International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:8, No:5, 2014

Forecasting Optimal Production Program Using Profitability Optimization by Genetic Algorithm and Neural Network

Galal H. Senussi, Muamar Benisa, Sanja Vasin

Abstract**—**In our business field today, one of the most important issues for any enterprises is cost minimization and profit maximization. Second issue is how to develop a strong and capable model that is able to give us desired forecasting of these two issues. Many researches deal with these issues using different methods. In this study, we developed a model for multi-criteria production program optimization, integrated with Artificial Neural Network.

The prediction of the production cost and profit per unit of a product, dealing with two obverse functions at same time can be extremely difficult, especially if there is a great amount of conflict information about production parameters.

Feed-Forward Neural Networks are suitable for generalization, which means that the network will generate a proper output as a result to input it has never seen. Therefore, with small set of examples the network will adjust its weight coefficients so the input will generate a proper output.

This essential characteristic is of the most important abilities enabling this network to be used in variety of problems spreading from engineering to finance etc.

From our results as we will see later, Feed-Forward Neural Networks has a strong ability and capability to map inputs into desired outputs.

Keywords**—**Project profitability, multi-objective optimization, genetic algorithm, Pareto set, Neural Networks.

I. Introduction

ARTIFICIAL neural networks have increased attention for solving many real complex world problems. ANN compared to traditional methods have solved many complex problems successfully where traditional methods have failed. In addition, numbers of research and development works are increasing rapidly in recent years.

A lot of research about Artificial Neural networks dealt with predicting and decision making problems in last few decades. These researches have improved and developed in methods of forecasting which led to make better decisions. Many forecasting and decision modeling problems have used ANN as solving process [1]. Artificial neural networks have strong potential advantages over than statistical methods and can be strongly deal with non-linear functions.

Hawley, Johnson, and Raina were one of the first who have applied neural networks in real business world [2].

In the last ten years, neural network are particularly applicable to risk management and forecasting [3].

ANN also has property of non-linear behavior where it can estimate non-linear functions well and extract any remaining of non-linear elements from the data as well [4], [5].

Hornik [6] has found that artificial neural networks are able to make the best approximation functional form as long as use good data characteristics.

In addition, ANN has potential power to transform the input data [7].

Kang [8] found that artificial neural networks give better forecasting in monthly and quarterly period than in the annual. However, recent studies including our study on artificial neural networks have proved the opposite.

Artificial neural network has strong ability to generate fitting to the data as good as the fit of the true functional forms where's performed with high noise and low sample sizes by high reliability [9].

Artificial neural network has ability to provide good solution, of which regenerate an existing system behavior that leads to right decision-making [10].

Crone [11] and Wang [12] have found that artificial neural network training is able to inaugurate the behavior of the original system. They applied that successfully to predict cost of manufacturing, control of system quality, etc.

II. Methodology

With the aim of solving a concrete problem of multi-criteria production program optimization in an industrial enterprise, a methodology that includes the application of genetic algorithms has been developed and objective nonlinear functions in previous work. The developed model created a Pareto set curve and generated a set of optimal solutions as well. The generated solutions were approximate to the real solutions.

The diagonal font in Fig. 1 shows the steps that were developed in our previous work. The first step was the definition of the problem, i.e. the segregation of products, machinery capacities, human and other production resources whose optimization we want to perform. The following step was generation of the criteria, whose maximum or minimum value we want to accomplish. The criteria in the production program optimization can be profit maximization, minimum production costs, the maximum utilization of production capacities and like. After this step, we set up the objective

Galal is with the faculty of Mechanical Engineering, Omar Al-Mokhtar University, El-Baida, Libya, (phone: 381-62-179-1864; e-mail: galalsanussi@yahoo.com).

Muamar Benisa is with the Faculty of Mechanical Engineering, Al-Merghab University, Alkomes, Libya, (e-mail: muamarisa@yahoo.com).

S. Vasin is with the Faculty of Mechanical Engineering, Belgrade University, Belgrade, Serbia, (e-mail: vasinsanja@gmail.com).

function in linear or nonlinear form, depending on how much they represent the real model in the best possible way. Defining constraints was the following step, which can derive from production capacities, i.e. the constraints of machinery capacities, human resources, material resources, but also from demands on the market for the observed product. The final step, applying GA, the Pareto front generated and optimal solutions were tested. If the optimization solutions met, the criteria stop; otherwise choose other population or change parameters in the algorithm.

