Nov 1st, 3:15 PM - 3:30 PM

ANN in Financial Prediction

Roneda Mucaj
Ismail Qemali University

Alketa Hyso
Ismail Qemali University

ANN in Financial Prediction

Roneda Mucaj, Alketa Hyso
Department of Computer Science, University of Vlora "Ismail Qemali"
Vlore, Albania

Abstract. This paper focuses on the treatment of intelligent systems and their application in the financial area. Types of intelligent systems are numerous, but we will focus on those systems, which based on their ability to learn, are able to predict. The concept of inductive reasoning, how these systems learn and reason inductively, the role and their integration in financial services are some of the concepts that will be addressed. The second and the main part focuses on the application developed in the design of an artificial neural network for financial forecasts. Recognizing the need for better predictive models, not just traditional statistical model, we considered with interest the development of an application that will predict currency exchange rates, USD-ALL, given the time series of real data in years 1995-2012. We test some of the learning algorithms in our system and conclude that one of them is most suitable for this problem. This intelligent system reached to create a relational model of data, on the basis of which is able to output satisfactory results forecast. After the presentation of experimental results, the paper closes with a discussion on possible improvements that could be made in the future.

Keywords: prediction, time series, currency exchange monetary, neural network

1 Introduction

It is said that a computer program learns from experiences E with respect to a particular task T and performance measure P, if its performance at this task T, improves with experience E. This is the definition, given for machine learning and the answer to the question how to create intelligent systems that improve their performance in certain tasks through experience.
One popular technique of implementing the machine learning in prediction is Artificial Neural Networks (ANN). ANN is actually an information processing system that consists of a graph representing the processing system as well as various algorithms. ANN is a complex and sophisticated computer program. It is able to adapt, to recognize patterns, to generalize, and to cluster or to organize data.

2 Knowing the Problem

Our project consists in the *currency exchange* rate forecasts. Monetary exchange market is the largest financial and liquid market. It is influenced by a variety of factors including economic, political events, and even psychological state of traders and investors. These factors are linked and interact with each other in a very complex way. All these interactions are highly volatile and dynamic. This complexity makes prediction harder in international markets.
The ability to predict accurately the rate exchange changes, affects in increasing profits. Trading on time with relatively precise strategies may create huge profit, but a trade based on wrong moves could risk big losses. Using the right analytical tool and appropriate methods, the effect of errors can be reduced and the profitability can be increased.
Pattern recognition performs on monthly data of monetary exchange rates to forecast future exchange rate. This value is forecasting using time series analysis. A suitable type of error analysis is used to determine which function is more efficient.

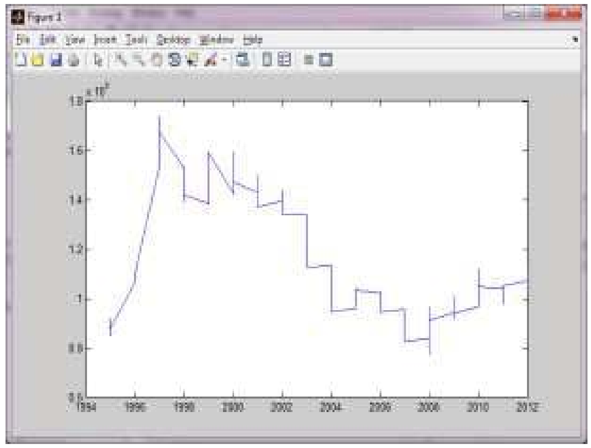


Fig.1 Time series data of exchange rate USD-ALL the years 1995-2012

2.1 Description of the data

We use monthly series of monetary exchange rates. The training and test data are real monthly rates from 1995 to 2012. There are two vectors of data. Time series are sequentially data. The inputs have two dimensions, one is year and the other is month. The output data is with one dimension, the exchange rate.

