Predicting students' performance using artificial neural networks

Ioannis E. Livieris[1,2], Konstantina Drakopoulou[1,2], Panagiotis Pintelas[1,2]
livieris@upatras.gr, kdrak@math.upatras.gr, pintelas@upatras.gr
[1] Department of Mathematics, University of Patras, GR 265 00, Greece.
[2] Educational Software Laboratory, University of Patras, GR 265 00, Greece.

Abstract

Artificial intelligence has enabled the development of more sophisticated and more efficient student models which represent and detect a broader range of student behavior than was previously possible.
In this work, we describe the implementation of a user-friendly software tool for predicting the students' performance in the course of "Mathematics" which is based on a neural network classifier. This tool has a simple interface and can be used by an educator for classifying students and distinguishing students with low achievements or weak students who are likely to have low achievements.

Keywords: Artificial neural networks, educational data mining, student's performance.

1. Introduction

During the last few years, the application of artificial intelligence in education has grown exponentially, spurred by the fact that it allows us to discover new, interesting and useful knowledge about students. Educational data mining (EDM) is an emerging discipline, concerned with developing methods for exploring the unique types of data that come from educational context. While traditional database queries can only answer questions such as "*find the students who failed the examinations*", data mining can provide answers to more abstract questions like "*find the students who will possibly succeed the examinations*". One of the key areas of the application of EDM is the development of student models that would predict student characteristics or performances in their educational institutions. Hence, researchers have begun to investigate various data mining methods to help educators to evaluate and improve the structure of their course context (see Romero & Ventura 2007; Romero et al. 2008 and the references therein).

The academic achievement of higher secondary school education (Lyceum) in Greece is a deciding factor in the life of any student. In fact, Lyceum acts like a bridge between school education and higher learning specializations that are offered by universities and higher technological educational institutes. Limiting the students that fail in the final examinations is considered essential and therefore the ability to predict weak students could be useful in a great number of different ways. More specifically, the ability of predicting the students' performance with high accuracy in the middle of the academic period is very significant for an educator for identifying slow learners and distinguishing students with low achievements or weak students who are likely to have low achievements. By recognizing the students' weaknesses the educators are able to inform the students during their study and offer them additional support such as additional learning activities, resources and learning tasks and therefore increase the quality of education received by their students. Thus, a tool

which could automatically recognize in time students' performance and especially students with learning problems is really important for educators.

However, the idea of developing an accurate prediction model based on a classifier for automatically identifying weak students is a very attractive and challenging task. Generally, datasets from this domain skewed class distribution in which most cases are usually located to the one class. Hence, a classifier induced from an imbalanced dataset has typically a low error rate at the majority class and an unacceptable error rate for the minority classes.

Related works can be found in (Kotsiantis & Pintelas, 2004) and (Kotsiantis et al. 2004). The first presents a high level architecture and a case study for a prototype machine learning tool for automatically recognizing dropout-prone students and how many students will submit a written assignment (project) in university level distance learning classes, while the second uses algorithms which base their predictions on demographic data and a small number of project assignments rather than class performance data as is the case in this paper.

In this work, we propose the application of an artificial neural network for predicting student's performance at the final examinations in the course of "Mathematics". Our aim is to identify the best training algorithm for constructing an accurate prediction model. We have also evaluated the classification accuracy of our neural network approach by comparing it with other well-known classifiers such as decision trees, Bayesian networks, classification rules and support vector machines. Moreover, we have incorporated our neural network classifier in a user-friendly software tool for the prediction of student's performance in order to making this task easier for educators to identify weak students with learning problems in time.

The remainder of this paper is organized as follows. In Section 2, we briefly describe the feedforward neural networks and in Section 3, we present the dataset of our study. Section 4 reports the experimental results while Section 5 presents our software tool and its main features. Finally, Section 6 presents our concluding remarks and our proposals for future research.

2. Artificial neural networks

Artificial neural networks (ANNs) are parallel computational models comprised of densely interconnected, adaptive processing units, characterized by an inherent propensity for learning from experience and also discovering new knowledge. Due to their excellent capability of self-learning and self-adapting, they have been extensively studied and have been successfully utilized to tackle difficult real-world problems (Bishop 1995; Haykin 1999) and are often found to be more efficient and more accurate than other classification techniques (Lerner et al. 1999). Classification with a neural network takes place in two distinct phases. First, the network is trained on a set of paired data to determine the input-output mapping. The weights of the connections between neurons are then fixed and the network is used to determine the classifications of a new set of data. Although many different models of ANNs have been proposed, the feedforward neural networks (FNNs) are the most common and widely used in a variety of applications.