In this work, we extended our previous work and developed a combined model of genetic algorithms and neural network as shown in Fig. 1. This combined model has been tested and given satisfying results. Taking into account that a set of optimal solutions which offers from applying genetic algorithms has been employed and used in neural network to produce good predicted and satisfied results.

Fig. 1 shows steps in generating the combined model of multi-criteria production program optimization. Data, which produced from genetic algorithm, have entered to neural network after building the model. Part of this data has used as training and other two parts for testing and validation. After running the model and getting solution, the next step is to compare the solution with the target. If the output meets with the target, end the program. Otherwise, modify the model by changing hidden layer and/or number of neuron and/or Use other learning parameters and/or adjusts the weights.

III. Math

In our previous work, we started to examine possibility of production program optimization in pilot factory, Industry of Precision Mechanics. We examine production process of three lines of products: Clocks, Water meter, and gas meters.

By longitudinal investigations and monitoring of production data, we have analyzed the available data, formed nonlinear functions of the TR and the TC for the three products, and generated next objective functions:

a) Clocks (X1)

Revenue function

$$f(x)_{11} = TR(Q) = -0.04Q^2 + 686Q - 1375.3 \quad (1)$$

Cost function

$$f(x)_{21} = TC(Q) = -0.024Q^2 + 410.Q - 4342 \quad (2)$$

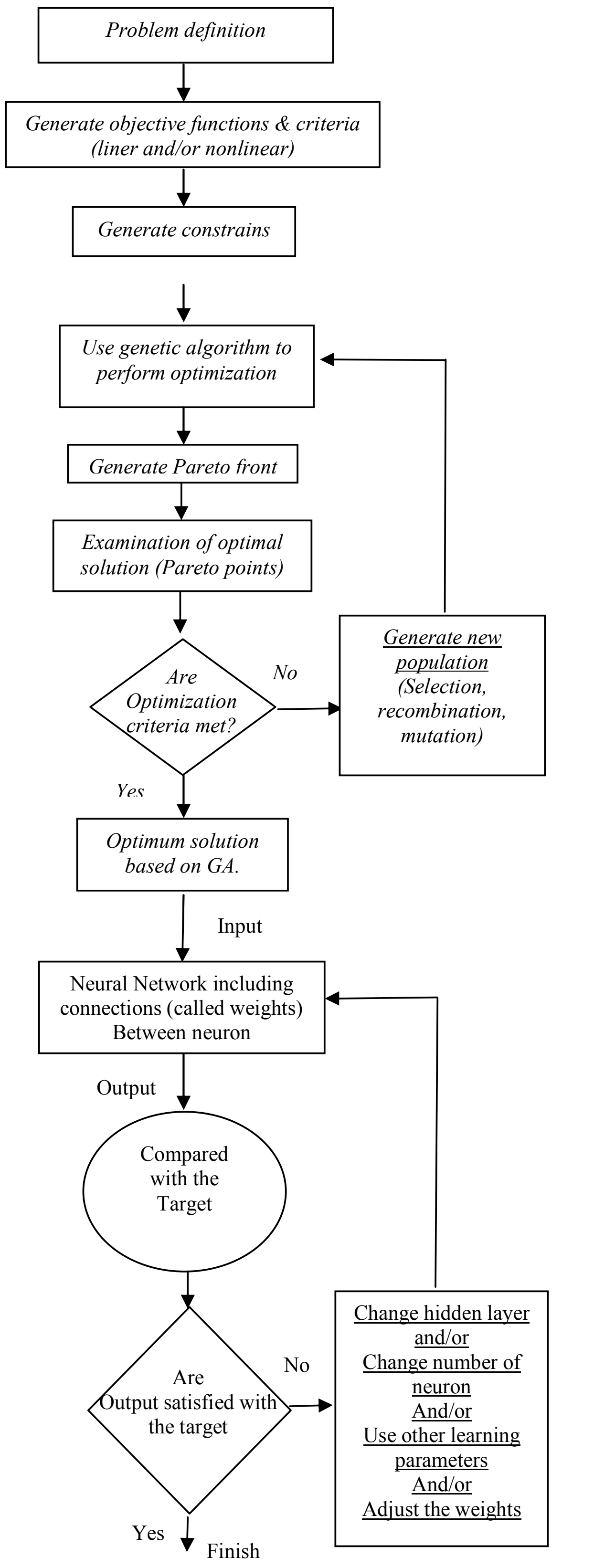

Problem definition
Generate objective functions & criteria (liner and/or nonlinear)
Generate constrains
Use genetic algorithm to perform optimization
Generate Pareto front
Examination of optimal solution (Pareto points)
Are Optimization criteria met?
No
Generate new population (Selection, recombination, mutation)
Yes
Optimum solution based on GA.
Input
Neural Network including connections (called weights) Between neuron
Output
Compared with the Target
Are Output satisfied with the target
No
Change hidden layer and/or Change number of neuron And/or Use other learning parameters And/or Adjust the weights