2.2 Data Processing

The rule of data processing is the learning rule.

$$\Delta\omega = (\text{target} - \text{output})\,\text{input}' = \text{gabimi} * \text{input}'$$

The larger the input vector, the larger is its effect on the vector of weights. Its solution is to normalize the data, thus reducing them. This normalization sets the data in the range 0-1. This improves the performance of the network.

2.3 Partition of the data

In our application, about 65-75% of the data will be used for training and the rest will be used for testing and evaluation. Evaluation set is used to determine the performance of the neural network when data are not trained before. Its primary aim is to avoid overlapping of data during the training phase. Test set used to check the overall performance of the network.

2.4 The choice of transformation functions

Each input vector is weighed with a particular matrix. Bias b is added with the weighted entry and set as entry transfer function. The transfer function gives an output value which can have a value between minus and plus infinity and then normalize it to a value between 0 and 1 or between -1 and 1 depending on the type of function used. Another reason why a function is used is to prevent the influence of noise in the training process of the network. We will use tan-sigmoidal function as a function of the hidden layer and linear function as a function of output layer.

2.5 Creating neural network

This section presents the network architecture with multi-layer feed-forward. We will use the network with two layers, based on research for this application a large and more complicated network would cause overlapping.
The hidden layer of network has three neurons and output layer has one neuron. The transfer function of hidden layer is tansig and output layer function is purelin. The training function is traind.

2.6 Training

The learning procedure is determined by different training algorithms. Choosing the training algorithm is very important to build the best model possible. It affects network performance and its prediction. We will train different algorithms, testing each of them to find then the best predict model for our problem.We use supervised learning rule in which the weights and bias are modified according the error. Feed-forward networks have several algorithms. It is this variation that allows us to build different models to fit the problem.The based implementation of backpropagation network consists in updating weights and bias in the opposite direction with error. There are two methods to implement the gradient descending algorithm: incremental method and grouping way. In the first mode the gradient is calculated and bias with weights are updating after each input. In the second method the gradient is calculated and weights and bias are updated after all inputs are applied to the network.

3 The Experiment Results

3.1 Comparing the performance of different algorithms used for training the network.

There are some parameters, different values of which affect positively or negatively in network performance. We are focused on two key parameters: accuracy and training speed. Backpropagation algorithm has several different types of training. We will test some of them to monitor each performance, during training, to choose then the most appropriate algorithm for our application.

- Levenberg-Marquardt algorithm
- Descent gradient in series algorithm
- Variable learning rate algorithm

3.2 Descent gradient in series algorithm

Gradients are evaluated at each trained entrance and are summarized to determine changes in weights and bias. These two change the opposite direction with the gradient.

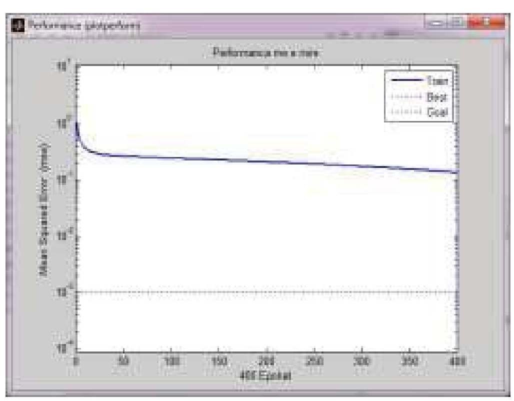


Fig.3.2.1 Performance curve algorithm training with discount gradients in series

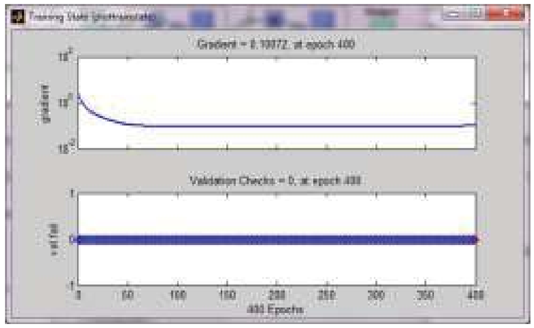


Fig.3.2.2 Curve of gradient changes

We see that the MSE is approximately 0.1 for the parameters that we set.

