WAVELET TRANSFORM, NEURAL NETWORKS AND THE PREDICTION OF S&P PRICE INDEX: A COMPARATIVE STUDY OF BACK PROPAGATION NUMERICAL ALGORITHMS

Salim Lahmiri
Department of Computer Science, University of Quebec at Montreal, Canada.
Email: lahmiri.salim@courrier.uqam.ca

Abstract

In this article, we explore the effectiveness of different numerical techniques in the training of backpropagation neural networks (BPNN) which are fed with wavelet-transformed data to capture useful information on various time scales. The purpose is to predict S&P500 future prices using BPNN trained with conjugate gradient (Fletcher-Reeves update, Polak-Ribière update, Powell-Beale restart), quasi-Newton (Broyden-Fletcher-Goldfarb-Shanno, BFGS), and Levenberg-Marquardt (L-M) algorithm. The simulations results show strong evidence of the superiority of the BFGS algorithm followed by the L-M algorithm. Also, it is found that the L-M algorithm is faster than the other algorithms. Finally, we found that previous price index values outperform wavelet-based information to predict future prices of the S&P500 market. As a result, we conclude that the prediction system based on previous lags of S&P500 as inputs to the BPNN trained with BFGS provide the lowest prediction errors. Key words: Wavelet Transform, Neural Networks, Numerical Optimization, Stock Market, Forecasting.

It is hard to predict the stock market since financial time series are highly irregular and nonlinear. Therefore, traditional linear models such as autoregressive integrated moving average (ARIMA) are not suited to model financial time series. Indeed, ARIMA processes are based on the assumptions that the time series are stationary (or that they can be made stationary), and that the error variables are normally distributed (Box and Jenkins, 1970). Unfortunately, financial data does not appear to meet those criteria. As a result, there has been a large literature on the effectiveness of various soft computing techniques to predict future stock returns (Atsalakis and Valavanis, 2008). For instance, artificial neural networks are effective in realizing the input–output mapping and can approximate any continuous function given an arbitrarily desired accuracy (Cybenko, 1989; Hornik et al., 1989). In addition, there are no prior assumptions on the underlying process from which data are generated (Zhang et al., 1998). In addition, there are no prior assumptions on the model form required in the model building process. Because of the attractiveness of artificial neural networks, a large number of applications have been proposed in recent decades for predicting stock markets using artificial neural networks (Enke et al., 2005; Giordano et al., 2007; Huang et al., 2007). The backpropagation (BP) (Rumelhart et al., 1986) is the most widely used algorithm to train the artificial neural networks (Atsalakis and Valavanis, 2008). However, the BP algorithm suffers from two major drawbacks: low convergence rate and instability. They are caused by a risk of being trapped in a local minimum (Ahmed et al., 2001) and possibility of overshooting the minimum of the error surface (Wen et al., 2000). Over the last years, many numerical optimization techniques have been employed to improve the efficiency of the backpropagation algorithm including the conjugate gradient descent. In addition, some papers have been proposed in the literature to compare the prediction accuracy of different backpropagation algorithms with applications in engineering and science. However, there is a need to do so in the case of financial time series prediction in order to identify which algorithm allows achieving higher prediction accuracy. On the other hand, multiresolution analysis techniques such as the wavelet transform which is widely employed in pure science and engineering have received a very little attention in finance. Therefore, it would be interesting to examine the effect of wavelet processed data on the performance of numerical algorithms used to train backpropagation algorithm.

The purpose of our paper is straightforward. First, we aim to compare the financial prediction accuracy of backpropagation algorithm trained with different numerical techniques. Second, we examine the effectiveness of wavelet transform coefficients on the accuracy of prediction. For...
instance, a reference prediction model is designed. It uses previous price index values as inputs to neural networks. Then, the forecasting accuracy of the reference model is compared to the accuracy of neural networks that use wavelet coefficients as predictors.

The remainder of this paper is structured as follows: In section 2 related works are presented. Section 3 deals with our methodology. Section 4 presents the results of simulations, and section 5 concludes.

