AINT303 - Neural Network Report

Alexander Lea

Problem Analysis

The problem

Predicting accurately GBP/USD Exchange rates based on three influencing factors: interest rates, exchange rates and the rate of inflation. Interest and inflation rates are deemed to be the most significant in influencing the exchange rate (Bergen, 2004), and historic exchange rate information is included for reference to long-term trends within the market. Due to the size of the study, the decision was taken only to include three input factors, and ignore other influencing factors such as political stability, and national debt.

How have others addressed it?

Predicting exchange rates is a common challenge undertaken by neural network developers, due to the potential financial benefits, if the prediction is correct. Commonly, a Multi-Layer Perceptron (MLP) is used for this purpose, due to its simplicity and "powerful non-linear mapping capabilities" (Bullinaria, 2004). Simon, 2002, used a MLP to predict the exchange rates, as did Huang et. al, 2008.

Alternatively, another implementation that has been used is the Recurrent Neural Network (RNN), which, in its simplest form is "an MLP with the previous set of hidden unit activations feeding back into the network along with the inputs" (Bullinaria, 2004). This has the benefit that it uses the outputs from the previous iteration to alter the weights for the input, therefore allowing a fine-tuning process to take place. This technique has been used by Tenti, 1996.

For the reasons mentioned above, this study will be conducted using a MLP neural network.

Work Completed

Network Details

When compiling the data set for this study, there was uncertainty regarding the importance of date (or month) data, regarding the prediction of exchange rates in the future. It was not certain whether including date information would allow seasonal insights, and therefore provide a valuable input variable into the network, or if the time of year had very little impact on the exchange rates. As a result, it was decided to implement two MLP networks and compare the results. Two networks were planned:

- Network 1 taking inputs:
    - month number
    - historical inflation rate
    - historical exchange rate
    - historical interest rate

    - current exchange rate
- Network 2, taking inputs:
    - historical inflation rate
    - historical exchange rate
    - historical interest rate
    - current exchange rate

The question still existed, however, about the format of the date to be included as an input. Month and year was deemed too specific, and had little value predicting values far in the future - a set of inputs that matched exactly, other than the date would be unlikely to produce a similar output in the system, due to the fact the dates were so different. An alternative would be to organise the data into months within years, so each month had a numeric value between 1 and 12. The date categorisation method that was chosen was to sequentially number the dates, starting with month 0 in January 1990. This decision was taken because it provided an overall trend to the data, where the categorisation alternative did not.

In addition to two network types (dates and no dates), three different networks will be implemented for each type, with a varied number of hidden nodes each time, in order to find the optimal solution. There will be a network trained with 10, 20, and 40 hidden nodes, for each network type. This aims to find the ideal number of hidden nodes, in order to produce the lowest error rate.

The learning rate for the MLP networks was decided as 0.004, momentum of 0.95 and boost of 6.

The nodes used for the network were as follows: INPUT nodes were used for the inputs into the network; BACKPROP nodes as the hidden layer; and a LINEAR node as the output. The LINEAR node was used, because it copes better with skewed data, and the normalised output of the system is focussed on values around 1. A standard output node may struggle to fully represent these values accurately.

Data Preprocessing

The following details the steps taken to create the data set for this study.

1. Access monthly data, starting from 1990.
2. Normalise values:
    - $Normalised\ ER = Exchange\ Rates / 2$
    - $Normalised\ Inflation = Inflation / 5$
    - $Normalised\ Interest = Interest / 10$
3. Numbered months, starting with January 1990 as month 0
    - $Normalised\ month\ \# = month\ \# / 240$

