Using Artificial Neural Networks to Predict First-Year Traditional Students Second Year Retention Rates

Mark Plagge
Columbus State University
4225 University Ave
Columbus, GA 31907
1-706-507-8800
plagge_mark@columbusstate.edu

ABSTRACT

This research investigates the use of Artificial Neural Networks (ANNs) to predict first year student retention rates. Based on a significant body of previous research, this work expands on previous attempts to predict student outcomes using machine-learning techniques. Using a large data set provided by Columbus State University's Information Technology department, ANNs were used to analyze incoming first-year traditional freshmen students' data over a period from 2005–2011. Using several different network designs, the students' data was analyzed, and a basic predictive network was devised. While the overall accuracy was high when including the first and second semesters worth of data, once the data set was reduced to a single semester, the overall accuracy dropped significantly. Using different network designs, more complex learning algorithms, and better training strategies, the prediction accuracy rate for a student's return to the second year approached 75% overall. Since the rate is still low, there is room for improvements, and several techniques that might increase the reliability of these networks are discussed.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence]:** Learning – *Connectionism and Neural Nets*

General Terms

Human Factors

Keywords

Artificial Neural Networks, Student Retention

1. INTRODUCTION

Student retention, progression and graduation rates are a significant concern for institutions in the United States. Students who started seeking a four-year degree in 2004 had about a 58% completion rate after six years, according to the U.S. Department of Education [8]. College attrition rates are of a concern throughout the United States. Prediction of a student's outcome would be extremely valuable for educators and educational institutions. There have been several attempts to predict student rates, using various methods to tackle the problem [5]. Wongkhamdi et al. in [9] have used neural networks to a reasonable degree of success, and the use of machine learning as a whole seems promising [1, 2, 3].

The success of students and a well-educated work force will be one of the biggest driving forces on our country's future and relevance in the modern world. By enabling faculty and advisers in higher education institutions to better identify at-risk students sooner, they will have the tools necessary to drastically improve student outcomes.

This research investigates the application of feed-forward neural networks and cascade feed-forward neural networks to predict the outcomes of first year freshman students. This research uses a sample pool of data from Columbus State University from 2005–2010, provided by the university's information technology services department. The department is also using statistical analysis on the data, the results of which will help verify the results of this work. The goals of this work were to create a predictive neural network for student RPG rates and provide insights into the weights of factors that affect student RPG rates. The artificial neural network will work with data points available on a student at the end of the first semester. This research shows that given enough data points, an artificial neural network is suited for the task.

2. NEURAL NETWORKS

An Artificial Neural Network (ANN) is a well-documented and developed machine learning technique. This algorithm has been used with a good degree of success by several researchers, including [6, 7].More recently, [9] found very good results when applying an ANN to predict student graduation outcomes. The solid body of research suggests that an ANN is an excellent tool to use for this data.

ANNs are constructs that roughly mimic the structure of a biological brain. The structure of an ANN is best represented by an ordered graph. In the most common model, a feed-forward or multilayer perceptron, the left side of the graph contains the neurons that represent the input layer. These neurons generally relay the data given to them further into the network. The next N layers in the network are known as the Hidden Layer. Each neuron in this layer takes the summation of all of the previous layer's output, and applies an activation function to the data. If the result is larger than a specific threshold, the neuron will fire, and send a numeric value to the next layer. An ANN can have multiple hidden layers, and any number of neurons per layer. This type of network is a very rough approximation of the complexity of an actual biological neural network.

ACMSE'13, April 4-6, 2013, Savannah, GA, USA.

Of the many types of neural network models currently in common use, this research uses two. The first model is known as a feed-forward back-propagation neural network. This network consists of an input layer, one or more hidden layers, and an output layer. This network trains through a process called back-propagation. Data is carried from the input layer, through the hidden layer(s), and then output. Once there, the output is compared with a set of expected values. The weights of the neurons are then adjusted based on the training algorithm.

In the case of this research, two primary learning algorithms were used. The first algorithm is known as Levenberg-Marquardt back-propagation. This algorithm is a fast and accurate training algorithm and is commonly used with generally good results[4]. The other training algorithm uses a nonlinear conjugate gradient method (NCG), with Polak-Ribire updates.

