14204

Predicting Car Production using a Neural Network

Technical Paper –Vetronics (In-house)
24-Apr-03
Mike Del Rose

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
24 APR 2003		

4. TITLE AND SUBTITLE
Predicting Car Production using a Neural Network

5a. CONTRACT NUMBER
5b. GRANT NUMBER
5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Michael Del Rose

5d. PROJECT NUMBER
5e. TASK NUMBER
5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army TARDEC,6501 East Eleven Mile Rd,Warren,Mi,48397-5000

8. PERFORMING ORGANIZATION REPORT NUMBER
#14204

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Neural networks are used for many reasons, like predicting stocks in the stock markets [1], recognizing objects [2, 3], brain theory [4], and countless other areas. In this project, a simple neural network is used to predict the yearly production of a car company based on past productions, market conditions, and other indicators. The data was retrieved from [5, 6, 7]. In this paper, the methodology is discussed in section 2. Section 3 describes the files and how to run the program. Section 4 presents the results, and section 5 concludes the paper. Section 6 lists the references.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**6**	

1. Introduction

Neural networks are used for many reasons, like predicting stocks in the stock markets [1], recognizing objects [2, 3], brain theory [4], and countless other areas. In this project, a simple neural network is used to predict the yearly production of a car company based on past productions, market conditions, and other indicators. The data was retrieved from [5, 6, 7].

In this paper, the methodology is discussed in section 2. Section 3 describes the files and how to run the program. Section 4 presents the results, and section 5 concludes the paper. Section 6 lists the references.

2. Methodology

The neural network used in this exercise is a 3 layer (two hidden layers) neural network with a unipolar sigmoid activation function. During the training of the network, the weights were changes using the back propagation method until a desired completion run was executed. In this example, 100,000 cycles (epochs) were used to train it. The initial weights were randomly selected from values between 1 and -1. Visual basic .NET was used to program the neural network [8]. The neural network algorithm followed the steps outlined in [9].

As stated above, a 3 layer neural network was used. The reason for using two hidden layers rather than one is due to a problem with training a two layer (one hidden layer) network with the XOR function. After consulting many sources [9, 10, 11] it was decided to change the single hidden layer to two hidden layers. The improvement was minimal, but the remainder of the debugging was done with the 3 layer network.

Since the XOR function is not linear, this offered an easy way to test the network for convergence (or close to convergence). Changing the alpha (momentum) and ro (learning step) values showed decreases in the sum-squared error where there were no changes before. This took the network out of small local minimums so that the network could converge close to the actual output. From there, modifications to the network were made to improve the speed by increasing and decreasing the number of nodes in each of the hidden layers.

For the prediction of the yearly production for a certain car manufacture, a second part was added to do the prediction. Functions to read in the xml files were added and many new classes and class containers were added to make the program easily changeable (hopefully). A few modifications were made to make global changes easy.

3. Description of files and running the code

The following list is a brief description of the programming files in the project:

- carClass.vb – class to hold the car information
- carContainerClass.vb – a collection of all the carClasses
- carNN.vb – the car production neural network predictor.
- econClass.vb – a class to hold all the economic information.
- econContainerClass.vb – a collection of all the econClass.
- form1.vb – the running (and exiting) window program
- fuelClass.vb – a class for the fuel information per company per year
- fuelContainerClass.vb – contains all the classes fuelClass
- NNClass.vb – a class to hold all the information specific to the neural network.
- totalFuelClass.vb – a class to hold all the fuel prices per year
- XOR3LayerMod.vb – the xor neural network.
- xorClass.vb – a class to hold all the xor information
- xorCollection.vb – contains all the xorClass classes

The following is a list of xml files. These files are read into the program.

- dowAve.xml –used in the econClass. It's the Dow Jones opens, closes, highs and lows for each year
- fuel.xml – used in carClass and fuelCLass. It's the mpg average per car company per year
- fuelPrice.xml – used in totalFuelClass. It's the average cost of fuel per year
- totalProduction.xml – used in carClass. It's the production data for each company per year
- xorFile.xml – used in the xor neural network. It's the xor function results.

To run the program, start up the project and under debug, click on start or hit F5. A small window will be displayed with a run button and an exit button. Hit the run button to run the neural network and the prediction. The output will be displayed on the output window (or console). Hit the exit button to quit.

