Neural Network Analysis of Business Data
A Power Tool for Data Mining

Timothy G. Woodcock [a], Thomas P. Bradley [b], Shauna Bewicke [c]
[a] Computer Information Systems, Texas A&M Central Texas, Killeen, Texas
woodcock@tarleton.edu
[b] College of Business Administration, Tarleton State University, Stephenville, Texas
tbradley@tarleton.edu
[c] Information Technology Department, Alfred State College, Alfred, New York
sbewicke@hotmail.com

Abstract

In order to prosper, businesses need to be able to get the most information out of the data they already have without increasing costs. Many non-technical professionals do not have the tools they need to make fast and informed decisions. Large Enterprise Resource Planning (ERP) systems are a financial burden due to both hardware costs and software and licensing costs. This paper shows that free or inexpensive neural network programs can be effective at making predictions with limited knowledge of how a neural network functions. Online occupational data was used to compare neural network predictions to the predictions made by the U. S. Bureau of Labor Statistics. Although the predictions are generalizations of job growth, with only five variables, these general predictions were calculated within six hours at an accuracy of 94.96%. While this result may not be optimal, it does show that with no expense or prior expertise information can easily be gleaned from current data. Having the benefit of making rapid informed decisions can give a company the boost it needs to be competitive.

Key Words: Data mining; Neural Networks; Business Predictions; Labor Statistics

Introduction

This review covers the benefits of using neural networks in business data analysis predictions, and decision making specifically using data widely available online. Organizations in a variety of industries have used the same methods for predictions for many years. For example, the Bureau of Labor Statistics has used the same factoring analysis for over 30 years to predict growth for various occupations (U.S. Department of Labor, Bureau of Labor Statistics, 1997). The growth of technology in business has opened up new opportunities for organizations to develop customized prediction and decision making tools. The goal of this study was to determine if useful predictions could be made given data readily available online, no cost software; processed by a user with limited knowledge of neural network software.

The software available to create and run a neural network varies from no cost, downloadable programs to applications that cost thousands of dollars; which may be a module of an existing Enterprise System. While no cost software has the benefit of lowering overall project costs, the algorithms used in this software may not be as efficient as the algorithms in purchased

software. No cost software may have other limitations for example in the free version, analysis may be restricted. For example, NuClass is a no cost neural network program that has a limitation of ten nodes in the hidden layer. Even with these limitations we intend to show that even less efficient algorithms may prove useful to many organizations. For example in our analysis the inputs for this occupational growth prediction study were restricted to five to attain the optimum topology.

Background

Neural Networks are analysis tools that fall into the general class of data mining tools and are most commonly used in pattern recognition problems. Data mining is the process of extracting previously unknown information out of a set of data (Witten & Frank, 2005). Data mining includes many analysis tools including, but not limited to, linear regression analysis, decision trees, neural networks, and data clustering. These tools are quite useful for data classification (Pant & Srinivasan, 2005) (Pendharkar, 2006).

Neural networks have been proven to be effective tools to classify data from small to large data pools (Wasserman, 1989). Most of the studies using neural networks in the last ten years have focused on one type of neural network, the Multilayer Perception (MLP) model. This is arguably the most common type of neural network and is well adapted for classifying data. A recent example is the study by (Becerra, Galvao, & Abou-Seda, 2005) where they compared wavelet networks to neural networks as well as linear models. They concluded that neural networks were a valid alternative to linear techniques.

Use of data mining techniques to examine business data is not unique. For example, (Becerra, Galvao, & Abou-Seda, 2005) used data mining techniques to identify businesses in distress. Becerra used five common business financial ratios from 21 companies that failed and 21 companies that are enduring firms, to build and test several classification models. He concluded that non-linear modeling techniques, like neural networks, outperformed linear techniques such as linear discriminate analysis.

Likewise (Koh, 2004) used data mining techniques to predict auditing assessment of a business's health. Koh looked at five variables: 1) Working capital to total assets, 2) Retained earnings to total assets, 3) Earnings before interest and tax to total assets, 4) Market value of equity to book value of total debt, and 5) Sales to total assets. Using data from 33 failed and 33 enduring companies, Koh also compared several modeling techniques including neural networks and linear analysis. He concluded that the neural networks performed well in predicting a company's going concern status.

These studies used a variety of hardware systems and software to test neural networks and their functionality. Of the studies that stated the systems and software used, there was a wide variety of software. The software ranged from self written algorithms and software to neural network tools that were available through enterprise systems available to the researchers. The research that focused on new methodologies relied heavily on self written software, or open source software, such as Weka (Weka3 - Data mining with open source machine learning) that could be modified to test these theories. The research that focused on what a neural network could predict relied more heavily on purchased software that was readily available than on the low or no cost software available online.

