Using nonlinear neurogenetic models with profit related objective functions to trade the US Tbond future

Zac Harland
Krueger Research
zachar@dial.pipex.com

Trading decision models are often designed by optimizing standard statistical measures of error to create forecasts from which trading rules are then indirectly synthesized. In the context of a bond futures trading model, we find that by directly optimizing a number of profit related objective functions, superior results are attained than when using mean square error. Moreover, by using a genetic training method we gain flexibility with regards to the choice of objective function by dispensing with the need to operate within a gradient based framework.

1 Introduction

Although much has been written recently on the use of neural networks and computational learning techniques in financial forecasting and the design of trading systems, the majority of this research has concentrated on minimizing standard statistical measures of predictability rather than directly optimizing a profit related objective function – for recent examples see Refenes et al.(1993). This is not altogether surprising as many of these advanced techniques have their foundation in other fields where minimizing error functions such as mean square error (MSE) is more meaningful to the task at hand. However, in financial forecasting practitioners are more interested in real trading profits and the risk associated with attaining those profits than they are in 'inferred' profitability derived from intermediate measures of predictive accuracy, whose relationship to trading returns is somewhat tenuous.

The observation that statistical error measures may be unsuitable is not new; Leitch (1991) finds that, in the context of treasury bill forecasts, predictive accuracy as measured by MSE is not strongly related to returns. A study by Caldwell (1995) shows that lower measures of MSE and Mean Absolute Error(MAE) did not correlate well with trading profits and in some cases were positively correlated (lower error resulted in lower returns and vice versa). LeBaron (1993) also finds evidence of a weak connection between MSE measures and trading profits. When minimizing traditional error measures such as MSE more emphasis is placed on reducing an error where the network predicts a positive move of +3%, compared to an actual move of +1%, than in reducing the error in a predicted move of –0.5% for the same actual move. However, from a trading perspective, the latter error is least desirable especially if the decision is to sell when the predicted move is negative. This is one reason why metrics based on direction may be more closely correlated to actual returns. Using some kind of threshold on the network output may help to alleviate the problem but it is probably more efficient to optimize the desired trading rule more directly.

In this paper we describe a trading model that utilizes neural networks trained with a genetic training algorithm to trade the 30 year US Treasury bond futures contract. To address the issue of sub-optimal objective functions we directly optimize a number of profit related functions, including the well known Sharpe Ratio (Sharpe, 1966), and compare the results with a model which minimizes MSE. The main objective was to produce a tradable model and so to this end we endeavored to keep all aspects of the model as simple as possible.

Much of the other work on training neural networks with profit related error functions has tended to concentrate on modifying the standard gradient descent learning algorithms such as backpropagation - usually designed to minimize the MSE between target and output - to minimize an objective function more directly related to the trading process. Bentz and Refenes(1994) use a technique whereby the backpropagation algorithm is modified so that, using a sigmoidal weighting function, more emphasis is placed on observations where there is directional error. Choey and Weigend(1996) present a gradient ascent method for directly optimizing the Sharpe Ratio as the cost function. Other related work in this area includes that of Bengio(1996) and Moody(1996).

Approaching the problem from a gradient based standpoint, requiring modification of the gradient descent learning algorithm, is somewhat restrictive as it requires that the function which captures the problem's optimization objective be differentiable. Moreover, this procedure can be fairly complex and limits the space of possible objective functions. What is arguably more useful is a method that is able to train a neural network on any function of interest including those that are not continuous. For example, functions such as, "if network output > 0 then long else short" are not easy to accommodate within a gradient based framework. This is where stochastic training methods such as those using genetic algorithms can be very useful. In the case of the genetic training algorithm all that is required is a comparative measure of 'fitness', without any need for gradient based information. This allows more flexibility in the choice of objective function.

2 Genetic Algorithms

The concept of Genetic Algorithms (GAs) was defined by John Holland (1975). Holland showed that a variety of optimization problems could be solved by a method that he called the Genetic Algorithm. GAs are especially useful when the function to be optimized is discontinuous or noisy (such as a trading related function) and does not lend itself easily to other gradient based approaches. Its effectiveness is partly based on its ability to simultaneously search large regions of the parameter space and converge to regions of high fitness (although not necessarily the global optimum).

The technique is based on the principle of Darwinian evolution and the "survival of the fittest" and follows this general pattern:

1) First a suitable representation scheme is identified for the problem of interest. In this case it is the neural network weight vectors that are encoded in such a way as to represent candidate solutions to the problem.
2) An initial randomly created population of solution candidates is created.
3) Each member of the population is assigned a fitness measure depending on the nature of the problem.
4) A new generation is then created by repeatedly iterating through the following three steps:
    - Reproduce an existing member by copying it directly into the new population.
    - Create two new offspring solutions by performing a crossover operation on two parent solutions (parents are selected randomly but biased towards high fitness).
    - Create a new solution from an existing one by random mutation – this particular operation is generally constrained to have a low probability of occurrence.