If you desire to run the xor test, go into form1.vb and under button_click1(..) comment out the line "carNN.mainCarNN() ". Uncomment the line "xor3LayerMod.mainXOR3Layer()". Next, go into the NNClass.vb file and change the ERROR_VALUE to .1. This will give the testing phase the ability to run until the maximum number of cycles are completed or the sum-squared error is below ERROR_VALUE. Run the code following the steps above.

4. Results

These results reflect the effects caused by the addition and subtraction of nodes for a specific company's output. The Chrysler company was used to test the effects each network had towards the result. In the data there are 12 companies and 11 years of compiled information. The information gathered includes U.S. production; U.S. market share of production; average vehicle fuel for each company per year; average fuel cost per year; Dow Jones high, low, open and close for each year. The data was gathered through [5, 6, 7]. The years were from 1990 to 2001. Three types of node arrangements were used on the Chrysler company: 5 nodes in the first hidden layer and 3 nodes in the second hidden layer, 4 nodes in the first hidden layer and 2 nodes in the second hidden layer, and 7 nodes in the first hidden layer and 4 nodes in the second hidden layer. The network cycled 100,000 times through the training phase. Three runs were compiled for each test and the average percentage of differences are shown here.

For the network with 5 nodes in the first layer and 3 nodes in the second layer, the data used to train and test were U.S. production of all vehicle companies for year t, year t's average fuel cost, the average mpg for year t for all the companies, and the open and close of the Dow Jones for year t. Chrysler company accurately predicted within 6.9% of the total for year 2001. For the network with 4 nodes in the first hidden layer and 2 hidden nodes in the second layer, it was only 5.8% off. And the final network (7 nodes in the first hidden layer and 4 nodes in the second hidden layer) it predicted at 95.0% accuracy. Since this produced the best results, it was used to compare with the other company's. See table 1 for the results.

	Difference	% off	Actual Value
Chrysler	21963	5.0	438141
Ford	332519	33.6	989868
GM	383951	23.2	1656172
Toyota	41142	11.6	353381

Table 1 – Production prediction results of NN using all data

The network was changed to leave out the Dow Jones information and use 7 nodes in the first layer and 4 nodes in the second layer. Each company was tested three times. The results are in table 2.

	Difference	% off	Actual Value
Chrysler	4612	1.1	438141
Ford	468750	47.4	989868
GM	449335	27.1	1656172
Toyota	16806	4.8	353381

Table 2 – Production prediction results of NN leaving out Dow Jones data

As shown in table 1 and table 2, the Chrysler company was accurately predicted for all networks tested on. Toyota did great for the second network (leaving out the Dow Jones information) but only marginal for the first network. The rest did not fair so well. It is also worth noting that for Toyota and Chrysler, the accuracy increased when leaving out the Dow Jones information where the other companies in the survey had worse results. This leads me to believe that there are one of three things going on with this data or this neural network:

1. The data is meaningless
2. There is too much variability in the companies other than Chrysler and Toyota, and
3. The network does not work as good for higher producing companies then it does for lower producing companies.

5. Conclusion

As the results show, it is hard to predict production of vehicles based on common market indicators. The neural network was not as robust as I had planned it to be. It did very well in some cases, but poorly in most others. If a more robust predictor of U.S. production is needed, than much more time needs to be spent on what the causes for people buying cars are.

This project was challenging for me due to the fact that I am learning the programming language and learning neural networks. It was a very rewarding experience. It gave me a great appreciation of neural networks.

6. References

[1] R. Lawrence, *Using Neural Networks to Forecast Stock Market Prices*, University of Manitoba, 1997

[2] M. A. Sipe, D. Casasent, *Global Feature Space Neural Networks for Active Object Detection*, Carnegie Melon University

[3] L. Zhao, C. Thorpe, Stereo- *and Neural Network-Based Pedestrian Detection*, IEEE Transaction on Intelligent Transportation Systems, Volume 1, Number 3, September 2000.

[4] M. A. Arbib, *The Handbook of Brain Theory and Neural Networks*, MIT Press, 1998

[5] *Ward's Automotive Yearbook*, Ward's Communication, Detroit, 2002

[6] Web site: http://Finance.yahoo.com/

[7] *The World Almanac*, World Almanac Education Group, 2003

[8] E. Petroutsos, *Mastering Visual Basic .NET*, SYBEX Inc., 2002

[9] D. E. Rumelhart, J. L. McClelland, *Parallel Distributed Processing - Explorations in Microstructure of Cognition*, MIT Press, Volume 1, 1986

[10] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1998

[11] M. H. Hassoun, *Fundamentals of Artificial Neural Networks*, MIT Press, 1995