4. Worked out a 3 month moving average for each metric (inflation rate, exchange rate, interest rate) using the formula $(n-1 + n-2 + n-3)/3$, where n is the month to be calculated, as well as including exchange rate at n as the target value
5. Compiled single table of all normalised data between Jan-1990 and Dec-2009 as training and testing data
6. Compiled single table of all normalised data between Jan-2010 and Sept-2014 as unseen future values.
7. Randomised order of training/test set
   - 119 rows of data were assigned to the test set
   - 119 rows of data were assigned to the training set
8. Test set 1:
   - Selected the first 119 rows (columns: normalised date #, Prev Q. Interest, Prev Q. Inflation, Prev Q Exchange, Current Month Exchange) as test data. Saved as **TRDAT1.DAT**
   - Selected the next 118 rows as training data. Saved as **TEDAT1.DAT**
   - Selected normalised date #, Prev Q. Interest, Prev Q. Inflation, Prev Q Exchange, Current Month Exchange from the future set, and saved in file **FUTURE1.DAT**
9. Test set 2
   - Selected the first 119 rows (columns: Prev Q. Interest, Prev Q. Inflation, Prev Q Exchange, Current Month Exchange) as test data. Saved as **TRDAT2.DAT**
   - Selected the next 118 rows as training data. Saved as **TEDAT2.DAT**
   - Selected Prev Q. Interest, Prev Q. Inflation, Prev Q Exchange, Current Month Exchange from the future set, and saved in file **FUTURE2.DAT**

Problems Encountered

Initially, when creating the data set, and training an early prototype network, it came to light that the inputs had not been properly normalised, and thus there was a very high error rate. The data set was corrected, and the network re-trained.

When planning this study, the decision was taken to use figures from the previous three months, so if month x is being calculated, take the average values of months x-1, x-2 and x-3. However, this means that a value for month x cannot be calculated until month x-1 has completed, and once x-1 has been completed, it is already month x. In order to avoid this, the calculations should have included a latent month, to allow calculations for month x to take place in month x-1. Including a latent month (x-1), the previous months' figures would be calculated for months x-2, x-3 and x-4.

Results

There is a table of results, including the error rates at specified epochs in Appendix 1. A shortened version of this is included as figure 1, below. Screenshots of the network in action (after training has stopped) are included in Appendix 2.

Test	Stopped training at epoch	Test error @ stop
1.1	900	0.023963
1.2	2600	0.023804
1.3	5000	N/A
Avg 1.x (with month data)		**0.0238835**
2.1	1400	0.018899
2.2	2000	0.019137
2.3	1500	0.018621
Avg 2.x (no month data)		**0.01888566667**

Figure 1: Network results

Figure 1 shows the results from training and testing the networks. As demonstrated, the tests without month number as an input had a 20% lower average error rate (0.019) than when month number was an input (0.024). Aside from that, there were no general trends worthy of mention. Test 1.3 failed to reach an optimal solution by epoch 5000, and thus the decision was taken to stop training, and move on to another network implementation.

Test 2.3 had the lowest error rate of the conducted networks, with the test error when training was stopped of 0.0186, closely followed by test 2.1 with an error rate of 0.0189. This demonstrates that the network is a significantly more accurate predictor without the month number as an input.

Using the remaining data values (that were not used in testing and training), an evaluation set was built, consisting of 56 data points (56 months' worth of data). This set was then applied to network 2.3, in order to assess its predictive accuracy. The network was able to match the target value relatively accurate, as shown in figure 2, with normalised target value plotted in green and normalised output in red. Interestingly, in the majority of the evaluation cases, the actual output was below the target value, but the general trend was matched.