5) The process is then halted when the population has converged on an acceptable solution.

GAs can be used in a number of ways in ANN design including but not limited to, input feature selection, optimization of network architecture, training algorithms etc. In this particular application we use GAs as a training algorithm with all other network parameters being decided a priori.

3 Objective Functions

The decision as to which objective function to use is not so clear cut - it is unlikely that a single objective function will capture all the features relevant to the trading process. In this paper we use four different objective functions: MSE, the Sharpe Ratio (SR), Total log returns (TLR) and a variation of TLR which includes a penalty term (TLRP). Although the SR is used extensively in the financial community as measure of risk adjusted returns, it is generally criticized as it penalizes both negative and positive returns equally. TLR is somewhat similar to the SR objective but without the risk adjustment however; its drawback is that it does not take into account the total number of trades – a factor that becomes increasingly important when transaction costs are taken into account. In the end the choice of objective functions was guided by the need for simplicity.

For the MSE, SR and TLR functions the models take on either a Long or Short position $S_t \in \{1,-1\}$ in the futures contract and as such are always in the market. The MSE model simply minimizes the mean square error between the network output and the target, which in this case is one day log returns. After training the following rule is applied:

If output ≥ 0.0 then long, else short.

The TLR model maximizes the following:

Define daily log returns as,

$$r_t = \log(price_{t+1} / price_t) \tag{1}$$

$$\text{Total log returns} = \sum_{t=1}^{N} p_t(r_t) \tag{2}$$

where $p_t = \begin{cases} 1 \text{ if } o_t \geq 0.0 \\ -1 \text{ otherwise and} \end{cases}$

o_t is the network output.

The SR model maximizes the Sharpe Ratio defined as,

$$\text{SR} = \frac{r^a - r^f}{\sigma} \tag{3}$$

where r^a = actual returns,

r^f = risk free returns, and

σ = standard deviation of r^a.

The SR provides a measure of risk adjusted returns. Actual returns are calculated on a monthly basis, averaged and then annualized. We actually remove the risk free returns term as it cancels with the interest that is ordinary earned from margin in a trading account.
The TLRP model can be long, short or neutral $S_t \in \{1,-1,0\}$. It has a penalty term which simply penalizes incorrect daily direction by increasing the resultant daily loss by a certain percentage in an attempt to raise overall trading accuracy.

$$\text{TLRP} = \sum_{t=1}^{N} d_t((r_t) + w_t(r_t)) \quad (4)$$

where $d_t = \begin{cases} 1 \text{ if } o_t > 0.5 \\ -1 \text{ if } o_t < -0.5 \\ 0 \text{ otherwise} \end{cases}$

and o_t is the network output,

and $w_t = \begin{cases} g \text{ if } d_t(r_t) < 0.0 \\ 0 \text{ otherwise} \end{cases}$

Four values for the penalty term, g are chosen: 0.0, 0.1, 0.15 and 0.20.

4 Data

The data used in this study consist of the open, high, low and close of both the US Tbond and S&P500 nearest to expiry futures contracts, covering the period from 12th April 1983 to 30th June 1998 - 62 separate contracts in all. As explained later, S&P500 data is used for additional inputs to the models. It was necessary to create two adjusted time series from the raw prices. For total return calculations in contract points the Bond series is converted into a continuous contract series – raw contract prices adjusted to take into account the spread at rollover between the nearest contract to expiry and the next contract. This is accomplished by establishing the spread between the nearest contract and the next nearest contract at rollover and then adding the cumulative spread up to that contract to the new contract prices. Contracts are rolled over when the trading volume of the next contract is equal to, or greater than, that of the present contract. This splicing creates a new series with the contract rollover distortions removed. All exchange holidays are removed from the data.

Where first log differences are required we use the unadjusted raw prices except on the two days at rollover, where the appropriate one day overlapping prices of each contract are used to calculate the respective log difference of the data (see Eq.1).

Because futures price data are already inherently noisy, adjusting for the spread is essential. Without this adjustment a potentially large artificial drop/rise at each rollover can result. In the case of the US Bond futures price this spread is mostly negative in the period under study due to the financing rate at the time being less than the income or coupon received from the underlying bond. Over the period of 1983-1998 the total spread results in a 40 point artificial drop in the unadjusted prices. This artificial drop will tend to cause a negative bias in a model using only unadjusted prices.

