance loss is due to communication overhead, which can best be explained by the famous potato peeling problem: Imagine a bunch of potatoes that have to be peeled. A single person will work for a long time. If a second person helps the peeling will probably be twice as fast. But if more and more persons join in the peeling time will be spent on handing the potatoes from one to another and the entire process will be more and more inefficient. Finally the largest potato determines the minimum time achievable.

4 IDSS FOR FINANCIAL SERVICES PROVIDERS — EXAMPLES

The following examples are chosen with the background that they cover important tasks for financial services providers. These include forecasting applications or determining fair prices for financial products. The presented examples are probably mostly supported by software although not using ANNs. Applying ANN to any sort of problem involves two methodological steps. First, a mathematical model of the problem has to be created which can be fed into the neurosimulator. This is typically achieved by observ-

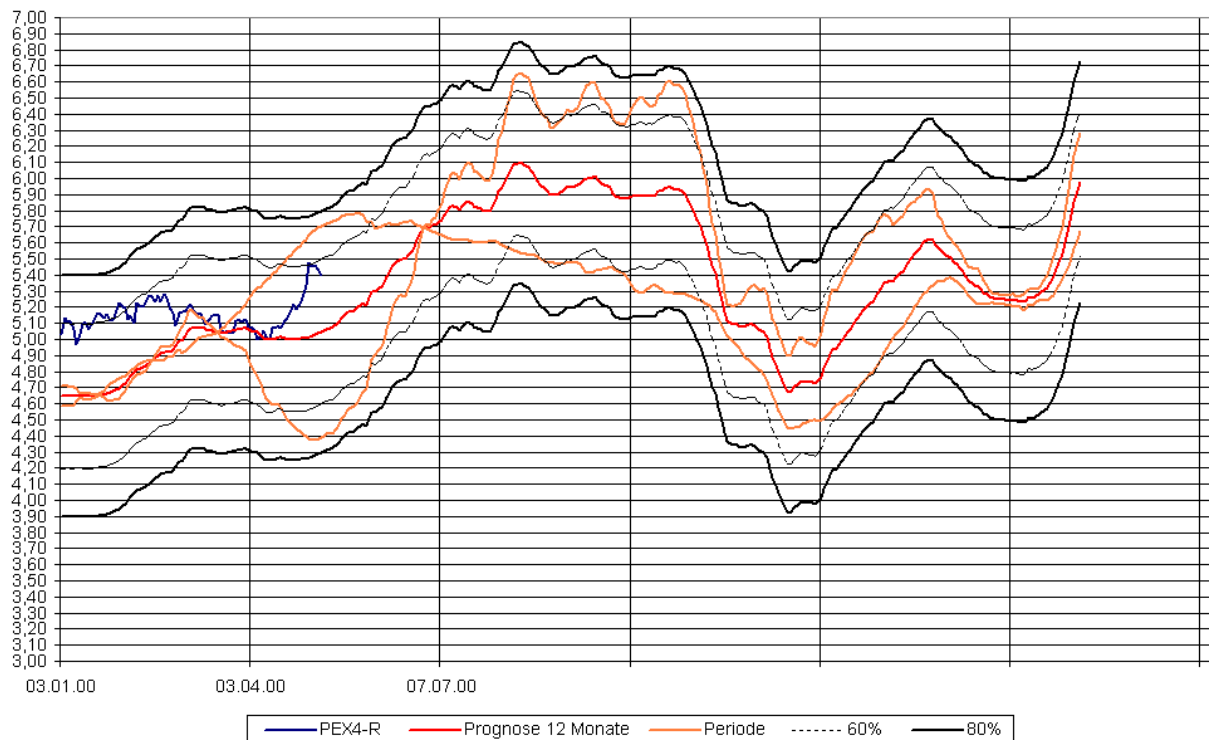


Figure 3: Excel graphics of an interest rate forecast model for the German PEX-4 based on ANN. The model has been developed and upgraded 1999–2001. Various fluctuation bands computed with long-term time series (more than 15 years) are visible. This forecast model is part of an IDSS of the Sparkasse Goslar/Harz. Sub models for the PEX-1, PEX-4 and PEX-10 and for horizons of 3, 6 and 12 months are implemented.

ing the processes. Second, the resulting ANNs are empirically analyzed. Their results are compared to real-life figures. If necessary the model is modified and new ANNs are trained.

