Neural Networks Refined: Using a Genetic Algorithm to Identify Predictors of IS Student Success

Randall S. Sexton
Computer Information Systems
Southwest Missouri State University
901 South National
Springfield, Missouri 65804
Office: 417-836-6453
Fax: 417-836-6907
Email: rss000f@mail.smsu.edu
Web: http://www.faculty.smsu.edu/r/rss000f/

Michael A. Hignite
Computer Information Systems
Southwest Missouri State University
901 South National
Springfield, Missouri 65804

Tom Margavio
Computer Information Systems
Southwest Missouri State University
901 South National
Springfield, Missouri 65804

John Satzinger
Computer Information Systems
Southwest Missouri State University
901 South National
Springfield, Missouri 65804

Neural Networks Refined: Using a Genetic Algorithm to Identify Predictors of IS Student Success

Abstract

In this study, a genetic algorithm trained neural network was used to identify relevant characteristics of successful IS students. Those variables identified as predictors of student success included a student's age, gender, and the Myers-Briggs' attributes of extraversion/introversion (EI) and sensing/intuition (SN). During the past decade, neural networks have gained in popularity, as they work exceedingly well for mapping unknown functions from historical data. With a neural network, a researcher can simply include a wide variety of inputs in the model and the neural network will learn to discriminate between the relevant inputs and those that are irrelevant. Since the majority of neural network research uses gradient search techniques, usually some form of backpropagation, there is no way to identify the inputs that actually contribute to the prediction. By using the genetic algorithm as an alternative search technique, these contributing relevant variables can be identified.

KEY WORDS: Genetic Algorithm, Neural Networks, Artificial Intelligence, Myers-Briggs.

Introduction

As interest and enrollment in Information Systems (IS) programs has grown, so too has grown the problem of identifying those traits or characteristics of students that contribute to success in the field. In this study, a genetic algorithm trained neural network was used to identify relevant characteristics of IS students that could serve as predictors of student success. By successfully identifying such factors, the researchers believe, the opportunity exists to improve or enhance student success rates.

Neural Networks

The ability of neural networks (NNs) to approximate unknown functions to any degree of desired accuracy, as demonstrated by Funahashi [5] and Hornik et al. [6], is the foremost reason why NNs are increasing in popularity. Since NNs can do this without making any unnecessary assumptions about the distributions of the data, researchers can include any input variables that they believe could possibly contribute to the NN model.

Although it is likely that irrelevant variables are introduced to the model, the NN is expected to sufficiently learn to ignore these variables during the training process. The NN does this by finding a solution, or set of weights, that when plugged into the network will zero out any effects of irrelevant variables. Since irrelevant variables have nothing to do with the output, there is no underlying function for the NN to map. Therefore, to find weights that negate the effects of the irrelevant variables, the NN memorizes the values that are seen in the training data.

A problem can occur when new observations are introduced to the NN model. Since the NN memorized the values of the irrelevant variables in the training data in order to zero them out, any deviation of irrelevant input values in the new observations will likely introduce additional error. A better approach is to identify the irrelevant inputs in order to eliminate them entirely from the model. This prevents the introduction of additional error for out-of-sample observations and provides the researcher with a better understanding of a given problem.

For example, it would be possible to train a NN that would predict whether an employee would be likely to terminate their employment with a company within the next year. Employee information, such as gender, age, length of employment, etc., then, would be designated as inputs to the model. If the NN model accurately predicted this outcome, identified gender and age as relevant variables, and discarded length of employment as irrelevant, specific reasons why employees were leaving their jobs could then be identified.

The majority of previous NN research has utilized gradient search techniques, such as backpropagation (BP), which require differentiability of the objective function. Unfortunately, with this technique the ability of researchers to identify relevant variables with other than trail-and-error methods is eliminated.

In this paper, however, a modified genetic algorithm (GA), which does not require differentiability of the objective function, is used for training a NN; that NN, then, being designed to correctly distinguish relevant from irrelevant variables and simultaneously search for a global problem solution.

The Genetic Algorithm

Genetic algorithms are computational models that mimic natural evolution to solve complex optimization problems. The algorithm's similarity to natural selection inspires its name. As the GA progresses through generations, the weights most favorable for optimizing the objective function will reproduce and thrive in future generations, while poorly performing weights die out, as in "survival of the fittest".

A recent survey of research pertaining to the combination of genetic algorithms and neural networks can be found in Yao [12]. In past research, the GA has been used for finding optimal NN architectures and as an alternative to BP for training. This paper uses the GA for both.