Our claim in this paper is not that neural networks are new, nor that using neural networks to analyze business data is new. Rather our claim is that neural networks have matured to the point where laymen can use them successfully with little knowledge of how the neural

network works. While we used labor statistics in this paper, this was just a vehicle to show how to use neural networks on business data, and we further claim that any non-linear business data can be used with similar results.

Data

The data selected was a list of 744 occupations taken from the U.S. Department of Labor, Bureau of Labor Statistics (1997). The five predictor variables selected were: number currently employed; median annual earnings; educational attainment; most significant source of postsecondary education or training; and unemployment rate for that specific occupation.

Since there is interest in forecasting future employment trends and since the data is readily available (Franklin, 2007) (Chentrens, n.d.), we chose U.S. Department of Labor, Bureau of Labor Statistics data to test downloadable neural network software to predict employment trends. The U.S. Bureau of Labor Statistics has been using a regression methodology for thirty years to make occupation growth predictions. The data used in these predictions is available online for anyone to review and use. These data are available in a few different formats, pdf, Excel, and text files. There are also .dat files available for some of the larger data files. Occupational growth was predicted using data from this source and downloadable no cost neural network software in order to show that managers and business researchers can use widely available data to make fast and accurate predictions.

Neural Network Model

The following section is a simple introduction to neural networks provided for those unfamiliar with neural networks. This information will help managers and researchers understand the terms and workings of neural network software. This description is not exhaustive and interested readers should direct their attention to the references or one of the many books on the subject (Chawla, 2005) (Wasserman, 1989).

A neural network is set up in a pattern that loosely resembles the neural network of a human brain. In the human brain neurons receive inputs from as many as thousands of synapses and become active when the input reaches a set threshold. The neuron output may then become input to another neuron or send its output to the nervous system. The most common neural network is the multi-layer perception model or more commonly referred to as simply a neural network (Wasserman, 1989). The neural network consists of several layers of nodes connected by weights. The weights are values calculated during the training process or preset by the researcher. The nodes are simply data variables, where each node holds one value. The user supplies some node values and others are computed by the network. The neural network nodes are grouped into three layers: input, hidden and output. The first layer, the input layer, is where the user will input the data. The input layer will contain one node for each variable value used for the classification. The second layer is called the hidden layer because the user normally does not see or use this layer. The number of nodes necessary for ideal performance in this layer is not known at the start of the project. In general the number of nodes in the hidden layer should be between 1.5 and 2.0 times the number of nodes in the input layer. The exact number is often found by trial and error. The last layer is the output layer where the model's results can be observed. Typically the output layer contains one node for each category being classified, but sometimes only one output node is used. When multiple nodes are used for output one node is

expected to have the value 1 and the rest of the nodes have the value of zero to indicate the classification.

A neural network can have more than three layers and if so the extra layers are hidden layers. However, seldom does the performance of extra hidden layers exceed the performance of neural networks with one hidden layer especially when considering the additional complexity required. Thus the three layer architecture is typical. Examples of the nodes of a neural network are shown in Figure 1.

To recognize and properly classify data, neural networks need training. The most common method of training is back-propagation. Back-propagation is a technique where previously classified (known) data is applied to the input nodes and the value of the output nodes are recorded for comparison to the expected output. These matched data establish the weights of the hidden nodes. In other words the data and known solutions are given to the network. The network uses the inputs and outputs to determine how to sort future inputs.

Training starts with all the weights in the neural network assigned random values between 0 and 1. Then the known sample input is fed into the network, as described above, and the output node values are computed. The network output node values are then compared to the known classifications. The comparison results in an error signal that is used to adjust the networks weights. This process is repeated with the sample input data until the network reaches stability.

It should be noted that every neural network, including two created using the same data, will not be exactly the same. The models can differ due to the random starting weights, the order in which the training data is presented, and the randomness built into the training process. However, two networks trained with the same data should perform exactly the same.