Related Works

Mokhnache and Boubakeur (2002) compared the performance of three back-propagation algorithms, Levenberg-Marquardt, backpropagation with momentum and backpropagation with momentum and adaptive learning rate to classify the transformer oil dielectric and cooling state in four classes: change the oil, regenerate it, filter it, or keep it. The simulations showed that the back-propagation with momentum and adaptive learning rate improves the accuracy of the backpropagation with momentum and also gives a fast convergence to the net. Kisi and Uncuoglu (2005) compared Levenberg-Marquardt, conjugate gradient and resilient algorithm for stream-flow forecasting and determination of lateral stress in cohesionless soils. They found that Levenberg-Marquardt algorithm was faster and achieved better performance than the other algorithms in training. On the other hand, resilient backpropagation achieved the best test accuracy. In addition, the results showed that the resilient backpropagation and conjugate gradient algorithms are, respectively, the most robust in stream-flow prediction and lateral stress estimation. In sum, they conclude that it is very difficult to conclude which algorithm performs the best for a given problem. Indeed, the performance depends on the problem complexity, on the size of the dataset, and on the number of weights and biases in the network. In the problem of breast cancer diagnosis, Esugasini et al. (2005) compared the classification accuracy of the standard steepest descent back-propagation algorithm against the classification accuracy of the gradient descent with momentum and adaptive learning, resilient back propagation, Quasi-Newton and Levenberg-Marquardt algorithm. The simulations show that the neural network using the Levenberg-Marquardt algorithm achieved the best classification performance. On the other hand, the gradient descent with momentum and adaptive learning rate algorithm produced the lowest accuracy. Finally, the moderate accuracy was obtained with the Quasi-Newton and resilient back propagation algorithm models. The authors concluded that the Levenberg-Marquardt algorithm provides the best performance and it is also efficient compared to the other networks since it requires a lower number of hidden nodes. Iftikhar et al. (2008) employed three neural networks with different algorithms to the problem of intrusion detection in computer and network systems. The learning algorithms considered by the authors were the standard, the batch, and the resilient backpropagation algorithm. They found that initially standard and batch backpropagation algorithms converge more quickly. However, as the time passes the resilient algorithm outperforms the others in terms of time of convergence and minimum error. They conclude that the resilient algorithm had a better performance to the application. Nouir et al. (2008) compared the performance of the standard backpropagation with and Levenberg-Marquardt algorithms to the prediction of a radio network planning tool. They found that the standard backpropagation algorithm achieved the minimum error and then outperforms the Levenberg-Marquardt algorithm.

The wavelet transform is a multi-resolution approximation technique based on low-pass filters capable of providing coarser approximation of the signal and high-pass filters capable of providing finer approximation when it is applied to a non-stationary time series. Wavelets combine powerful properties such as different degrees of smoothness and localization in time and scale. Indeed, wavelet coefficients are capable of revealing aspects of the data such as changes in variance, level changes, discontinuities, sharp spikes, detection of outliers. The wavelet analysis is widely used in science and engineering; however, it is not largely employed in stock market forecasting; indeed a limited number of papers considered wavelet transform in the problem of stock market prediction. For instance, Li and kuo (2008) combined K-chart technical analysis for feature representation of stock price movements, discrete wavelet transform for feature extraction, and a self-organizing map network for to forecast Taiwan Weighted Stock Index (TAIEX). They concluded that the proposed system can help finance professionals making profitable decisions due to the contribution of DWT. Hsieh et al. (2011) integrated wavelet transforms, recurrent neural network (RNN) and artificial bee colony algorithm to forecast Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX). The authors employed the Haar wavelet family to decompose the stock price time series and thus eliminate noise. The recurrent neural network uses numerous fundamental, technical indicators, and denoised signals as predictors. On the other hand, the artificial bee colony algorithm is utilized to optimize the RNN weights and biases. The authors found
that their approach outperformed previous methods found in the literature to predict TAIEX. Wang et al. (2011) used discrete wavelet transform and backpropagation neural networks to predict monthly closing price of the Shanghai Composite Index. For instance, low-frequency signals were fed to neural networks to predict the future value of the stock index. They found that wavelet signals improve the accuracy of neural networks in comparison with previous studies.

