1-1-2011

Forecasting exchange rates using neural networks: A trader's approach

Adam Stokes
Eastern Illinois University

THESIS MAINTENANCE AND REPRODUCTION CERTIFICATE

TO: Graduate Degree Candidates (who have written formal theses)

SUBJECT: Permission to Reproduce Theses

The University Library is receiving a number of request from other institutions asking permission to reproduce dissertations for inclusion in their library holdings. Although no copyright laws are involved, we feel that professional courtesy demands that permission be obtained from the author before we allow these to be copied.

PLEASE SIGN ONE OF THE FOLLOWING STATEMENTS:

Booth Library of Eastern Illinois University has my permission to lend my thesis to a reputable college or university for the purpose of copying it for inclusion in that institution's library or research holdings.

5-20-11

Author's Signature Date

I respectfully request Booth Library of Eastern Illinois University **NOT** allow my thesis to be reproduced because:

Author's Signature Date

This form must be submitted in duplicate.

Forecasting Exchange Rates Using Neural Networks:

A Trader's Approach

(TITLE)

BY

Adam Stokes

THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

Master of Arts in Economics

IN THE GRADUATE SCHOOL, EASTERN ILLINOIS UNIVERSITY
CHARLESTON, ILLINOIS

2011

YEAR

I HEREBY RECOMMEND THAT THIS THESIS BE ACCEPTED AS FULFILLING
THIS PART OF THE GRADUATE DEGREE CITED ABOVE

Ahmed Abou-Zaid 5/9/11
THESIS COMMITTEE CHAIR DATE

5/9/11
DEPARTMENT/SCHOOL CHAIR OR CHAIR'S DESIGNEE DATE

5/9/11
THESIS COMMITTEE MEMBER DATE

5/9/11
THESIS COMMITTEE MEMBER DATE

THESIS COMMITTEE MEMBER DATE

THESIS COMMITTEE MEMBER DATE

Abstract

This study is to investigate the use of Artificial Neural Networks (ANN) for forecasting exchange rates. The currencies examined in the paper are the Euro, Pound, and Yen. Two different types of ANNs are used in this paper: Feedforward and Nonlinear Autoregressive with Exogenous Input (NARX). Forecasts are made for daily and weekly exchange rates using the open, high, low, and close of the exchange rates.

Many standard econometric models cannot deal with daily and weekly forecasts due to many macro economic variables not being available at such frequencies. ANNs are able to deal with daily and weekly data as well as the nonlinearities in exchange rate movements.

Acknowledgments

I would like to thank Dr. Ahmed Abou-Zaid for his valuable input and wiliness to advise a thesis outside the economic norms. I would also like to thank my committee members Dr. Upadhyay and Dr. Bruehler for their comments and suggestions. Lastly, I owe thanks to Derek Serrano for bringing this idea to me and helping me on some of the coding.

Table of Contents

List of Figures

I. Introduction

Being able to accurately predict movement in exchange rates has been a "holy grail" in finance and economics. Many valuable things can be done and implemented if an individual, government, or business is able to accurately forecast future exchange rates. Leung *et al* (2000) note that for large multinational firms which conduct substantial currency transfers in the course of business, being able to accurately forecast the movements of exchange rates can result in considerable improvement in the overall profitability of the firm. However because attempts to directly model exchange rates have been mostly unsuccessful, many companies started employing the use of state-of-the-art artificial intelligence technologies to help in accurately forecasting exchange rate movements (Leung *et al* 2000).

Before the 1980's, time series prediction was mainly based on linear parametric Auto Regressive(AR), Moving Average(MA), or Autoregressive Moving Average(ARMA) models (Diaconescu 2008). The major disadvantage of these common econometric models is that they are not able to forecast at a significantly higher level of accuracy than that of a naïve random walk (Leung *et al* 2000, Meese and Rogoff 1988). The most common econometric models are linear and are not able to deal with non-stationary signals and signals which are not linear. Jamal *et al* (1997) point that foreign exchange rate movements are often based on short term factors and since many standard econometric models do not deal with the short term very well it is important to explore other methods. They also point out that the major shortcoming of the standard models is that much of the economic variables are not available at the frequency needed for short term projections (Jamal *et al* 1997).

The use of machine learning models, such as Artificial Neural Networks (ANN), is helpful in overcoming these issues. ANNs belong to a field of research aimed at building a computational machine-based cognitive system that tries to capture the key aspects of the human cognitive process (Nag and Mitra 2002). While the microprocessor may compute faster than the human brain ever will, the advantage of the human brain rests in its adaptability, self-organization, and parallelism. An ANN is simply a way to try and mimic these aspects of the human cognitive process (Nag and Mitra 2002). Using terminology from statistics, an ANN is a nonlinear, nonparametric, multivariate, and wholly data driven inference procedure (Nag and Mitra 2002). Neural networks are powerful when applied to problems whose solutions require complex relationships which are difficult to specify, but for which there is an abundance of examples (Diaconescu 2008). As opposed to traditional models, ANN are data driven, self adaptive, and can approximate any continuous function to any desired accuracy (Nag and Mitra 2002). ANN models also have an advantage over standard models in that they do not require a specified functional relationship between the input and output variables (Nag and Mitra 2002). Leung *et al* (2000) point out the many possibilities for ANN uses in the areas of finance including bankruptcy prediction, savings and loan association failures, and corporate distress diagnosis. Common uses outside the business world are facial recognition, image processing, and artificial intelligence programming in robots.

The focus of this paper is to forecast the exchange rate of the U.S. Dollar against three major currencies: Euro, British Pound, and Japanese Yen. This paper will adopt an

ANN methodology, rather than classic techniques, as it is a more accurate method of forecasting exchange rates. Due to the differing nature of ANNs to normal econometric methods, a different test will need to be used to assess the significance of the results of the ANNs. A non-parametric market timing test developed by Nobel Laureate Robert Merton and Roy Henriksson (1981) will be used to assess the strength of the ANNs market timing ability. Based on past literature two types of neural networks will be used: Feed-forward (FF) neural networks and Non-linear Autoregressive with Exogenous Inputs (NARX) neural networks.

The currencies examined in this paper are defined as: Euro (Euro/USD), British pound (pound/USD), and the Japanese Yen (Yen/USD). Both neural network types will be estimated for daily and weekly data from 1980-2010.

The remainder of the paper will be divided into Literature Review, How Neural Networks Work, Network Types, Results, and Conclusion. The literature review section will give a summary of relevant past literature on Neural Networks and exchange rates. The section How Neural Networks Work will describe the mechanisms behind ANNs and show how they function for forecasting exchange rates. The Network Types section will describe the two ANN types used in this paper. Following the description of network types will be the Results section and conclusion.