5 The BDS test

Before embarking on the relatively costly and time intensive task of designing a prediction model for financial time series it is prudent to ascertain whether or not the data will lend itself to such a task. With this in mind we use the test of Brock, Dechert, and Scheinkman (1987), the BDS test, as an initial diagnostic of potential predictability within the data series. The BDS test is a powerful statistic and tests the null hypothesis that a time series $\{x_t\}$ is independent and identically distributed (iid) against a non-specific alternative. Let $\{x_t, t = 1,...,T\}$ be a time series, and denote $X_t^M = (x_t, x_{t+1}, ..., x_{t+M-1})$ a point in the M-dimensional Euclidean space. The BDS test develops a statistic based on the correlation integral, defined as:

$$C_M(\varepsilon,T) = \frac{2}{T_M(T_M-1)} \sum_{t<s} I_\varepsilon(X_t^M, X_s^M) \quad (5)$$

where $T_M = T - M + 1$ is the number of M-histories constructed from the sample of length T, and $I_\varepsilon(X_t^M, X_s^M)$ is an indicator function defined as,

$$I_\varepsilon(X_t^M, X_s^M) = 1, \text{if } \left\| X_t^M - X_s^M \right\| < \varepsilon \quad (6)$$
$$= 0, \text{ otherwise}$$

and $\| \ \|$ denotes the sup norm.

The correlation integral gives us the fraction of all possible pairs of points that are within a distance ε of eachother.

The test statistic is

$$\text{BDS}_\text{M}(\varepsilon, \text{T}) = \frac{\sqrt{T_M}\left[C_M(\varepsilon,T) - C_1(\varepsilon,T)^M\right]}{\sigma_M(\varepsilon,T)} \quad (7)$$

and has a limiting standard normal distribution. Under the null hypothesis that $\{X_t\}$ is iid, the term, $\sqrt{T_M}\left[C_M(\varepsilon,T) - C_1(\varepsilon,T)^M\right]$, has a normal limiting distribution with mean zero and standard deviation, $\sigma_M(\varepsilon,T)$. The null hypothesis is rejected if the probability of any two *M*-histories being close together is greater than the *M*th power of the probability of any two points being close together.

The test can also serve as a general model specification test by applying the test to the residuals of any time series model – a properly specified model should produce iid residuals. In order to test for possible nonlinear dependencies it is common practice to remove any linear dependence which may be present by filtering or prewhitening the series by fitting a linear autoregressive model and then analyzing the residuals to check for structure beyond linearity. In this case we test the log difference series, the residuals of an AR(1) and AR(4) model fitted to the series and randomly shuffled versions of all three.

Table 1 shows the results of the BDS statistic for embedding dimensions M ranging from 2 to 5, with ε = 1 and 0.5 times the standard deviation for the log first differences of the original series, the whitened AR(1) and AR(4) residuals and, for ε=1, the mean of 50 shuffled (with replacement) versions of all three series.

Table 1.

ε	M	*Log Diff*	*AR(1)*	*AR(4)*	*LogDiff shuffled*	*AR(1) Shuffled*	*AR(4) Shuffled*
1	2	2.16	2.32	2.33	0.35	0.166	-0.039
1	3	3.79	3.93	3.94	0.17	0.128	-0.096
1	4	4.90	5.07	5.09	0.20	0.179	-0.0165
1	5	6.15	6.33	6.35	0.24	0.207	-0.193
0.5	2	2.04	2.43	2.41			
0.5	3	3.57	3.86	3.87			
0.5	4	4.41	4.71	4.72			
0.5	5	5.36	5.64	5.64			

All statistics excluding the shuffled series are significant at the 5% level (>1.96) with the majority significant at the 1% level. In the absence of a significant difference between the statistics for the raw data and the AR residuals this points to nonlinearities in the underlying data generating process of the series. The possible presence of nonlinear dependence suggests that linear methods will not be sufficient to model the data adequately. The results for the shuffled series show that the underlying structure is destroyed by this process and lends further support to the results.

6 Inputs

Given that many of today's financial markets are interrelated and global in nature it is possible that including inputs from related markets may help to improve performance. We decided to use both Bond and S&P500 data in the model. There is a 0.3 linear correlation between daily log differences of the Bond and S&P500 daily futures data series from 83-98 – an indication that certain interrelationships may be exploitable in an effort to improve the model.

