مجله علوم اجتماعی و انسانی دانشگاه شیراز
دوره نوزدهم، شماره اول، زمستان ۱۳۸۱ (پیاپی ۳۷)

استفاده از شبکهٔ عصبی هوشمند در تفسیر نتایج مدل سازی سیستم اطلاع رسانی کتابخانهٔ منطقه ای علوم و تکنولوژی شیراز

سارا کلینی **دکتر اقبالی جهرمی** **دکتر جعفر مهراد**
دانشگاه شیراز

چکیده

در این مقاله، روش تفسیر نتایج شبیه سازی سیستم های گسسته با استفاده از شبکه عصبی با یادگیری نظارت شده به عنوان تکنیک یادگیزی ماشین مورد مطالعه قرار گرفته است.

برای تحقق این مهم، مجموعه‌ای از داده‌های واقعی متشکل از زیر مجموعه نمونه گیری از سیستم اطلاع رسانی کتابخانهٔ منطقه ای علوم و تکنولوژی شیراز جمع آوری گردید. سیستم اطلاع رسانی کتابخانه منطقه ای با استفاده از اطلاعات گردآوری شده توسط برنامه ای که در محیط GPSS/H نوشته شد، شبیه سازی گردید و نتایج آن به عنوان ورودی برای تعلیم شبکه عصبی پس – انتشار خطا استفاده شد. این بررسی نشان می‌دهد که الگوی شبکه عصبی که به عنوان الگوریتم یادگیری ماشین به کار گرفته شده است دارای دقتی است که با شبیه سازی انجام شده توسط GPSS/H قابل مقایسه می باشد و با تعلیم گرفتن شبکه، سیستم می تواند در زمان واقعی عملیات خود را به انجام برساند. کارآیی سیستم شبکه عصبی در مورد تخمین زمان سرویس دهی به هر کاربر در سیستم اطلاع رسانی با کارآیی سیستم شبیه ساز مقایسه گردید و نشان داده شد که نتایج مطلوب است. شبکه عصبی پس- انتشار خطا در زمینه بررسی نتیجه شبیه سازی دارای قابلیت های خوبی است که می توان از آن بهینه سازی و پیش بینی سیستم‌های- سیستم اطلاع رسانی کتابخانه استفاده نمود.

*JOURNAL OF SOCIAL SCIENCES & HUMANITIES
OF SHIRAZ UNIVERSITY*
VOL.19 . NO. 1, Winter 2003 (Ser. 37)

Using Artificial Neural Network for Interpreting the Results of RLST Information Center's Discrete Event System Simulation

S. Koleinee* **Dr. H.J. Eghbali**** **Dr. J. Mehrad*****
Shiraz University

ABSTRACT

This paper describes an approach to interpretation of discrete event simulation using supervised neural network. A data set of ten replicas were generated from the information center of RLST (Regional Library of Science and Technology). Data was first processed with GPSS/H simulation package. The simulation results were used as the input to a neural network. A backpropagation neural network was used to train the results of the system. It is shown that the neural network metamodel is quite competitive in accuracy when compared to the simulation by itself and can operate in nearly real time. The performance of the neural network metamodel is compared to the simulation performance for estimating the user mean time in information center. The overall results can be used to optimize the performance of library information center. The BBS (Bulletin Board Services) was also simulated and trained.

Key words: 1.Discrete event simulation 2.GPSS 3.Artificial neural network 4.Library 5.Information center 6.BBS 7.RLST.

1. Introduction

A simulation is the operation of a real-world system and the behavior of the system over time. Whether done by hand or on a computer, simulation involves the generation of an artificial history of a system, and the observation of that artificial history to draw inferences concerning the operating characteristics of the real system.

Any number of definitions can be used from a variety of distinguished texts. Overall system simulation is defined as "The use of a mathematical/logical model as an experimental vehicle to answer questions about a given system and accept the correct result by using a proper Learning Machine."

The behavior of a system, as it passes over time, is studied by developing a simulation model. This model usually takes the form of a set of assumptions concerning the operation of the system. These assumptions are expressed in mathematical, logical, and symbolic relationships between the entities, or objects of interest, of the system. Once developed and validated, a model can be used to investigate a wide variety of "what if" questions about the real-world system. Potential changes to the system can first be simulated in order to predict their impact on system performance. Simulation can also be used to study systems in the design stage, before such systems are built. Thus, simulation modeling can be used both as an analysis tool for predicting the effect

---

* Rlst's Depct. Of Planning and Programming, Director
** Associate Professor of Computer Engineering
*** Professor of Library Sciences and Information

of changes to existing systems, and as a design tool to predict the performance of new systems under varying sets of circumstances.(Banks, et.al, 1996).