II. Literature Review

Exchange rate forecasting is widely covered throughout the economic literature. Different models have been developed over the years to try and more accurately predict exchange rates. One model widely used is the purchasing power parity (PPP) model.

The PPP model does have some support as an effective forecasting tool for long horizon predictions, but fails to be effective for short horizon predictions (Cheung *et al* 2005). In their paper, Cheung *et al* (2005) found that the directional forecasts of the PPP and uncovered interest rate parity models performed somewhat favorably for long term forecasts, but failed to have significant results when using quarterly forecasts. Also tested in their paper was a monetary model.

In the monetary model, Cheung *et al* (2005) modeled the log exchange rate as a function of the log values of money, log GDP, interest rate, and inflation rate. This was referred to as a "sticky price" model. A productivity differential exchange rate model was included and was similar to the sticky price model except for productivity differential was substituted in for the inflation rate. Cheung *et al* conclude that PPP and interest rate parity models outperform a random walk for the 20-quarter time horizon when MSE is the device used to measure accuracy between different model types. However, no specific model was found to be dominant over a random walk model for all currencies or even a single currency at different horizons.

Going even farther back in time, we find the results of Meese and Rogoff (1988). In this seminal paper, the authors tried to find a better alternative to a random walk model. Meese and Rogoff tested a random walk with drift model, ARIMA, and a local trend predictor. Neither of these methods was able to significantly outperform the random walk model when using monthly forecasts. Based on these finding, Meese and Rogoff (1988) conclude that there is no evidence whatsoever that a model using sticky prices and monetary disturbances and predict swings in the exchange rate.

Although exchange rates are widely covered in economic literature, forecasting using ANN's is very sparse in economics. Most literature that adopts ANN's, including those that forecast exchange rates, are found in computer science, mathematics, and computer learning journals. There is, however, a couple published in economic journals. The most influential to this thesis is found in *The Journal of Applied Econometrics*.

In their paper, Kuan and Liu (1995) experiment with using feed forward and recurrent neural networks to forecast the exchange rate of British Pound, Canadian Dollar, Deutsche Mark, Yen, and Swiss Franc. More specifically, Kuan and Liu were trying to evaluate if the predictive stochastic complexity criterion was an effective way of determining an accurate neural network for out of sample prediction. After building and selecting their networks, they use a market timing test derived by Robert Merton (1981) to see if the network has significant power for predicting the markets.

Yao and Tan (2000) provide further evidence of the usefulness of ANN's in exchange rate forecasting. Some technical indicators such as moving averages are added to their list of inputs to aid in forecasting. They argue that the addition of technical indicators can help provide a basis to further capture the underlying rules of exchange rate movements. Francis and Van Homelan (1998) also suggest that the addition of technical indicators may provide results more favorable to ANNs that do not use them. The addition of the technical indicators is listed as a possible way to eliminate the ANN picking up GARCH signals. Neglected GARCH does not lead to falsely successful ANNs in out of sample forecasting and if nonlinearity of exchange rates does exist, the ANN can exploit it for improved forecasting (Francis and Van Homelan 1998).

As a benchmark comparison, Yao and Tan compare the correctness (total percent of out-of-sample predictions that were correct) to an ARIMA model. They found that the ARIMA model was able to only get a 50% correctness in sign prediction while the ANN's were achieving as high as 73% correctness. The ANN correctness rate attained by Yao and Tan falls in line with those attained by Kuan and Liu(1995). Frances and Van Homelen (1998) also used the fraction of times the ANN was able to adequately forecast the sign of the next day's return as the way to evaluate their networks performance. In their paper, Yao and Tan (2000) where it was aimed at trading profits, they find that profits from the ANN were nearly 6 times greater than that of ARIMA (36% vs. 6%). Similar to Yao and Tan's ARIMA comparison, Nag and Mitra (2002) note that ANN outperform ARIMA models and exponential smoothing methods.

In the study of Leung *et al* (2000), General Regression Neural Networks (GRNN) are compared to multi layer neural networks, models based on multivariate transfer functions (a common econometric forecasting tool), and a random walk forecast model. The motivations for Leung's study was the findings from Meese and Roghoff in the 1970's and 1980's that sticky price models, flexible price models, and the sticky price asset model all performed unsatisfactory in exchange rate forecasting. The selection criterion used was the root mean squared error and then the mean absolute error was also used for comparison in selection of networks. Both criteria led to the same network selections for Leung *et al* (2000). After running and comparing all the four models chosen by Leung *et al*(2000), the authors conclude that the GRNN statistically outperform all other tested models. More importantly, the GRNN is more accurate than a random walk which, based on previous literature observations, is equal to most econometric techniques.

Kamruzzaman and Sarker (2003) use ANN's to predict six currencies against the Australian dollar. They use the ARIMA model as their benchmark for comparison. Similar to Yao and Tan (2000), Kamruzzaman speaks of the benefit to using some technical indicators as input into the networks. The reason listed is that exchange rates will exhibit their own trend, cycle, season, and irregularity. To try and capture these behaviors, Kamruzzaman and Sarker input 5 moving averages in their input stream to go along with the closing price. The ANNs outperformed the ARIMA model in every case for each of the six currencies.

As mentioned in many of the previous studies, the only data input in the ANNs are past values of the exchange rates or various representations of past exchange rates such as moving averages and other technical indicators. The lower data requirement is one of the advantages ANNs have over standard econometric models (Jamal *et al* 1997). By keeping the inputs to the ANNs various forms of the past exchange rates, the problem of economic data frequency and data correctness is avoided. Many economic variables are subject to frequent revisions and thus may not be completely accurate (Jamal *et al* 1997).

Further, Nag and Mitra (2002) list the failure of standard linear econometric models to outperform a basic random walk model. They point out that ARCH and GARCH models provide evidence that small (large) changes in price are followed by small (large) changes in either direction. The ARCH/GARCH model's inclusion of such nonlinearities lead to some possible better forecasts of monthly exchange rates (Nag and Mitra 2002).

One problem pointed out in the literature with ANNs is the parameters used in the ANN structure. There are no guidelines for what to use as parameters. Deciding on what parameters to use is often found to depend on the experience of the researcher (knowing what has worked in the past for ANN structure) and trial and error (Nag and Mitra 2002). To address this trial and error problem, some research is being done on the benefits to optimizing ANN structure/parameters. Methods such as genetic algorithms have been shown to have favorable results on improving ANN performance (Nag and Mitra 2002).

III. How Neural Networks Work.

This section is devoted to explaining the techniques behind ANNs and describing how they work. The methods described in this section are applicable to the two network types used in this paper: Feed Forward and NARX networks. In its most basic structure, ANN's are composed of three layers: input layer, hidden layer, and an output layer. The input layer is where the data (the exchange rates) are passed into the network. The second layer, often referred to as the hidden layer or the neuron layer, is made up of a set amount of "neurons" that perform calculations and summations based on the input layer. The output layer is where the neuron layers calculations are converted to the specified output form.