**Methodology**

**Neural Networks and Numerical Algorithms**

The Multi-layer perceptron (MLP) networks trained using backpropagation (BP) algorithm are the most popular choice in neural network applications in finance (Atsalakis and Valavanis, 2008). The MLP networks are feed forward neural networks with one or more hidden layers which is capable to approximate any continuous function up to certain accuracy just with one hidden layer (Cybenko, 1989; Funahashi, 1989). The MLP consists of three types of layers. The first layer is the input layer and corresponds to the problem input variables with one node for each input variable. The second layer is the hidden layer used to capture non-linear relationships among variables. The third layer is the output layer used to provide predicted values. In this paper, the output layer has only one neuron corresponding to the prediction result. The relationship between the output \( y_t \) and the input \( x_i \) is given by:

\[
y_t = w_0 + \sum_{j=1}^{q} w_{0,j} \cdot f \left( w_{0,j} + \sum_{i=1}^{p} w_{i,j} \cdot x_i \right)
\]

where \( w_{ij} \) (\( i=0,1,2,\ldots,p; j=1,2,\ldots,q \)) and \( w_j \) (\( j=0,1,2,\ldots,q \)) are the connection weights, \( p \) is the number of input nodes, \( q \) is the number of hidden nodes, and \( f \) is a nonlinear activation function that enables the system to learn nonlinear features. The most widely used activation function for the output layer are the sigmoid and hyperbolic functions. In this paper, the hyperbolic transfer function is employed and is given by:

\[
f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}
\]

The MLP is trained using the backpropagation (BP) algorithm and the weights are optimized. The objective function to minimize is the sum of the squares of the difference between the desirable output \( (y_{t,p}) \) and the predicted output \( (y_{t,d}) \) given by:

\[
E = 0.5 \sum_{t=1}^{n} (y_{t,p} - y_{t,d})^2 = 0.5 \sum_{t=1}^{n} e^2
\]

The training of the network is performed by the well-known Backpropagation (Rumelhart et al., 1986) algorithm trained with the steepest descent algorithm given as follows:

\[
\Delta w_k = -\alpha_k g_k
\]

where, \( \Delta w_k \) is a vector of weights changes, \( g_k \) is the current gradient, \( \alpha_k \) is the learning rate that determines the length of the weight update. Thus, in the gradient descent learning rule, the update is done in the negative gradient direction. In order to avoid oscillations and to reduce the sensitivity of the network to fast changes of the error surface (Jang and Mizutani, 1997), the change in weight is made dependent of the past weight change by adding a momentum term:

\[
\Delta w_k = -\alpha_k g_k + p\Delta w_{k-1}
\]

where, \( p \) is the momentum parameter. Furthermore, the momentum allows escaping from small local minima on the error surface (Castillo and Soria, 2003). Unfortunately, the gradient descent and gradient descent with momentum do not produce the fastest convergence, and even are often too slow to converge. One solution to speed up rate of convergence is to use numerical optimization techniques which can be broken into three categories: conjugate gradient algorithms, quasi-Newton algorithms, and Levenberg-Marquardt algorithm. In particular, this paper compares the prediction performance of the following algorithms: quasi-Newton (Broyden-Fletcher-Goldfarb-Shanno, BFGS), conjugate gradient (Fletcher-Reeves update, Polak-Ribière update, Powell-Beale restart), and Levenberg-Marquardt algorithm. The algorithms are briefly described in Table 1, and are well presented in Scales (1985) and Jorge and Wright (2006).
Table 1. Description of the conjugate, quasi-Newton (secant), and L-M algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Computation of search direction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fletcher-Reeves (conjugate)</td>
<td>$p_0 = -g_0$</td>
<td>(1) iteration starts by searching in the steepest descent direction. (2) Charalambous (1992) search line method is employed to find the optimal current search direction $\alpha$. (3) Next (update) search direction $\beta$ is found such that it is conjugate to previous search directions.</td>
</tr>
<tr>
<td></td>
<td>$\Delta w_k = \alpha_k p_k$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$p_k = -g_k + \beta_k p_{k-1}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\beta_k = \frac{g'<em>k g_k}{g'</em>{k-1} g_{k-1}}$</td>
<td></td>
</tr>
<tr>
<td>Polak-Ribiere (conjugate)</td>
<td>$p_0 = -g_0$</td>
<td>Update is made by computing the product of the previous change in the gradient with the current gradient divided by the square of the previous gradient.</td>
</tr>
<tr>
<td></td>
<td>$\Delta w_k = \alpha_k p_k$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$p_k = -g_k + \beta_k p_{k-1}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\beta_k = \frac{\Delta g'<em>k g_k}{g'</em>{k-1} g_{k-1}}$</td>
<td></td>
</tr>
<tr>
<td>Powell-Beale restarts</td>
<td>$</td>
<td>g'_{k-1} g_k</td>
</tr>
<tr>
<td>(conjugate)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BFGS (quasi-Newton)</td>
<td>$\Delta w_k = -H_k g_k$</td>
<td>$H$ is the Hessian (second derivatives) matrix.</td>
</tr>
<tr>
<td>Levenberg-Marquardt (L-M)</td>
<td>$\Delta w_k = -H_k g_k$</td>
<td>$J$ is the Jacobian matrix (first derivatives) and $e$ is a vector of network errors.</td>
</tr>
<tr>
<td></td>
<td>$H' = JJ'$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$g = J'e$</td>
<td></td>
</tr>
</tbody>
</table>