All examples are *real*, i. e., real (live) data are taken and the results are verified on-site where appropriate. We will happily provide all data used for the training as well as a FAUN test account to the interested reader. Please contact us.

4.1 Example: Interest Rate Forecast

IDSS typically are used for decisions under uncertainty, e. g. financing decisions and risk hedging decisions. The technique presented here for interest rates forecasts is mathematically an extrapolation from known data. ANNs can be used for the trend approximation of unknown values, see Figure 3. So in particular the employment of ANNs in fiscal problems becomes interesting, because a reliable forecast of market data, e. g. stock prices or interest rates, facilitates reasonable business decisions, see also, e. g., (Aminian, Suarez, Aminian, and Walz, 2006; Oh, Kim, Lee, and Lee, 2005). This leads to possible increases in profit and can limit the financial loss, respectively. Here ANNs serve as short to medium term forecast of the returns from the german covered bond Pfandbrief for three, six and twelve months. These returns result from the market prices of the Pfandbrief index PEX. The historical market price series of the PEX from the beginning of 1988 forms the basis of the training for the ANNs. So the presented forecast model is a technical analysis of the market price process. The forecast quality is calculated for the forecast horizons with terms of one, four and ten years. The resulting forecast model displays a better forecast capability compared to the naive forecast, see (Breitner, 2003; Mehmert, 2000).

4.2 Example: Heuristic Option Pricing

A second technique for IDSS is a regression of known data, e. g., market analyzes like the approximation of over the counter (OTC) option prices with available prices for standardized options. In today's option and warrant pricing based on deterministic option models the underlying price is mirrored inadequately. Particularly the very important option price sensitivities (option Greeks) usually are very inaccurate. The estimation of options' risk and chance and a hedging for their issuers suffer from these inaccuracies. In contrast to the theoretical pricing models a heuristic pricing model based on highly accurate ANN approximations can learn true market pricing of options and warrants (options confirmed by a security), see (Andreou, Charalambous, and Martzoukos, 2006; Breitner, 2003; Chen and Magdon-Ismail, 2006; Priddy and Keller, 2005; Tzastoudis, Thomaidis, and Dounias, 2006; Wang, 2005; Zell, 2003) for theory and applications of ANNs. Like the theoretical option price the heuristic option price depends on the permanently available underlying price, strike price and time to expiration.

- **BASF Call Warrant:** Here topological appropriate ANNs are given, e. g. by three-layer perceptrons with or without shortcuts. We take true market prices (Frankfurter Wertpapierbörse) for BASF stocks and 74 BASF stock call warrants over a period of one and a half year. As usual the perceptron is trained iteratively, i. e. ε_t is decreased by adaption of W , as long as $\varepsilon_v < \varepsilon_t$ or $\varepsilon_v \approx \varepsilon_t$ holds (prevention of overtraining). Thousands of multi-layer perceptrons with various topologies and with different weight initializations are trained with the fast SQP method. With an expert council ANN all options and warrants can be compared to single out overpriced and underpriced ones for each trading day. The neural model gains deep insight into the market price sensitivities (option Greeks). The good approximation properties of the expert council ANN for up to 2 future weeks facilitate the estimation of options' risk and chance and the hedging of/with options, see (Breitner, 2000b, 2003) for comparisons of our model with the warrant development.
- **Euro/US-Dollar Call Warrants:** In the example above ANNs are trained over the whole period of one and a half year and were evaluated on the last day. In contrast to this method it is possible to successively calculate empirical models, which are based on a data set of a few days, see (Köller, 2002). We take true market prices for 267 EUR/USD call warrants for the period of one year. For the training of the ANNs the 255 trading days cuts into 250 several overlapping time intervals, each with a length of five days. In doing so the statistical evaluation and the comparison to the Black/Scholes-model have been carried out on the fifth trading day and on the following sixth trading day. The consideration of the respective sixth trading day ought to emphasize how far ANNs are able to approximate the future market prices. It turns out that the empirical model calculated better values for approximately 82% of the 250 considered respective fifth trading days with a lesser average deviation to the market prices as the deterministic Black/Scholes model.