Mathematically, the problem of training a FNN can be formulated as the minimization of an error function E; that is to find a minimizer $w^* = (w_1^*, w_2^*, ..., w_n^*) \in \mathbf{R}^n$ such that

$$w^* = \min_{w \in \mathbf{R}^n} E(w),$$

where E is the batch error measure defined by the sum of square differences over all examples of the training set, namely

$$E(w) = \sum_{p=1}^{P} \sum_{j=1}^{N_L} (y_{j,p}^L - t_{j,p})^2,$$

where $y_{j,p}^L$ is the actual output of the j-th neuron that belongs to the L-th (output) layer, N_L is the number of neurons of the output layer, $t_{j,p}$ is the desired response at the j-th neuron of the output layer at the input pattern p and P represents the total number of patterns used in the training set.

A traditional way to solve this problem is by an iterative gradient-based training algorithm which generates a sequence of weights $\{w_k\}$ starting from an initial point $w_0 \in \mathbf{R}^n$ using the iterative formula

$$w_{k+1} = w_k + \eta_k d_k,$$

where k is the current iteration usually called epoch, $\eta_k > 0$ is the learning rate and d_k is a descent search direction. Since the appearance of backpropagation (Rumelhart et al. 1986) a variety of approaches that use second order derivative related information was suggested for improving the efficiency of the minimization error process.

Here, we have utilized three well-known and widely used classical methods, namely the Broyden-Fletcher-Goldfarb-Shanno (BFGS) (Nocedal & Wright 1999), the Levenberg-Marquardt (LM) (Hagan & Menhaj 1994) and the Resilient Backpropagation (Rprop) (Riedmiller & Braun 1993). Moreover, we have also used a new efficient conjugate gradient algorithm, the modified spectral Perry (MSP) (Livieris & Pintelas 2012). Due to space limitations, we are not in a position to briefly describe here the above methods. The interested reader is referred to (Hagan & Menhaj 1994; Livieris & Pintelas 2012; Nocedal & Wright 1999; Riedmiller & Braun93).

3. Dataset and data extraction

The source of data for this study is obtained by the students' performance in the course of "Mathematics" of the first year of Lyceum. The data have been collected by the private Lyceum "Avgoulea-Linardatou" during the years 2007-2010 and consists of 279 different patterns.

Table 1: List of features used in our study

Attributes	Range Values
The oral grade of the 1st semester	[0,20]
The grade of the 1st test of the 1st semester	[0,20]
The grade of the 2nd test of the 1st semester	[0,20]
The grade of the final examination of the 1st semester	[0,20]
The final grade of the 1st semester	[0,20]
The oral grade of the 2nd semester	[0,20]
The grade of the 1st test of the 2nd semester	[0,20]
The grade of the 2nd test of the 2nd semester	[0,20]
The grade of the final examination of the 2nd semester	[0,20]
The final grade of the 2nd semester	[0,20]

Grade of the final examination	[0,20]

Since it is of great importance for an educator to recognize weak students in the middle of the academic period, the attributes concerned information about the students' performance from the 1st and the 2nd semester such as tests grades, final examination grade and oral grade (Table 1).

Moreover, we use two different approaches to classify the students based on their grade of the final examination.

- **2-level classification:** The students were categorized in two groups i.e. if the student's performance is between 0 and 9 then it is characterized as "fail" and if the student's performance is between 10 and 20 then it is characterized as "pass".
- **3-level classification:** The students were categorized in three groups i.e. if the student's performance is between 0 and 9 then it is characterized as "fail", if the student's performance is between 10 and 15 then it is characterized as "good" and if the student's performance is between 16 and 20 then it is characterized as "very good".

Based on the previous, we have created the following datasets:

DATA_2^A: It contains the features from Table 1 which concerns the student's performance of the 1st semester and the students were categorized using 2-level classification.

DATA_3^A: It contains the features from Table 1 which concerns the student's performance of both semesters and the students were categorized using 2-level classification.

DATA_2^{AB}: It contains the features from Table 1 which concerns the student's performance of the 1st semester and the students were categorized using 3-level classification.

DATA_3^{AB}: It contains the features from Table 1 which concerns the student's performance of both semesters and the students were categorized using 3-level classification.

4. Experimental results

In this section, we present experimental results in order to evaluate the classification capability of neural networks using four different training algorithms: BFGS, LM, Rprop and MSP. All networks have received the same sequence of input patterns and for evaluating classification accuracy we have used the standard procedure called 10-fold cross-validation (Kohavi 1995). All simulations have been carried out on a computer (2.66GHz, Quad-Core processor) with 4Gbyte RAM and the implementation code was written in Matlab 2008a.