Kuan and Liu (1995) describe neural network as a nonlinear regression function characterizing the relationship between the dependent variable (y) and an n-vector of explanatory variables. However, instead of postulating a specific nonlinear function, the neural network is constructed by combining multiple basic nonlinear functions via a multilayer structure.

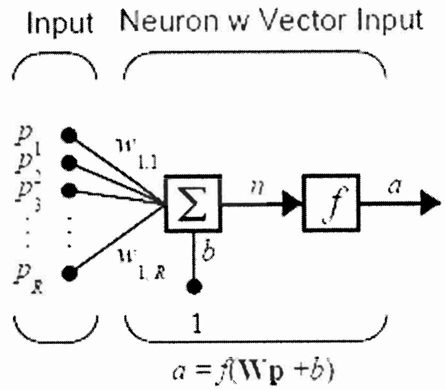
The constructing of these various "basic" nonlinear functions is done using several tools that characterize an ANN. These tools are the transfer functions, neuron properties, and network architecture. First the neuron's makeup will be explained followed by the explaining of the transfer functions and the network architectures.

An unbiased neuron takes a single input *(p)* from the input layer and assigns a weight*(w)* to p to give wp. The initial weights assigned are random and are adjusted using the training algorithm assigned to the network. This wp value is passed onto the transfer function which outputs a scalar value a (Demuth *et al* 2010). There is also a biased neuron model which acts the same as the unbiased except it adds a constant *(b)* to wp thus passing the value $wp+b$ to the transfer function (Demuth *et al* 2010). These two neuron structures are what are normally used for feedforward networks. Other neuron structures do exist, such as the radial basis neuron used in a radial basis network[1], but these structures do not pertain to the models used for forecasting exchange rates (Demuth *et al* 2010).

The above neuron description is for just one input value. For the forecasting of exchange rates we have a vector input structure. Instead of just a single p there are $p_1,p_2,p_3,\ldots p_n$=**P**. When a vector is used, all p_n inputs are taken as inputs for the neuron. Thus, the single neuron will assign n different weights to the n different p*'s* giving a vector of weights $w_{1,1},w,_{1,2},\ldots,w_{1,n}$=**W**. The neuron then sums all these weights giving **Wp** which is actually the dot product (single row) of the matrix **W** and the vector **P** (Demuth *et al* 2010).

[1] For more information on radial neurons/networks see Demuth *et al* 2005

Figure1. Neuron Diagram



Where

R = number of elements in input vector

Source: Demuth *et al* (2010)

The above diagram is for one neuron only. If there are multiple neurons, they would each behave the same way with each neuron receiving all the input values. Because of this and the way the transfer functions behave, it is very hard to interpret the *w's* in a manner that is reminiscent of coefficients in econometric models. For example, if a neural network has 10 neurons then there are 10 unique w's for each *p*. After the neurons have finished their calculations, the neuron output is sent to the transfer functions.

The transfer functions are what determine the input and output behavior of the neural network. After the neuron is finished summing all of the weights, the transfer function decides what form the neuron output will take. Some of the most commonly used transfer functions are: hardlim, linear, and log-sigmoid. A hardlim transfer functions forces the output to be either 0 or 1. If the weighted sum, $\mathbf{Wp} + b$, is less than 0 a value of 0 is assigned and if the sum is greater than 0 a value of 1 is assigned. For a linear transfer function the output is allowed to take the form of any number, compared to hardlim which is limited to 0 or 1. The log-sigmoid function allows for values between 0 and 1 and is a differentiable function. A differentiable function is required for some network types (Demuth *et al* 2010).

It has been observed that the forecasting of a time series will be better when simultaneously analyzing similar time series. More specifically, to forecast the next day it is useful to input all previous input values into the network (Diaconescu 2008). So for a single input of p_i there are p_{i-1} time series being analyzed, where i represents the i^{th} observation in a given time series. Therefore, by analyzing several time series at once, the ANN is able to more accurately predict the future value of p. This is better thought of as the "memory" of the ANN. By analyzing all past relationships of p on the future value, the ANN is able to remember what happened in the past.

The above has described how a ANN generates its output. However, the uniqueness in a ANN is in the way it works. An ANN is trained and then compared to some benchmark target. If the ANN is not at the target, the network is retrained. One of the most common measures is the root mean squared error(RMSE) and the ANN aims at reaching a RMSE equal to zero. The ANN tries to minimize its error in prediction by minimizing the RMSE and unless given certain rules, the training of an ANN will run until the RMSE is 0[2]. To accomplish this, neural networks use learning algorithms (also called training algorithms) to minimize the RMSE (Diaconescu 2008). Most ANN employ a gradient based learning algorithm such as a Kalman filter, Newton type algorithm, or annealing algorithm (Diaconescu 2008).

Typically data is segmented into three parts, the training set, the validation set, and the test set. Usually around 70% of the observations are used for training, 15% for

[2] The RMSE will never reach 0. Constraints are implemented in ANNs to stop this from happening since an ANN with 0 RMSE is highly likely to be "over fit", thus perfectly mimicking the past while not truly capturing the patterns/rules underlying the exchange rates.

validation, and 15% for testing. Only the training section of the data is used to calculate the weights as discussed above while the validation and test sets are used to see if the assigned weights need to be adjusted. The best model is then chosen from those run in the training process. The "best" is the one that performs the best in both the validation and test phases. If the model performs well on the validation and test samples of data, it is assumed to be the best for future forecasting (Yao and Tan 2000). This "best" model will be the one used for forecasting using the out-of-sample data.

There are some limitations to using ANNs. One is the influence of outliers. Outliers make it difficult for ANNs to model the true underlying function (Diaconescu 2008). Dianconescu (2008) also notes that sometimes a larger set of observations may not yield more accurate results. On a separate note, classical time series prediction required the time series to be stationary, even when many time-series are not stationary. ANN allows the original time-series to be used (Diaconescu 2008).