Although Schaffer [10] found that most of the past research using the GA as an alternative training algorithm has not been competitive with the best gradient learning methods, Sexton et al. [11] found that the problem with this past research is in the implementation of the GA and not its inability to perform the task. The majority of past implementations of the GA encode each candidate solution of weights into binary strings. This approach works well for optimization of problems with only a few variables but for neural networks with a large number of weights, binary encoding results in extremely long strings. Consequently, the patterns that are essential to the GA's effectiveness are virtually impossible to maintain with the standard GA operators such as crossover and mutation. It has been shown by Davis [2] and Michalewicz [8] that this type of encoding is not necessary or beneficial.

A more effective approach is to allow the GA to operate over real valued parameters [11]. The alternative approach described in Sexton et al. [11] also successfully outperformed backpropagation on a variety of problems. This line of research in based on the algorithm developed by Dorsey & Mayer [4] and Dorsey, Johnson and Mayer [3].

The GA, then, is a global search procedure that searches from one population of solutions to another, focusing on the area of the best solution so far, while continuously sampling the total parameter space. The following is a general outline of how the GA searches for global solutions. A formal description of the algorithm can be found in Dorsey & Mayer [4].

(Place Figure 1 approximately here)

Each of the 20 randomly drawn solutions in the population is evaluated based on a pre-selected objective function, which is not necessarily differentiable. Since our objective is to find a global solution that can identify relevant variables, an objective function is needed that will reduce the error between estimates and real outputs and the number of connections in the model. To do this, it becomes necessary to use an objective function that is not differentiable. The following objective function, Equation 1, takes the sum of squared errors (SSE) and adds an additional penalty error for every connection, or weight, that is non-zero. The additional error is set to the root mean squared error (RMSE) or the typical error of one observation for the particular solution being evaluated.

Consequently, a connection or weight is only eliminated if the SSE is increased by less than the RMSE. By setting the penalty equal to the RMSE, the NN is prevented from eliminating too many weights in the solution. Although, this objective function seems to work well for the problem in this study, the penalty value assigned for non-zero connections is arbitrary. Additional research, beyond the scope of this study, is warranted for finding an optimal penalty assignment.

Equation 1

$$E = \sum_{i=1}^{N} \left(O_i - \hat{O}_i\right)^2 + C\sqrt{\frac{\sum_{i=1}^{N} \left(O_i - \hat{O}_i\right)^2}{N}}$$

Data Collection

The students who participated in the study came from IS majors at a large mid-western university and consisted of 200 students from a pool of 700 total majors. Since the goal of the research effort was to predict how well a student would do in the IS major (in terms of GPA), only those students with nine or more hours of coursework in the IS major were selected for study. In addition, several students who did not report their ACT scores were excluded from the study. The resulting data set included 128 students.

In an effort to determine if a common profile existed for the successful IS student, data related to student demographic characteristics was collected along with students' scores on a standardized personality test.

The personality assessment instrument used for this study was the Myers-Briggs Type Indicator (MBTI). This instrument was used because of its proven validity as an appropriate means of accessing personality and its prior successful use in academic and business settings in identifying personality types most prevalent in given professions [9,7,1].

The MBTI

The MBTI is an instrument designed to identify a test subject's preferences directing their use of perception and judgment. These preferences indicate not only what a person would be most attentive to in a given situation, but also the way in which that person would draw conclusions from what was observed.

The four categories of preferences are Extraversion or Introversion (EI), Sensing or Intuitive perception (SN), Thinking or Feeling judgment (TF), and Judgment or Perception (JP).

Extraverts are characterized as individuals who gather information or perceptions from the external world by observing outside individuals or events. Introverts' perceptions, conversely, are shaped largely through internal or inner thought processes and ideas.

The SN preference index indicates how an individual gathers information, either through use of the five senses, Sensing (S) or through intuition (N).

Thinking-Feeling (TF) refers to the manner in which an individual reaches conclusions regarding a given situation. A Thinking (T) individual would rely most heavily on logical processes to make a judgment, while a Feeling (F) individual would rely more on personal values or feelings.

The Judgment and Perception category (JP) speaks to an individual's manner of interacting with the outside world. A person with a J preference score would deal with the world from a judging perspective, regardless of whether those judgments are derived either through thinking or feeling (TF). They would seek closure and tend to make decisions as soon as possible and could, therefore, tend to be viewed as decisive. An individual with a P preference score would interact with the world from a basis of using some process of perception and would be characterized as open, adaptable, and in no hurry to reach conclusions [9].