In Figure 1 are summarized the mean generalization results of each neural network training method, measured by the percentage of testing patterns that were classified correctly in the presented datasets. We point out that the training algorithm MSP is an excellent generalizer since it manages to exhibit the highest generalization performance, followed by the Rprop algorithm. Unfavorably, the neural networks that were trained with the training algorithms BFGS and LM achieved the worst performance relative to all datasets.
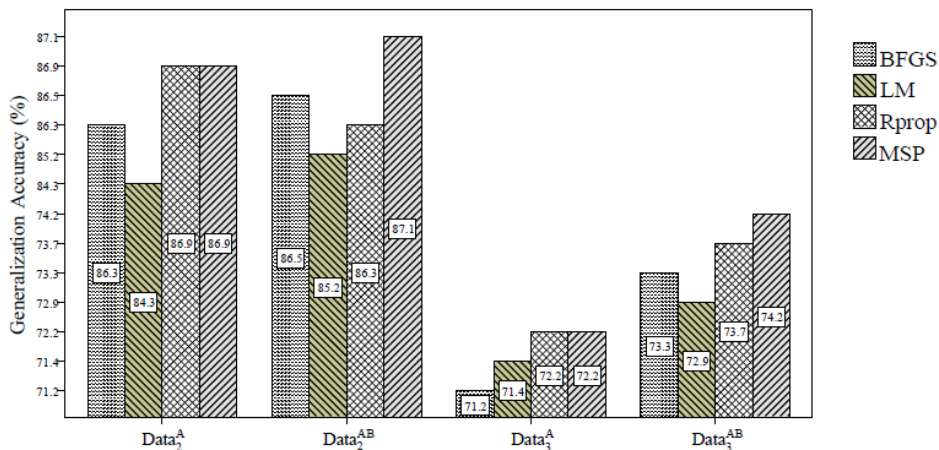
Figure 1. Mean generalization accuracy of all training algorithms for each dataset.

We conclude our analysis by comparing the generalization accuracy of the neural networks that were trained with MSP method with other well-known classifiers such as decision trees, Bayesian networks, classification rules and support vector machines. For this purpose, we have selected a representative algorithm for each machine learning technique. The most commonly used C4.5 algorithm (Quinlan 1993) was the representative of the decision trees. Naive Bayes (NB) algorithm was the representative of the Bayesian networks (Domingos & Pazzani 1997). The Ripper algorithm (Cohen 1995) was the representative of the rule-learning techniques because it is one of the most usually used methods that produce classification rules. Finally, the Sequential Minimal Optimization (SMO) algorithm was the representative of the support vector machines in our study because it is one of the fastest training methods (Platt 1998). Notice that all algorithms have been implemented in WEKA toolbox (Hall et al. 2009).
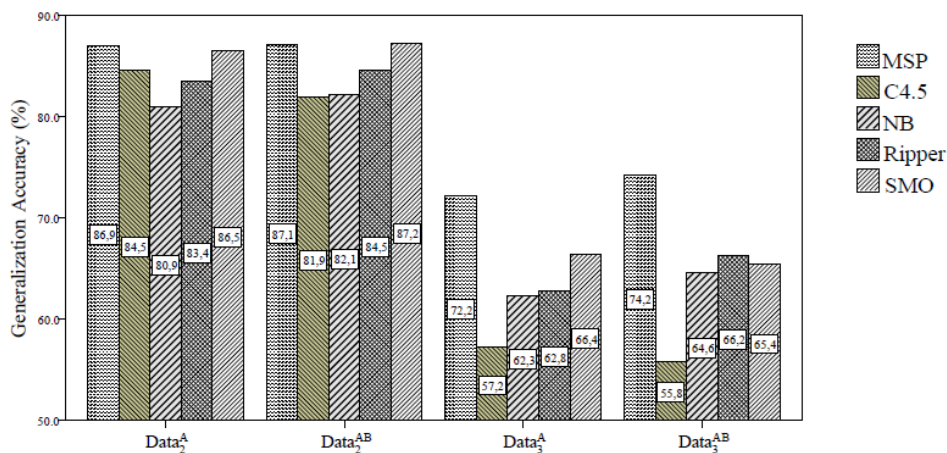

Figure 2. Mean generalization accuracy of all classifiers for each dataset.