IV. The Networks

a. Feed Forward

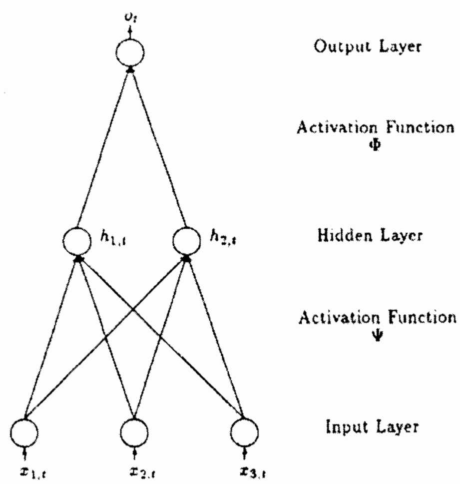
The first of the networks examined in this paper is the Feed Forward Neural Network (FF). In a FF there is no "recursion", more specifically, the amount of error in the training as well as the output is not taken as an input when the network is retrained. Kuan and Liu(1995) describe a FF network as one where the explanatory variables (the inputs) activate q hidden units in the hidden layer through a function Ψ, and the resulting hidden-unit activations h_i, $i=1,2,..q$, then activate output units through some function Φ to produce the network output o. Mathematically this is represented as

$$h_{i,t} = \Psi\left(\gamma_{i0} + \sum_{j=1}^{n} \gamma_{ij} x_{j,t}\right) \qquad i = 1, \ldots, q$$

$$o_t = \Phi\left(\beta_0 + \sum_{i=1}^{q} \beta_i h_{i,t}\right)$$

and graphically as

Figure 2. Feedforward Network



Source: Kuan & Liu (1995)

where the graphic represents a Simple Feed Forward Network with one output unit, two hidden units, and three input units. The activation functions Ψ and Φ can be chosen arbitrarily except Ψ is usually a bounded function. Kuan and Liu (1995) also state that it is normal to include lagged dependent variables as inputs, however the amount to include is unknown and because of this, the lagged dependent variables may not be enough to characterize the behavior of the dependent variable. To deal with this deficiency, recurrent networks (networks with feedbacks) have been created. The NARX network is one such network and will be described in the following section.

The FF network used in this paper has one neuron layer giving a single output. The output, y, follows the format where -1<y<1. Any value of y greater than 0 signifies a predicted increase in the exchange rate of the currency. Similarly, a value less than 0 would be a predicted decrease in the exchange rate. For example, if the current exchange rate of the Euro (Euro/USD) is 1.310 and the network outputs a value of y<0 then the next period is predicted to have an exchange rate less than 1.310. The inputs used in the FF are lags of the open, high, low, and close of the weekly/daily exchange rate.

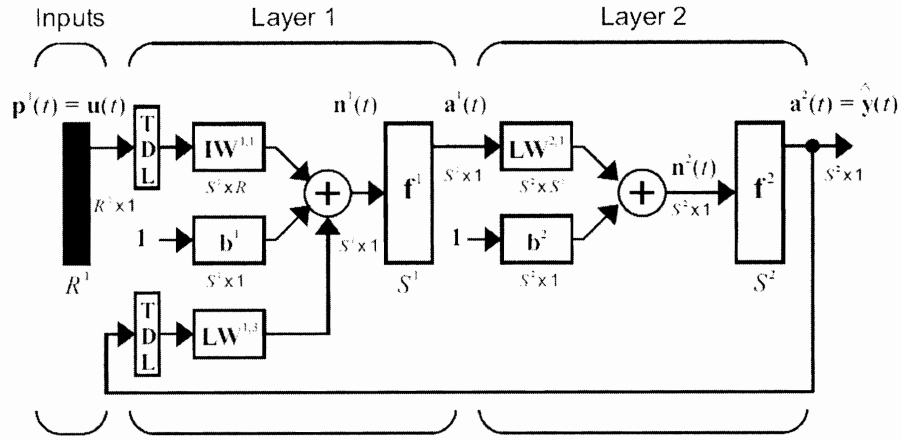
b. NARX

The NARX network is a form of a recurrent network. A recurrent network is a network that has feedback delays. This is where the output from one or more neurons in the hidden layer is used as input in addition to the normal time-series input. This feedback to the input layer with delay serves as a way to "memorize" the past information (Kuan and Liu 1995). Diaconescu (2008) states that NARX networks are well suited for modeling nonlinear system and specifically time series. One reason for using a NARX network is that they converge faster than many other ANNs and learning is more effective in NARX networks (Diaconescu 2008). Mathematically, the NARX network is represented by

$$y(t) = f(y(t-1), y(t-2), \ldots, y(t-n_y), u(t-1), u(t-2), \ldots, u(t-n_u))$$

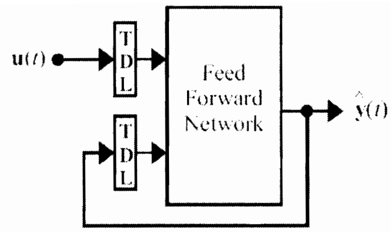
where the next value of the dependant output signal *y(t)* is regressed on previous values of output and previous values of independent (exogenous) input *u* (Demuth *et al* 2010). Graphically the NARX network is represented by

Figure 3. NARX Network



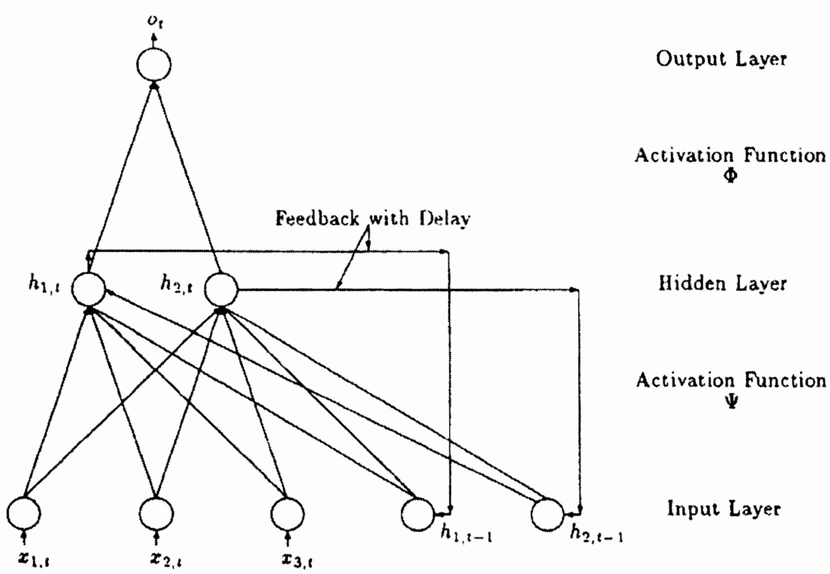
Source: Demuth *et al* (2010)

where the output of the NARX network is fed back into the NARX network as an input for the next series. The diagram for the NARX network above may look complicated but it can basically be viewed as the feed forward network from above where the output is fed back in as input. The diagram above shows a network with two hidden layers: Layer 1 and Layer 2. Layer 1 takes the input, represented by the TDL box, and assigns weights *IW* it then adds a bias of 1 through the *b* box and then sums up all weights and b's which is then passed on to the transfer function f^1. The output from the f^1 function is then passed as input into the second layer in a manner similar to the first layer. Where Layer 2 differs from Layer 1 is that Layer 2 passes its output (f^2) back into Layer 1 as an input and this feedback input gets assigned a weight of LW. To view this diagram in a less complex way, imagine Layer 1 and Layer 2(excluding the feedback) as a feedforward network. Thus, a simpler diagram of a NARX network can be drawn as the following:



Source: Demuth *et al* (2010)

where the first diagram is simplified down to show the FF network hidden inside and the output from the FF network feeding back into itself. The NARX network is similar in structure to the recurrent network used by Kuan and Liu (1995) which is represented below. In Kuan and Liu's (1995) paper the neuron output from $h_{1,t}$ and $h_{2,t}$ is taken as input for the next periods prediction. Thus, the $h_{1,t-1}$ is the output from the neuron from the last period.