Choice of inputs is of paramount importance when building financial prediction models. Although neural networks have been proven to be powerful function approximators it is still necessary to use extensive pre-processing of the input variables – it is unlikely that simply presenting inputs derived from price data of the target series at arbitrary lags will produce usable results. Theoretical concerns aside, were this the case these 'anomalies' would quickly be priced out of the market as various participants discovered them. Almost by definition, any pre-processed variable which contains predictive accuracy must be difficult to discover or it would no longer exist. In an effort to find meaningful input features we tested a number of the popular technical indicators such as, MACD, RSI, Bollinger bands etc. We came to the conclusion that many of these more commonly known technical indicators owe their use to the promotional skills of their creators, rather than from proper analysis of their correlation to future price changes. In light of this, we ended up in using a number of digital filter design techniques from the field of signal processing in an attempt to arrive at more useful and robust input features.

Given that the models are based on daily data and will produce an output at the close of each day, we focused our attention on inputs that showed correlation with future daily prices. We reasoned that if it was possible to find inputs with stable correlation to 1 and 2 day future returns it should help to circumvent the problem of the model being biased to either trending or sideways markets – a common problem in many conventional trading models. Various statistical measures were used including parametric and non-parametric correlation, mutual information and a technique using ANOVA, similar to that proposed by Burgess and Refenes (1995). The main criteria being that any correlation exhibited by a potential input candidate had to be as constant as possible over the full length of the training set.

In all, 5 inputs were formed from 3 separate price transformations. Price transformations derived from the S&P500 futures contract consisted of a momentum type indicator at time t, along with lags at $t-20$ and $t-25$. The two inputs from the bond series consisted of another momentum type indictor at time t, together with an input that measured the predictive correlation of this input with future prices (lagged appropriately). All inputs were standardized to zero mean and unit variance. In order to lessen the effect of outliers the inputs were then put through the tanh function, which has the effect of compressing outliers.

7 Model design and methodology

Neural networks can be powerful function approximators and thus lend themselves well to financial time series prediction, a form of 'weak' modeling in which the underlying equations of the system are not readily available (unlike so called 'strong' models). However, weak models gain their flexibility via the use of many parameters which can lead to the danger of overfitting the data. This concept is embodied in the so called bias/variance trade off. The modeler's objective is to produce a statistical model of the underlying data generating process which then generalizes well out of sample. If too many parameters are used there is a tendency for the model to overfit the data which leads to poor generalization (the model has a high variance in the estimated parameters and a low bias in that it has the ability to fit a wide variety of functions). Conversely, if too few parameters are used it may be too inflexible to extract the relevant features of the underlying process (it is said to have a high bias and low variance). The ultimate goal is to have a model with low variance and low bias. Given that financial time series tend to have a very low signal to noise ratio, overfitting the data is a real possibility. As such, we lean towards the low-variance/high-bias choice and constrain the network's ability to overfit the data by restricting the number of hidden neurons used.

There are different approaches when it comes to deciding how much data to use for training a network. One view is that the market is always changing and therefore one does not want to use data too far back in history as there is a danger that much of it will be redundant. The other approach is to use as much data as is available, reasoning that the only way to have confidence in the model's final results is if it has acceptable performance over as long a data history as possible. We subscribe to the latter approach and therefore use as much of the available data to train the networks whilst making sure to leave a large enough out of sample period for final analysis.

Initially, the data set was divided into training, validation and test sets as shown in table 2. All networks had 1 hidden layer with 5 input units and 1 output unit. Input units were linear while tanh units were used for the remaining layers. In an attempt to deal with the bias/variance trade off and to achieve *structural stabilization* within the model, networks with hidden layers of 1 to 10 units were tested. The objective was to find the network architecture which relied on the least amount of parameters (hidden units) yet delivered the most consistent performance. As we wished to use as much data to train the final networks as possible the following training regime was decided on.

First, all potential network architectures for each objective function were trained to convergence on the training set data with the resulting validation set performance analyzed. Then both sets were alternated and the process repeated. This allowed out of sample performance to be monitored on both training and validation data in an effort to make sure that the final architecture chosen had stable performance. It was found that networks with 2 hidden units produced the most stable results over all objective functions. Although some networks with more hidden units produced better performance, this performance varied too widely across individual trials.

Finally, networks for each objective function were trained to convergence on the joined training and validation sets (see Table 3) and their performance analyzed on the final 935 day testing set. Twenty training runs, each with populations of 150 networks, were used for each objective function (the total of 140 runs took approximately 170 hours on a 200Mhz PC). The outputs of the top 10 performing networks were then averaged to produce the final results for each objective.

Table 2.

Set	Dates	Length
Training set	12th April 1983 to 6th January 1989	1456 days
Validation set	7th January 1989 to 7th October 1994	1456 days
Final testing set	10th October 1994 to 30th June 1998	935 days

Table 3.