In some instances, a model can be developed which is simple enough to be solved by mathematical methods. Such solutions may be found by the use of differential calculus, probability theory, algebraic methods, or other mathematical techniques. The solution usually consists of one or more numerical parameters, which are called the system performance measures. However. many real-world systems are so complex that models of these systems are virtually impossible to solve mathematically. In these instances, numerical, computer-based simulation can be used to show the behavior of the system over time. From the simulation, data are collected as if a real system were being observed. This simulation-generated data is used to estimate the system performance measures.

Discrete-event system simulation is the modeling of systems in which the state variable changes only at a discrete set of points in time. Essentially, a simulation is the basis for making some decision. This decision is based on the "answers" provided by the simulation.(Kleijnen, 1993).

The simulation provides an assessment of some system which is not readily amenable to other types of analysis; thus the simulation provides the only means by which to assess a given situation. Arriving at the correct decision is the singular overriding objective of simulation, where in this work this concept is considered. One may want a simulation to provide a variety of behaviors and possess a multitude of characteristics, but none of these can be achieved at the expense of a correct decision.

Knowledge acquisition, for an understanding of discrete event simulation systems is a difficult task. Machine Learning by the use of neural network is used to help in the knowledge acquisition process.

Takefuji (1993) used artificial neural networks to construct empirical metamodel of computer simulations. The problem is how to perform basic simulation tasks, such as prediction and comparison of alternatives, with neural approximations from the output results of simulation. Obtaining the training and testing pairs for a stochastic computer simulation is computationally costly, thus motivating the need to form the most accurate neural network model from a limited data set. In both new training approaches, learning is iterative. A subset of the training data is obtained from simulation output as the input to a neural network for training so that getting the optimum result. The feedforward multilayer perception network trained by backpropagation is the type of neural network which is more popular in simulation.

2. Background

Flood (1996) evaluated a neural network approach for modeling the dynamics of construction processes that exhibit both discrete and stochastic behavior, providing an alternative to the more conventional method of discrete-event simulation.

Fishwick (1999) described that traditional computer simulation terminology includes taxonomic divisions with terms such as "discrete event," "continuous," and "process oriented." Even thou such terms have become familiar to simulation researchers, the terminology is distinct from other disciplines [such as artificial intelligence and software engineering] which have similar goals relating specifically to modeling dynamic systems. There is a need to unify terminology among these disciplines so that system modeling is formalized in a common framework.

Assistant (Cestnik,et.al,1987) is an attribute-value learning algorithm for the construction of binary classification trees. It belongs to the Top Down Induction of Decision Trees family of algorithms. Reties is an attribute-value learning algorithm for the construction of regression trees. Regression trees are similar to binary classification trees constructed by the Assistant algorithm. The most important difference is that, while classification trees are used to classify objects into discrete classes, regression trees are used when the class is continuous. A regression tree actually implements a function $y(x_1, x_2, \dots, x_n)$ of n continuous or discrete attributes $x_1, x_2, \dots, x_n$. Both algorithms incorporate several noise handling mechanisms that are used in their experiments.

Markus is an Inductive Logic Programming (ILP) system designed for inferring Prolog programs from examples and counterexamples of their behavior. The Markus algorithm has its basis in Shapiro's model inference system [7], but extends it in several ways. A Prolog program introduced by Markus uses background-knowledge predicates defined by the user. Used predicates can be very simple such as the "greater-than" (>) predicate or complex procedures written in Prolog.

Kilmer and Smith[8] described that a computer simulation model may be regarded as a stochastic function that maps a set of inputs to a set of outputs. In many cases, computer simulation models are quite computationally expensive. It would be beneficial to have fast, accurate approximations of computer simulation models to perform such tasks as quick turn-around studies, sensitivity analyses, model aggregation/reduction, and simulation optimization. They examined the use of two methods, artificial neural networks (ANN) and multiple linear regression, for approximating a lot size-reorder point inventory system simulation.