4.3 Example: Optimization of a General Queuing System's Performance in Contact Centers at Banks, Loan Associations and Insurances

More and more often banks or insurances sell their products only online. In this case direct contact to customers is generally limited to the telephone. Carefully planned call or contact centers are necessary. Various simulations for inbound call centers show that ANNs can approximate the system performance, e. g., waiting times, accurately compared to an analytic solution, see (Köller and Breitner, 2005). This paragraph outlines the following step: ANNs are also applied to general queuing problems, for which no explicit solutions exist, see (Köller and Breitner, 2006). In real life almost all queuing problems are solved either with complex, discrete simulations. Or the problem is simplified so that it becomes analytically solvable, see (Bassamboo, Harrison, and Zeevi, 2006; Borst, Mandelbaum, and Reiman, 2004; Koole, 2004; Koole and van der Sluis, 2003).

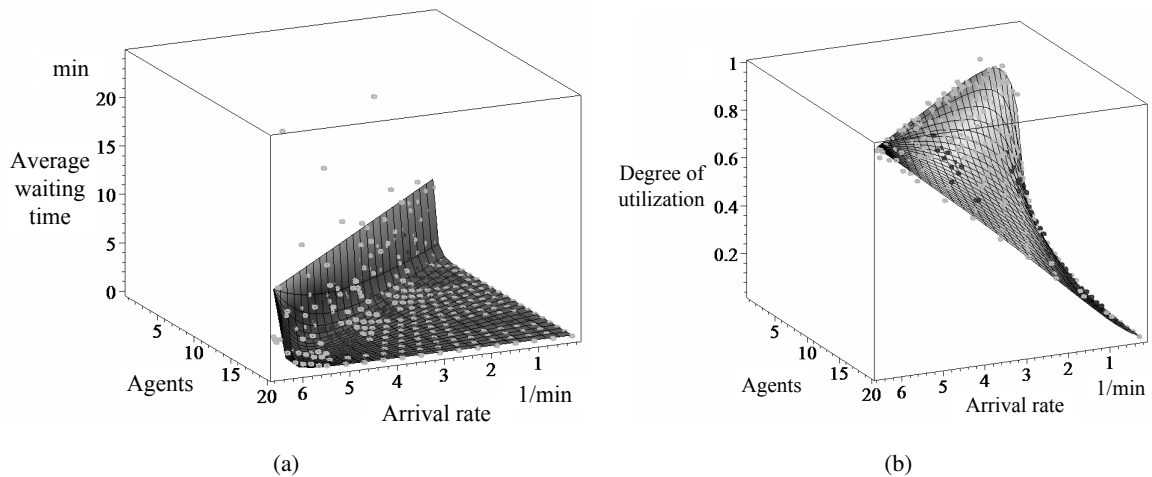


Figure 4: Average waiting time of customers in a queue (a) and corresponding staff utilization (b) in a contact/call center (ANN approximation). Implemented in a contact/call center IDSS these important characteristics help to supervise and optimize the IDSS performance permanently. (a) is interpreted as follows: The arrival rate is exogenous and determined by the number of customers calling at the moment. By ordering an appropriate number of agents it is possible to control the waiting time.

For an ANN training and approximation the structure of the problem does not have to be simplified. And only “few” simulation points, compared to standard techniques, need to be generated in contrast to an evaluation for the whole surface. Inevitable noise in the simulation data is smoothed by the continuous, approximated solution, i. e. the system performance is available quickly and accurately. The number of agents, e. g., can be optimized in real time.