These networks were trained using a selection of training data, testing data, and validation data. For initial experiments, a 75% /15% /15% mixture of testing, training, and validation data was used. This ratio was kept at a similar ratio through most of the work. The final model had a ratio of 60% training, 20% validation, and

20% testing data selected for its training. The validation and test data was selected using a random selection algorithm. For many data sets, a random selection of data for the validation and testing selections works very well.

3. METHODS

3.1 Data Set

The data used was a result of a collaboration between the computer science department and the Columbus State University Information Technology Services department. The IT department at Columbus State was doing a major data mining operation-involving student RPG rates. They provided us with samples of data from entering freshman from 2005 - 2010. This data contained information on traditional, first year entering freshmen. This dataset consisted of a large number of variables per student, too many for an effective neural network. Furthermore, for this prediction to be useful, it must predict a student's outcome at or before the end of the first semester. This data contained information on the student's second semester outcomes. Removing irrelevant data, and combining redundant information reduced the dataset to 16 points per student, as seen in Table 1.Reducing the data and converting it into something appropriate for a neural network entailed several phases. The first issue tackled was converting the data into usable numeric values. Textual descriptions were converted into numerical categories. The next data reduction technique involved summarizing the class data used. Rather than an individual variable for each class taken in a semester, the data here is a count of the number of core courses taken during the semester. This provided the neural network with a better set of data for its analysis.

Since some of the data's numerical values were very large, and others were small (GPA and Distance from Home), the data had to be normalized. Each point was converted to a set between 1 and 0, ensuring that there would not be any variables that dominated the neural network's data.

For the analysis, we only considered one outcome: a binary value representing a student's return to the university after first year. This narrowed the scope of this analysis to prediction of a student's second year return. Analysis that was attempting to predict the first year outcome is acceptable since the majority of dropouts at Columbus State University happen during the first to second year transition. This is not unique to Columbus State University; a majority of student dropouts happen between the first and second year of school in universities across the United States as well [3, 7]. The factors given here represent a sampling of data that the information technology department gave us. The research, in addition to predicting the outcome of a student, was interested in determining if any of this data was particularly influential on the student.

Table 1: Data Categories used for Neural Network

Category	Data Type
Returned 2nd Year	Boolean
Student Age	Integer
Gender	Boolean
Ethnicity	Integer (Categorical)
International Status	Boolean
In State Status	Boolean
Major	Integer (Categorical)
Minor	Integer (Categorical)
CSU Entrance Test Aggregate	Float
Fall Semester Core Course Count	Integer
Distance from Home	Integer
High School GPA	Float
Father Highest Education Level	Integer (Categorical)
Mother Highest Education Category	Integer (Categorical)
Estimated Family Contribution	Float
Financial Need Difference	Float
Fall GPA	Float

3.2 Student Data

It should be noted that the student data used in this research was made anonymous by the information technology department at Columbus State University before use. Unique identifiable traits, such as student ID numbers, addresses, and names were removed from the data set. The students' ID numbers were replaced with a randomly generated unique identifier that allowed for data analysis without being able to identify a particular student.

3.3 Experimentation Results

Determining the optimal neural network model required training and testing several neural network models and algorithms. The general results of the network runs are summarized in Table 2. In this table, networks A and B contained both first semester data and second semester data. Including two semesters worth of data significantly increased the accuracy of the neural network predictions.

Table 2 shows a selection of interesting results from the neural network training and testing runs. The network type is simply the network design. For two layer networks, there is only an input and output layer, so the networks have only one layer of variable size. The total prediction accuracy is an overall percentage that reflects

Table 2: Neural Network Experimentation Summary

Network	Network Type	Input Layer 1 Size	Layer 2 Size	Total Prediction Accuracy	Dropout Prediction Accuracy	Training Algorithm
A	Feed Forward 2 Layer	8	N/A	81.90%	75.80%	Levenberg-Marquardt
B	Feed Forward 2 Layer	16	N/A	87%	77.40%	Levenberg-Marquardt
C	Feed Forward 2 Layer	16	N/A	75.70%	10.30%	Levenberg-Marquardt
D	Feed Forward 2 Layer	32	N/A	66.20%	10.05%	Levenberg-Marquardt
E	Feed Forward 3 Layer	32	64	63.10%	12.01%	Levenberg-Marquardt
F	Feed Forward 3 Layer	64	128	68.83%	1.03%	NCG
G	Feed Forward 3 Layer Cascade	32	64	76.09%	40.88%	NCG
H	Feed Forward 3 Layer Cascade	64	128	69.00%	69.99%	NCG

how accurate the network was in classifying a student as a returning student or a non-returning student. The dropout prediction is a subset of the total accuracy that shows how accurate the network was in identifying non-returning students. The last column shows the training algorithm used for the particular network. The table shows a summary of the various training methods and networks used for this research.