3. Procedure Review

Discrete event system simulation of RLST's Information Center (IC) together with its BBS and a metamodel of that simulation is developed. The metamodeling technique used is an artificial neural network, which is trained using the simulation result. The overall procedure used in this work is illustrated in figure1.

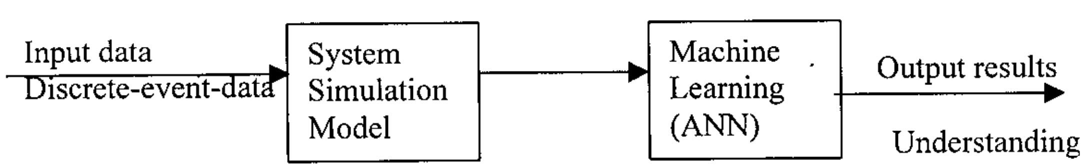

Figure 1. Overall Procedure

3.1. Overview of the RLST's Information Center (IC) Simulation

3.1.1 The RLST Environment: RLST has over 45 computers and two CD-Towers. Users utilize RLST's services by one of the following ways:

I. Attending at RLST Personally;

II. Using remote distance services by completing the search forms;

III. Using dial up network through RLST's BBS.

All request of the users in part A and B are distributed among four servers. The servers may utilize a CD-Tower through the RLST's Local Area Network. When a user arrives, she/he enters in a queue. The user will receive prompt response if the server is idle, otherwise she/he must wait. The BBS users connect to RLST's through one of six phone lines. There is an intermediate machine between users and BBS servers. This center provides information services for more than 400 users every month. To consider possible changes to operating policies and procedures without disrupting the user services, a system simulation model is constructed. An important output variable of interest is the mean service time for each user.

3.1.2. The Simulation Structure: In this work, the RLST's Information Center (IC) and its BBS are modeled by GPSS/H package that runs on a PC. This model contains all of the main structures that could be used in describing or presenting the actual system. User related services and location in the RLST models are represented in separate sections called servers and lines. The servers include computers that are used for servicing in the part A or B. Lines include phone lines that are available for servicing via BBS. The model is shown in figure2.

Service time of servers and lines are modeled with Poisson distribution. IC simulation begins from 8:00 AM until 4:00 PM there is one hour break from 12:00 PM -13:00 PM. A form was used to collect information and data from users attending RLST.

There might be some waiting time for users when they arrive until they receive a response. The service time is defined as the time between search ending time and search beginning time. Forms filled out by operators in group B (remote distance) is the same as form A except that the sending date is added to this form to describe the waiting time. For group C (BBS users) the connection time, and service times are obtained from a log file which is generated by BBS software.

print out receiving for users in group A and mailing (or faxing) search results in group B determine the search ending time. Line disconnection time determines ending time for users in group C. Sampling was done in ten days to produce ten replicas. Building the simulation model in this modular manner allows the simulation to be more easily verified and validated and, if necessary, to be modified or expanded. The experiment frame contains data distribution such as arrival time; service time and the user flow pattern as well as the construct for obtaining output date from the simulation. Since data related changes are made only to the experiment phase, the system manager can perform different simulation experiment without actually changing the structure of the system as represented in the model. The simulation program runs for ten replicas. The expected value of each replica was used as the input of the simulation program. The comparison between real data and simulation results was found to be comparable. The utilization diagrams for IC and BBS are shown in figure 3 and figure 4.