Figure 2 presents the mean generalization results of all classifiers, relative to all datasets. Obviously, MSP and SMO report the best results while for the datasets DATA_2^A and DATA_2^{AB}. Furthermore, MSP illustrates the highest generalization accuracy for the datasets DATA_3^A and DATA_3^{AB}. Moreover, it is worth mentioning that the MSP-trained FNNs exhibit more consistent behavior and achieved better generalization accuracy than the rest of the classification algorithms.

5. Prediction tool

In this section, we present our developed software tool for predicting the student's performance at the final examinations (Figure 3) which has a simple user-friendly interface. The main features of our software tool are:

- **Neural network:** This module is dedicated for importing the dataset in a specific format (txt). Once the dataset is loaded the user can ask the tool to construct the neural network classifier.
- **Neural network parameters:** This module is used for selecting the neural network training algorithm and the classification level using the corresponding pop-up menus.
- **Student's grades:** This module allows the user to insert the grades of a new student for predicting its performance.
- **Prediction of student's performance at the final examinations:** This module displays the prediction of the classifier for the new student.
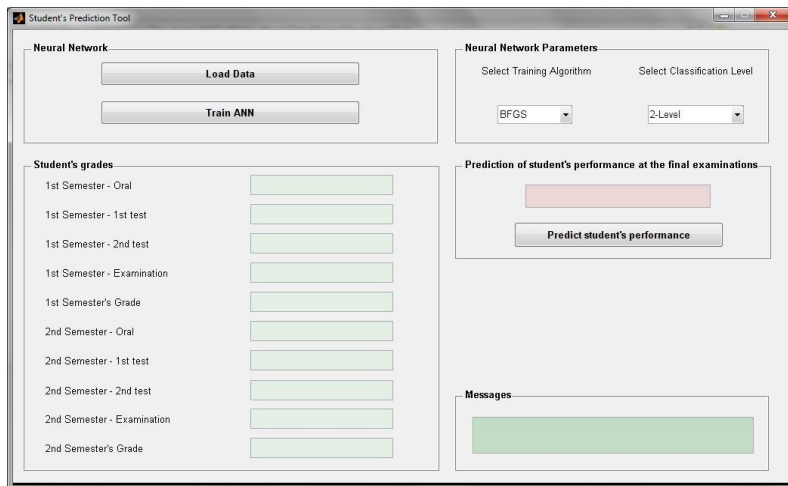

Figure 3. Prediction tool.

Subsequently, we present a use-case study to illustrate the functionality of our tool and the experiment set up process. Firstly, by clicking on the button "*Load Data*" the user can load his data collected from his own course (Figure 4). Next, the user can select the training algorithm and the classification level using the corresponding pop-up menus from the "Neural network parameters" module. Our tool always recommends one algorithm and a

classification level by default in order to facilitate its usage/execution for beginners. In our example, we have chosen the MSP as training algorithm and the 3-level classification. Then, the user can ask the tool to construct the neural network classifier by clicking on the button "*Train ANN*". Notice that the classification accuracy of our model is evaluated using 10-fold cross-validation. After the training process is complete, the user can import the new student's grades in order for predicting the student's performance at the final examinations by clicking on "*Predict student's performance*" (Figure 5). In this example, the model predicts that the student's performance will be GOOD at the final examinations with probability 74.1%.
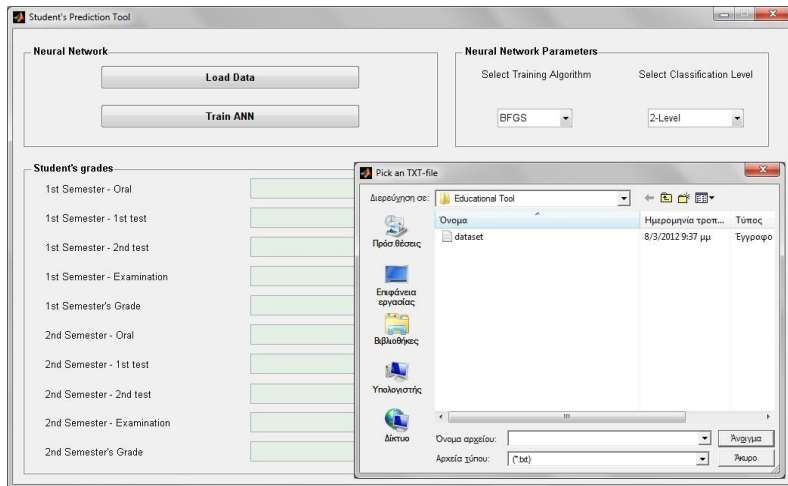