The Wavelet Approach

The wavelet transform (WT) (Daubechies, 1990) is designed to address the problem of nonstationary signals. The original signal is decomposed into two types of components; namely approximation of the signal (high-scale and low-frequency components), and details of the signal (low-scale and high-frequency components). The continuous wavelet transform (CWT) is defined by:

$$CWT(a, b) = \int_{-\infty}^{+\infty} x(t) \psi^*_{a,b}(t) dt$$

where $x(t)$ represents the analyzed signal, $a$ and $b$ represent the scaling factor and translation along the time axis, respectively, and the superscript (*) denotes the complex conjugate. The function $\psi_{a,b}(t)$ is obtained by scaling the wavelet at time $b$ and scale $a$ as follows:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right)$$

where $\psi(t)$ represents the wavelet. Following the methodology in (Wang et al., 2011), the price index time series are decomposed using the discrete wavelet transform (DWT) with Daubechies (db3) as mother wavelet and with six decomposition levels. Finally, the low-frequency components are used in this paper as inputs to artificial neural networks to predict future stock market price index. Figure 1 shows the block diagram of the wavelet-BP approach.
Lags-BP Approach

This model uses lagged price index values as inputs to the neural networks in order to predict future price values. To find the appropriate lags the following methodology is considered. First, the stock market price level is linearly transformed according to:

\[ R(t) = \log(P(t)) - \log(P(t)) \]

Then, the auto-correlation function \( \tau \) is computed as follows:

\[ \tau_k = \frac{1}{T-k+1} \sum_{t=k}^{T} (R(t) - \bar{R})(R(t-k) - \bar{R}) \left( \sum_{t=1}^{T} (R(t-k) - \bar{R}) \right)^{-1} \]

where \( t \) is time script, \( k \) is a lag order which is determined using the auto-correlation function, and \( \bar{R} \) is the sample mean of \( R(t) \). The appropriate \( k \) is determined following the methodology of Greene (2002) and Brockwell and Davis (2002). Indeed, it is important to include past returns to predict future market directions if the return series are auto-correlated. In other words, history of the returns may help predicting future returns. Figure 2 shows the block diagram of the Lags-BP approach.

Performance Measures

The prediction performance is evaluated using the following common statistics: root mean of squared errors (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), average relative variance (ARV), coefficient of variation (CoV), and mean absolute deviation (MAD). They are given as follows:

\[ RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^{N} (P_t - F_t)^2} \]

\[ MAE = \frac{1}{N} \sum_{t=1}^{N} |P_t - F_t| \]

\[ MAPE = \frac{1}{N} \sum_{t=1}^{N} \left| \frac{P_t - F_t}{P_t} \right| \]

\[ ARV = \frac{\sum_{t=1}^{N} (P_t - F_t)^2}{\sum_{t=1}^{N} (F_t - \bar{F})^2} \]

\[ CoV = \sqrt{\frac{1}{N} \sum_{t=1}^{N} (F_t - \bar{F})^2} \]

\[ MAD = \frac{1}{N} \sum_{t=1}^{N} |F_t - \bar{F}| \]
where \( P_t, F_t, \) and \( \overline{F} \) are respectively the true price, forecasted price and the average of forecasted prices over the testing (out-of-sample) period \( t = 1 \) to \( N \).