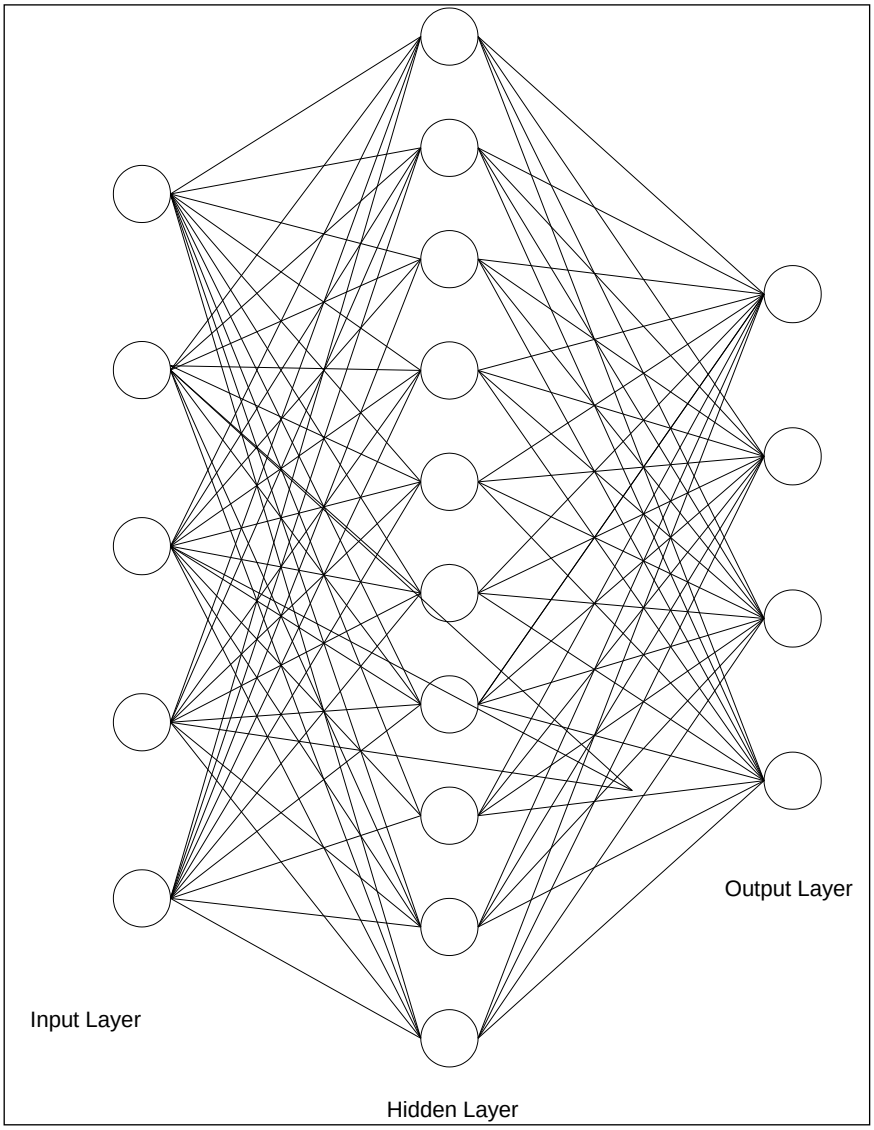

Figure 1 - Neural network topology.

Once the neural network is trained, it must be validated. Validation is the process of verifying that the model is in fact a good model of the data. Most researchers take a random portion of their original input data and remove it from the training set for validation. Once the network is trained this data is applied to the network to measure how well the network works on original, but previously unseen data. If the validation step does not show that the model adequately classifies this previously unseen data, the model can be modified by changing the number of nodes in the hidden layer and repeating the training process. This validation step is required in establishing the best model fit.

One issue that the validation step can expose is neural network over fitting (Belhadjali, Whaley, & Abbasi, Training a neural network for knowledge, 2006). When trained correctly, neural networks can generalize well. That is they can properly classify never before seen data. However, if over trained, the network remembers the exact training set and fails to perform well on new data. To avoid this condition a sufficiently large training data set is required along with a reasonable training cut off criteria.

The training cut off criteria is often dynamically calculated by the neural network tool. This is done by looking at the rate of change of the error rate and the overall error rate. Any stable error rate between 0.01 and 0.001 is considered good.

A properly trained and validated neural network is considered a valid alternative to linear regression especially when the original data violate the assumptions that linear regression rely on. Specifically when the data is non-linear, regression fails to perform satisfactorily. Neural networks have exceptional performance with non-linear data because neural networks are themselves non-linear.

Method

The neural network tool used in this study was a free tool called NuClass that was down loaded from the internet (Image Processing and Neural Network Lab at the University of Texas at Arlington). This tool is a complete system that can build train and apply several different kinds of neural networks. Since we are only interested in multi-layer perception models, we will focus our discussion in that area.