Source: Kuan & Liu (1995)

A NARX network has many applications. It can be used to predict the next value of the input signal (a forecast 1 period into the future) as well as for nonlinear filtering where the target output is a noise-free version of the input signals (Demuth *et al* 2010). The NARX network is also able to model nonlinear dynamic systems. Demuth *et al* (2010) say the output of the NARX network can be considered as an estimate of the output of some nonlinear dynamic system.

The output and targets of the NARX network are the same as the FF. That is, the targets are either -1 or 1 where -1 represents a fall in the exchange rate and 1 represents an increase in the exchange rate. The output is thus either positive, representing an expected increase, or negative which represents an expected decrease. Where the NARX differs from the FF is in the inputs. Whereas the FF was just inputting the open,high, low, and close of the weeks exchange rate, the NARX network will include those as well as some technical indicators commonly used in finance for exchange rate trading. These technical indicators act as the "exogenous" inputs in the NARX network.[3]

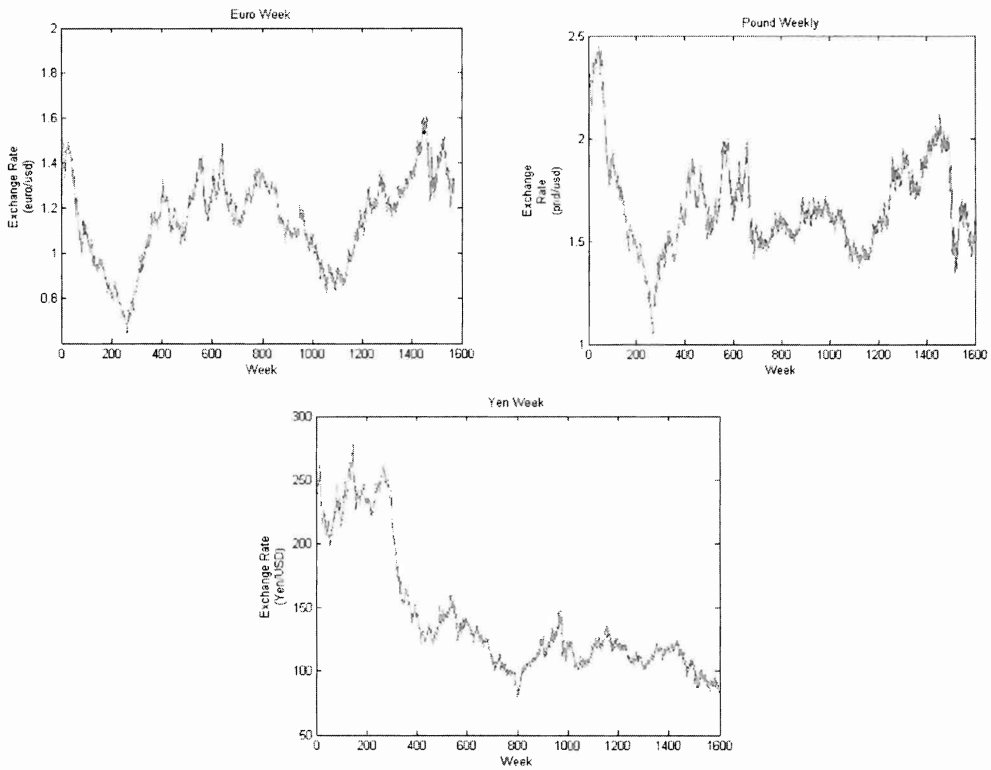
V. Data

Data on the three currencies was provided by a Bloomberg terminal in August 2010. Data is recorded as the Open, High, Low, and Close for each day of the week. It should be noted that the FOREX markets are open 24 hours a day, 5 days a week. This provides for a rather large sample size. The daily data is then prepared to be weekly. This is done by using the last day of trading in the week as the closing price, high is the

[3] The technical indicators are: ADX, ATR, MACD, Momentum between times, Ultimate Oscillator, Percentage Price Oscillator, Fast Stochastic, Slow Stochastic, RSI, Stochastic Oscillator, Acceleration between times, William Accumulation/Distribution, Williams %R, Highest High, and Lowest Low.

maximum for the week, low is the low for the week, open is where the currency started trading at when the markets opened for the week. All target data in this study has been normalized to be either 1 or -1.

Figure 4. Currency Charts



VI. Results

Kuan and Liu (1995) define the ARMA(0,0) model as the “random walk” model to compare the results of the network output to. In their paper, the sign prediction of the random walk models never broke the 50% barrier. This paper seeks results that more accurately predict the exchange rates of the Euro, Pound, and Yen when compared to a random walk model. Ideally the best result would have a 99% significance level for the Merton market timing test and have a high sign prediction.

In this paper, NARX networks are designed and trained[4] for the three currencies (pound, euro, and yen)[5]. Lags are set from 1-5 meaning that 10 networks are trained(5 for weekly and 5 for daily) for each currency at each out of sample size. The lags are applied to the 22 inputs: Open, High, Low, and close of the currency as well as the technical analysis indicators. The NARX models have 15 neurons in their hidden layer.

In the table below, the results are posted for various networks. To calculate the significance levels of the ANN's market timing we need to know how many predictions it made, N, how many times it predicted a decrease,n, how many times a decrease was actually observed during the sampling period N1, how many times an increase was observed, N2, and how many of the predicted decreases represented a correct prediction ,x (Henriksson and Merton 1981, Henriksson and Lessard 1982, Henriksson 1984). Once these values are obtained, the market timing ability can be calculated using the following:

$$\sum_{x=x^*}^{\bar{n}_1} \binom{N_1}{x}\binom{N_2}{n-x} \Big/ \binom{N}{n} = 1 - c.$$

The last value of x in the above formula that is less than 1-c, where c is the confidence level desired, is the number of correct predictions needed for significant market timing.

[4] All networks are designed and trained using Matlab version 2010b using the Neural Network Toolbox.
[5] "Design and training" refers to the manner described in section III. If the network is not accurate (low RMSE), the process repeats up to 1000 times.

