Forex Trading Using Geometry Sensitive Neural Networks

Gene I. Sher
University of Central Florida
gsher@eecs.ucf.edu

ABSTRACT

When neural network based systems are used within the field of financial analysis, either as price oracles or autonomous traders, they are primarily used with a sliding price window. This paper presents a novel approach where the indirectly encoded neural network system, just like the technical analysts, looks directly at the candlestick style sliding chart instead, the actual geometrical-patterns within it, to make its predictions. The results presented demonstrate that this approach results in a higher and more consistent generalization to previously unseen financial data, while maintaining a profit level on par with the neuroevolutionary system which uses a standard sliding window.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Artificial Intelligence – Connectionism and neural nets.

Keywords

Neural Network, Evolutionary Computation, Neuroevolution, Memetic Algorithm, Financial Analysis, Forex, DXNN.

1. INTRODUCTION

This summary concentrates on presenting the results of applying the geometrical pattern sensitive, substrate encoded, memetic algorithm based topology and weight evolving artificial neural networks, to currency trading on the foreign exchange market. Foreign exchange (Forex, FX) is a global and decentralized financial market for currency trading. It is the largest financial market, with a daily turnover of 4 trillion US dollars.

Neural Networks, particularly those trained with the the error backpropagation algorithm, have seen a lot of use and success in the financial market, with many papers published on the subject. One of the main strengths of NN systems, which makes them so popular as market predictors, is that they are naturally non linear, and can learn non linear data correlation and mapping. Artificial neural networks are also data driven, can be on-line-trained, are adaptive and can be easily retrained when the markets shift, and finally, they can deal well with data that has some errors; neural networks are robust.

When traders which specialize in technical analysis look at the financial data, they do not usually look just at raw price lists, instead they look at the actual chart patterns. The technical analyst uses the various technical indicators to look for geometrical patterns and emerging trends in these charts. There are many recurrent patterns within the charts, some of which have even been given names. Like for example the “head and shoulders” pattern, in which the time series has 3 hills, resembling head and shoulders. Other such patterns are the “cup and handle”, the “double tops and bottoms”, the “triangles”... Each of these geometrical patterns has a meaning to a trader, and is used by the trader to make predictions about the market. Whether these patterns really do have a meaning or not, is under debate. It is possible that the fact that so many traders do use these techniques, results in a self fulfilling prophecy, where a large number of the traders act similarly when encountering similar geometrical chart patterns, thus making that pattern and its consequence a reality, by all acting in a similar manner that is proscribed by the trend predicting rule of that pattern. What is important though, is that a trading system that does not take these geometrical patterns into consideration, is missing out on the information used by other technical analysts, and is thus under a disadvantage due to not seeing the whole picture.

This summary presents the results of the first of its kind application and benchmarking (to the author's knowledge) of a topology and weight evolving neural network (TWEANN) algorithm for the evolution of geometry-pattern sensitive, substrate encoded (method popularized in HyperNEAT) trading agents that use the actual candlestick style price charts as input, rather than the price based sliding window. In this summary I present the benchmark results of the evolved substrate encoded neural network (NN) based traders which use the Price Chart Input (PCI), and the results of the evolved, standard direct encoded NN based trading agents, which use Price List Input (PLI), in which the input is a simple list of exchange rates. Finally, all of these NN based trading agents will be evolved using the memetic algorithm based TWEANN system called: Deus Ex Neural Network (DXNN) [1, 2].

2. BENCHMARK SETUP

The Forex simulator used in this benchmark simulates the market using 1000 real EURUSD currency pair closing exchange rates, stretching from 2009-11-5-22:15 to 2009-11-20-10:15, using a 15 minute time frame. The simulator uses a price spread of $0.00015. This 1000 point dataset is split into the training set, and a generalization set. The training set is the first 800 time steps, ranging from: 2009-11-5-22:15 to 2009:11-18-8:15, and the generalization data set is the immediately following 200 time steps from 2009-11-18-8:15 to 2009-11-20-10:15. When an agent opens a position, it is always done with $100, leveraged by x50 to $5000. Finally, each agent is started with $300, with the fitness being the agent's final net worth. An evaluation was considered complete when an agent's net worth dropped below $100, or when the agent had passed through all the 800 points (during training), or 200 points (during testing).

In total, 13 experiments were performed to test the theory of a better generalization of a substrate encoded, sliding chart pattern based input NN based systems as compared to the standard directly encoded sliding price window based NN. Each experiment is composed of 10 evolutionary runs, from which a fitness average, max, and min are calculated. The experiments demonstrate and compare the performance of PCI based NNs to the PLI based NNs. Both of these input type experiments were tested with different sensors of comparable dimensionality.

Experiments 1-5 were performed using with PLI NNs. Each experiment differed in the resolution of the sliding window input that the NNs used. Each NN started with the sliding window sensor, and the vector: [Position, Entry, PercentageChange]. The

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

networks were allowed to evolve recurrent connections. These 5 experiments used the sliding windows of size: 5, 10, 20, 50, and 100 respectively.

Experiments 6-14 were performed using the PCI NNs. In these experiments each PCI based NN used a Jordan Recurrent 4 dimensional substrate. The input hyperlayer to the substrate was composed of the candle stick styled *price chart,* the vector: [CurrentPosition, EntryPrice, PercentageChange], and the substrate's own output, making the substrate Jordan Recurrent. The substrate architecture of these CPI NN is shown in Fig-2.1.