3.3 Variable learning rate algorithm

This algorithm uses the guidelines techniques and is developed from a performance analysis of the descending standard algorithm. The difference is that the learning coefficient is not constant during training. For this reason it is not possible that the learning rate achieve optimum value before completion of training. This also affects performance.The algorithm tries to improve performance by increasing learning rate step by step. This value is responsible for the local error. If the new error exceeds the old error more than a predetermined value-max-perf-inc, new weights are decreased. Also the learning rate is decreased. If a new error is smaller than the old error, then this rate is increased.

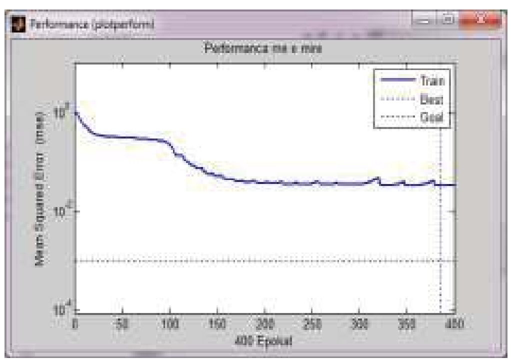


Fig.3.3.1 Training performance curve by backpropagation algorithm with variable learning rate

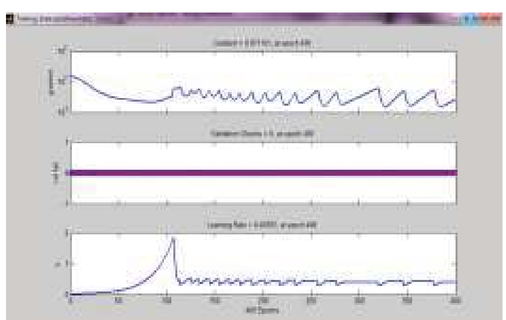


Fig.3.3.2 Changed gradient curves

We see from the figure that its convergence is faster than the previous algorithm, but quadratic error is not stable. This is because the training process has not big increase of error. The function of this algorithm combines a rate adaptive training with momentum coefficient.

3.4 Levenberg-Marquardt algorithm

Levenberg-Marquardt is an algorithm very usable for nonlinear problems. It follows an iterative procedure to locate the minimum of a function which is expressed as a sum of squares of nonlinear functions. When the current solution is far from the real solution, the algorithm behaves like a gradient descent, but it is certain that it will converge. When the current solution is close to the exact solution it behaves as GNA algorithm.

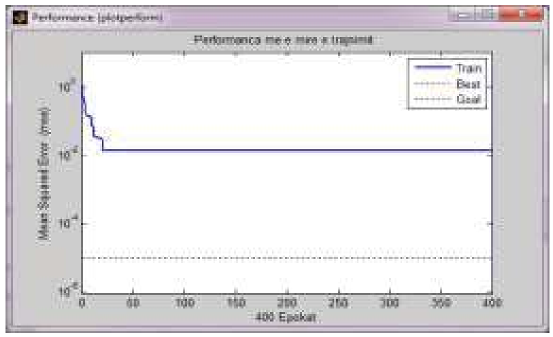


Fig.3.4.1 Training performance curve by Levenberg-Marquardt algorithm

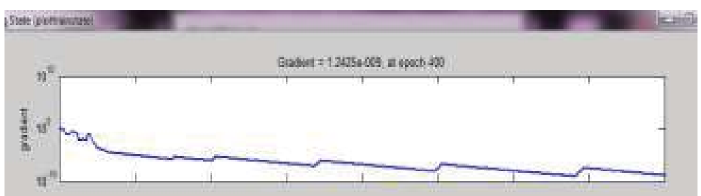


Fig.3.4.2 Variability of gradient

We see that the selected algorithm Levenberg-Marquardt has better performance and converges faster than other algorithms.