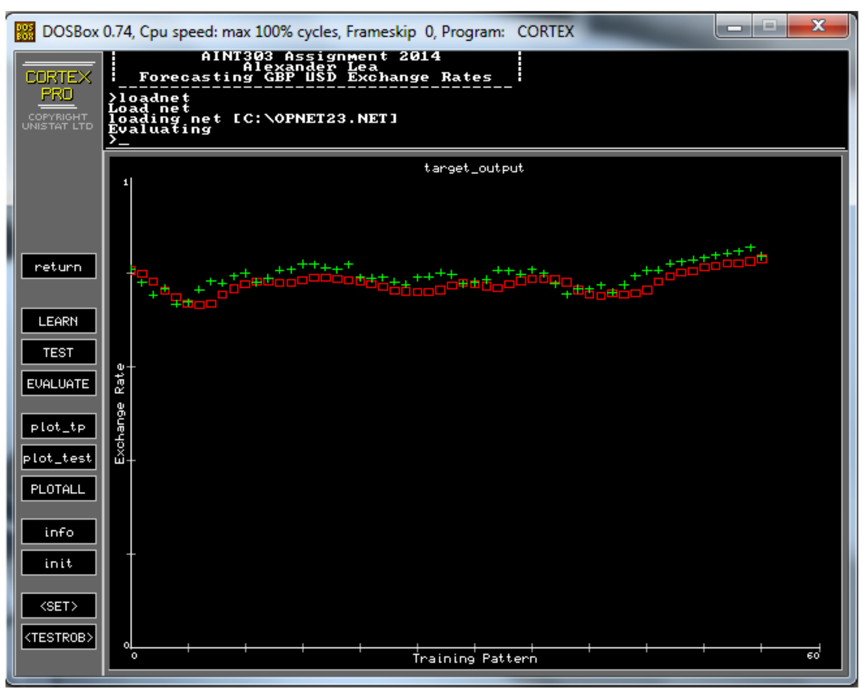


Figure 2: Network 2.3 Evaluation

Conclusions

In conclusion, to improve the quality of the results found by this study, two suggestions are offered. Firstly, rather than using the same period of time for all moving averages (previous quarter), it might be beneficial to use a varied time period for sampling. This would look to remove any idiosyncrasies from the data set (like quarterly trends), and instead offer a more accurate representation of the exchange rate market over a period of time.

Secondly, it would be worth considering the use of more input variables. Due to the complexity of exchange rates, and the large number of influencing factors, it would be impossible to include all factors in the network, but it is slightly naive to suggest that the exchange rate can be predicted solely by interest rates and inflation rates. Easily measurable factors could include economic growth expectations, trade balance, employment outlook (OANDA.com, 2014), current-accout deficit, and perhaps more complex measures such as political stability or market speculation (Tutor2u.net, 2014).

Overall, though, this study was able to relatively accurately predict the exchange rate between the GBP and USD, and predicts that the average exchange rate for December 2014 be 1.578 GBP/USD. At the time of writing this report, the average exchange rate for December (thus far) is 1.578 (Xe.com, 2014).

References

Bergen, J. (2004). *6 Factors That Influence Exchange Rates*. [online] Investopedia. Available at: http://www.investopedia.com/articles/basics/04/050704.asp [Accessed 17 Dec. 2014].

Bullinaria, J. (2014). *Recurrent Neural Networks*. [online] Available at: http://www.cs.bham.ac.uk/~jxb/INC/l12.pdf [Accessed 17 Dec. 2014].

Huang, W., Keung Lai, K., Zhang, J. and Bao, Y. (2008). *Foreign Exchange Rates Forecasting with Multilayer Perceptrons Neural Network by Bayesian Learning*. 1st ed. [ebook] IEEE Computer Society, p.28. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4667939&tag=1 [Accessed 17 Dec. 2014].

ONANDA.Com, (2014). *Exchange Rate Factors*. [online] Available at: http://fxtrade.oanda.co.uk/learn/top-5-factors-that-affect-exchange-rates [Accessed 17 Dec. 2014].

Simon, E. (2002). *Forecasting Foreign Exchange Rates with Neural Networks*. 1st ed. [ebook] Neuchatel: University of Neuchatel. Available at: http://liawww.epfl.ch/uploads/project_descriptions/project_282.pdf [Accessed 17 Dec. 2014].

Tenti, P. (1996). *Forecasting Foreign Exchange Rates using Recurrent Neural Networks*. 1st ed. [ebook] Switzerland: Taylor & Francis. Available at: http://pdf.aminer.org/000/340/264/fractal_structure_of_linear_recurrent_neural_networks_with_temporal_input.pdf [Accessed 17 Dec. 2014].