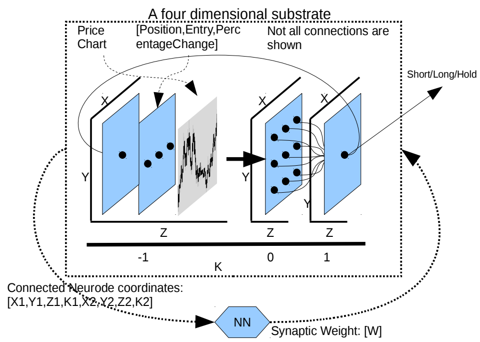

Fig-2.1 Jordan Recurrent, 4 dimensional substrate encoded NN

Thus for the PCI based NNs, I created a 4 dimensional substrate (the 4th dimension was called K) with an input hyperplane composed of the noted 3 planes and located at K = -1, all of which connected to the 5X5 hidden plane positioned at K = 0, which then further connected to the 1X1 output plane (a single neurode) located at K = 1, which output the short/hold/long signal and which was also fed back to the substrate's input hyperplane. Each of these 9 experiments used a price chart input of a differing resolution. The different experiments used the input chart resolution of: 5X10, 5X20, 10X10, 10X20, 20X10, 20X20, 50X10, 50X20, and 100X10 respectively.

3. BENCHMARK RESULTS

The generalization results for the price chart and price list experiments are shown in figures 3.1 and 3.2 respectively. The generalization test (testing the champion NN of a population, on a financial data it has not previously seen, the last 200 price ticks) was performed every 500 evaluations during the training, which lasted for 25000 evaluations in total. The highest achieved generalization fitness scores, amongst the 10 evolutionary runs of every experiment, is what the following figures present. In a production system, as one performs generalization testing, one would immediately extract a champion NN based agent which was able to generalize well, and then give that champion real money to trade with.

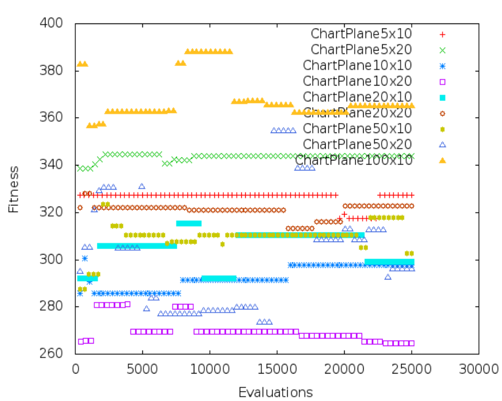

Fig-3.1 PCI "Generalization Testing Fitness Vs. Evaluations"

What can be clearly noticed from these two figures (3.1 & 3.2), is that though both approaches were able to produce NNs capable of making profit (final net worth being above $300) when tested on financial data that the systems were not trained on, the PCI was able to generalize much better. Whereas in the PLI experiments the profit is achieved rarely, and is only held for a brief amount of time, the PCI based champion NNs are on average more often profitable.

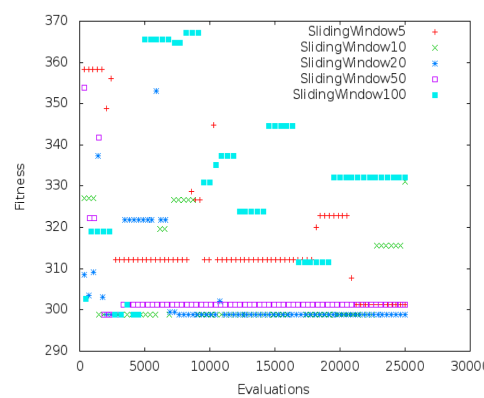

Fig-3.2 PLI "Generalization Testing Fitness Vs. Evaluations"

4. CONCLUSION

In this summary I presented the performance, profitability, and generalization of *Price List Input using NNs,* directly encoded, and the *Price Candle-Stick Chart Input using, geometrical pattern sensitive NNs*. I presented a never before published and completely new, type of trading and prediction system that uses the actual candle-stick charts of financial instruments as input, thus letting the evolved NNs take into account the geometrical patterns of the financial data when making its predictions and trades. The hypothesis that Topology and Weight Evolving Artificial Neural Network systems could indeed effectively evolve currency trading agents was shown, based on the generalization results, to be **correct**. The hypothesis that geometrical pattern sensitive NN systems could indeed trade profitably and generalize much better and more consistently than standard PLI NNs, was also proved to be **correct**.

5. REFERENCES

[1] Sher G "DXNN Platform: Shedding the Biological Inefficiencies", arXiv:1011.6022v3 [cs.NE] 2011

[2] Sher G "DXNN: Evolving Complex Organisms in Complex Environments Using a Novel TWEANN System", GECCO '11 Proceedings of the 13th annual conference companion on Genetic and evolutionary computation. 2011

[3] Sneha S "Applications of ANNs in Stock Market Prediction: A Survey" IJCET, 71-83. 2005

[4] Li Y. Weihua M "Applications of Artificial Neural Networks in Financial Economics: A Survey" *Neural Networks*: 211-214. 2010