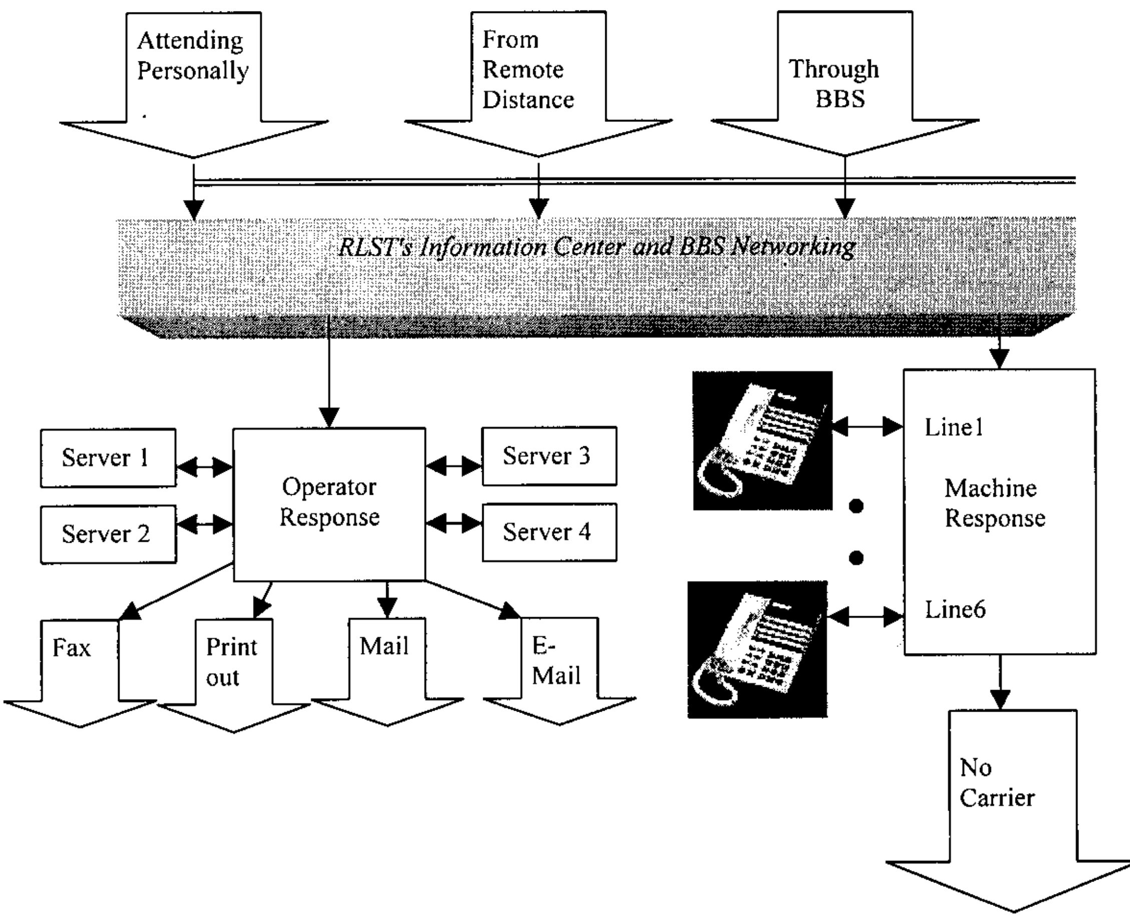

Figure 2. User Flow Through the RLST's Information Center Simulation.