Tutor2u.net, (2014). *Tutor2u - what determines the value of an exchange rate*. [online] Available at: http://www.tutor2u.net/economics/content/topics/exchangerates/forex_markets.htm [Accessed 17 Dec. 2014].

Xe.com, (2014). *XE.com - GBP/USD Chart*. [online] Available at: http://www.xe.com/currencycharts/?from=GBP&to=USD&view=1D [Accessed 17 Dec. 2014].

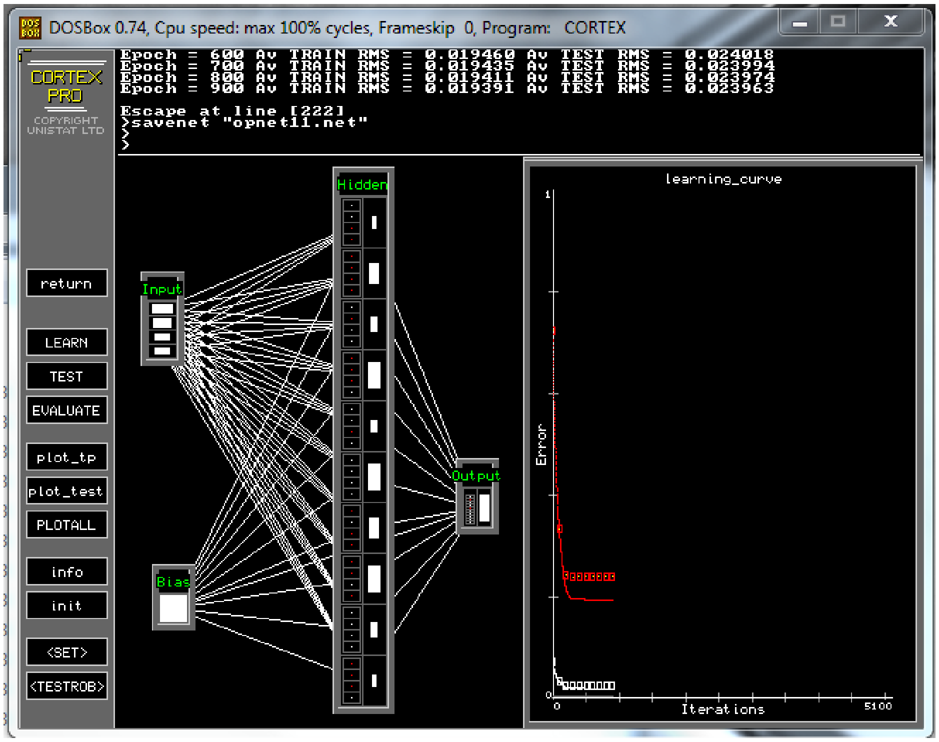
Appendix

Appendix 1

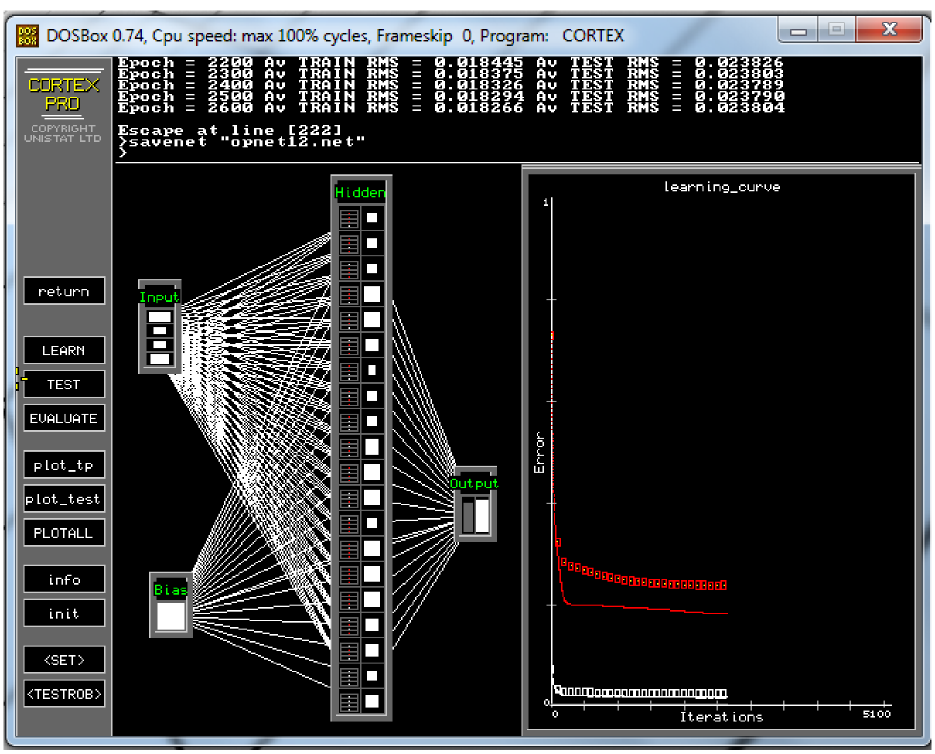
	200 Epoch		500 Epoch		1000 Epoch		1500 Epoch		3000 Epoch		5000 Epoch			
	Train Error	Test Error	Train Error	Test Error	Train Error	Test Error	Train Error	Test Error	Train Error	Test Error	Train Error	Test Error	Stopped training at epoch	Test error @ stop
Test 1.1	0.021983	0.024556	0.019493	0.024006	0.019386	0.024003	0.019322	0.024018	0.018739	0.024104	0.018107	0.024537	900	0.023963
Test 1.2	0.021406	0.28981	0.020033	0.029357	0.019777	0.025316	0.019382	0.02443	0.019041	0.023976	0.018561	0.024391	2600	0.023804
Test 1.3	0.021052	0.027272	0.020402	0.028271	0.020132	0.02777	0.019844	0.02732	0.019158	0.027108	0.19023	0.026115	5000	N/A
													Avg 1.x (with month data)	0.0238835
Test 2.1	0.026365	0.019341	0.026043	0.018936	0.025857	0.018872	0.025672	0.018954	0.025161	0.019614	0.023264	0.022567	1400	0.018899
Test 2.2	0.027933	0.020824	0.027309	0.020842	0.026641	0.019722	0.026225	0.019153	0.024214	0.021241	0.022882	0.021921	2000	0.019137
Test 2.3	0.029963	0.019039	0.029559	0.019577	0.028533	0.01892	0.028013	0.018596	0.026712	0.022953	0.023792	0.020415	1500	0.018621
													Avg 2.x (no month data)	0.01888566667