Once we downloaded the NuClass tool and ran setup, we prepared our data for analysis. The tool requires the data to be in a text file that is comma or tab-delimited. For ease of use, we put our data in a spreadsheet and then saved it in a comma-delimited file. The data needs to be organized with the inputs (in the same order) followed by the expected output. The number of inputs is determined by the data and by the restrictions of the NuClass tool.

As an example of the tool limiting our data choices, the downloadable NuClass tool only allows ten nodes in the hidden layer (upgrades are available). Since we wanted to start with a model that had twice as many nodes in the hidden layer as in the input layer, we were restricted to five input nodes. Buying the upgrade or using a different input to hidden layer ratio would have allowed us to use more input data.

We used five data inputs for this study, based on current occupational figures: number currently employed; median annual earnings; educational attainment; most significant source of postsecondary education or training; and unemployment rate for that specific occupation. The neural network topology chosen is consistent with previous studies. We started with the standard topology of five input nodes; one hidden layer with ten nodes; and one output classification node.

Results were classified into four job growth categories based on the change in the number of jobs per occupation: VH = Very high, H = High, L = Low, and VL = Very low. These are the same classification categories that were used by the Bureau of Labor Statistics in their predictions of 10-year job growth (U.S. Department of Labor, Bureau of Labor Statistice, 2007).

Recall that the neural network needs to be trained and then validated to ensure performance. To prepare the data for training and validation we randomly divided the data into two groups, a training set of 377 values and a validation set of 377 values. Once we separated the data, the neural network training started.

Training starts by choosing the training menu in the NuClass tool as shown in Figure 2.