Yes
Finish

Fig. 1 Flow chart of developing model for multi-criteria production program optimization combined with ANN

b) Water meter (X2)

Revenue function

$$f(x)_{12} = TR(Q) = -0.18Q^2 + 4298Q - 343884 \quad (3)$$

Cost function

$$f(x)_{22} = TC(Q) = -0.49Q^2 + 3382.4Q - 463764 \quad (4)$$

c) Gas meter (X3)

Revenue function

$$f(x)_{13} = TR(Q) = -0.87Q^2 + 5984.5Q - 5715.1 \quad (5)$$

Cost function

$$f(x)_{23} = TC(Q) = -0.58Q^2 + 3818.2Q - 3643.6 \quad (6)$$

The functions of criteria for profit maximization will have the form:

$$\max f(x) = \sum_{i=1}^{3} f_{1i} = f(x)_{11} + f(x)_{12} + f(x)_{13}$$

$$\min f(x) = \sum_{i=1}^{3} f_{2i} = f(x)_{21} + f(x)_{22} + f(x)_{23} \quad (7)$$

Respectively,

f(1)= -0.04*x(1)^2 + 686*x(1) - 0.18*x(2)^2 + 4298*x(2) - 0.87*x(3)^2 + 5984.5*x(3) - 350975.4; (8)

f(2)= -0.024*x(1)^2 +410*x(1) - 0.49*x(2)^2 + 3382.4*x(2) -0.58*x(3)^2 + 3818.2*x(3) - 463066; (9)

The production capacity considered as a key constraint in the production quantity of some products, temporarily ignoring the structure of demand for mentioned products on the market, the restrictions is:

$$0 \le x1 \le 4400$$
$$0 \le x2 \le 2444$$
$$0 \le x3 \le 1100$$

Taking into account the Employees and raw material in the observed company are not of limiting character.

The second step of experiment was applying genetic algorithms, where we obtained the following results, as shown in Table I.

IV. Neural Network

Neural network model consist of three stages: Building, Training, and Testing.

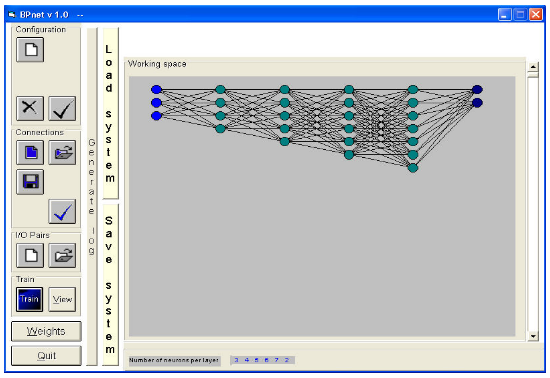
1) Network Building:

After testing of many different models, built with different number of hiding layers and/or different numbers of neurons at same time, we found that the model below gave us satisfied results. The model type is (3 4 5 6 7 2) as shown in figure below. This means, fist layer consist of three neurons which represent input data (production quantity one, production quantity two, and production quantity three respectively). The second, third, fourth, and fifth are represented the hidden layers which contained four, five, six, and seven neurons respectively. The last layer represents the output and consists of two neurons (total profit and total cost respectively).