Queuing problems can be solved by ANNs, using real data as basis for the training. In principle two ways are possible. On the one hand the distribution functions can be determined from the real data for the arrival process and for the different control processes depending on the appropriate business processes. With the resulting practical distributions more realistic training data can be simulated. On the other hand real data can be used directly as input for the training of the ANNs. The neurosimulator FAUN offers the possibility to generate approximated, explicit solutions of the characteristics for all queuing systems, see Figure 4. For difficult queuing problems advantages of an approximation compared to the analysis by discrete simulations are the followings. An analytical function for short-term manpower planning and cost minimising is generated. It is evaluable rapidly. The inevitable noise of the simulation data is smoothed.

5 INSIGHTS AND MANAGEMENT RECOMMENDATIONS

Can ANN live up to the high hopes that are put into them? Can they offer significant advantages to financial services providers, especially compared to established technical forecasting methods like linear, quadratic or cubic regression? — The answer is: It depends. It is probably not a good idea to use ANN as the sole source for making a decision. But ANN can support decisions based on other factors like fundamental or empirical analysis. Let’s revisit our examples 4.1 to 4.3.

In 4.1 we use ANNs and High Performance Computing to forecast *long term* interests. This works remarkably well as the underlying long term data are smooth and not noisy. It is, of course, necessary

to rebuild the model from time to time. This way the ANN can adapt to changing situations. This forecasting model is a real-life example and has been implemented for the Sparkasse Goslar, i. e. a German bank, see (Breitner, 2000a). However, we have to confess, that ANN are not especially well suited for *short term* forecasting. It is, for example, not lucrative to forecast short term currency rates. The data are too erratic.

Example 4.2 is of interest both for banks issuing options as well as for (professional) traders who want to compare options with seemingly similar properties. Here, ANNs offer business ratios that are more adequate than the traditional Black/Scholes model. Banks can use the results to issue options or warrants that are only *slightly* more advantageous than those of the competitors. They draw capital to their products without giving away too much margin. Investors or hedgers, on the other hand, can thoroughly compare the market and choose the objectively best option or warrant fitting their needs. This example is computed using real data, too.

At first glance, example 4.3 might seem thematically distant for financial services providers. However, contact and call centers are an important expense and also a major revenue source for banks and other institutions with many clients. The application of ANN allows to optimize the number of contact agents. Especially the expensive experts in the second level support are bothered less. The experts have more time left for administrative tasks. The present queueing model uses live data from an important German loan association and bank, see (Köller and Breitner, 2005). The model has been tested on-site with real-time data and produces accurate results.

Other real-life applications realized with FAUN include robust optimal reentry guidance for a space shuttle. This application is the result of a cooperation with the European aeronautic defence and space company EADS (BGT), see (Breitner, 2000b). Research on project capacity forecasting via function points in cooperation with IBM is on-going.

We also want to emphasize the potential of high performance and grid computing for neurosimulator applications. Our system offers the possibility to reuse existing computing capacities without impacting the performance. It is therefore possible to use FAUN without significantly investing in hardware. FAUN is designed as thin client solution with a web interface. Direct software and hardware support is provided via the FAUN project team consisting of the authors. A manual and examples can be provided, too.

6 CONCLUSIONS AND OUTLOOK

State-of-the-art neurosimulation, e. g., with the neurosimulator FAUN presented here, enables standard and new promising applications in today's business and management IDSS. In corporate finance, e. g., often crucial time series forecasts for market supply and demand, for interest rates and for exchange rates are necessary in real time. Market valuations of derivatives and assets often must be available in seconds or at most in a few minutes. In production and service delivery processes, e. g., expensive waiting queues must be avoided by farsighted planning with so-called synthesized production and service delivery functions.