**Experimental Results**

The initial sample of the S&P500 daily prices is from October 2003 to January 2008, with no missing values. All neural networks are trained with 80% of the entire sample and tested with the remaining 20%. Figure 3 shows the

\[ P(t) = s + a_i + d_i \]

...decomposition of the index price series – \( P(t) \) – using the wavelet transform.; where \( s, a_i \) and \( d_i \) represent respectively the price index, approximation signal, and details signals at level \( i \). Figure 4 shows the auto-correlation function up to 10 lags. Since \( \tau \) is nonzero for \( k=1 \) and \( k=2 \), it means that the series is serially correlated. In addition, the values of the auto-correlation function die quickly in the first three lags which is a sign that the series obeys a low-order autoregressive (AR) process; for example an AR(2). Therefore, the number of lags to be included in the prediction models is up to \( k=2 \).

The performance of each numerical optimization technique given the type of input (lags or DWT coefficients) in terms of MAD, RMSE, and MAE is shown in Figure 5; in terms of ARV and MAPE is shown in figure 6; and in terms of CoV is shown in Figure 7. For instance, Figure 5 shows clearly that BFGS, L-M, and Powell-Beale outperform Fletcher-Reeves and Polak-Ribiére algorithm whether using lags or DWT approach according to MAD, RMSE, and MAE statistics. In addition, BFGS perform slightly better than the L-M algorithm. On the other hand, the simulations results show evidence that BFGS, L-M, and Powell-Beale provide higher accuracy with previous price index values as inputs than with DWT approximation coefficients. On the other hand, based on ARV and MAPE statistics, BFGS, L-M, and Powell-Beale outperform Fletcher-Reeves and Polak-Ribiére algorithm both for lags and DWT approach (Figure 6). In addition, BFGS perform clearly much better than the L-M algorithm. The simulations results also show...
strong evidence of the superiority of Lags-BP over the DWT-BP approach. Finally, as shown in Figure 7, the coefficient of variation statistic confirms all previous results. In sum, the BFGS algorithm has shown its superiority over the other numerical techniques to approximate the nonlinear relationship between the inputs and the S&P500 price index values. In addition, previous price index values outperforms wavelet approximation signals to predict future prices of the S&P500 market. In terms of time of convergence, Figure 8 shows that the L-M algorithm is in general faster than conjugate gradient algorithms. On the other hand, BFGS quasi-Newton technique is not fast to converge. However, all these algorithms can be implemented for real time forecasting tasks since the maximum time of convergence is 6.29 seconds.

Figure 4. Auto-correlation function $\tau_k$ of S&P500 returns at different lags $k$.

Figure 5. Performance in terms of MAD, RMSE, and MAE

Figure 6. Performance in terms of ARV and MAPE
Conclusion

In this paper, the problem of S&P500 price index prediction using backpropagation algorithm is considered. Gradient descent and gradient descent with momentum are often too slow to converge. One solution to speed up rate of convergence is to use numerical optimization techniques. Numerical methods can be broken into three categories: conjugate gradient, quasi-Newton, and Levenberg-Marquardt algorithm. This paper compares the prediction performance of the following algorithms: conjugate gradient (Fletcher-Reeves update, Polak-Ribiére update, Powell-Beale restart), quasi-Newton (Broyden-Fletcher-Goldfarb-Shanno, BFGS), and Levenberg-Marquardt (L-M) algorithm. In addition, we examine the effectiveness of the discrete wavelet transform as a multiresolution technique to extract valuable information from the S&P500 time series. The performances of the learning algorithms are evaluated by comparing the statistical measures of the prediction error.

The simulations results show strong evidence of the superiority of the BFGS algorithm followed by the L-M algorithm. However, the difference between the two numerical algorithms is very small when ARV, MAPE, and CoV statistic are taken into account. In other words, L-M algorithm is very powerful too. Thus, our findings confirm the results obtained by Kisi and Uncuoglu (2005) and Esugasini et al. (2005). On the other hand, the L-M algorithm is faster than the other algorithms; except Polack-Ribiére using DWT approximation coefficients. Indeed, the LM converges faster than conjugate gradient methods since the Hessian matrix is not computed but only approximated and the use of the Jacobian requires less computation than the use of the Hessian matrix. Finally, unlike the literature, we found that, previous price index values outperforms wavelet approximation signals to predict future prices of the S&P500 market. In other words, in comparison with wavelet approximation coefficients; more recent values of the price index contain more information about its behaviour the following day. As a result, we conclude that the prediction system based on previous lags of S&P500 as inputs to the backpropagation neural networks trained with BFGS provide the lowest prediction errors. For future works, the effectiveness of wavelet extracted signals in finance should be more investigated.
References