Appendix 2

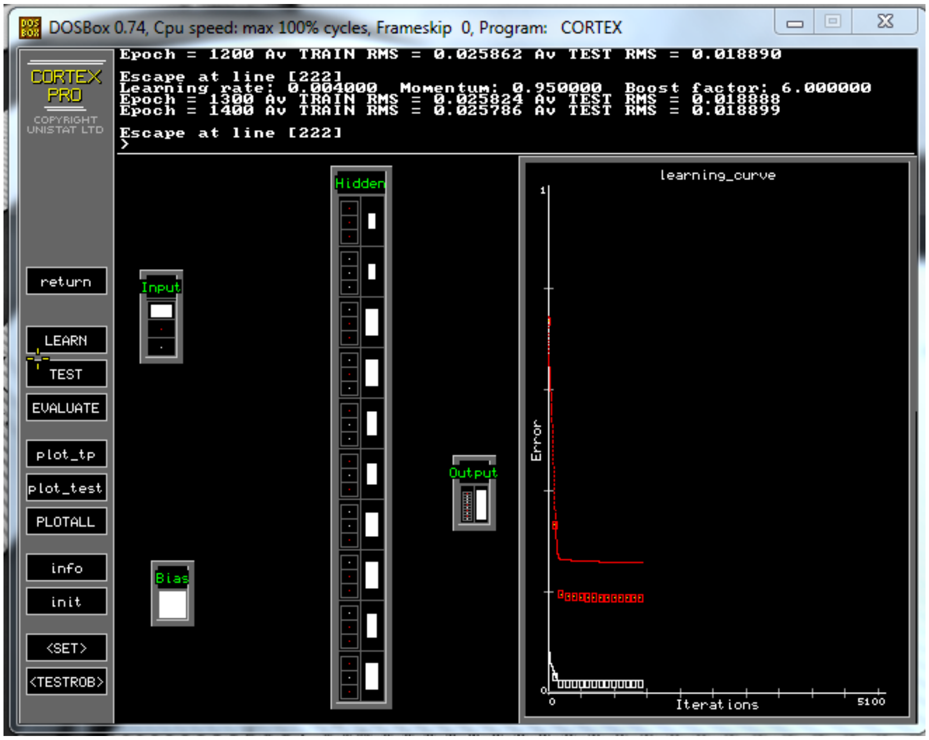
Test 1.1: 4 input nodes; 10 hidden nodes; 1 output node



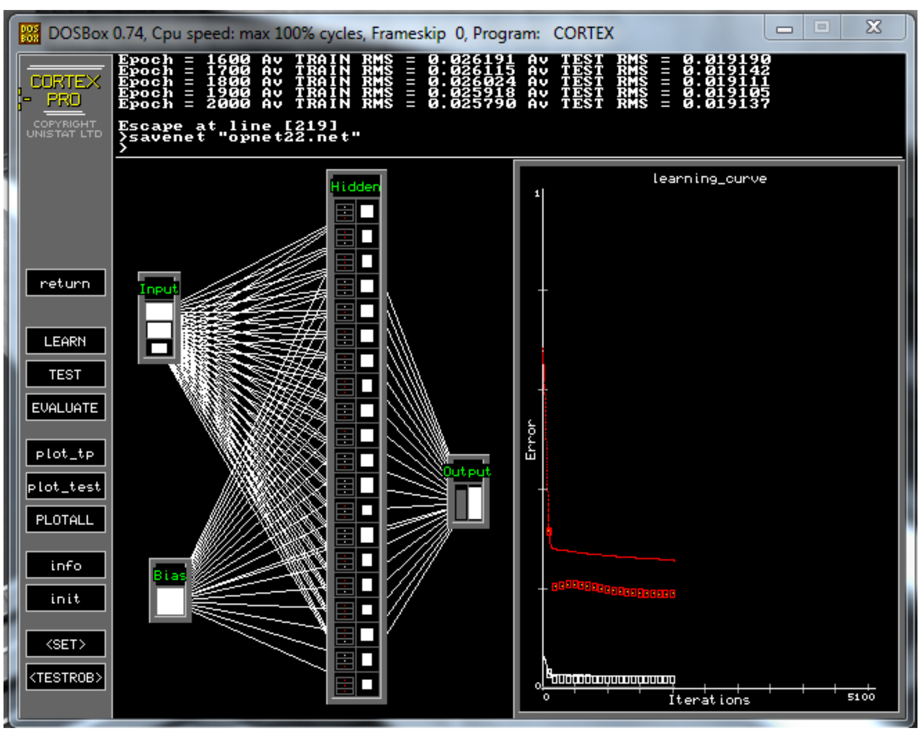
Test 1.2: 4 input nodes; 20 hidden nodes; 1 output node



Test 2.1: 3 input nodes; 10 hidden nodes; 1 output node



Test 2.2: 3 input nodes; 20 hidden nodes; 1 output node



Test 2.3: 3 input nodes; 40 hidden nodes; 1 output node