TABLE I
Mat-lab Results by Applying GA

	Input data			Output data	
No	Quantity of product one(X1)	Quantity of product two(X2)	Quantity of product three(X3)	Total profit(f1)* 10^4	Total cost(f2)* 10^4
2	2312.1	2192.8	944.4	1446.5	-850.6
13	2312.1	2192.8	944.4	1446.5	-850.6
17	2193.0	1804.7	838.9	1254	-762.4
5	1973.0	1678.2	775.6	1167.1	-716.1
15	2087.2	1561.3	624.9	1057.8	-653.4
19	1799.6	1408.7	317.1	826.2	-514.2
18	367.9	1168.4	269.1	622	-395.3
7	2008.5	1157.9	735.7	953.3	-601.8
9	907.5	1050.0	816.4	886	-563.1
6	269.9	1020.1	297.5	573.2	-367.1
4	1926.2	899.0	513.4	738.4	-468.9
3	461.2	671.9	161.3	370.6	-237.3
10	68.2	547.2	24.2	213.8	-136.1
20	124.1	392.5	249.4	283.2	-175.6
12	1395.3	266.3	545.7	466.7	-283.8
8	345.0	225.2	90.2	137.3	-75.2
14	9.8	161.3	184.4	141.8	-75.8
11	197.1	153.5	23.6	58	-21.4

* *Data used as input data in neural network model*



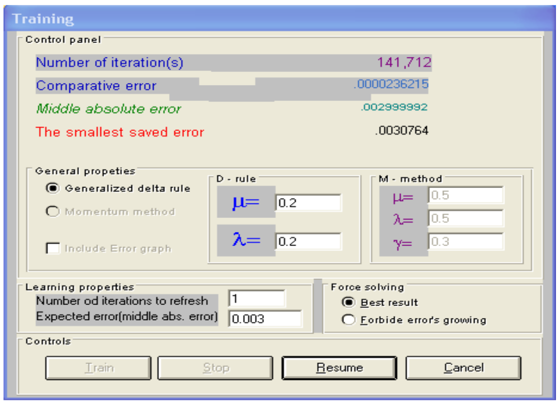
Number of neurons per layer 3 4 5 6 7 2

Fig. 2 Network Building

2) Network Training:

The training sets were established in MS Excel from [-1, 1] step 0.2 and the input values were normalized. There are 64 sample examples. The 42 samples used as training examples (red color) and the rest used as testing samples (black color) as shown in Table II.

After trying different learning parameters with our model (3 4 5 6 7 2), learning parameters μ=λ=0.2 and middle absolute error = 0.003 have been chosen. The figure below showed the BPN screen in training stage.



```
Number of iteration(s)                   141,712
Comparative error                        .0000236215
Middle absolute error                    .002999992
The smallest saved error                 .0030764

(●) Generalized delta rule      μ= 0.2        μ= 0.5
( ) Momentum method             λ= 0.2        λ= 0.5
[ ] Include Error graph                       γ= 0.3

Number od iterations to refresh      1        (●) Best result
Expected error(middle abs. error)    0.003    ( ) Forbide error's growing
```

Fig. 3 Network Training

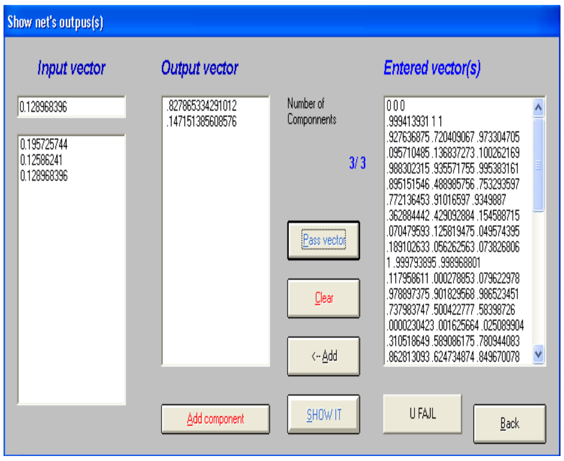
The table below showed the final run of statistical results for BPN.