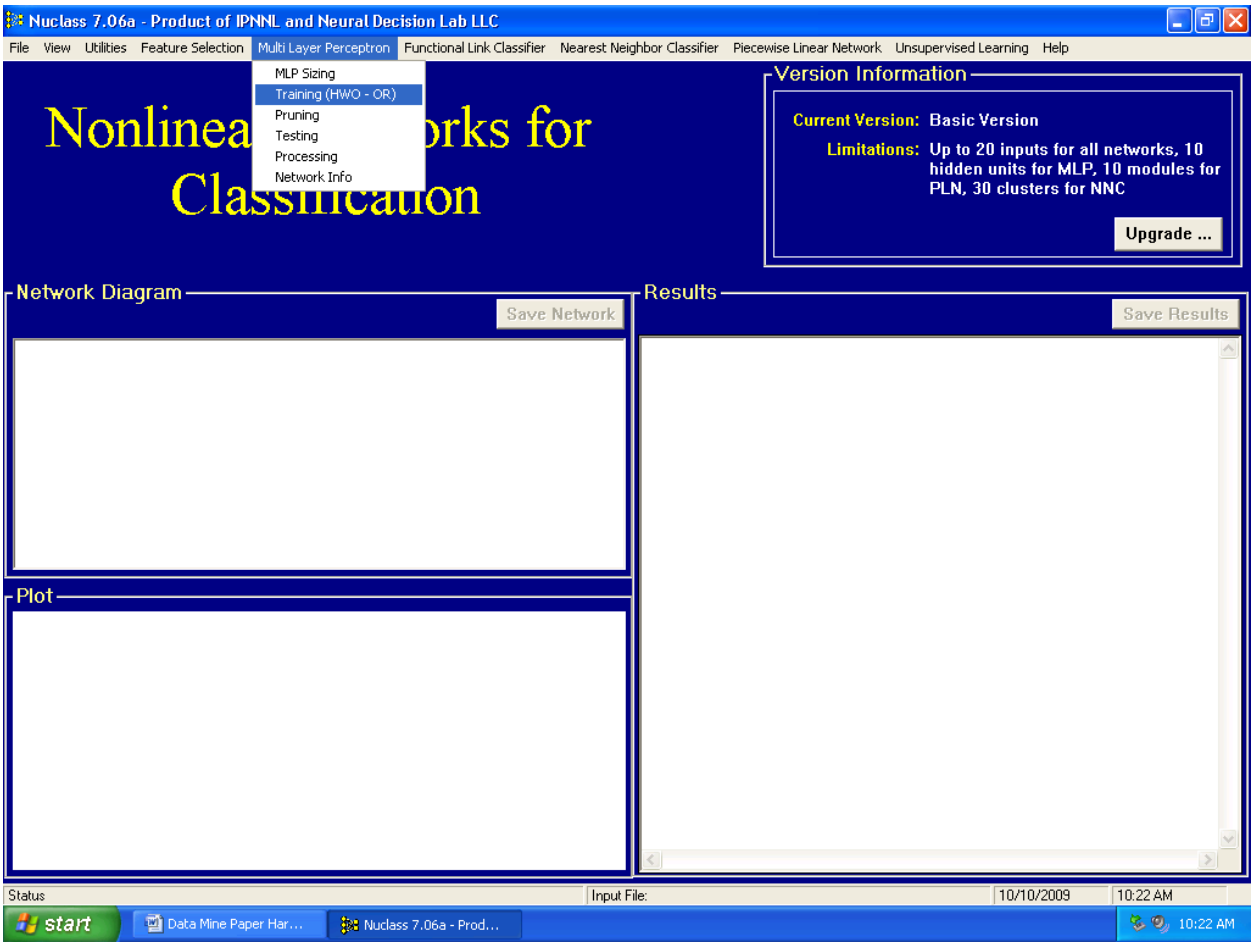

Figure 2 Starting Training

Once the training data is loaded and the training process has been run the screen looks like the screen shot in Figure 3. At this point the model should be saved.

Network training was performed until an error rate of 0.002919 had been achieved. This is lower than the desired training error rate of 0.001 or lower. The neural network tool trained the network using the standard back propagation algorithm. This took about an hour to complete. Once it was completed, the validation data set was used to validate the model.

To validate the model the testing menu was chosen. Once completed the tool should look like the screen shot in Figure 4.