Final sets		
Training set	12th April 1983 to 7th October 1994	2912 days
Final testing set	10th October 1994 to 30th June 1998	935 days

8 Results

The final results for all objective functions over the training period of 2912 days are presented in Table 4. Also shown are the results for the benchmark buy and hold (B&H) strategy (in practice this would involve rolling over a long position at each rollover date). No slippage or transaction costs were taken into account.

Table 4. In sample results for all objective functions based on data from 1983 to 1994.

	TLR	*MSE*	*SR*	*TLRP 0%*	*TLRP 10%*	*TLRP 15%*	*TLRP 20%*	*B&H*
Total Log Returns	88.08	44.08	79.29	66.46	52.62	45.22	50.2	28.4
Avg log trade	0.17	0.09	0.15	0.13	0.15	0.19	0.2	na
Total net profit	$177.08	$97.78	$166.22	$139.23	$113.81	$100.84	$104.80	$58.43
Gross profit	$385.14	$318.12	$381.58	$313.28	$222.29	$164.88	$162.12	na
Gross loss	-$208.06	-$220.34	-$215.36	-$174.05	-$108.48	-$64.04	-$57.32	na
Sharpe Ratio	1.46	0.98	1.30	1.23	1.23	1.26	1.43	0.54
% of time in Market	100%	100%	100%	68%	42%	30%	22%	100%
Total # of trades	511	463	538	496	349	237	252	na
Number winning trades	301	255	319	300	216	163	168	na
Number losing trades	210	208	219	196	133	74	84	na
Percent profitable	59%	55%	59%	60%	62%	69%	67%	na
Largest winning trade	$9.75	$11.63	$9.75	$9.53	$9.53	$9.53	$9.75	na
Average winning trade	$1.28	$1.25	$1.20	$1.04	$1.03	$1.01	$0.97	na
Largest losing trade	-$4.19	-$5.59	-$3.91	-$4.09	-$3.13	-$2.97	-$3.00	na
Average losing trade	-$0.99	-$1.06	-$0.98	-$0.89	-$0.82	-$0.87	-$0.68	na
Ratio avg win/avg loss	1.29	1.18	1.22	1.18	1.26	1.17	1.41	na
Avg trade(win & loss)	$0.35	$0.21	$0.31	$0.28	$0.33	$0.43	$0.42	na
Max consec. winners	9	9	10	10	12	15	13	na
Max consec. losers	9	8	6	5	7	3	8	na
Avg # days in winners	5	6	5	4	4	3	3	na
Avg # days in losers	6	6	5	4	4	4	3	na
Max intraday drawdown	-$9.67	-$15.45	-$11.65	-$9.98	-$8.47	-$5.46	-$4.98	-$20.21

Notes: All log figures are multiplied by 100. Currency figures are in thousands of US dollars. All results are based on trading one US Treasury bond futures contract.

Of those models that are always in the market, it is clear that the MSE objective delivers the worst performance. Apart from lower total return (approximately half that of the TLR objective) and average trade statistics, it also suffers from a comparatively large drawdown of $15450.00. Inspection of the cumulative equity curves for the MSE, SR and TLR objectives in Figure 1 also show the poor performance of the MSE objective, including a flat period in the cumulative equity curve from 1988 to 1993, followed by the large drawdown previously mentioned (the vertical line at the right of the graph shows the division between in and out of sample results). The SR and TLR objectives fair much better but are closely correlated. This is evident in the similarity of their equity curves and also in the percentage profitable figures, which are equal at 59%, with the TLR network delivering the superior performance overall. All the models outperform the B&H strategy. The fact that the SR model did not perform as well as the TLR model is a possible indication that there may not be enough information within the input features to take advantage of the inherent SR risk adjustment.

The out of sample performance of these models (see table 5 and figure 3) is fairly consistent with the in sample results. The MSE model continues to deliver relatively poor performance, including another large drawdown of $20700.00 – far too high a figure for this model to be traded with real money.

Examination of the TLRP models in Tables 4 and 5, along with their respective equity curves in figure 2, show that as the penalty term increases from 0% to 20%, the percentage of trades that were profitable rises from 60% to 67% in sample. Average trade figures rise from $280.00 to $420.00, along with a decrease in the average days spent in each trade. Like the other models, there seems to be a fair amount of consistency between the in and out of sample results except for the 20% TLRP model, where performance drops out of sample (see figure 4). What is interesting is that it continues to have near 70% accuracy but the average trade decreases from $420.00 to $270.00 as the avg. win/loss ratio drops from 1.41 to 1.03. Even though the TLRP models spend less time in the market than the B&H strategy, all except the 20% objective deliver superior out of sample performance.