Specifically, the nine inputs to the neural network model data set derived from the data collection process included the students' Age, Gender, Information Systems Hours Taken (HrsCIS), ACT score, Cumulative GPA (cumGPA) and the four Myers-Briggs scores of extraversion/introversion (EI), sensing/intuition (SN), thinking/feeling (TF), and judging/perceiving (JP). The output to the network was the student's GPA for the IS major.

Methodology

A ten-fold cross validation was conducted by splitting the data set into 10 training and 10 testing sets. For example, the first training set included the first 115 observations and the corresponding test set included the remaining 13 observations. The next test set was constructed by putting the 13 testing observations at the beginning of the training set and removing the last 13 observations for the second training and test sets. This was repeated for 10 data sets, with the last set containing 117 observations for training and the final 11 observations for testing. In this way, every observation was utilized once as a testing observation. By splitting the data and running 10 different NNs on these data sets, we can be more confident with our results. All processing of data was conducted using one-hidden layer and the standard sigmoid function.

Model Validation

Each data set was also analyzed using regression in order to establish baseline data for comparison analysis. Although the purpose of the current study was to identify relevant variables in the model, identification of these variables becomes much less meaningful if regression analysis techniques consistently outperform the NN. Table 1 includes the root mean squared errors (RMSEs) for the in-sample and out-of-sample data sets, as well as the averages and standard deviations for the 10 data sets. Since the in-sample or training data has already been seen by regression and the NN, the most relevant comparisons are for the out-of-sample data.

As can be seen, the NNs found solutions that produced better out-of-sample estimates for all 10 data sets. Once it was determined that the NN models were relevant in prediction, the focus of the research effort shifted to the identification of the relevant inputs to the model. Fortunately, this identification occurs automatically during training through the use of the modified objective function (Equation 1) and the additional genetic algorithm operator (Mutation2, as shown in Figure 1).

Table 1 – Neural Network and Regression RMSE Comparison

Data	NN In-Sample	Regression In-Sample	NN Out-of-Sample	Regression Out-of-Sample
1	0.3726	0.3761	0.2597	0.3611
2	0.3683	0.3767	0.2957	0.3454
3	0.3714	0.3827	0.2637	0.2812
4	0.3731	0.3874	0.2423	0.2530
5	0.3785	0.3849	0.2089	0.2450
6	0.3546	0.3601	0.4138	0.4902
7	0.3584	0.3609	0.3916	0.5009
8	0.3469	0.3598	0.4721	0.4764
9	0.3578	0.3668	0.3970	0.4402
10	0.3445	0.3544	0.5098	0.5424
Avg.	0.3626	0.3710	0.3455	0.3936
Stdev.	0.0118	0.0120	0.1045	0.1105

Results

Figure 2 illustrates the neural network (NN) results for the first data set and is typical for all the NNs in this study of IS student success. Only five input variables were found to have connections to the hidden nodes indicating that they were contributing to prediction of student success. All of the weights were zeroed out for the MBTI variables of TF and JP, as well as the variables of hours in the IS major (HrsCIS), and ACT indicating that these variables made no contribution to the prediction of success. What is interesting is that all 10 NNs zeroed out these same inputs and kept as relevant Age, Gender, CumGPA, and the MBTI variables of EI and SN, all of which had at least one non-zero weight in every NN model.

Figure 2 - Neural Network Example 1

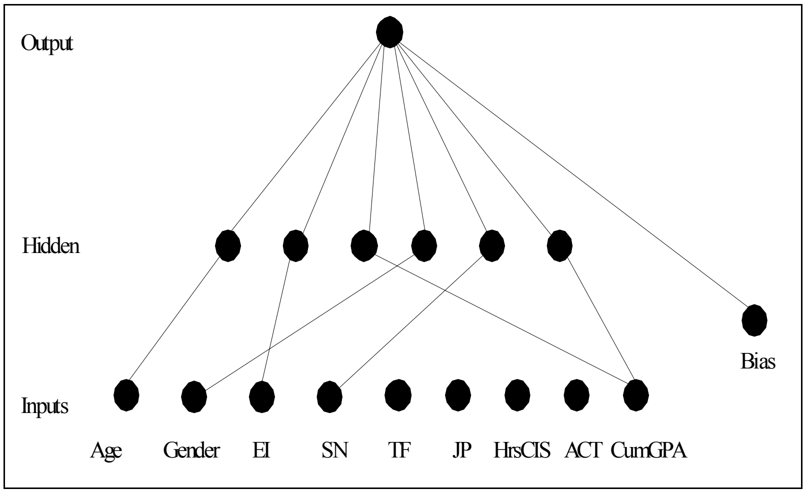


Table 2 lists the input variables as well as the number of times each variable had non-zero weights out of the 10 different networks. For example, the Age variable had at least 1 non-zero weight for all 10 networks and the ACT variable had no weights that were non-zero.

