PREDICTING LEARNERS PERFORMANCE USING ARTIFICIAL NEURAL NETWORKS IN LINEAR PROGRAMMING INTELLIGENT TUTORING SYSTEM

Samy S. Abu Naser

Faculty of Engineering and Information technology, Al-Azhar University-Gaza,
Palestine.

samy@abunasser.com

ABSTRACT

In this paper we present a technique that employ Artificial Neural Networks and expert systems to obtain knowledge for the learner model in the Linear Programming Intelligent Tutoring System(LP-ITS) to be able to determine the academic performance level of the learners in order to offer him/her the proper difficulty level of linear programming problems to solve. LP-ITS uses Feed forward Back-propagation algorithm to be trained with a group of learners data to predict their academic performance. Furthermore, LP-ITS uses an Expert System to decide the proper difficulty level that is suitable with the predicted academic performance of the learner. Several tests have been carried out to examine adherence to real time data. The accuracy of predicting the performance of the learners is very high and thus states that the Artificial Neural Network is skilled enough to make suitable predictions.

KEYWORDS

Linear Programming, Intelligent Tutoring System, backprobagation, Artificial Neural Network

1. INTRODUCTION

Intelligent Tutoring Systems (ITS) are computer based educational systems which are designed to increase the learning rate and retention of learners. Recent tutoring systems of this type use software algorithms that adapt to the learner's anticipated knowledge of the material to deliver tailored instruction. With these adaptations the tutoring system is able to more efficiently direct the learner through solving problems[1,2,3].

There has been little research in the field of learner models that can precisely predict the learner performance to inspect themselves.

Liner Programming Intelligent Tutoring system was trained by Neural Network using backpropagation algorithm for predicting the learner academic performance from the log file stored in the LP-ITS.

An expert system is used for selecting the proper linear programming problems using the output of the Artificial Neural Networks is presented. Neural network and expert systems were examined for their capabilities in making prediction in a simple setting. This paper discusses the outcome of this work.

DOI : 10.5121/ijaia.2012.3206

Problem difficulty level in linear programming has been used to determine the knowledge a learner has acquired during his/her learning with LP-ITS, but there are situations when being presented with increasingly challenging problems does not aid the learning process. The learner may need to be presented with easier or similar difficulty problems to boost his/her confidence and make their experience more enjoyable.

The reminder of this paper is organized as follows: Background information about LP-ITS, expert systems and artificial neural network is presented in section 2, Design of the NN is presented in section 3, Testing and training the neural network is outlined in section 4, and the accuracy of prediction of learner performance is discussed in section 5. Section 6 presents the conclusion.

LITERATURE REVIEW

Kanakana and Olanrewaju used Artificial Neural Network and linear regression models to predict student performance after access to higher education. Data received from the Tshwane University of Technology was utilized for the study. The total Average Point Scores (APS) students obtained in grade 12 was employed as input variable. The results indicated a better agreement between ANN model prediction and observed values compared to those in the linear regression[14].

Kyndt et. al. in their study predicted general academic performance in the first bachelor year educational sciences, based on students' motivation, approaches to learning, working memory capacity and attention using a neural network analysis. Participants in this study were 128 university students. Results showed that working memory capacity and attention are both good predictors of academic performance, especially for the best and weakest performers of the group. Students' motivation and approaches to learning were good predictors for the group of students whose performance was in the middle 60% [15].

Mukta and Usha carried out an analysis to predict the academic performance of business school graduates using neural networks and traditional statistical techniques and the results were compared to evaluate the performance of these techniques. The underlying constructs in a traditional business school curriculum were also identified and its relevance with the various elements of admission process were presented[16].

Croy et, al. used data for student in discrete math course at North Carolina State University in order to understand student behaviour. This data was extracted from engineering and computer science students in four semesters during 2003 – 2006. As an average 223 students were included in the study. They found that 90% of all student errors related to explaining their actions, and a great majority of these were on the simplification rule. As a conclusion visualizations can be useful in learning about other problem-solving processes from student data, instead of creating expert systems by hand [17].

2. BACKGROUND OF LP-ITS, EXPERT SYSTEMS AND ARTIFICIAL NEURAL NETWORKS

2.1 LP-ITS

LP-ITS has been developed to students enrolled in Operations Research in the Faculty of Engineering and Information Technology at Al-Azhar University in Gaza, Palestine. LP-ITS gradually introduces students to the concept of Linear Programming and automatically generates problems for the students to solve[1].