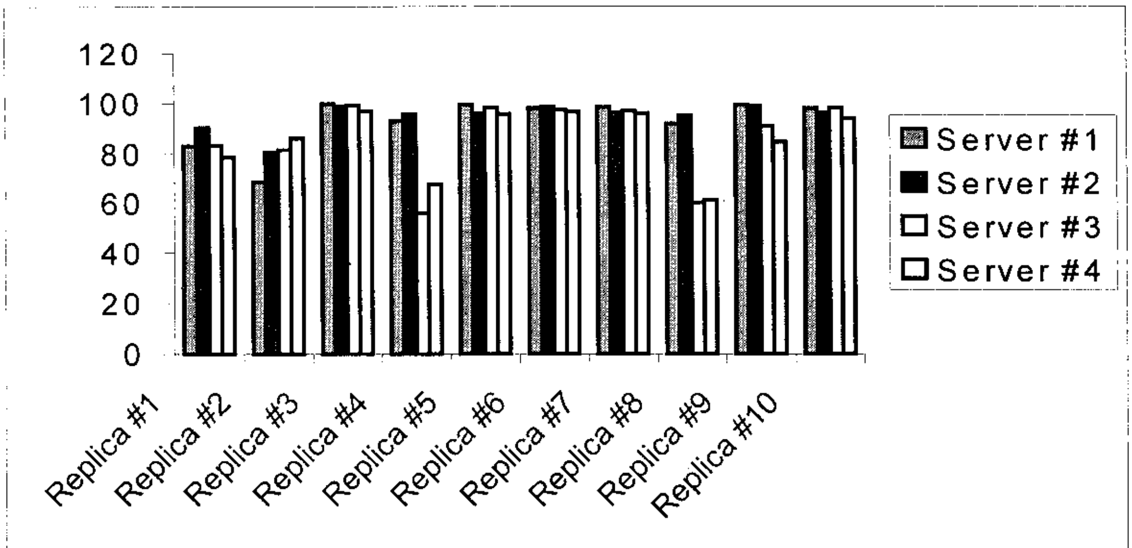

Figure 3: IC Utilization Diagram for 10 Replicas

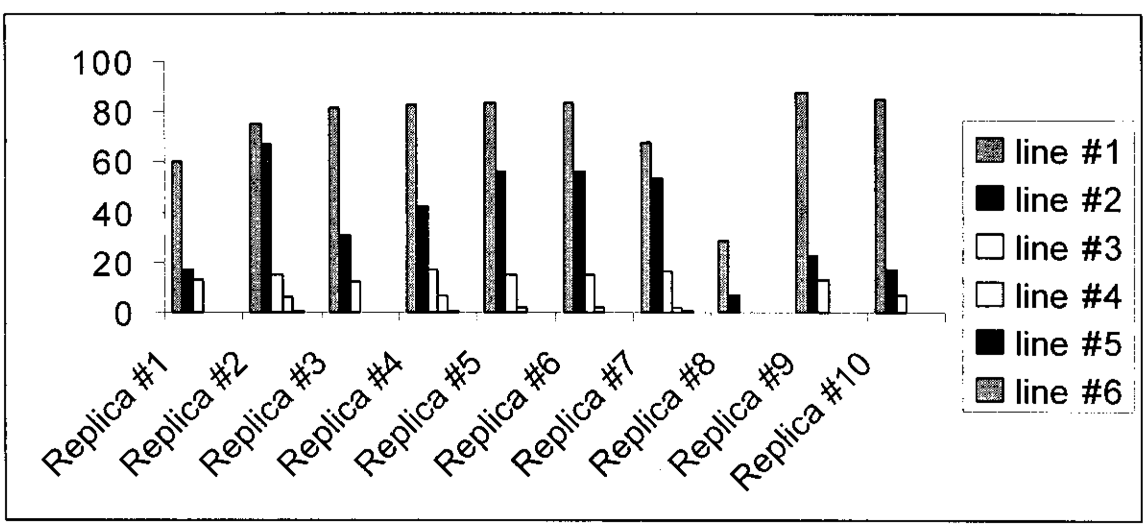

Figure 4. BBS Utilization Diagram for 10 Replicas

4. Development of the Simulation Metamodel

To reduce the long running times necessary to explore and optimize a simulated scenario, a metamodel was developed. This is an especially important step for an environment such as Information Center, where the stochastic nature causes queues to build up frequently and with little warning,.and decisions need to be made quickly to relieve the resultant congestion. If a metamodel were available, it could then be used as a "real-time" decision aid to determine the best alternative to resolve the problem. There are various approaches to metamodeling of simulations, but most are based on simplifying algorithmic or functional assumptions, such as polynomial regressions. Another disadvantage of traditional metamodeling techniques is that they are often limited to a subset of the simulation domain, and must be redeveloped or discarded when exploring other ranges. Neural networks offer universal function approximation capability based wholly on the data itself, i.e. they are purely empirical models which can theoretically mimic any relation to any degree of precision. The neural network metamodels for the IC and BBS system were developed. These networks predict expected time in the respective system. The IC's neural network input variables were the means of exponential random variables representing server#1 to server#4 expected time. Four continuous input neuron were used, one for each input variable. There was one output variable of interest - expected time in IC for user. This was represented by one continuous output neuron. Figure 5 shows the architecture of this model for IC. There are two hidden layers with two neuron in each layer. All neural network simulation in this work was implemented with neural network toolbox of MATLAB software. Fewer hidden layer nodes and number of neuron in each layer caused insufficient accuracy and more hidden layer or more neuron in each layer caused a little improvement but mean while caused the slowing training time, therefore the values of the number of hidden neurons were identified by experiment. The network was trained using a fast backpropagation algorithm, with using a smoothing activation function. The network which achieved the minimum mean square error over the entire training set was kept as the final trained network. Sum squared error goal was set to 0.02 and the value of threshold was set to be less than mean squared error. The value of momentum set to 0.2. Initial weights were random numbers between -0.1 to +0.1. Figure 6 shows the accuracy of the neural network training run and the changes of learning rate as a function of iteration number (Epoch). Figure 7 and figure 8 are illustrated for the BBS with two hidden layers and four neuron in each layer, other parameters are the same as IC network.