3.5 Comparison of algorithms

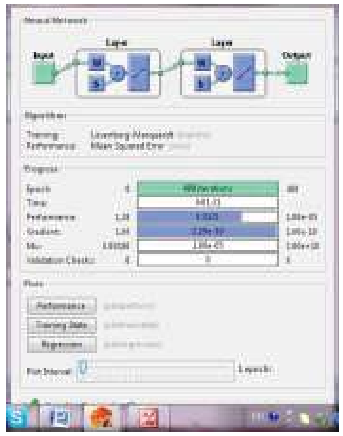
Training results showed that gradient descent algorithms, generally, are very slow, because they require small training rates to have stable training. Conjugate gradient algorithms converge quickly, but their performance is not very well. Levenberg-Marquardt algorithm was the algorithm that results faster and more accurate. This algorithm appears to be best suited for problems that need very accurate models. We need this in our problem, an algorithm that creates more accurate model under which we may perform forecasting and for more in a shorter time. Based on this empirical analysis, we decided to train our application with Levenberg-Marquardt algorithm.We give some features and differences of tested algorithms. Levenberg-Marquardt algorithm will converge faster in those networks which have less than a hundred weights. However, with the increasing number of weights efficiency of this algorithm can be reduced.Backpropagation algorithm with variable learning coefficient generally is slower than other algorithms as we saw during testing.

3.6 Simulation

In this section you will see all the steps in which the solution passes.

We use real data: monetary exchange rate from USD to ALL from 1995 to 2012. The main steps:

1. Creation of data input and target variable.
2. Normalization of data.
3. Creation and training



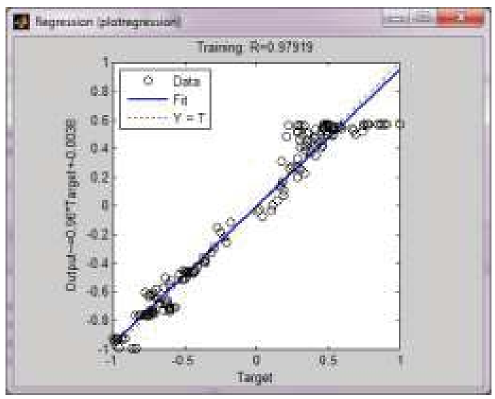


Fig.3.6.2 Linear Regression curve between the network outputs and target outputs.

4. We have to convert output values in real values. So, we use a built-in function.

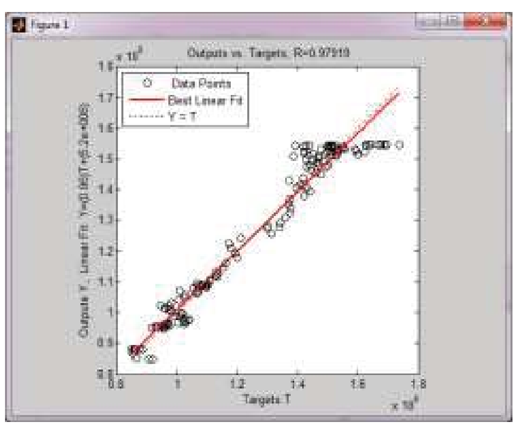


Fig.3.6.3 Comparison between the output of the network with real value and target series.

5. After training the network, the required pattern is detected. Now is test phase

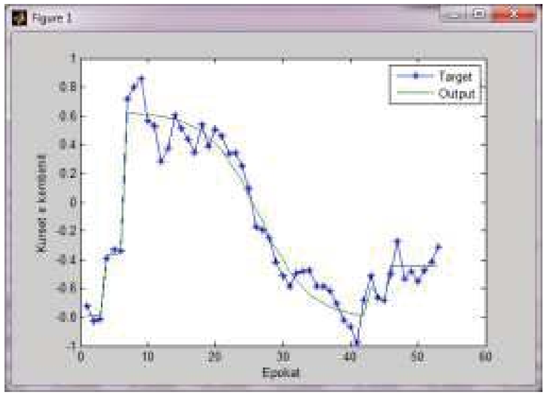


Fig.3.6.4 Forecasting Simulation

As seen from the figure, the testing process gives satisfactory results. Curve network outputs, during the test phase, matches satisfactorily with the curve of the target outputs. So, the network has managed to create a forecast model successfully.

