# Recurrent Neural-Genetic Hybrids in Corporate Financial Evaluation

**Loukeris Nikolaos [a], Matsatsinis Nikolaos [b]**

*Technical University of Crete*
*Department of Production Engineering and Management*
*Akrotiri Campus, 73100 Chania, Crete, Greece*

*(a)PhD student University of Essex, (b)Associate Professor Techn. Univ. of Crete*

*Abstract:*-Financial management implements a variety of effective techniques to determine the economic health of corporations. The vast amount of financial data and the complexity of accounting statements require improved techniques not only of econometrics, but also of artificial intelligence. Effective decisions on financial management and corporate management can be supported by hybrid systems. Recurrent neural networks with neural with genetic algorithms optimization have provided results of significant precision and therefore may be considered in financial analysis.

*Key-words:* Financial Management, Hybrid Systems, Recurrent Neural Networks, Genetic Algorithms

## 1 Introduction

Asset management from managers and portfolio management from investment administrators requires a thorough financial evaluation of corporations. Corporate financial evaluation and bankruptcy prediction achieved significant results through advanced methods of artificial intelligence, data mining, finance, and operations research, **[1]**, whilst new effective tools may be used to support the decision makers. The objective of this article is to investigate the efficiency of hybrid forms in Recurrent Neural Networks, optimized by Genetic Algorithms. Paragraph 2 provides a description of hybrid algorithms. Paragraph 3 analyzes the Recurrent Neural Networks. Paragraph 4 presents the optimization process that Genetic Algorithms offer to Recurrent Neural Nets. Paragraph 5 has a description of the problem and the tools of financial analysis that will be used. Paragraph 6 analyzes the results of each method thoroughly. Paragraph 7 includes the conclusions and the future research.

## 2 Hybrid Algorithms

Hybrid algorithms compare genetic algorithms with the most successful optimization methods to a problem, ensuring their correct application. Genetic algorithms, despite their vigorous, are not always effective in optimization, **[2]**. **[3]** used non-linear coding, specializing operations of genetic algorithms to combinations with search models based on genetic search models. **[4]** describing a parallel genetic algorithm, concluded that, whilst an initial population is created, each individual each individual elaborates a local ascension and after each descendant is created it activates a local ascension as well. Researchers diverse their conclusions on hybridization matter. Addition of ascensions or hybridization with other optimisation methods, learning is added in the process of evolution. Coding of acquired information, in chromosomes indicates a form of Lamarck search. Optimized chromosomes by local ascension or other methods are put on the total population, and there are allowed to compete aiming to obtain reproduction opportunities.

## 3 Recurrent neural networks

A recurrent neural network (RNN) is a neural network where the connections between the units form a directed cycle, creating a network of neurons with feedback connections. Recurrent neural networks must operate differently from feedforward neural networks, during their computational behavior and during their training. Recurrent neural networks may behave chaotically, thus dynamical systems theory is used to model and analyze them.

The human brain is a recurrent neural network (RNN) or feedback neural network. RNNs from training examples can learn to map input sequences to output sequences. In principle they can implement almost arbitrary sequential behavior, ideal for adaptive robotics, speech recognition, music composition, attentive vision, and many other applications. RNNs are biologically more plausible and computationally more powerful than other adaptive models such as Hidden Markov Models (with no continuous internal states), feedforward networks and Support Vector Machines (without internal states). The network components used for recurrent neural networks are not different from the feedforward topologies. The topology is characterized by feedback loops that go from the processing elements to themselves and to other processing elements. Feedback loops must include some form of delay to implement a realistic network, because there is no instantaneous transmission of information in dynamic systems.

RNNs could not learn to look far back into the past. Their problems were first rigorously analyzed on Schmidhuber's RNN long time lag project by **[6]**, **[7]**. A novel feedback network called "Long Short-Term Memory" (LSTM, Neural Comp., 1997) overcomes the fundamental problems of traditional RNNs, and efficiently learns to solve many previously unlearnable tasks involving:

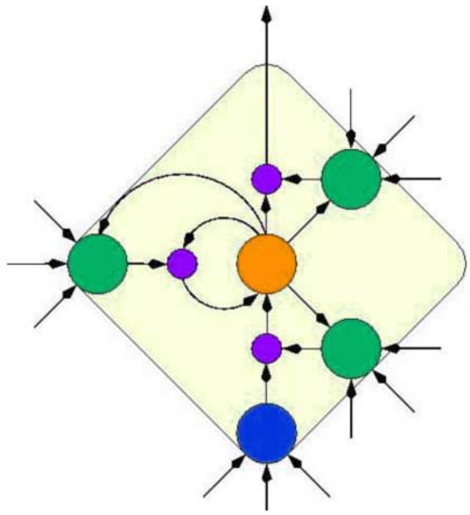

**Chart 1. A recurrent neural network [5]**

1. Recognition of temporally extended patterns in noisy input sequences

2. Recognition of the temporal order of widely separated events in noisy input streams

3. Extraction of information conveyed by the temporal distance between events

4. Stable generation of precisely timed rhythms, smooth and non-smooth periodic trajectories

5. Robust storage of high-precision real numbers across extended time intervals

## 4 Recurrent Neural Networks with Genetic Algorithms Optimization

The importance of each one of the 16 financial inputs in hybrid RBF network is not predefined thus Genetic Algorithms select the significant inputs. The network must be trained multiple times to find the inputs combination with the lowest error. Genetic Algorithms were used on each layer in RBF with different topologies. On-Line learning were chosen to update the weights of hybrid neuro-genetic RBF, after the presentation of each exemplar. Genetic Algorithms were used providing solution to the problem of optimal values in a) Processing Elements, b) Step Size and c) Momentum Rate. RBF network must be trained multiple times to achieve the settings that result the lowest error. Output layer was chosen to implement Genetic Algorithms optimizing the value of the Step size and the Momentum.

## 5 Description of the problem

In this research 4 different topologies of Recurrent Neural Networks with Genetic Algorithms optimization. Data came by 1411 companies from the loan department of a Greek commercial bank, with the following 16 financial indices:

1) EBIT/Total Assets**,**
2) Net Income/Net Worth,
3) Sales/Total Assets,
4) Gross Profit/Total Assets,
5) Net Income/Working Capital,
6) Net Worth/Total Liabilities,
7) Total Liabilities/Total assets,
8) Long Term Liabilities /(Long Term Liabilities + Net Worth),
9) Quick Assets/ Curent Liabilities,
10) (Quick Assets - Inventories)/Curent Liabilities,
11) Floating Assets/Current Liabilities,
12) Current Liabilities/Net Worth,
13) Cash Flow/Total Assets,
14) Total Liabilities/Working Capital,
15) Working Capital/Total Assets,
16) Inventories/ Quick Assets,

and a 17th index with initial classification, done by bank executives. Data test set was 50% of overall data, and training set 50% as well.

## 6 Results

NeuroSolutions 4.3 was deployed with many different network topologies. Initially Genetic Algorithms were chosen to be used the intermediate steps of solutions that neural networks produced, providing choice of solution genes in neural network, consuming significant time periods. The method of data representation on each genetic session of offspring was on-line, because it offers the optimal solution in each generation, although sometimes it is exposed to the higher risk of falling in local minima/maxima. Batch representation was rejected because confusion matrix changes on each new chromosome, without adjusting instantly, whilst it has lower risk of getting trapped in local minima/maxima. Recurrent network with 1 hidden layer, 400 epochs and genetic optimization had a slower convergence, compared to Recurrent network of 500 epochs. In all neural networks 500 epochs were chosen for each generation of solutions that Genetic Algorithms used to, aiming to give the necessary time for convergence in the optimal offspring set, without wasting surplus time when the optimal set of solutions was found in each repeat. In the following table 1, are presented the results of each different topology in the hybrid Recurrent neural-genetic networks. Different topologies, used the same data set of 1411 companies and the training set 50% of overall data. The amount of hidden layers were increased, after the previous topology converged to a solution, noticing that while a change in the number of hidden layers the results such as confusion matrix, MSE varied significantly. Neural nets cannot produce the same output, given the same input. Analytically the performance of each recurrent neural network topology is in table1.