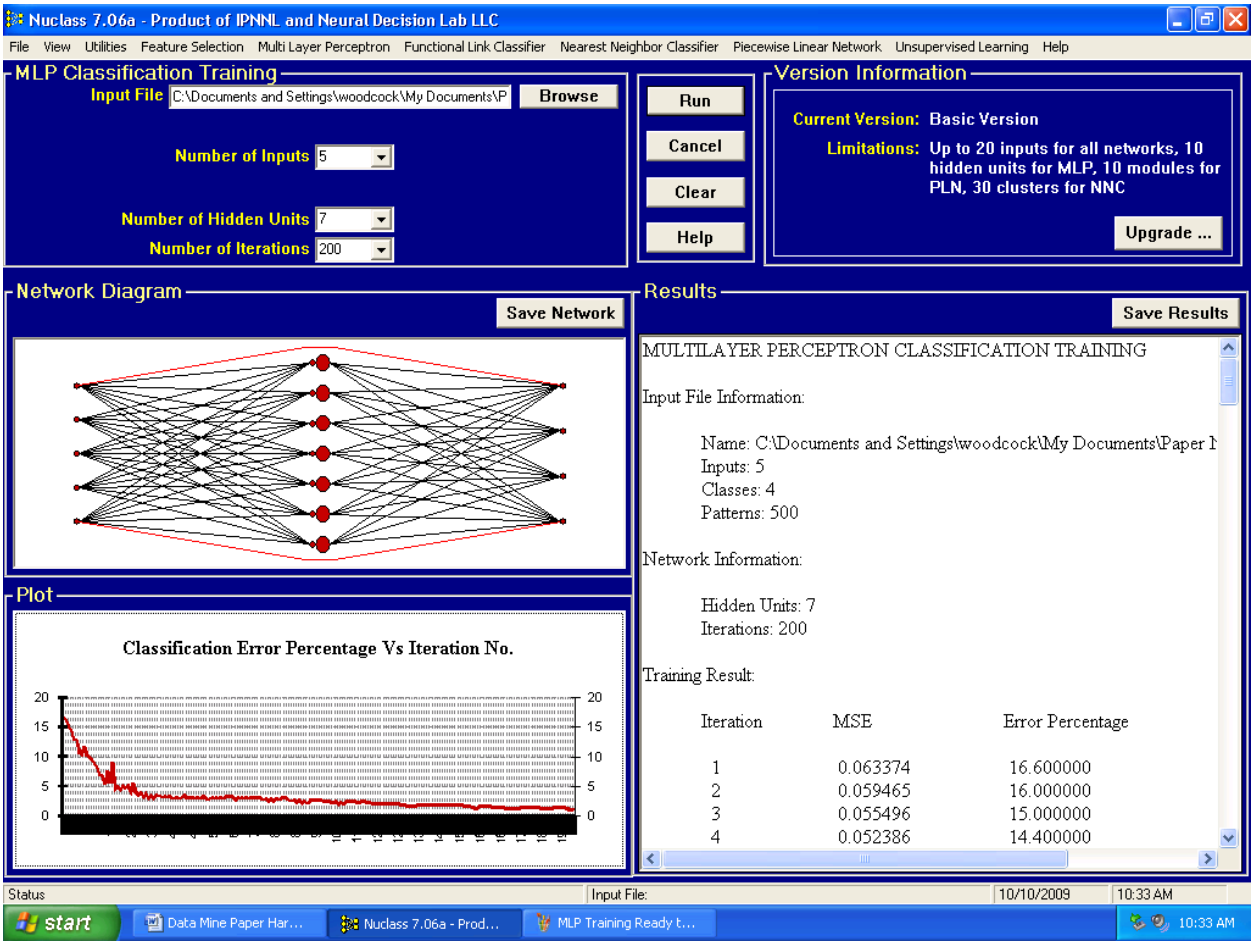

Figure 3 Training Complete

The measure of the performance of the model is calculated by counting how many times it misclassifies the validation set data. Fewer misclassifications mean that one model is better than another is. Our first model had an overall misclassification rate of 27.85%.

The disappointing outcome of the original model was not unexpected. Neural networks can be affected by many factors that are difficult to impossible to see ahead of time. Factors like over training, memorizing one data set, are usually handled in the NuClass tool. More likely, disappointing results are caused by the input to hidden layer ratio. Choosing a different ratio often results in a better model.

Since we had marginal results, we chose to reduce the number of nodes in the hidden layer and repeated the analysis. This is a common practice in neural network analysis. It is common to adjust the model and perform multiple runs to find the best model fit. The second model performed up to our expectations.