4 Conclusions

The application, that we discussed, presents the implementation of an intelligent system in prediction of monetary exchange of USD in ALL. We chose artificial neural network, as an intelligent technique, and designed it according to the data and requirements of the problem. Since it was a problem where the data were entered in the series (time series) we chose the feedforward network and trained it by one of back-propagation algorithm. Training is based on supervised learning, in Levenberg-Marquardt algorithm. We tested some of back-propagation algorithms and according to their performance, it resulted that Levenberg-Marquardt algorithm was most suitable for our problem. As a simulation environment was used MATLAB. It provided functions for designing, implementing, visualizing, and simulating neural networks. It speeded up training and handle large data sets. From the simulation of this system was noted that the forecasting created model resulted successful. We saw that the network was able to train very well and did this in a short time. Also during the test phase, which is the main phase of the evaluation, the results were very close to reality. However the network still needs further improvements. Restrictions that we may encounter in forecast accuracy or space complexity are some of the elements need to be improved.

References

1. Machine Learning. Tom. M Mitchell
2. Foreign Exhange Market Prediction Using Standard Backpropagation Neural Network. M. Ibrahim Ruslan.
3. Using Recurrent Neural Networks to Forecasting of Forex. V.V. Kotrandenko, Yu. A. Kuperin
4. Artificial Neural Network Model for Forecasting Foreign Exchange Rate. A. Philip, A. Taofiki, A. Bidemi
5. Using Artificial Neural Networks To Forecast Financial Time Series. Rune Aamont
6. Foreign Exchange Rates Forecasting with Neurals Networks. J. Yao, H. Poh, Tiao. Jasic
7. Regression Neural Network for Error Correction in Foreign Exchange Forecasting and Trading. Chen, An-Sing, Leung, Mark T. (2003)
8. Time Series prediction with Feed-Forward Neural Networks - A Beginners Guide and Tutorial for Neuroph. Laura E. Carter-Greaves
9. How Efective are Neural Networks at Forecasting and Prediction? A Review and Evaluation. M. Adya , F. Collopy

10. Neural Network for Short Term Load Forecasting. P. Murto
11. Prediction Markets for Machine Learning. Krzysztof Jerzy Geras
12. Machine learning applied to the prediction of transcriptional regulation in prokaryotes. D. Arends
13. Cognitive Engineering, A distributed approach to Machine Intelligence. A. Konar, L. Jain. Springer, 2005.
14. ANN-Based Forecasting of Foreign Currency Exchange Rates. Joarder Kamruzzaman, Ruhul A. Sarker
15. Statistics & Data Analysis using Neural Network
16. Neural networks for decision support, Decision Support Systems, 11 (1994), 389 - 392.
17. Artificial Intelligence, A Modern Approach. Stuart J. Russell and Peter Norvig
18. Economic prediction using neural networks: a case of IBM daily stock returns, IEEE International Conference on Neural Networks, II, 1988, 451 - 458. White, H
19. Computation and Cognition: Toward a Foundation for Cognitive Science. MIT Press, Cambridge, Massachusetts. Pylyshyn, Z. W.
20. A System of Logic, Ratiocinative and Inductive: Being a Connected View of the Principles of Evidence, and Methods of Scientific Investigation. J. W. Parker, London.
21. Inductive Logic Programming. Academic Press, New York. Muggleton, S.
22. Automatic Methods of Inductive Inference. PhD thesis, Edinburgh University.
23. A formal theory of inductive inference. Information and Control, 1:1-22, 224-254.
24. http://www.exchangerates.org.uk/ALL-EUR-exchange-rate-history.html
25. http://fxtop.com/en/historates.php?A=1&C1=USD&C2=ALL&DD1=01&MM1=01&YYYY1=1995&B=1&P=&I=1&DD2=01&MM2=09&YYYY2=2012&btnOK=Go