Table 5. Out of sample results for all objective functions based on data from 1994 to 1998.

	TLR	*MSE*	*SR*	*TLRP 0%*	*TLRP 10%*	*TLRP 15%*	*TLRP 20%*	*B&H*
Total Log Returns	24.22	17.26	18.35	22.4	15.42	15.02	10.83	11.94
Avg log trade	0.14	0.11	0.09	0.13	0.12	0.14	0.1	na
Total net profit	$63.04	$44.41	$48.16	$57.26	$39.48	$39.97	$28.21	$30.31
Gross profit	$120.71	$104.89	$115.14	$101.89	$77.85	$67.94	$52.63	na
Gross loss	-$57.67	-$60.48	-$66.98	-$44.63	-$38.37	-$27.97	-$24.42	na
Sharpe Ratio	1.80	1.23	1.35	1.92	1.63	2.00	1.49	0.82
% of time in Market	100%	100%	100%	72%	54%	45%	33%	100%
Total # of trades	175	154	199	166	131	113	105	na
Number winning trades	103	86	107	105	81	76	71	na
Number losing trades	72	68	92	61	50	37	34	na
Percent profitable	59%	56%	54%	63%	62%	67%	68%	na
Largest winning trade	$4.88	$4.60	$4.59	$4.59	$3.13	$3.56	$2.53	na
Average winning trade	$1.17	$1.22	$1.08	$0.97	$0.96	$0.89	$0.74	na
Largest losing trade	-$3.22	-$3.31	-$3.22	-$3.16	-$3.16	-$3.12	-$3.00	na
Average losing trade	-$0.80	-$0.89	-$0.73	-$0.73	-$0.77	-$0.76	-$0.72	na
Ratio avg win/avg loss	1.46	1.37	1.48	1.33	1.25	1.18	1.03	na
Avg trade(win & loss)	$0.36	$0.29	$0.24	$0.34	$0.30	$0.35	$0.27	na
Max consec. winners	8	14	7	7	13	10	15	na
Max consec. losers	4	5	5	4	4	5	3	na
Avg # days in winners	5	6	5	4	6	4	3	na
Avg # days in losers	5	6	4	4	4	4	3	na
Max intraday drawdown	-$6.47	-$20.70	-$5.81	-$7.75	-$9.93	-$5.59	-$5.56	-$14.78

Notes: All log figures are multiplied by 100. Currency figures are in thousands of US dollars. All results are based on trading one US Treasury bond futures contract.

With an out of sample sharpe ratio of 2.0, the TLRP15% objective seems to perform well however, on closer inspection it has an associated drawdown of $5.59, compared with that of $6.47 for the TLR objective which gains an extra 40% in profits by being constantly in the market. Moreover, the TLRP15% objective, with a win/loss ratio of 1.17, relies on high trade accuracy for its profit as opposed to the TLR model, which has a win/loss ratio of 1.30 and hence a more desirable balance between trade accuracy and amount gained per trade. The amount of days spent in losers and winners seems to be fairly consistent at 5 days for the TLR objective. It may be possible to the use stops to improve this ratio.

An interesting corollary to these results can be observed by looking closely at the cumulative equity curves for all objectives in figures 1 and 2. From 1985 to 1987 all equity curves experience a steep rise until the time of the October 1987 crash, at which point the curves flatten out, something that is especially noticeable with the MSE objective. This flat period persists until about 1993 at which point the equity curves begin to rise again at a steeper rate. This would seem to indicate that there was a structural shift in the relationship between stocks and bonds at that time. Further research is needed to gain more insight into this observation.

9 Implementation

It should be noted that there are some implementation problems with the final models. The day session of S&P500 futures closes 75 minutes after that of the US Tbond futures. In order to put the models into practice closing prices from both contracts are required at the time of the Tbond close, therefore a modification in the trading strategy is needed. There are a number of possible solutions to this. The simplest method is to trade the Tbond during the evening session at the close of the S&P500 day session. To give an indication of the resulting difference that could be expected we recalculated the TLR results on daily data by assuming trades were made on the following day's opening price and found a difference of less than 1% in total log returns. This difference is likely to be even less if trading took place at the time of the S&P500 close.

10 Conclusion

We have researched the feasibility of using genetically trained neural networks in conjunction with profit oriented objective functions to trade the US Tbond future. The results show some promise in this endeavor in that the networks seem to have captured an on going relationship between the two series, leading to profitable trading of the US Tbond futures contract from 1983 to 1998. Further research is required into the exact nature of this predictive relationship in order to gain additional confidence in the underlying models. Our results lend further support to previous findings that traditional statistical measures of error may be inappropriate for use as objective functions when designing trading models in this way.