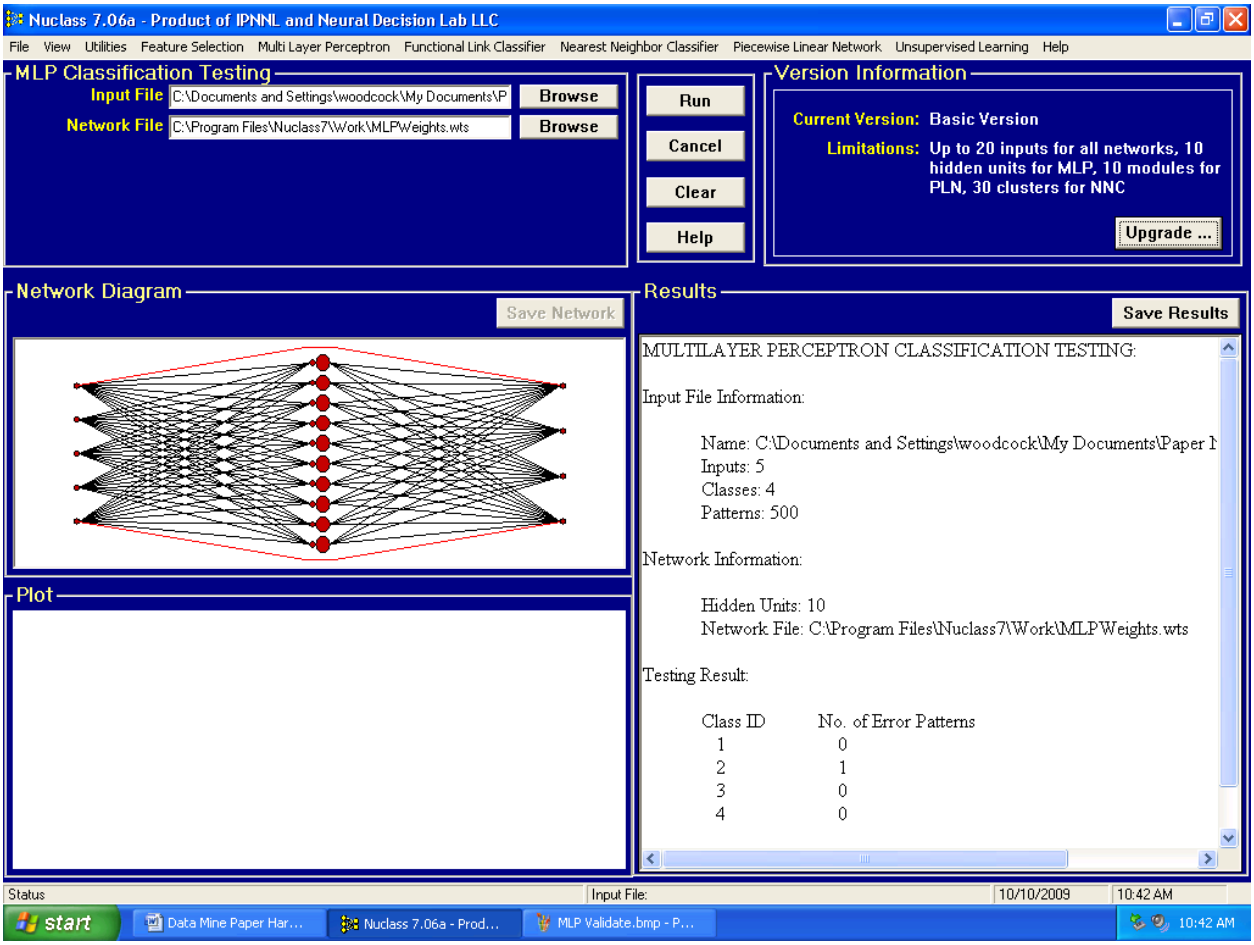

Figure 4 Validation finished

Since we had marginal results, we chose to reduce the number of nodes in the hidden layer and repeated the analysis. This is a common practice in neural network analysis. It is common to adjust the model and perform multiple runs to find the best model fit. The second model performed up to our expectations.

Results

Below are the results of the validation set in the trained neural network. The first table shows the results of testing the original network with the validation data. The classification and Class ID both show the four groups that the input can belong to. The number of error patterns is the number of inputs that the neural network should have had in that output category, but were misclassified to another category. The % of Total Errors column is the percent of errors for that classification that were classified incorrectly by the model.

NuClass Results with 10 nodes in the hidden layer

Testing Result for :	Inputs: 5	Hidden Units: 10	Patterns: 377
Classification	**Class ID**	**No. of Error Patterns**	**% of Total Errors**
VL (-155,000 to +18,000)	1	29	7.69%
L (19,000 to 77,000)	2	57	15.12%
H (78,000 to 201,000)	3	19	5.04%
VH (202,000 to 736,000)	4	0	0%
Totals		105	27.85%

Table 1: Misclassifications of neural network in the original NuClass 5-10-4 model

NuClass Results with 7 nodes in the hidden layer

Testing Result for :	Inputs: 5	Hidden Units: 7	Patterns: 377
Classification	**Class ID**	**No. of Error Patterns**	**% of Total Errors**
VL (-155,000 to +18,000)	1	6	1.59%
L (19,000 to 77,000)	2	11	2.92%
H (78,000 to 201,000)	3	2	0.53%
VH (202,000 to 736,000)	4	0	0%
Totals		19	5.04%

Table 2: Misclassifications of neural network in the second NuClass 5-7-4 model

Conclusion

The results of the validation set for the neural network with the original ten nodes in the hidden layer show a classification accuracy of 72.15% (see table 1). A majority of the classification errors occurred in category 2 which represents Low job growth between 19,000 and 77,000 more jobs predicted in 2014 (U.S. Department of Labor, Bureau of Labor Statistice, 2007)

Then the second neural network with seven nodes in the hidden layer (5-7-4 network) was trained and tested. The results were greatly improved (see table 2). The results of the validation set show a classification accuracy of 94.96% a gain of 22% accuracy over the original 5-10-4 network topology.

This neural network tool quickly created a model that can be used on future data to classify predicted job growth. This tool is easy to use, robust, and accurate. This tool is available at no cost and can be used by people with very limited knowledge of the inner workings of neural networks.

Business managers and researchers do not need to make a large investment in order to see if the neural network methodology can improve their processes and decisions. The United States

Government and related entities publish large amounts of data on their websites. Organizations can combine published data with their own data to make customized predictions using low or no cost neural network applications. These predictions can support decision makers in coming to informed decisions quickly.

Neural network tools exist that make it easy for building a classification system for business data. These tools are robust and do a good job on classifying data, especially when the underlying data is non-linear.

In this paper, we showed that freely available, easily down loaded neural network tools can be used to do business data analysis and classification. We showed this using some standard information available on a government website and we were able to classify it into occupation growth predictions. We claim that these tools have matured to the point where anyone can use them to classify raw data.

References

Becerra, V. M., Galvao, R. K., & Abou-Seda, M. (2005). Neural and Wavelet Network Models for Financial Distress Classification. (G. Webb, Ed.) *Data Mining and Knowledge Discovery , 11*, 35-55.

Belhadjali, M., & Waley, G. L. (2004). A data mining approach to neural network training. *Information Management & Computer Security* , 117.

Belhadjali, M., Whaley, G. L., & Abbasi, S. M. (2006). Training a neural network for knowledge. *Competition Forum*, (pp. 131-135). Indiana.

Chawla, N. (2005). Discovering Knowledge in Data: An Introduction to Data Mining. *Briefings in Bioinformatics* , 411-412.