In Sec. 2 a glance on modern ANN and neurosimulation, especially on supervised learning, is given. This includes software quality aspects, also mentioned in Sec. 1 shortly, and thin client architectures. These architectures — usually realized with web-frontends — enable to source out computationally expensive computations to dedicated compute servers. In Sec. 3 important high performance computing approaches, i. e. mainly parallel and grid computing approaches, are discussed. Decision problems and processes and the dedicated decision support models become more complex every day. An increasing number of companies and organizations need high performance computing. The three real life IDSS examples in Sec. 4 show and explain typical processes where IDSS based on ANN are already used

or can be used by financial services providers. As a research result we want to emphasize that the combination of ANN and high performance/grid computing offers valuable potential.

It has to be noted that ANNs also have some inherent limitations. If the relationship between input and output data is known a priori it is much better to exploit this knowledge than to use ANNs. It wouldn't make sense, e. g., to model obvious linear or quadratic relations with ANNs. For this task classical regression techniques are better suited. On the other hand ANNs are interesting for complex, unstructured problems. ANNs are no black-box and can't create miracles. If a basic understanding of the underlying problem isn't available, ANNs often will lead to poor results. FAUN especially has some training parameters which have to be tuned appropriately. This is easy for simple problems but difficult problems require some preliminary test runs.

The authors' research also focuses on various other decision problems for which ANNs are promising. Examples are estimation and management of a credit portfolio's risk, hedging of foreign currency cash-flow risks and robust optimal control of spacecraft, aircraft and missiles. The neurosimulator FAUN developed by the authors is permanently under reengineering to improve maintainability and portability. Among the features of FAUN the thin client architecture is especially important. It offers the possibility to use FAUN without local installation. Grid computing functionality enables low cost high performance computing by harvesting unused computation capacities on standard personal computer networks. The interested reader is invited to contact us for a FAUN test account on our compute server. A comprehensive user manual and ready to use examples are available. Formulation languages and standard interfaces for common AI and optimization problems are under construction. High performance computing always is a challenging task: New hardware, new software and new operating systems permanently cause FAUN maintenance and further development.

REFERENCES

- Abbas, A. (2004). *Grid Computing: A Practical Guide to Technology and Applications*. Charles River Media, Massachusetts.
- Aminian, F., E. D. Suarez, M. Aminian, and D. T. Walz (2006, August). Forecasting economic data with neural networks. *Computational Economics* 28(1), 71–88.
- Andreou, P. C., C. Charalambous, and S. H. Martzoukos (2006, May). Robust artificial neural networks for pricing of european options. *Computational Economics* 27(2-3), 329–351.
- Bassamboo, A., J. M. Harrison, and A. Zeevi (2006). Design and control of a large call center: Asymptotic analysis of an LP-based method. *Operations Research* 54(3), 419–435.
- Beck, K. (2002). *Test Driven Development. By Example*. Addison Wesley, Boston, Massachusetts.
- Borst, S., A. Mandelbaum, and M. I. Reiman (2004). Dimensioning large call centers. *Operations Research* 52(1), 17–34.
- Breitner, M. H. (2000a). Mittelfristige Prognose der deutschen Kapitalmarktzinskurve (in German). In H.-P. Beck (Ed.), *Brücken in die Zukunft – Forschung an der Technischen Universität Clausthal, Clausthal-Zellerfeld*, pp. 180–183. TU Clausthal.
- Breitner, M. H. (2000b). Robust optimal onboard reentry guidance of a space shuttle: Dynamic game approach and guidance synthesis via neural networks. *Journal of Optimization Theory and Applications* 107, 484–505.
- Breitner, M. H. (2003). *Nichtlineare, multivariate Approximation mit Perzeptrons und anderen Funktionen auf verschiedenen Hochleistungsrechnern (in German)*. Akademische Verlagsgesellschaft, Berlin.
- Chen, H.-C. J. and M. Magdon-Ismail (2006). NN-OPT: Neural network for option pricing using multinomial tree. In *Neural Information Processing*, Volume 4234, pp. 360–369. Springer Berlin / Heidelberg.