Networks A and B had very good rates of confusion after training. Once the second semester was removed, as can be seen in network C, the accuracy rate dropped significantly. Despite several attempts at re-training, this neural network model was not able to recover the high accuracy rates observed with both semesters worth of data. Next, a second layer of neurons was added to the neural network model. Keeping the design of the network similar, except for the added layer of neurons, provided accuracy that indicated the second layer was beneficial. These network training runs, D and E, provided a higher rate of accuracy than were seen with the previous network design. However, the network was still not providing a significantly accurate prediction rate of non-returning students. By the time additional neurons were added to the network, the feed forward network was not providing accurate results with the students who were dropping out. The network achieved just about a 1% accuracy rate for predicting these students. This was the point where the network design was changed. This is shown as network F in Table 2.

Networks G and H show the data accuracy as run through a feed forward neural network with a cascade function. These networks started to give better accuracy rates with the same data, and were able to give better dropout prediction rates overall. This type of network seems to be the most promising for processing student data.

3.4 Experimental Work

Several different neural network models were applied to the data during the course of this research. The initial network was a single layer neural network with feed-forward features. This network had to be adapted to achieve a better accuracy rate than what was initially produced. After some experimentation with this data, a two hidden layer neural network was found to produce a significantly more accurate result.

Wongkhamdi et al. [9] argued that the best way to determine the number of middle layers is trial-and-error. From the initial neural network consisting of 16 neurons, the number was increased until training results started to plateau. By the end of testing, the optimal neural network size was 64 and 128 neurons in each layer. Past this number and the neural network began to learn the pattern of the data, and below this number of neurons the network could not provide a reasonable accuracy rate.
The design of the network was also determined by trial-and-error. Initially, a feed-forward network was used, based on promising results from [2, 3]. However, the training results achieved here were initially not as promising. The experiments ranged from 16 to 256 hidden layer nodes, with a peak in accuracy at 128 nodes.

When analyzing the data and the confusion matrix from the single hidden layer models, it started to appear that the networks were learning the data rather than finding patterns in the data. Eventually, the network settled on a 68% accuracy rate; this, however, closely matched the percentage of returning students. The neural network started to return a 1 value for almost any input values given. This is why network F's dropout prediction accuracy is at 1%. Table 3 shows the confusion matrix of network F. This confusion matrix represents 0 as a non-returning student, and 1 as a returning student. The network predicted 4072 students returning out of 4086 students, but only correctly predicted 27 students dropping out.

Table 3: Network F Confusion Matrix

from \ to	0	1	Total	Accuracy
0	27	1842	1869	1.44%
1	14	4072	4086	99.66%
Total	41	5914	5955	68.83%