In the design of the Linear programming Intelligent System, we have used the traditional modules of an ITS: Pedagogical Module Design, Expert module, Learner module, Problems generator module, and Tutoring process module (See Figure 1) [1,2,3].

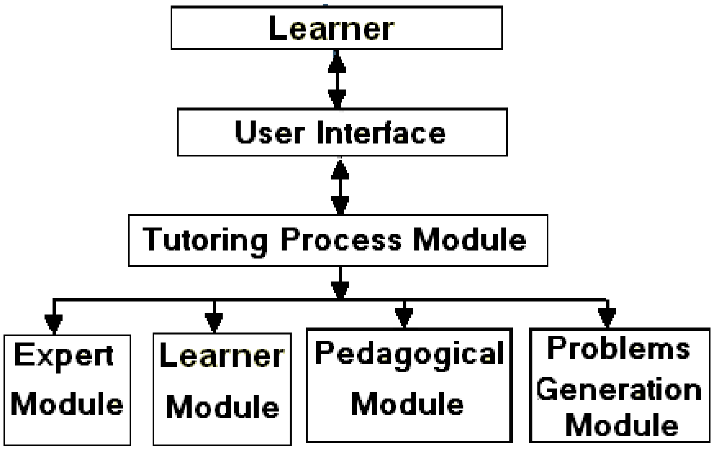

Figure 1: Architecture of LP-ITS

The learner module is the most critical module because it stores information of the learner's current state of knowledge that are used in predicting academic performance of the learner.

2.2 Expert Systems

An expert system is computer software that attempts to act like a human expert on a particular subject area such as production, classification, accounting, medical diagnosis, and identification. The expert module contains a an expert system to identify the proper difficulty level for learner according to the prediction of the neural network[4,5].

2.3 Artificial Neural Networks

An Artificial Neural Network (ANN) is a arithmetical model that is motivated by the organization and/or functional feature of biological neural networks. A neural network contains an interrelated set of artificial neurons, and it processes information using a connectionist form to computation. As a general rule an ANN is an adaptive system that adjusts its structure based on external or internal information that runs through the network during the learning process. Recent neural networks are non-linear numerical data modeling tools. They are usually used to model intricate relationships among inputs and outputs or to uncover patterns in data[6,8].

ANN has been applied in numerous applications with considerable attainment[6,7,13]. For example, ANN have been effectively applied in the area of prediction [8], handwritten character recognition[9,10], evaluating prices of lodging [11], disease categorization[12].

3. DESIGN OF THE NN

3.1 Log File Data

Data log file of the problem domain is required by the network to learn the task. From the data in the log file we must determine what is necessary and what is not, since log file can be huge and difficult to analyze, for the training data set. This can be difficult to determine. A good approach is to include everything that could be useful from the log file, build and test the network with all the possible data and remove what is not necessary (See figure 2).

```
...
Stat Date : 18/12/2011
Stat Time : 11:33
Pre Process
Help Level : 2
Feedback Option: Partial Solution
Type of Problem: Maximize
Problem number: 21
No of  attempt: 2
Difficulty Level : 3
Topic :  Big M
Special Case of LP : Infeasibility
Sensitivity Analysis:  Changes in Objective Function Coefficient
Expertise: 3

End Date : 18/12/2011
End Time : 11:43
Post Process
User  Result : 0.91

User logged out
……
```

Figure 2: An example of a problem submission within a learner log file

The data available to us for the investigation into problem Generation consisted from log files of previous evaluations of LP-ITS Tutor in 2011. We had log files for 67 learners that have worked with LP-ITS Tutor. 13 of the log files were invalid as there were no problems solved by the learner and no data could be extracted from the log file. There were 14 cases were the learners logged themselves onto LP-ITS Tutor and logged out immediately, thus not completing any problems.

Training files were generated from these learner logs (Figure2). The following information was extracted:

☒ Problem number, the identification of the problem, ranging from 1 to m.

☒ Problem Difficulty Level, which shows the problem difficulty. Ranging from 1 (easiest) to 6 (most difficult).

☒ Student Expertise Level, which is the current level that the student has achieve. The student level ranges from 1 (novice) to 6 (experienced).