At first Recurrent neural network produced successful results. Four Recurrent nets were deployed with 1 hidden layer for the first one, 2, 3, and 4 layers for each one of the other networks. Their cost function expressed by Mean Square Error were very low: 0.122 for the net with 1 layer, 0.208 with 2 layers, 0.116 with 3 layers, and 0.218 with 4 layers, offering a satisfactory fitness of the network output to the desired output. Their correlation coefficient r were 0.771 for the hybrid RNN with 3 layers that presented the optimal fitness of the model to the data, followed by the hybrid RNN of 1 layer with 0.731, whilst the RNN with 2 layers and 4 layers had a much inferior correlation coefficient at 0.558 and 0.521 respectively. Recurrent nets provided adequate classification results for companies of mark 1 (in distress) and were either 33.33%, 33.34% or 66.66%in one case (3 layers). Convergence time vas very low and varied from 1 hour 36' (1 layer), 2 hours 44' (layers), 5 hours 55' (3 layers) and 9 hours 8' (4 layers). Although their performance was of adequate convergence, very fast, and with very low statistical error (MSE), Recurrent networks could not approach the effectiveness of SVM networks **[1]**.

### Table 1. Overall results in Neural Networks per architecture

| Neural Network | | Active Confusion Matrix | | | | Performance | | | | | | Time |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *Layers* | **0->0** | **0->1** | **1->0** | **1->1** | **MSE** | **NMSE** | **r** | **%error** | **AIC** | **MDL** | |
| Recurent | 1 | 95.83 | 4.16 | 66.66 | 33.34 | 0.122 | 0.476 | 0.731 | 9701647 | -20.63 | -26.97 | 1 h 36'00 |
| Recurent | 2 | 100 | 0 | 66.66 | 33.33 | 0.208 | 0.811 | 0.558 | 7034926 | 163.71 | 127.44 | 2 h 44'00 |
| **Recurent** | **3** | **100** | **0** | **33.34** | **66.66** | **0.116** | **0.454** | **0.771** | **11092527** | **940.05** | **764.36** | **5 h 25'00** |
| Recurent | 4 | 100 | 0 | 66.66 | 33.33 | 0.218 | 0.8504 | 0.521 | 7702640 | 3664.9 | 3012.57 | 9 h 08'00 |

## 7 Conclusions-Future Research.

Optimal performance was achieved by Recurrent neural network of 3 hidden layers, as the confusion matrix classified companies of characterization 0 to 0 with a percentage of 100% whilst companies with characterization 1 were classified as 1 in a rate 66.66%, Mean Square Error was the lowest of all RNNs' at 0.116, correlation coefficient r was the highest among the RNNs at 0.771 revealing a fine fitness of the model to the data, whilst the computing time was the longest of all the Recurrent neural-genetic hybrids at 5 hours and 25 minutes. Thus Recurrent Neural Networks with Genetic Algorithms optimization have a quite good performance in classifying financial data, producing reliable analyses, whilst the optimal Recurrent neuro-genetic hybrid is with 3 hidden layers among the 4 examined hybrids.

## 8 Bibliography

**[1]** Loukeris N., Matsatsinis N., 'Corporate Financial Evaluation and Bankruptcy Prediction implementing Artificial Intelligence methods', Journal of Business and Economics, Issue 4, Volume 3, April 2006, WSEAS Transactions ISSN 1109-9526

**[2]** Davis. L. "Handbook of Genetic Algorithms"- Van Nostrand Reinhold, New York, 1991

**[3]** Michalewitz Z."Genetic algorithms+data structures=Evolution programs", Springer Verlag, 1992

**[4]** Mühlenbein H. (1991)-Evolution in Time and Space. The Parallel Genetic Algorithm- Foundtions of Genetic Algorithms, G. Raawline ed. Morgan-Kaufmann pp 316-337

**[5]** Schmidhuber J., Wierstra D. and Gomez F. (2005) -'Hybrid Neuroevolution/ Regression Search for Sequence Prediction'. Proceedings of the

19th International Joint Conference on Artificial Intelligence (IJCAI).

**[6]** Hochreiter S. and J. Schmidhuber. Bridging long time lags by weight guessing and ``Long Short-Term Memory". In F. L. Silva, J. C. Principe, L. B. Almeida, eds., Frontiers in Artificial Intelligence and Applications, Volume 37, pages 65-72, IOS Press, Amsterdam, Netherlands, 1996.

**[7]** S. Hochreiter and J. Schmidhuber. LSTM can solve hard long time lag problems. In M. C. Mozer, M. I. Jordan, T. Petsche, eds., *Advances in Neural Information Processing Systems 9*, *NIPS'9*, pages 473-479, MIT Press, Cambridge MA, 1997.