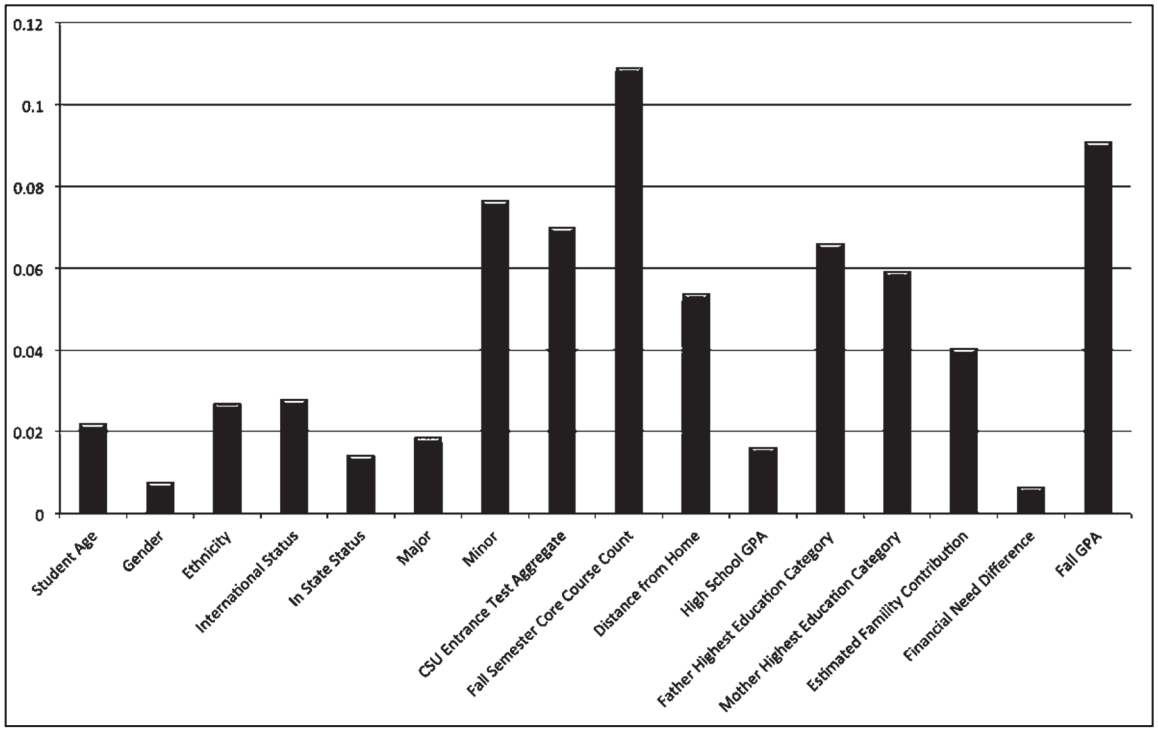

Figure 1: Average Weight Magnitude from Neural Network H

At this point, a new network design was used. Based on the experimental runs of the data, a cascade forward neural network was decided on. This network significantly increased the accuracy of prediction of a student not returning in the second year. The best-case scenario provided a 76% overall accuracy rate, and a 40% accuracy rate for predicting student dropout rates. The cascade forward network was trained using a larger set of testing and validation data, 20% each. In addition, the network underwent multiple training sessions in order to minimize the potential for data skew. Since the test and validation data was selected randomly from the whole set, there would have been the potential for not selecting enough of the student fail conditions during selection. Multiple training runs, with new sets of training and validation data, helped to reduce the potential for this type of data skew, and provided better accuracy overall. Table 4 shows the confusion matrix for this network result.

An increase in the size of the new network design resulted in the network recognizing a higher rate of non-continuing students, but at the cost of some of the accuracy of the positive result predictions. The end result was a network with less than 70% accuracy for both the returning and non-returning students. Table 4 shows the confusion matrix for the neural network H.

Table 4: Network G Confusion Matrix

from \ to	0	1	Total	Accuracy
0	764	1105	1869	40.88%
1	319	3767	4086	92.19%
Total	1083	4872	5955	76.09%

3.5 Neural Network Weights

A series of weights were extracted from the more accurate neural network. This was done in an attempt to increase the accuracy of the neural network - by removing data that was not relevant to the classification of the result the accuracy of the network would hopefully increase. These weights influence how each input value is applied to a neuron in the neural network. The averages of these weights, shown in Figure 1 with magnitudes included, indicate that there are some variables that might be removed to increase the accuracy of the network's predictive model. Further analysis using this lighter data, along with new neural network training will be, however, reserved for future research.

The weights in Figure 1 are from the first layer of the neural network. One of the challenges facing an analysis of a neural network is the interpretation of the weights. With two layers, the second layer's complexity is extremely high. Figure 1, therefore, shows the input weights carried to the first layer of the neural network.