☒ Problem attempt, whether the student has attempted the problem before. The attempt is equal to zero if the problem is new and one if they have seen the problem before.

☒ Time spent solving the problem, When the solution is correct how much time was spent for solving the problem. It ranges : (1 – 5 min, 6 – 10 min, 11-15 min, 16-20 min)

☒ Help Level Provided, which ranges from 0 (no Help provided) to 6 (the solution was given to the solution)

☒ Number of errors the student made in the last attempt on the current problem.

3.2 Training Data Set

There were 67 log files from a previous evaluation of LP-ITS Tutor. 13 of those learner logs were invalid because there were no problem submissions. The learner logged onto the system and did not complete any tasks, logged off and did not return. These files were omitted. For training purposes we trained the networks with 25 students and tested the networks with the remaining 25. This split was arbitrarily based on the size of the training and testing set. The training set had 1120 submissions, and the testing set had 1024.

3.3 Problem Difficulty Predictor

We needed to predict whether the learner will have difficulties with the problem at hand. That means we predict the actual number of errors a learner is going to make. The more the errors is the more difficulty the learner is having. A suitable threshold of error per problem is chosen to be 4.

3.4 Network Architecture

The network is a multilayer perceptron neural network using the linear sigmoid activation function.

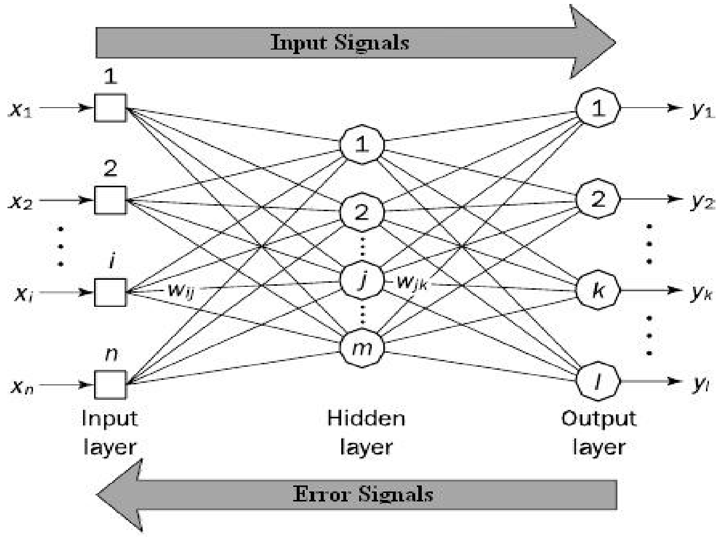

Figure 3. NN System Architecture

3.5 The back-propagation training algorithm

Step 1: Initialization

All the weights and threshold levels of the network are random numbers uniformly distributed inside a small range:

$$\left(-\frac{2.4}{F_i},\ +\frac{2.4}{F_i}\right)$$

where Fi is the total number of inputs of neuron i in the network. The weight initialisation is done on a neuron-by-neuron basis.

Step 2: Activation

Activate the back-propagation neural network by applying inputs $x_1(p)$, $x_2(p)$,…, $x_n(p)$ and desired outputs $y_{d,1}(p)$, $y_{d,2}(p)$,…, $y_{d,n}(p)$.

(a) Calculate the actual outputs of the neurons in the hidden layer:

$$y_j(p) = sigmoid\left[\sum_{i=1}^{n} x_i(p) \cdot w_{ij}(p) - \theta_j\right]$$

Where *n* is the number of inputs of neuron *j* in the hidden layer, and *sigmoid* is the *sigmoid* activation function.

(*b*) Calculate the actual outputs of the neurons in the output layer:

$$y_k(p) = sigmoid\left[\sum_{j=1}^{m} x_{jk}(p) \cdot w_{jk}(p) - \theta_k\right]$$

where *m* is the number of inputs of neuron *k* in the output layer.

Step 3: Weight training

Update the weights in the back-propagation network propagating backward the errors associated with output neurons.