Lastly, there is always a danger of 'datasnooping' when building financial forecasting models based on finite datasets. Even if all preventive measures are taken in an attempt to eliminate this possibility, one can never be absolutely certain that is has not taken place. Research and design of trading models tends to be a continual process and it is unlikely that a researcher will arrive at a successful model on the first attempt. It is quite possible that during this ongoing process, a priori knowledge of the full dataset could lead to inadvertent 'datasnooping'. Indeed, supporters of the EMH would argue that most, if not all trading models which outperform the market on a risk adjusted basis, are the result of inadvertent 'datasnooping'.

Figure 1. Cumulative equity curves for TLR, SR and MSE objectives.

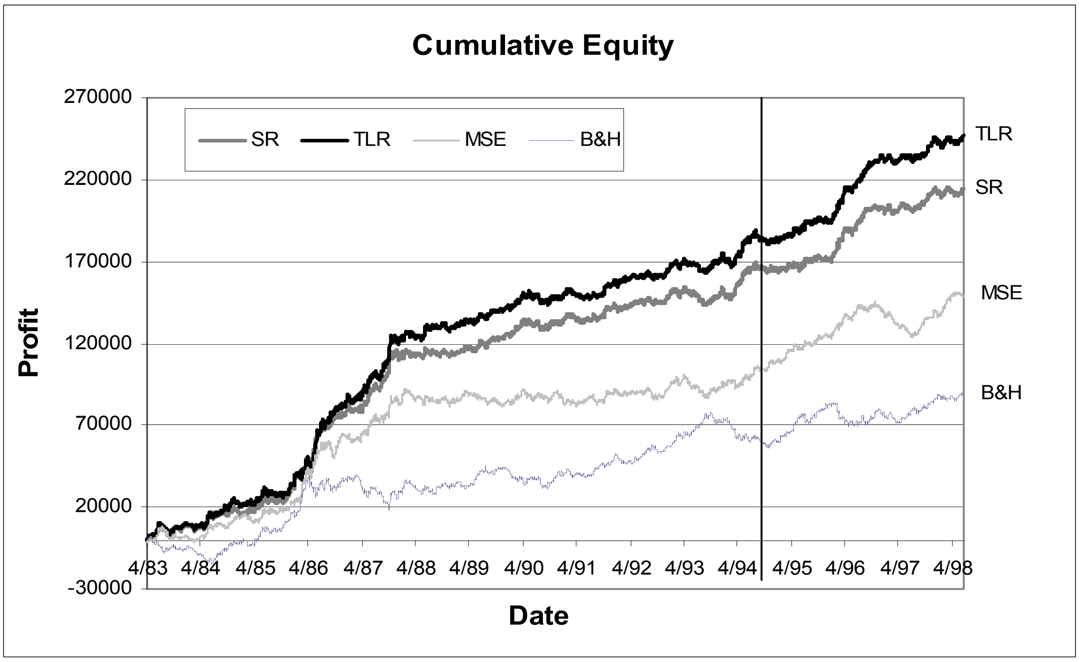


Figure 2. Cumulative equity curves for TLRP objectives.