The sign prediction column in the result tables below reports what percent of the out of sample predictions were correct. It should be noted that this is the overall accuracy of both directional forecasts (positive and negative). While the above formula has x representing only the correct decrease predictions, the sign prediction uses the sum of $x_{decrease}$ and $x_{increase}$. For the NARX networks the column "Correct (99%)" lists the number of correct predictions required for a 99% significance level while the "correct (Total)" column lists the networks actual number of correct predictions.

Table 1. NARX-Euro-Weekly

Lag	Out of Sample	Observed Depreciation	Predicted Depreciation	Correct (99%)	correct (Total)	Market Timing	Sign Prediction
1	48	26	28	18	18	99%	62.5%
2	47	26	31	20	19	92%	60%
3	46	26	28	19	19	99%	65%
4	45	25	25	17	16	94%	60%
5	44	25	25	17	14	57%	50%
Lag	Out of Sample	Observed Depreciation	Predicted Depreciation	Correct (99%)	correct (Total)	Market Timing	Sign Prediction
1	99	51	52	32	32	99%	61%
2	98	51	50	31	27	73%	52%
3	97	50	51	31	27	69%	52%
4	96	49	44	27	28	99%	61%
5	95	49	51	31	25	37%	47%

Table 2. NARX-Euro-Daily

Lag	Out of Sample	Observed Depreciation	Predicted Depreciation	Correct (99%)	correct (Total)	Market Timing	Sign Prediction
1	99	51	51	31	26	54%	49%
2	98	50	49	30	25	58%	50%
3	97	49	48	29	25	69%	52%
4	96	48	49	29	24	50%	49%
5	95	47	44	26	23	76%	53%
Lag	Out of Sample	Observed Depreciation	Predicted Depreciation	Correct (99%)	correct (Total)	Market Timing	Sign Prediction
1	49	22	22	13	11	83%	55%
2	48	22	20	12	10	78%	54%
3	47	21	23	13	11	76%	53%
4	46	20	22	12	12	99%	61%
5	45	20	22	13	11	85%	56%

Table 3. NARX-Pound-Weekly

Lag	Out of Sample	Observed Depreciation	Predicted Depreciation	Correct (99%)	correct (Total)	Market Timing	Sign Prediction
1	48	29	25	18	18	99%	63%
2	47	28	25	18	16	83%	55%
3	46	27	25	18	15	69%	52%
4	45	26	26	18	17	93%	60%
5	44	25	24	16	14	80%	52%
Lag	Out of Sample	Observed Depreciation	Predicted Depreciation	Correct (99%)	correct (Total)	Market Timing	Sign Prediction
1	99	55	50	33	30	86%	55%
2	98	55	53	35	36	99%	63%
3	97	54	52	34	29	59%	51%
4	96	53	50	32	28	64%	51%
5	95	53	46	30	23	18%	44%

Table 4. NARX-Pound-Daily

Lag	Out of Sample	Observed Depreciation	Predicted Depreciation	Correct (99%)	correct (Total)	Market Timing	Sign Prediction
1	99	48	54	31	29	91%	54%
2	98	47	47	27	24	78%	51%
3	97	46	47	27	26	96%	56%
4	96	46	47	27	24	79%	51%
5	95	46	47	27	26	94%	55%
Lag	Out of Sample	Observed Depreciation	Predicted Depreciation	Correct (99%)	% correct (Total)	Market Timing	Sign Prediction
1	49	20	23	12	11	89%	57%
2	48	20	22	12	10	78%	54%
3	47	20	21	12	11	94%	60%
4	46	19	27	14	12	79%	52%
5	45	18	23	12	12	99%	62%

Table 5. NARX-Yen-Weekly

Lag	Out of Sample	Observed Depreciation	Predicted Depreciation	Correct (99%)	correct (Total)	Market Timing	Sign Prediction
1	48	26	24	16	10	7%	44%
2	47	26	27	18	14	40%	53%
3	46	25	23	15	11	27%	50%
4	45	25	23	16	11	22%	53%
5	44	24	27	17	12	8%	55%
Lag	Out of Sample	Observed Depreciation	Predicted Depreciation	Correct (99%)	correct (Total)	Market Timing	Sign Prediction
1	98	56	56	37	26	1%	43%
2	97	55	51	34	24	3%	40%
3	96	54	51	33	28	47%	47%
4	95	54	47	31	24	18%	41%
5	94	53	46	31	28	86%	50%

Table 6. NARX-Yen-Daily

Lag	Out of Sample	Observed Depreciation	Predicted Depreciation	Correct (99%)	correct (Total)	Market Timing	Sign Prediction
1	99	54	56	35	27	11%	40%
2	98	53	56	35	28	23%	43%
3	97	53	56	35	29	33%	44%
4	96	53	55	35	31	68%	49%
5	95	52	54	34	27	20%	42%
Lag	Out of Sample	Observed Depreciation	Predicted Depreciation	Correct (99%)	correct (Total)	Market Timing	Sign Prediction
1	49	27	26	17	12	14%	39%
2	48	26	27	18	14	47%	46%
3	47	26	29	19	12	2%	32%
4	46	25	26	17	12	17%	39%
5	45	24	25	16	13	54%	47%

Of all three currencies, the Euro performed the best in the NARX networks. When comparing the overall out of sample accuracy and significance, the lower out of sample sizes seemed to perform slightly better. This suggests that the patterns memorized by the Neural Network change frequently and the trained network doesn't apply as well for longer periods. Also of note for the Euro was that the 1 and 3 lag for the shorter weekly out of sample length and the 1 and 4 lag for the longer weekly out of sample were the networks with significant market timing ability. This could possibly provide evidence for the support that 1 lag is an adequate network structure for the Euro as it may capture the underlying behavior of the currency. Likewise, 4 lags may also be suitable as it was at a 94% significance level in the shorter period and 99% in the longer out of sample period.

The reason the Yen has such poor results could be due to the rather rapid appreciation in the last 2 years, compared to the previous 10 years where it was rather steady in the 115-130 yen/USD range. However starting in 2008 we see a sharp drop to the high 80's and low 90's in the exchange rate. Since the NN is doing out of sample predictions for the last 50-150 weeks, that is a 1-3 year span stretching back from august of 2010. This means that the pattern from this decrease would not be "memorized" and therefore could be potentially causing problems since the yen is not acting like it has in the past. Yao and Tan (2000) also found that neural networks forecasted 5 major currencies accurately but did not find favorable results for the Yen.

One interesting result is the turnaround in performance when the Yen is input into the FF network with no lags. In the table below are selected results from the FF network results for the weekly Yen. While the FF networks are given lagged data in some of the models, the term "regular FF" in this paper refers to a FF network with no lag inputs.