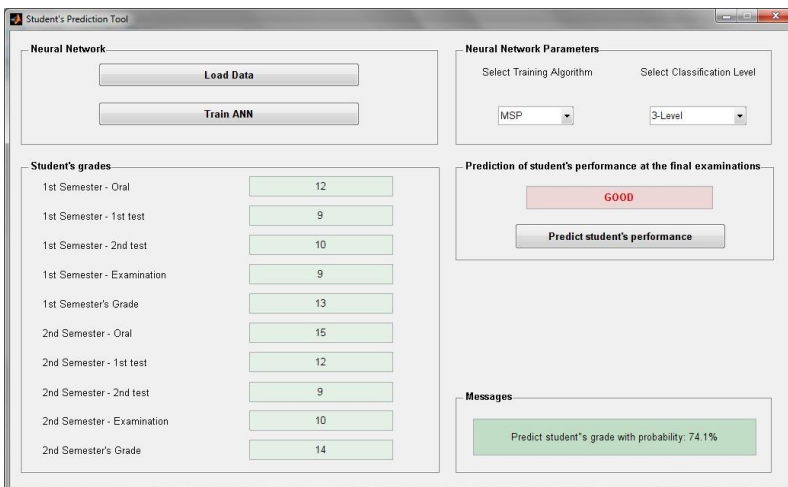**Figure 4. Loading the data in the prediction tool.**

Figure 5. Tool's prediction about the performance of a student at the final examination.

5. Conclusions & future research

In this work, we developed a user-friendly software tool which is based on neural network classifiers for predicting the student's performance in the course of "Mathematics" of the first year of Lyceum. Based on our numerical experiments, we conclude that the MSP-trained neural networks exhibit more consistent behavior and illustrate better classification results than the other classifiers. Furthermore, we have shown the main features of our software tool and a use-case study to illustrate its functionalities and the experiment set up process. Our tool is still under development but our first results show that we can gain insights about student progress and recommend possible actions such as further study or additional learning activities, resources and learning tasks. The tool was tested by a small number of teachers who were enthusiastic with its predictions as they felt they were close to their own based on their extensive teaching experience.

In our plans for the next academic year is to do a systematic and extensive evaluation of the tool by teachers in two schools, one private and one public. Furthermore, our future research is concentrated on collecting data from all three years of Lyceum and applying our methodology for predicting the students' performance on the Panhellenic national level examinations.

References

Bishop C.M. *Neural Networks for Pattern Recognition*. Oxford, 1995.

Cohen W. Fast effective rule induction. In *International Conference on Machine Learning*, p.p. 115–123, 1995.

Domingos P. and Pazzani M. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29, p.p. 103–130, 1997.

Hagan M.T. and Menhaj M.B. Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6), p.p. 989–993, 1994.

Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., Witten I.H. The WEKA Data Mining Software: An Update; *SIGKDD Explorations*, 11(1), 2009.

Haykin S. *Neural Networks: A Comprehensive Foundation*. Macmillan College Publishing Company, 1994.

Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IEEE International Joint Conference on Artificial Intelligence*, p.p. 223–228, 1995.

Kotsiantis S.B., Pierrakeas C. and Pintelas P., Predicting students' performance in distance learning using machine learning techniques, *Journal of Applied Artificial Intelligence*, 18(5), p.p. 411-426, 2004.

Kotsiantis S.B. and Pintelas P., A decision support prototype tool for predicting student performance in an ODL environment, *International Journal of Interactive Technology and Smart Education*, 1(4), p.p. 253-263, 2004.

Lerner B., Guterman H., Aladjem M., and Dinstein I. A comparative study of neural network based feature extraction paradigms. *Pattern Recognition Letters*, 20(1), p.p. 7–14, 1999.

Livieris I.E. and Pintelas P. An improved spectral conjugate gradient neural network training algorithm. *International Journal on Artificial Intelligence Tools*, 21(1), 2012.

Nocedal J. and Wright S.J. *Numerical Optimization*. Springer-Verlag, New York, 1999.

Platt J.C. Advances in Kernel Methods - Support Vector Learning. *MIT Press, Cambridge*, 1998.

Quinlan J.R. *C4.5: Programs for machine learning*. Morgan Kaufmann, San Francisco, 1993.

Riedmiller M. and Braun H. A direct adaptive method for faster backpropagation learning: The Rprop algorithm. In *IEEE International Conference on Neural Networks*, p.p. 586–591, San Francisco, CA, 1993.

Romero C. and Ventura S. Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, 33, p.p. 135–146, 2007.

Romero C., Ventura S., and Garcia E. Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, 51(1), p.p. 368–384, 2008.

Rumelhart D.E., Hinton G.E., and Williams R.J. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, p.p. 318–362, 1986.