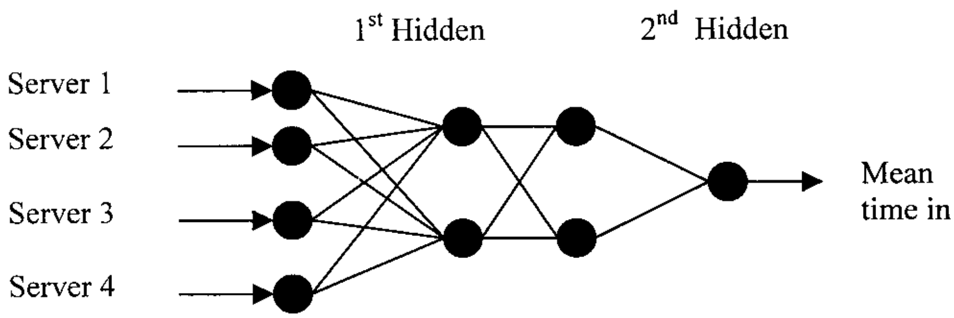

Figure 5. Architecture of IC's Neural Network.

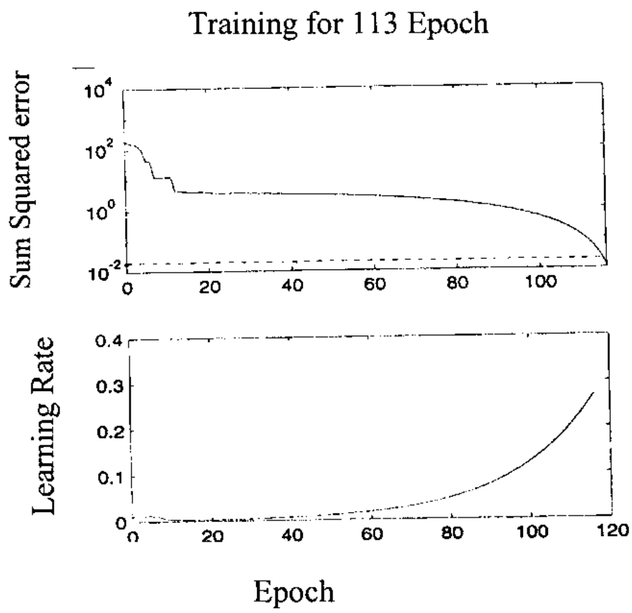

Figure 6. Accuracy of the Neural Network Training Run and the Changes of Learning Rate as Epoch.

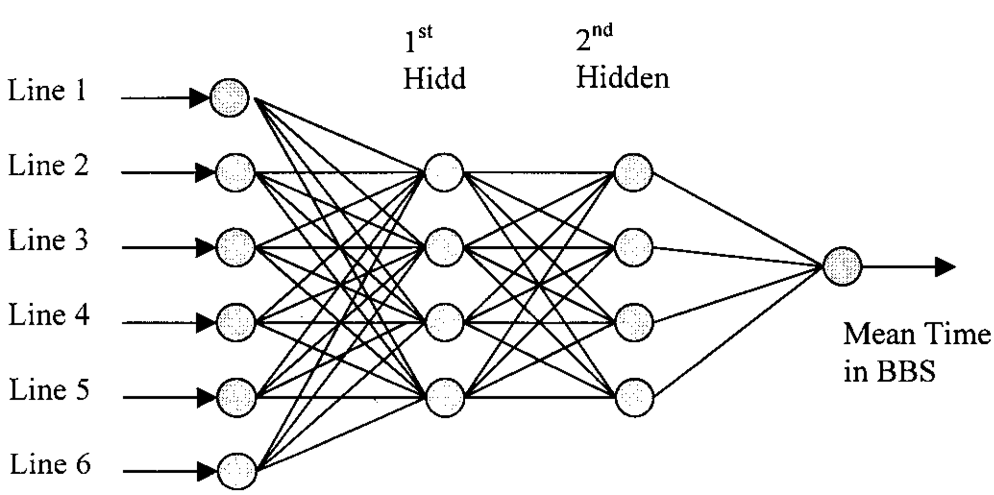

Figure 7. Architecture of BBS's Neural Network.