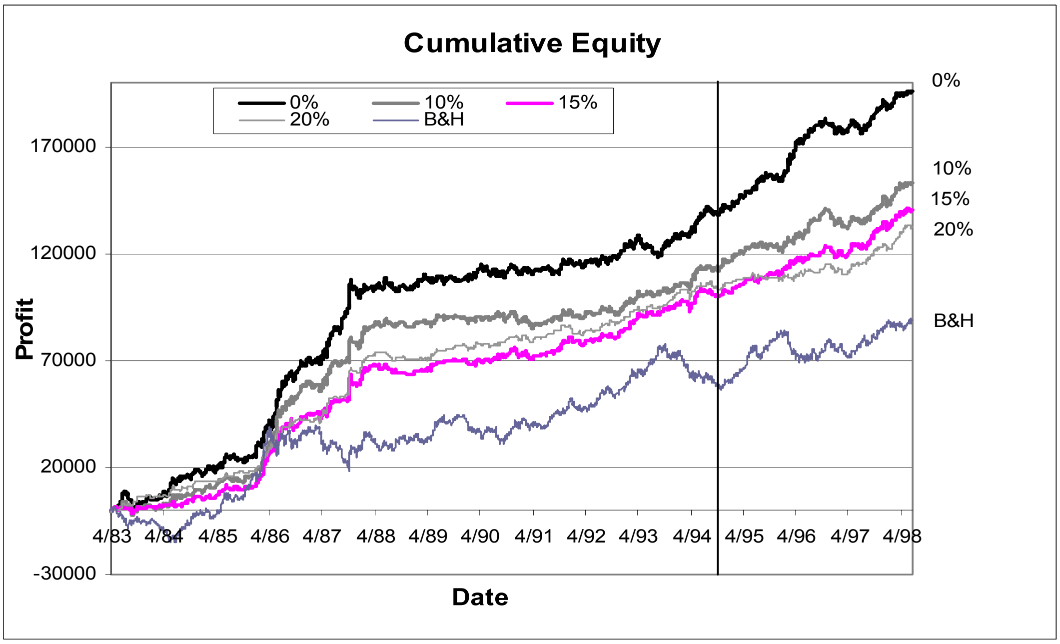
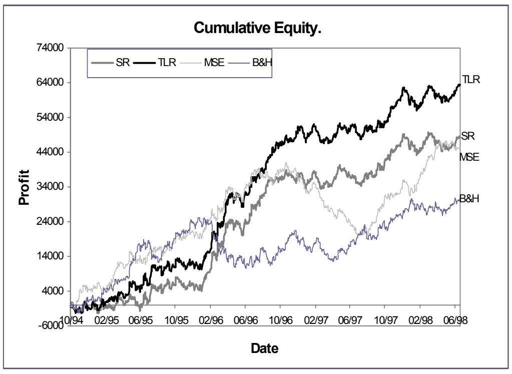
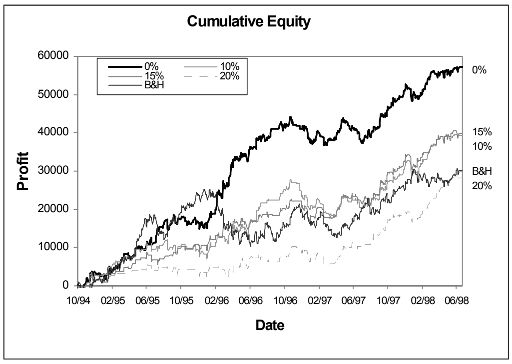


Figure 3.



Out of sample cumulative equity curves for the MSE, TLR and SR objectives.

Figure 4.



Out of sample cumulative equity curves for the TLRP objectives.

References

Bengio, Y. 1996. Training a neural network with a financial criterion rather than a prediction criterion, *Decision Technologies for Financial Engineering* (*Proceedings of the fourth international conference on Neural Networks in the Capital Markets 1996).* World Scientific.

Bentz, Y and A.N.Refenes 1994. Backpropagation with weighted signs and its application to financial time series, *Proceedings of neural networks in the Capital Markets* 1994.

Brock, W.A., W.D. Dechert, and J. A. Scheinkman 1987. A test for independence based on the correlation dimension, working paper, University of Wisconsin Press, Madison, WI.

Burgess, A.N. and A.N.Refenes. 1995. Modelling non-linear cointegration in international equity index futures, *Neural Networks in Financial Engineering* (*Proceedings of the Third International conference on neural networks in the Capital Markets 1995).* World Scientific.

Caldwell, R.B. 1995. Improved prediction performance metrics for neural network-based financial forecasting systems, *Neurove$t journal*, vol.3,No.5, pp. 13-23.

Choey, M. and A.S.Weigand. 1996. Nonlinear trading models through Sharpe Ratio Maximization, *Decision Technologies for Financial Engineering* (*Proceedings of the fourth international conference on Neural Networks in the Capital Markets 1996).* World Scientific.

Holland, J.H. 1975. Adaptation in Natural and Artificial Systems. Ann Arbor: The University of Michigan Press.

LeBaron, B. 1993. Nonlinear diagnostics and simple trading rules for high-frequency foreign exchange rates, *Time Series Prediction: Forecasting the Future and Understanding the Past.* Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol XV. Reading, MA: Addison-Wesley.

Leitch, G. and J.Tanner. 1991. Economic forecast evaluation: Profits versus the conventional error measures. *The American Economic Review* , 580-590.

Moody, J. and W. Lizhong. 1996. Optimization of trading systems and portfolios, *Decision Technologies for Financial Engineering* (*Proceedings of the fourth international conference on Neural Networks in the Capital Markets 1996).* World Scientific.

Refenes, A.N. 1995. *Neural networks in the capital markets,* ed. A. Refenes, John Wiley & Sons, Chichester.

Sharpe, W. F. Jan 1966. Mutual fund performance. *Journal of business ,* 119-138.