- Deuffhard, P. (2004). *Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms*. Springer, Berlin.
- Fletcher, R. (2000). *Practical Methods of Optimization*. John Wiley & Sons, New York.
- Fowler, M. (1999). *Refactoring. Improving the Design of Existing Code*. Addison-Wesley, Massachusetts.
- Gill, P. E., W. Murray, and M. H. Wright (2004). *Practical optimization*. Academic Press, London.
- Gupta, J. N. D. (2006). *Intelligent Decision-making Support Systems: Foundations, Applications and Challenges*. Springer, London.
- Koole, G. (2004). Performance analysis and optimization in customer contact centers. In *Proceedings of QEST 04, IEEE, Enschede*, pp. 2–5.
- Koole, G. and E. van der Sluis (2003). Optimal shift scheduling with a global service level constraint. *IIE 35*, 1049–1055.
- Köller, F. (2002). Statistische Analyse und Visualisierung der Genauigkeit empirischer und deterministischer Optionspreismodelle am Beispiel von Devisenoptionen (in German). Diploma thesis, Technische Universität Clausthal, Institut für Mathematik.
- Köller, F. (2007). *Optimale Echtzeit-Personaleinsatzplanung für Inbound Call Center durch Approximation von Warteschlangenkennzahlen mit Künstlichen Neuronalen Netzen (in preparation, in German)*. Ph. D. thesis, Leibniz Universität Hannover.
- Köller, F. and M. H. Breitner (2005). Optimierung von Warteschlangensystemen in Call Centern auf Basis von Kennzahlenapproximationen (in German). In H.-O. Günther (Ed.), *Supply Chain Management und Logistik*, pp. 459–483. Physica/Springer, Heidelberg.
- Köller, F. and M. H. Breitner (2006). Optimierung von Warteschlangensystemen durch Approximation mit Neuronalen Netzen (in German). In H.-D. Haasis, H. Kopfer, and J. Schönberger (Eds.), *Operations Research Proceedings 2005, Bremen*, pp. 263–268. Springer, Heidelberg.
- Matsatsinis, N. F. and Y. Siskos (2003). *Intelligent Support Systems for Marketing Decisions*. Kluwer Academic, Boston.
- Mehmert, P. (2000). Mittelfristige Zinsprognosen basierend auf technischen Ansätzen mit parallel trainierten Perzeptrons durch FAUN 0.2-PVM (in German). Diploma thesis, Technische Universität Clausthal.
- Mettenheim, H.-J. v. and M. H. Breitner (2005). Neural network forecasting with high performance computers. In E. P. Hofer and E. Reithmeier (Eds.), *Proceedings of the Thirteenth International Workshop on Dynamics and Control*, pp. 33–40. Shaker, Aachen.
- Nowak, U. and L. Weimann (1998). A family of newton codes for systems of highly nonlinear equations. algorithm, implementation. Technical Report TR 91-10, Konrad Zuse Zentrum, Berlin.
- Oh, K. J., T. Y. Kim, H. Y. Lee, and H. Lee (2005). Using neural networks to support early warning system for financial crisis forecasting. In *AI 2005: Advances in Artificial Intelligence*, Volume 3809/2005, pp. 284–296. Springer Berlin / Heidelberg.
- Priddy, K. L. and P. E. Keller (2005). *Artificial Neural Networks: An Introduction*. SPIE Press, Bellingham, Washington.
- Turban, E., J. E. Aronson, and T. P. Liang (2005). *Decision Support Systems and Intelligent Systems*. Pearson Prentice Hall, Upper Saddle River.
- Tzastoudis, V. S., N. S. Thomaidis, and G. D. Dounias (2006). Improving neural network based option price forecasting. In *Advances in Artificial Intelligence*, Volume 3955/2006, pp. 378–388. Springer Berlin / Heidelberg.
- Wang, J. (2005). Advances in neural networks. In *Proceedings of the Second International Symposium on Neural Networks, Chongqing, China*. Springer, Berlin.
- Zell, A. (2003). *Simulation Neuronaler Netze (in German)*. Oldenbourg, München.