(a) Calculate the error gradient for the neurons in the output layer:

$$\delta_k(p) = y_k(p) \cdot [1 - y_k(p)] \cdot e_k(p)$$

Where $e_k(p) = y_{d,k}(p) - y_k(p)$

Calculate the weight corrections:

$$\Delta w_{jk}(p) = \alpha \cdot y_j(p) \cdot \delta_k(p)$$

Update the weights at the output neurons:

$$w_{jk}(p+1) = w_{jk}(p) + \Delta w_{jk}(p)$$

(b) Calculate the error gradient for the neurons in the hidden layer:

$$\delta_j(p) = y_j(p) \cdot [1 - y_j(p)] \cdot \sum_{k=1}^{l} \delta_k(p)\, w_{jk}(p)$$

Calculate the weight corrections:

$$\Delta w_{ij}(p) = \alpha \cdot x_i(p) \cdot \delta_j(p)$$

Update the weights at the hidden neurons:

$$w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p)$$

Step 4: Iteration

Increase iteration p by one, go back to Step 2 and repeat the process until the selected error criterion is below 0.001.

4. TESTING AND TRAINING THE NEURAL NETWORK

As stated earlier, the purpose of this experiment was to determine whether the student will have difficulty solving the linear programming problem at hand or not. We used feed forward Backpropagation, which provides the facility to implement and test the neural networks and learning algorithm. Our neural network is a feed-forward network, with Single input layer (9 inputs), a single hidden layer (5 inputs) and a single Output layer (1 Output).

The output for the neural network is supposed to predict that a learner will have difficulty if the value is more than 0.4, thus requiring the intervention of the expert system to determine a suitable difficulty level in order to generate problem for the learner to solve. If the output value is less than or equal 0.4, then it is assumed that the learner will make four or less errors on a given problem, will not require the assistance of the expert system and can complete the current problem.

5. PREDICTION ACCURACY

Starting with the NN model as discussed in the previous sections, we trained it with data collected form the log files from last version of LP-ITS during the year of 2011. We tested the NN by generating linear programming problems for the learner to solve and at the same time the NN predict the academic performance of the learner for the same problem and store both results. That means the actual learner result and the predicted results are recorded in a file for determining the accuracy of the NN.

The average of incorrect learner academic performance predictions was 8%. The average prediction accuracy of the NN was 92%. The network produced the total of 11 incorrect predictions, out of 140 submissions. Fig.4 shows the comparison between the Actual Output of the learner and the predicted learner performance using the NN.

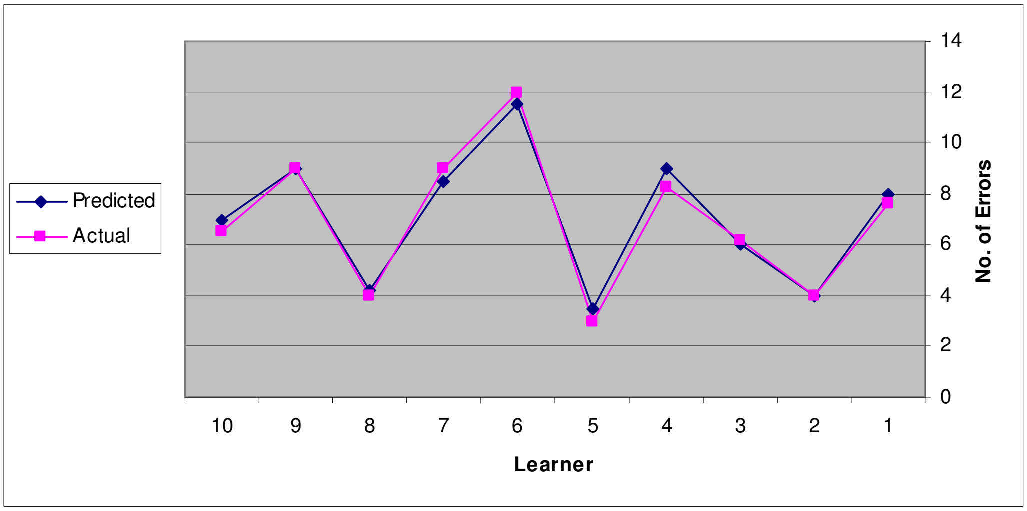

Figure 4. Comparison between Actual and NN Outputs

We were interested to know how quickly would the network adapts to the individual learner. Fig.3 shows how the number of incorrect predictions differs over time. It exhibits the number of incorrect predictions that happen on the first submission up to the tenth submission for each learner network to adapt to each learner.