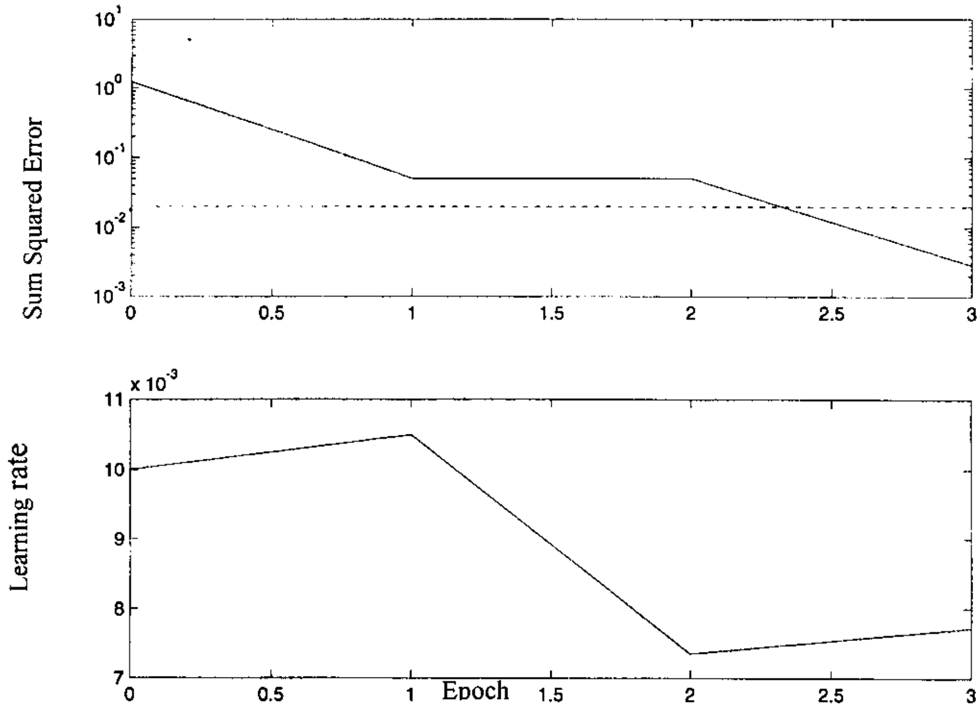

Figure 8. Accuracy of the Neural Network Training Run and the Changes of Learning Rate as Epoch

5. Conclusion

This work has shown the real world application of RLST simulation and its neural network based on metamodel. There is limitation of the metamodel. It is valid only for the specified parameter domains included in the training set. The number of replications for which the estimate is valid is fixed at ten. The metamodel, as developed, cannot reflect the abnormality of the output distribution. Similarly, the metamodel is completely deterministic so the stochastic variability of the simulation is output lost. For most day to day decision in the RLST, the functionality required of the simulation is to estimate the mean value of the output variable. The neural network does not need to remain static. It can be updated through additional training as more simulation replications become available. The neural network metamodel could also be updated through direct observation of the system, if that were possible. Both of these additional training methods could be applicable to the RLST. More simulation may be run as computational resources and time allow, and the system may be observed directly through special study or daily records. However, changes to the simulation (e.g. distribution parameters, additional or subtraction of variables) would invalidate the neural network metamodel, and a new metamodel reflecting the altered simulation would need to be developed.

The metamodeling of the RLST's IC was satisfactory as shown in figure 9 and figure 10 when simulation data was compared with neural network results. The discrepancy in some of the replicas was noted to be due to the abnormality observed in that replica and having higher variances compared to other replicas. The metamodel of the BBS was shown to be similar to the IC except low number of calls causes the simulation model do not to represent the real world correctly. Since the neural network model is built on the simulation results, the neural network model did not match with simulation as well. In conclusion the BBS system can not be modeled correctly due to limited number of calls from users and stochastically has high variance.