Chentrens, C. (n.d.). *Factors Affecting Occupational Demand Growth, 2006-2016*. Retrieved from ftp://ftp.bls.gov/pub/special.requests/ep/factor.analysis/fa_description.pdf

Cushing, J. B., Nadkarni, N., Finch, M., Fiala, A., Murphy-Hill, E., & Delcambre, L. (2007). Component-based end-user database design for ecologists. *Journal of Intelligent Information Systems* , 7.

Franklin, J. (2007). An overview of BLS projections to 2016. *Monthly Labor Review* , 3-12.

Hardgrave, B. C., Wilson, R. L., & Walstrom, K. A. (1994). Predicting Graduate student success: A comparison of neural networks and traditional techniques. *Computers and Operations Research* , 249-263.

Koh, H. (2004). Going concern prediction using data mining techniques. *Managerial Auditing Journal* , 462.

Lam, M. (2004). Neural network techniques for financial performance prediction: integrating fundamental and technical analysis. *Decision Support Systems* , 567-581.

Li, R., & Wang, Z. (2004). Mining classification rules using rough sets and neural networks. *European Journal of Operations Research* , 439-448.

Liu, Y., & Schumann, M. (2005). Data mining feature selection for credit scoring models. *The Journal of the Operational Research Society* , 1099.

Pant, P., & Srinivasan, P. (2005). Learning to crawl; Comparing classification schemes. *ACM Transactions on Information Systems* , 430.

Pendharkar, P. (2006). A data mining-constraint satisfaction optimization problem for cost effective classification. *Computers & Operations Research* , 3124-3135.

Sabbagh, A., & Darlu, P. (2006). Data-mining methods as useful tools for predicting individual drug response: Application to CYP2D6 Data. *Human Heredity* , 119-134.

Schikora, P. F., & Godfrey, M. R. (2003). Efficacy of end-userneural network and data mining software for predicting complex system performance. *International Journal of Production Economics* , 231-253.

Sexton, R. S., & Sikander, N. A. (2001). Data mining using a genetic algorithm-trained neural network. *International Journal of Intelligent Systems in Accounting* , 201.

Sherrod, P. (2008). *DTREG: Predictive Modeling Software*. Retrieved April 1, 2008, from http://www.dtreg.com/DTREG.pdf

Stolzer, A. J., & Halford, C. (2007). Data mining methods applied to flight operations quality assurance data: A comparison to standard statistical methods. *Journal of Air Transportation* , 6-24.

Strano, M. (2004). A neural network applied to criminal psychological profiling: An Italian initiative. *International Journal of Offender Theraphy and Comparative Criminology* , 495.

Su, C., Hsu, H., & Tsai, C. (2002). Knowledge mining from trained neural networks. *The Journal of Computer Information Systems* , 61-71.

Suzuki, K., Horiba, I., & Sugie, N. (2001). A simple neural network algorithm with application to filter synthesis. *Neural Processing Letters , 13* (1), 43-53.

Taylor, W. A. (2004). Computer-mediated knowledge sharing and individual user differences: an exploratory study. *European Journal of Information Systems* , 52-64.

U.S. Department of Labor, Bureau of Labor Statistice. (2007). *Table IV–1. Occupational employment and job openings data, 2004-14, and worker characteristics, 2004*. Retrieved March 23, 2008, from Http://www.bls.gov/emp/optd/optdtabiv_1.pdf

U.S. Department of Labor, Bureau of Labor Statistics. (1997). *BLS Handbook of Methods*. Retrieved March 29, 2008, from http://www.bls.gov/opub/hom/pdf/homch13.pdf

Vandamme, J., Meskens, N., & Superby, J. (2007). Predicting Academic Performance by Data Mining Methods. *Eduation Economics* , 405-419.

Vojinovic, Z., Kecman, V., & Seidel, R. (2001). A data mining approach to financial time series modelling and forecasting. *International Journal of Intelligent Systems in Accounting* , 225.

Wasserman, P. (1989). *Neural computing: Theory and Practice.* New York: Van Nostrand Reinhold.

Weka3 - Data mining with open source machine learning. (n.d.). Retrieved March 28, 2008, from http://www.cs.waikato.ac.nz/ml/weka/

Witten, I., & Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques.* San Francisco: Morgan Kaufmann.

Ye, N., & Li, X. (2002). A scalable, incremental learning algorithm for classification problems. *Computers & Industrial Engineering* , 677-692.

Yu, W. (2007). Hybrid Soft Computing Approach for Mining of Complex Construction Databases. *Journal of Computing in Civil Engineering* , 343-352.

Zhang, Y., Edwards, J., & Harding, J. (2007). Personalized online sales using web usage data mining. *Computers in Industry* , 772-782.