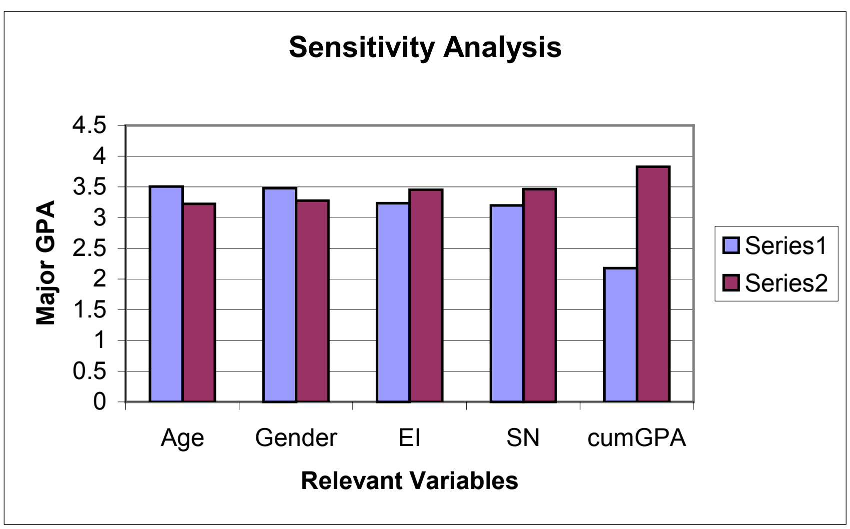
Table 2 – NN Relevant Variables

Variables	Found Relevant (Out of 10 networks)
Age	10
Gender	10
EI	10
SN	10
TF	0
JP	0
HrsCIS	0
ACT	0
CumGPA	10

Once the relevant variables were identified, from the initial nine variables that were included in the model, the specific effect of each of the relevant variables was examined utilizing sensitivity analysis. To do this, 5 artificial data sets were created, one for each of the identified relevant variables, to be processed by all 10 networks. For example, in analyzing the Age variable, a data set was created that included two observations. The first observation included the minimum value found in the data for Age, and the average input values for the other 8 variables (Gender, EI, SN, TF, JP, hrsCIS, ACT, CumGPA). The second observation included the maximum value found in the data for Age and, again, the average input values for the other 8 variables. By processing these two observations through the 10 networks, the effect (and the direction of the effect) Age had on the major GPA was determined. This processing/analysis was repeated for all 5 relevant variables.

Figure 3 illustrates the results of the sensitivity analysis. Series 1 were the observations with the minimum values for the specific relevant variable and series 2 were the observations with the maximum values for the specific relevant variable.

Figure 3 – Sensitivity Analysis of Relevant Variables



By examining Figure 3, we can see that for this particular data, five general themes emerged:

1. As students' age increases, the major GPA declines
2. Females' major GPA scores are higher than males
3. As the EI score increases, the major GPA increases
4. As the SN score increases, the major GPA increases
5. As the cumulative GPA (cumGPA) increases, the major GPA increases

It should be noted that results for this paper are specific to the data that was collected for this study. Without further research in this area, the results should not be considered generalizable to all universities. The main purpose for this analysis is to show that through the use of this type of NN (utilizing a genetic algorithm), variables can be analyzed and useful information extracted, where as with other NN models the results may not be as meaningful.

Conclusions and Recommendations

This study was designed primarily to illustrate/test the use of a neural network, employing a genetic algorithm, as an alternative to traditional methods of analysis for studying predictors of IS student success. The findings of this study, rather than being conclusive in their nature, are designed to provide a starting point from which further research can be conducted. Additional study of the relationships identified here, utilizing both NNs and traditional analyses, is recommended in order to determine whether cause and effect relationships might exist between the relevant variables.

For some of the variables, such as cumulative GPA, relationships appear obvious - it is apparent that as a student's cumGPA increases, their major GPA is likely to be higher. But for a number of the variables, additional research would appear appropriate. For example:

(1) Given the negative relationship between age and GPA, further investigation is needed to determine the specific variables that may cause a student's academic performance to decline as their age increases.

(2) It not readily apparent why being female might be associated with higher GPA scores.

(3) Having higher scores for EI (being more extraverted) or SN (relying more on senses rather than intuition for information gathering) on the Myers-Briggs indicators appears to indicate that a student may achieve a higher level of academic success.

(4) The ACT has been used for years as an indicator for how well students will do in college, but this model shows that it apparently does not contribute to predicting the major GPA for these students.