Table 7. Yen- Feed Forward /w no lag input- Weekly

Neurons	Out of Sample	Observed Depreciation	Predicted Depreciation	Depreciation Correct	Depreciation Incorrect	Sign Prediction	Market Timing
1	50	27	25	15	10	56%	87%
2	50	27	29	13	16	40%	7%
3	50	27	23	7	16	28%	0%
4	50	27	28	11	17	34%	2%
5	50	27	23	8	15	32%	1%
6	50	27	22	17	5	70%	99.9%
7	50	27	23	16	7	64%	99%
8	50	27	19	8	11	40%	15%
9	50	27	24	11	13	42%	20%
10	50	27	20	5	15	26%	0%
11	50	27	33	17	16	48%	43%
12	50	27	23	14	9	56%	88%
13	50	27	24	18	6	70%	99.9%
14	50	27	23	9	14	36%	4.7%
15	50	27	22	6	16	26%	0%
16	50	27	28	19	9	66%	99%
17	50	27	21	15	6	64%	99%
18	50	27	23	12	11	48%	51%
19	50	27	26	10	16	34%	2%
20	50	27	28	19	9	66%	99%

As stated above, the likely reason for the poor performance of the Yen in the NARX is due to the uncharacteristic behavior in the out of sample testing. The uncharacteristic drop in the Yen's exchange rate during the out-of-sample forecasting may not follow the "rules" the network had found in the previous periods. The FF network with no lag inputs might solve this problem. The results reported in table 8 show that when no lags are input into the network there are some very favorable results for the Yen. If the results for the Yen are better in the FF, what about the other currencies? Below are the tables for the Euro and Pound with various lag inputs as well as weekly and daily.

Table 8. Top Yen FeedForward Networks based on Lags and Out of Sample size - Weekly

Lags	Neurons	Out of Sample	Observed Depreciation	Predicted Depreciation	Depreciation Correct	Depreciation Incorrect	Sign Prediction	Market Timing
3	8	50	27	24	18	6	70%	99%
3	12	50	27	24	19	5	74%	99%
3	18	50	27	26	20	6	74%	99%
5	10	50	27	26	18	8	66%	99%
5	16	50	27	24	18	6	70%	99%
5	18	50	27	25	18	7	68%	99%
0	2	100	54	37	29	8	67%	99%
0	6	100	54	51	36	15	67%	99%
0	13	100	54	54	38	16	68%	99%
3	2	100	54	48	37	11	72%	99%
3	3	100	54	48	34	14	66%	99%
3	9	100	54	48	32	16	62%	99%
5	1	100	54	48	35	13	68%	99%
5	13	100	54	51	35	16	65%	99%
5	14	100	54	50	35	15	66%	99%

Table 9. Top Euro FeedForward Networks based on Lags and Out of Sample size - Weekly

Lags	Neurons	Out of Sample	Observed Depreciation	Predicted Depreciation	Depreciation Correct	Depreciation Incorrect	Sign Prediction	Market Timing
0	6	50	27	23	15	8	60%	96%
0	7	50	27	18	14	4	66%	99%
0	20	50	27	25	15	10	56%	87%
3	10	50	27	28	17	11	58%	91%
3	19	50	27	24	17	7	66%	99%
3	20	50	27	27	19	8	68%	99%
5	14	50	27	23	15	8	60%	96%
5	17	50	27	23	15	8	60%	96%
5	18	50	27	22	15	7	62%	98%
0	2	100	52	45	31	14	65%	99%
0	6	100	52	52	33	19	62%	99%
0	20	100	52	47	31	16	63%	99%
3	3	100	52	48	28	20	56%	92%
3	13	100	52	57	34	23	59%	97%
3	18	100	52	49	32	17	63%	99%
5	5	100	52	51	30	21	57%	94%
5	8	100	52	44	26	18	56%	93%
5	15	100	52	55	33	22	59%	97%

Table 10. Top Pound FeedForward Networks based on Lags and Out of Sample size - Weekly

Lags	Neurons	Out of Sample	Observed Depreciation	Predicted Depreciation	Depreciation Correct	Depreciation Incorrect	Sign Prediction	Market Timing
0	1	50	25	23	16	7	68%	99%
0	8	50	25	26	17	9	66%	99%
0	9	50	25	26	17	9	66%	99%
3	3	50	25	26	18	8	70%	99%
3	13	50	25	27	20	7	76%	99%
3	20	50	25	27	19	8	72%	99%
5	1	50	25	24	15	9	62%	97%
5	14	50	25	25	16	9	64%	98%
5	17	50	25	28	17	11	62%	98%
0	2	100	52	52	38	14	72%	99%
0	3	100	52	48	33	15	66%	99%
3	2	100	52	46	31	15	64%	99%
3	15	100	52	54	33	21	60%	98%
3	20	100	52	47	30	17	61%	99%
5	3	100	52	48	29	19	58%	96%
5	4	100	52	53	32	21	59%	97%
5	14	100	52	53	34	19	63%	99%

Table 11. Top Yen FeedForward Networks based on Lags and Out of Sample size- Daily

Lags	Neurons	Out of Sample	Observed Depreciation	Predicted Depreciation	Depreciation Correct	Depreciation Incorrect	Sign Prediction	Market Timing
0	8	50	25	22	17	5	74%	99%
0	19	50	25	27	18	9	68%	99%
0	20	50	25	24	16	8	66%	99%
3	9	50	25	25	17	8	68%	99%
3	15	50	25	24	16	8	66%	99%
3	16	50	25	27	16	11	60%	96%
5	3	50	25	25	17	8	68%	99%
5	17	50	25	25	16	9	64%	98%
5	20	50	25	25	16	9	64%	98%
0	1	100	52	50	34	16	66%	99%
0	3	100	52	53	35	18	65%	99%
0	5	100	52	50	34	16	66%	99%
3	13	100	52	46	33	13	68%	99%
3	14	100	52	49	32	17	63%	99%
3	17	100	52	49	32	17	63%	99%
5	11	100	52	47	37	10	75%	99%
5	12	100	52	45	34	11	71%	99%
5	14	100	52	49	34	15	67%	99%

Table 12. Top Euro FeedForward Networks based on Lags and Out of Sample size - Daily

Lags	Neurons	Out of Sample	Observed Depreciation	Predicted Depreciation	Depreciation Correct	Depreciation Incorrect	Sign Prediction	Market Timing
0	1	50	24	24	19	5	80%	99%
0	3	50	24	27	19	8	74%	99%
0	19	50	24	27	20	7	78%	99%
3	2	50	24	22	14	8	64%	98%
3	3	50	24	24	14	10	60%	95%
3	7	50	24	25	15	10	62%	97%
5	1	50	24	26	18	8	72%	99%
5	10	50	24	25	15	10	62%	97%
5	17	50	24	25	16	9	66%	99%
0	7	100	50	50	37	13	74%	99%
0	14	100	50	50	36	14	72%	99%
0	19	100	50	50	38	12	76%	99%
3	1	100	50	44	26	18	58%	96%
3	5	100	50	53	31	22	59%	97%
3	9	100	50	55	32	23	59%	98%
5	1	100	50	48	30	18	62%	99%
5	9	100	50	50	33	17	66%	99%
5	18	100	50	50	31	19	62%	99%