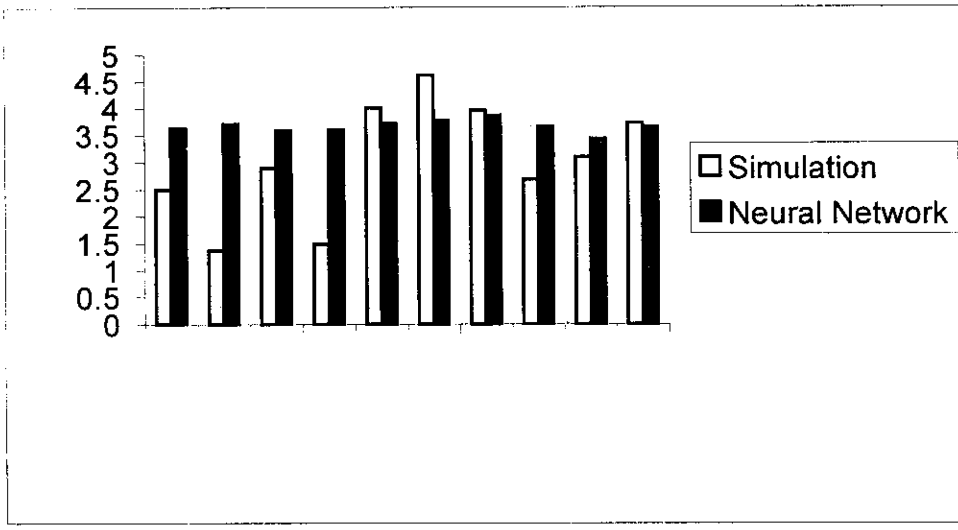

Figure 9. Simulation and Neural Network Comparison for Ten Replication of IC Time

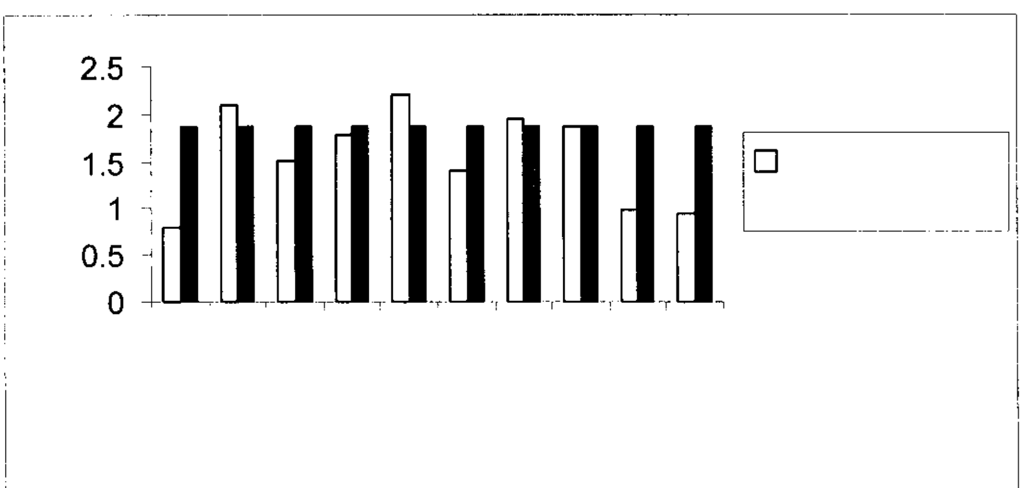

Figure 10. Simulation and Neural Network Comparison for Ten Replication of BBS Time.

References

Banks, J., Carson, J., and Nelson, B.L.(1996). **Discrete - event System Simulation,** London: Prentice Hall.

Cestnik, B., Kononenk, I., and Bratko, I. (1987). **ASSISTANT 86 : A Knowledge – Elicitation Tool for Sophisticated Users in Progress in Machine Learning,** Wilmslow: Sigma Press.

Fishwick, P.A. (1992). **An Integrated Approach to System Modeling Using a Synthesis of Artificial Intelligence Software Engineering,** http://www.cora.justresearch.com.

Flood, I.F. (1996). *Modeling Construction Processes Using Artificial Neural Networks,* **Automation in Construction,** 4(4), 307-320.

Kilmer, R.A., and Smith, A.E. (1996). **Using Artificial Neural Networks to Approximate a Discrete Event Stochastic Simulation Model,** http://www.csindex.com.

Kleijnen, J. P. C. (1993). *Simulation and Optimization in Production Planning,* **Decision Support Systems**, 9, 269 – 280.

Shapiro, E.Y. (1983). **Algorithmic Program Debugging**, Cambridge, MA : MIT Press.

Takefuji, Y. (1993). **Neural Network Parallel Computing,** Kluwer Academic Publisher, Boston: ISBN, 256