By using this type of NN for identifying relevant variables in a model, a researcher has the opportunity of not only predicting well, but also taking the information about the problem and applying it to an investigation of why a given effect occurs. The opportunity now exists to further explore the nature of these identified relationships.

References

[1] Bishop-Clark, C., & Wheeler, D., "The Myers-Briggs Personality Type And Its Relationship To Computer Programming", Journal of Research on Computing in Education, 26:3, 1994, 358-370

[2] Davis, L. (ed.), Handbook of Genetic Algorithms, New York, NY, Van Nostrand Reinhold, 1991

[3] Dorsey, R. E.; Johnson, J. D.; and Mayer, W. J., "A Genetic Algorithm For Training Feedforward Neural Networks", in Advances in Artificial Intelligence in Economics, Finance and Management, Greenwich, CT, JAI Press Inc., 1994, 93-111

[4] Dorsey, R. E., and Mayer, W. J., "Genetic Algorithms For Estimation Problems With Multiple Optima, Non-Differentiability, And Other Irregular Features", Journal of Business and Economic Statistics, 13:1, 1995, 53-66

[5] Funahashi, K. I., "On The Approximate Realization Of Continuous Mappings By Neural Networks", Neural Networks, 2:3, 1989, 183-192

[6] Hornik, K., Stinchcombe, M., and White, H., "Multilayer Feed-Forward Networks Are Universal Approximators", Neural Networks, 2:5, 1989, 359-366

[7] Jacobson, C., "Cognitive Styles Of Creativity: Relations Of Scores On The Kirton Adaption-Innovation Inventory And The Myers-Briggs Type Indicator Among Managers In USA", Psychological Reports, 72:3, 1993, 1131-1138

[8] Michalewicz, Z., Genetic Algorithms + Data Structures = Evolution Programs, Berlin Springer, 1992

[9] Myers, I., & McCaulley, M., "A Guide To The Development And Use Of The Myers-Briggs Type Indicator", Palo Alto, CA, Consulting Psychologists Press, 1985

[10] Schaffer, J. D., "Combinations Of Genetic Algorithms With Neural Networks Or Fuzzy Systems", in Computational Intelligence: Imitating Life, IEEE Press, 1994, 371-382

[11] Sexton, R. S., Dorsey, R. E., and Johnson, J. D., "Toward A Global Optimum For Neural Networks: A Comparison Of The Genetic Algorithm And Backpropagation", Decision Support Systems, 22:2, 1998, 171-186

[12] Yao, X., "Evolving Artificial Neural Networks", Proceedings IEEE 87, 1999, 1423-1447

Figure 1 – Outline of the Genetic Algorithm

Initialization. Choose an initial population containing 20 solutions to be the current population. Each solution consists of a string of real valued weights that are randomly drawn from a uniform distribution and then plugged into the NN for evaluation.

Evaluation. Each member of the current population is evaluated by a fitness function (Equation 1) in order to assign each solution a probability for being redrawn in the next generation. The probability is calculated by dividing the distance of the current string's error value from the worst error value in the generation by the sum of all distances in the current generation.

Reproduction. A mating pool of 20 solutions is created by selecting solutions from the current population based on their assigned probability. This is done by selecting a random number in the range of 0 and the sum of all probabilities (or 1) and comparing it to the cumulative probability of the current string. When it is found that the random value is less than the current string's cumulative probability, the current string is drawn for the next generation. This is repeated until the entire new generation is drawn.

Crossover. The solutions in the mating pool are then randomly paired constructing 10 sets of parent solutions. A point is randomly selected for each pair in which the parent solutions will switch the weights that are above that point, generating 20 new solutions or the next generation.

Mutation. For each weight in a generation a random number is drawn, if the random value is less than .05, the weight will be replaced by a randomly drawn value in the entire weight space. By doing this, the entire weight space is globally searched enhancing the algorithm's ability to find global solutions or at least the global valley.

Mutation2. Each solution in this new generation now has a small probability that any of its weights may be replaced with a hard zero. This is done to identify those weights in the solution that are not needed for estimating the underlying function. Once reproduction, crossover, mutation, and mutation2 have occurred, the new generation can now be evaluated to determine the new probabilities for the next generation. This process continues until 70% of the maximum number of generations set by the user is reached.

Convergence Enhancement. Once 70% of the maximum set of generations has been completed, the best solution so far replaces all the strings in the current generation. The weights of these 20 identical solutions are then modified by adding a small random value to the current weight. These random values decrease to an arbitrarily small number as the number of generations increase to its set maximum amount.

Termination. The algorithm will terminate on a user specified number of generations.