4. RESULTS

The neural network model was mostly able to predict the potential for a student to return the next semester. The neural network was not, however, as apt at predicting the rate that a student will not return the next semester. Using more advanced training algorithms and more layers in the neural network helped reduce this inaccuracy, however, the prediction rate was still fairly low. There is, however, a great deal of potential in these results. The fact that the neural network is able to predict, with extremely high accuracy, the second year outcome of a student when using two semesters worth of data is a very good indicator of the network's suitability to this type of problem. The predictions obtained using these models are potentially useful to an adviser to a student. If one of these models were to suggest that a student would not return during the second semester, then the probability is

extremely high that the student is at-risk. However, these models all tended to err on the positive side of things. If an adviser were to rely completely on these models, there would be a high incidence of students not being recognized as potential at-risk students. This model would have to be combined with other types of data mining and machine learning methods in order to become significantly useful.

4.1 Conclusion

An artificial neural network model has a phenomenal deal of potential when predicting student graduation rates. Given the importance of this issue, and the suitability of the ANN to this type of task, more research is deserved. Given that the accuracy rates potentially can be high, and that networks can be trained for different types of data, it is within the realm of possibility that this type of network will be able to drastically increase the retention rates at universities nationwide. There is room for future work, however. There is a huge amount of potential with this predictive neural network model.

One of the potential issues with this work is the selection of the validation and test data. A random selection algorithm was used to choose points from the primary data set to fill the training and validation groups. This is a fairly standard process for most data sets. However, this could have skewed the results of the network's training. Since the data is a binary result, and only 22% of the data is in one set, there is a good probability that the random data would not select enough, or even any, of the student failure conditions. Even though the training was run multiple times with a randomized selection from the provided data set, it still could be causing the inability of the neural network to accurately predict the student fail condition. Delen [3] mentions that it is extremely important to have a balanced data set when doing a binary classification with a neural network, so balancing the data could give significantly better results.

A solution for this would be to alter the set of data so that the training, testing, and validation data consisted of a 50%/50% mix of student outcomes. This would entail using all of the set of dropout outcomes, and as many randomly selected positive outcomes as there are dropout outcomes. Even though this would reduce the quantity of data, it would potentially improve the accuracy of the resulting network. Since a neural network is driven by weights, having a close to even selection of input data assists with the training of the network. This is something that should be looked at in future research, as this could potentially solve the problems with this particular model.

Further research avenues that should be considered include doing an exhaustive weight analysis. By removing noise from the model, accuracy rates should increase. Furthermore, the weights from this neural network could be applied to other machine learning methods.

Finally, it might be extremely valuable to cluster the data into groups that are relevant to each other. This type of analysis would provide advisers an insight into the neural network's black box prediction of a student's outcome. Using a Kohonen network, different classes of students could be determined, providing a deep analysis of the student's influences. This could provide invaluable information to an academic adviser, counselor, or other faculty at a university.

5. REFERENCES

[1] Barker, K. et al. 2004. Learning From Student Data. *Proceedings of the 2004 IEEE Systems and Information Engineering Design Symposium* (2004), 79–86.

[2] Bogard, M. et al. 2011. *A Comparison of Empirical Models for Predicting Student Retention*. White Paper. Office of Institutional Research, Western Kentucky University.

[3] Delen, D. 2010. A comparative analysis of machine learning techniques for student retention management. *Decision Support Systems*. 49, 4 (Nov. 2010), 498–506.

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

[5] Herzog, S. 2006. Estimating student retention and degree-completion time: Decision trees and neural networks vis-à-vis regression. *New Directions for Institutional Research*. 2006, 131 (2006), 17–33.

[6] Imbrie, P.K. et al. 2008. Artificial Intelligence Methods to Forecast Engineering Students' Retention Based on Cognitive and Non-cognitive Factors. *2008 Annual Conference & Exposition of the American Society for Engineering Education* (Pittsburg, PA, 2008).

[7] Karamouzis, S.T. and Vrettos, A. 2008. An Artificial Neural Network for Predicting Student Graduation Outcomes. *Proceedings of the World Congress on Engineering and Computer Science* (San Francisco, 2008), 22–25.

[8] The Condition of Education - Postsecondary Education - Completions - Postsecondary Graduation Rates: 2012. *http://nces.ed.gov/programs/coe/indicator_pgr.asp*.

[9] Wongkhamdi, T. and Seresangtakul, P. 2010. A Comparison of Classical Discriminant Analysis and Artificial Neural Networks in Predicting Student Graduation Outcomes. *Proceedings of the Second International Conference on Knowledge and Smart Technologies* (Chonburi, 2010), 29–34.