Table 13. Top Pound FeedForward Networks based on Lags and Out of Sample size – Daily

Lags	Neurons	Out of Sample	Observed Depreciation	Predicted Depreciation	Depreciation Correct	Depreciation Incorrect	Sign Prediction	Market Timing
0	1	50	24	25	16	9	66%	99%
0	7	50	24	24	16	8	68%	99%
0	11	50	24	24	16	8	68%	99%
3	1	50	24	22	13	9	60%	95%
3	10	50	24	26	15	11	60%	95%
3	14	50	24	20	12	8	60%	95%
5	1	50	24	22	11	11	52%	70%
5	7	50	24	26	13	13	52%	72%
5	20	50	24	28	17	11	64%	99%
0	2	100	47	50	32	18	67%	99%
0	4	100	47	47	31	16	68%	99%
0	7	100	47	50	35	15	73%	99%
3	1	100	47	52	28	24	57%	95%
3	5	100	47	52	29	23	59%	98%
3	16	100	47	48	27	21	59%	97%
5	15	100	47	48	26	22	57%	94%
5	16	100	47	53	29	24	58%	96%
5	19	100	47	48	23	25	51%	64%

From the above tables there is a general improvement over the NARX results when comparing sign prediction. The FF networks outperform the NARX networks when you compare the sign prediction and the number of networks with a 99% significance level for market timing.

Most interesting is the dramatic increase in performance of the predictive power of the Yen in both the daily and weekly data sets. The FF networks are able to achieve a high of 75% accuracy (sign prediction) in forecasting the Yen. Compared to the NARX networks where the high was 54% (daily) this is a dramatic increase. This perhaps concludes that the NARX network is not a favorable method for Yen forecasting while the FF networks seem to be a very good method for forecasting all three currencies.

Not only did the Yen improve in performance when compared to the NARX, but the other two currencies did as well. The Euro has a high sign prediction of 78% (0 lags, 19 neurons, daily) and the pound has a high of 73% (0 lag, 7 neuron, daily). All currencies had a greater number of networks with significant market timing. There were even more than listed above but only the top 3 for each of the lag/out of sample were reported.

VII. Conclusion

This paper examined the ability to forecast the exchange rate of the Euro, Pound, and Yen using NARX and feedforward neural networks. After networks are trained and out of sample forecasts made, ANNs show to be an effective method in the forecasting of exchange rates. The NARX networks exhibited that using technical indicators as input values can lead to significant market timing ability. Both network types proved to forecast at a higher accuracy (sign prediction) than random walk and ARMA models discussed in the literature.

However, further testing could be applied to potentially improve the predictive power of the NARX networks. One possible such improvement would be to optimize the NARX network's parameters (number of neurons, training algorithms, etc) to achieve better results. The NARX network structures applied in this paper have a limitation in that their structure is based on the intuition of the author (i.e. the number of neurons, lags, training algorithms, etc) which could bias the results downward by not having tested the best combination. For instance, the NARX network combinations were limited in testing and further research would be to explore other parameter set ups outside the 1-5 lag, 15 neuron set up. This could also fix the poor performance on Yen forecasting.

The feedforward network's results show that they are a great method for forecasting exchange rates of all three currencies. The number of networks with significant market timing power was numerous. The more favorable results in the FF networks may be attributed to the wider range of networks tested[6]. However further exploration into the NARX networks is needed to see if this is true or not.

The overall better performance of the Euro and Pound (NARX and FF) could be due to the similarities to the USD. Interest rates and trade are just two of the factors that play

[6] FF networks train significantly faster than NARX networks. For this reason, a wider range of networks were able to be tested.

a role in exchange rate movements and the United States interest rates move more in unison with the European rates over the 1980-2010 period than they do to Japanese rates. Similarly, the United States conducts a larger portion of trade with European countries over the time period studied and has a more equal trade share with Europe (imports are close to exports) whereas the United States imports a great deal more than it exports to Japan.

References

Diaconescu, E. (2008). The use of NARX Neural Networks to predict chaotic time series. *WSEAS Transactions on Computer Research.* Vol 3. No 3. 182-191.

Demuth, H., Beagle, M., and Hagan, M. (2010). Neural Network Toolbox 6: User's Guide. Natick, MA: The MathWorks, Inc.

Frances, P., & Van Homelen, P. (1998). On Forecasting Exchange Rates using Neural Networks. *Applied Financial Economics*. Vol 8. 589-596.

Henriksson, R., & Merton, R. (1981). On Market Timing and Investment Performance II: Statistical Procedures for Evaluating Forecasting Skill. *The Journal of Business.* Vol 54. No. 4. 513-533

Henriksson, R., & Lessard, D. (1982). The Efficiency of the Forward Exchange Market: A Conditional Nonparametric Test of Forecasting Ability. Research Program in Finance Working Papers 130, University of California at Berkeley.

Henriksson, R. (1984). Market Timing and Mutual Fund Performance: An Empirical Investigation. *The Journal of Business*. Vol 51 No. 1. 73-96.

Jamal, A. M., & Sundar, C. (1997). Modeling Exchange Rates with Neural Networks. *Journal of Applied Business Research.* Vol. 14. No 1. 1-5.

Kamruzzaman, J., & Sarker, R. (2003). Comparing ANN Based Models with ARIMA for Prediction of Exchange Rates. *ASBOR Bulletin.* Vol 22. No. 2

Kuan, C., & Liu, T. (1995). Forecasting Exchange Rates Using Feedforward and Recurrent Neural Networks. *Journal of Applied Econometrics.* Vol 10. No. 4. 347-364.

Leung, M., Chen, A., and Daouk, H. (2000). Forecasting exchange rates using general regression neural networks. *Computers & Operations Research.* Vol. 27, no. 11, pp. 1093-1110.

Meese, R., & Rogoff, K. (1983). The Out-of-Sample Failure of Empirical Exchange Rate Models: Sampling Error or Misspecification?. Exchange Rates and International Macroeconomics. Jacob Fenkel (Ed.). Chicago, IL: University of Chicago Press.

Nag, A., & Mitra, A., (2002). Forecasting Daily Foreign Exchange Rates Using Genetically Optimized Neural Networks. *Journal of Forecasting.* Vol 21. 501-511.

Yao, J., & Tan, C. (2000). A Case Study on Using Neural Networks to Perform Technical Forecasting of Forex. *Neurocomputing.* Vol 34. 79-98.