6. Conclusion and Future Work

In this paper, we presented linear programming intelligent tutoring system with the use of neural networks to predict learner's academic performance and an expert system to decide the proper difficulty level problem to the learner to solve. We trained a backpropagation multi layer neural network using the data retrieved from the LP-ITS log file in the fall of 2010/2011 in the faculty of Engineering and Information Technology, Al Azhar University in Gaza, Palestine. The neural network predicts the number of errors the learner will make in solving the problem at hand. The learner model was firstly produced using all the data, with the prediction accuracy of 92%.

The LP-ITS is useful tutor for the students to examining themselves and developing their skills according to their predicted academic performance by the neural network. Further evaluation and enhancement to the module is going to be done in near future to get more better results.

References

[1] Abu Naser, S., Ahmed, A., Al-Masri, N. and Abu Sultan,Y., (2011), Human Computer Interaction Design of the LP-ITS: Linear Programming Intelligent Tutoring Systems, International Journal of Artificial Intelligence & Applications, 2(3).

[2] Abu Naser, S., (2012).A Qualitative Study of LP-ITS: Linear Programming Intelligent Tutoring System, International Journal of Computer Science & Information Technology, 3(1).

[3] Roll, I., Aleven, V., McLaren, B. M., & Koedinger, K. R. (2011). Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learning and Instruction*, *21*(2).

[4] Abu Naser, S. and Abu Zaiter O., (2008). An Expert System For Diagnosing Eye Diseases Using Clips, Journal of Theoretical and Applied Information Technology, 5(4).

[5] Abu Naser, S. El- Hissi, H., Abu- Rass, M. and El- khozondar, N., (2010). An Expert System for Endocrine Diagnosis and Treatments using JESS, Journal of Artificial Intelligence, 3(4).

[6] Shakiba, M., Teshnehlab, M., Zokaie,S., and Zakermoshfegh M., (2008). Short-Term Prediction of Traffic Rate Interval Router Using Hybrid Training of Dynamic Synapse Neural Network Structure 8(8).

[7] Khatib, T. and AlSadi,S., (2011). Modeling of Wind Speed for Palestine Using Artificial Neural Network. Journal of Applied Sciences 11(4).

[8] Tanoh, A., Konan, K., Koffi, S., Yeo, Z., Kouacou, M., Koffi, B. and Nguessan S.,(2008). A Neural Network Application for Diagnosis of the Asynchronous Machine. Journal of Applied Sciences 8(19).

[9] Senol, D. and Ozturan,M., (2010). Stock price direction prediction using artificial neural network approach: The case of Turkey. J. Artif. Intell., 3: 261-268.

[10] Lotfi, A. and Benyettou, A., (2011). Using Probabilistic Neural Networks for Handwritten Digit Recognition. Journal of Artificial Intelligence 4(4).

[11] Khanale, P. and Chitnis, S.,(2011). Handwritten Devanagari Character Recognition using Artificial Neural Network. Journal of Artificial Intelligence 4(1).

[12] Eriki, P. and Udegbunam, R. (2010). Application of neural network in evaluating prices of housing units in Nigeria: A preliminary investigation. J. Artif. Intell., 3: 161-167

[13] Shahrabi, J., Mousavi, S. and Heydar, M., (2009). Supply Chain Demand Forecasting: A Comparison Of Machine Learning Techniques and Traditional Methods. Journal of Applied Sciences 9(3).

[14] Kanakana1, G. and Olanrewaju, A. (2011). Predicting student performance in Engineering Education using an artificial neural network at Tshwane university of technology, ISEM 2011 Proceedings, September 21-23, Stellenbosch, South Africa.

[15] Kyndt, E., Musso, M., Cascallar, E. and Dochy, F., (2011). Predicting academic performance in higher education: Role of cognitive, learning and motivation. Earli Conference 2011 edition:14th location:Exeter, UK date:30 August - 3 September 2011.

[16] Mukta P. and Usha A., (2009). A study of academic performance of business school graduates using neural network and statistical techniques, Expert Systems with Applications, volume: 36, Issue: 4, Elsevier Ltd, ; pp.: 7865-7872

[17] Croy, M., Barnes, T., and Stamper, J. (2008). Towards an Intelligent Tutoring System for Propositional Proof Construction, Computing and Philosophy, A. Briggle, K. Waelbers, and P. Brey (Eds.), IOS Press, Amsterdam, Netherlands pp. 145-15.