TABLE II
BPN FINAL STATISTICAL RESULTS

Number of iterations	Comparative error	Middle absolute error	The smallest error
141712	0.0000236215	0.002999992	0.0030764

3) Network Testing:

The final step was testing. Twenty-two data has chosen for testing stage. The red color data represent the forecasting results from BPN (output) whereas the black color data represent the results from GA (target). It is clear from the table below that the output is very close to the target. That means that the model generated satisfied results.



```
Input vector     Output vector         Entered vector(s)
0.128968396      .827865334291012      0 0 0
                 .147151385608576      .999413931 1 1
0.195725744                            .927636875 .720409067 .973304705
0.12586241                             .095710485 .136837273 .100262169
0.128968396                            .988302315 .935571755 .995383161
                                       .895151546 .488985756 .753293597
                                       .772136453 .91016597 .9349887
                                       .362884442 .429092884 .154588715
                                       .070479593 .125819475 .049574395
                                       .189102633 .056262563 .073826806
                                       1 .999793895 .998968801
                                       .117958611 .000278853 .079622978
                                       .978897375 .901829568 .986523451
                                       .737983747 .500422777 .58398726
                                       .0000230423 .001625664 .025089904
                                       .310518649 .589086175 .780944083
                                       .862813093 .624734874 .849670078
```

Fig. 4 Network Testing

V. DISCUSSION AND CONCLUSION

In our case, we have used model type (3 4 5 6 7 2) with Learning parameters μ=0.2, λ = 0.2 and expected error = 0.003. We have three inputs and two outputs. Our model consists of six layers. The first was for input data and the last one for output data. We have deployed four hidden layer ANN. The first hidden layer contain 4 neurons, 5 for the second, 6 for the third, and 7 neurons for the fourth one.

Using this model, we have trained our network on 70% of the data, 15% for validation, and tested on the remaining 15% data. Data division between the training and the validation made entirely random.

Using ANN for predicting net profit and total cost of production enterprise gave good results. The ANN learnt the data as following in Figs. 5 and 6 respectively:

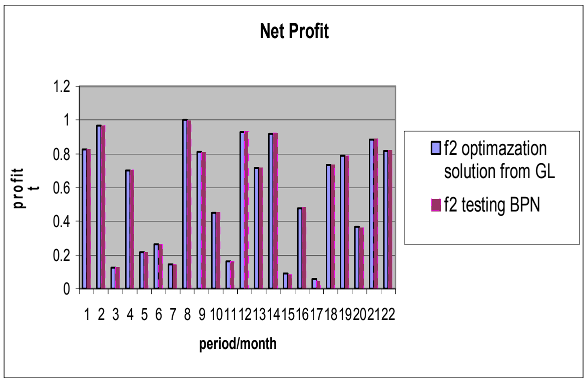


Fig. 5 Testing Net Profit compared with Real Net Profit

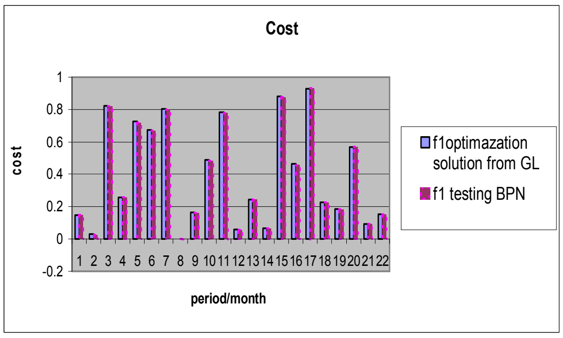


Fig. 5 Testing Cost compared with Real Cost

It is not hard to see that the output is not 100% what we actually wanted but much closer to our desire output. This means that our result does not come from the core of the Feed-Forward Neural Network. Namely, this network has meant to be universal. In other words, we have found the optimal architecture perfectly able to map inputs into desired outputs, also we insisted on mapping between translated patterns and outputs.

Feed-Forward Neural Network with six layers (3 neurons in the input layer, 4,5,6,7 neurons in the hidden layer and 2 neurons in the output layer) outperforms other tested networks. Compared with others neural network models needs more processing time but it is much closer to desired outputs. This is a price that has to be paid if a reliable solution is to be determined. Therefore, for this problem, the best Feed-Forward Neural Network architecture is 3x4x5x6x7x2.

The data fit is quite good. From figure above, we can note that the testing data (red curve) is very close to the real data (blue curve). Yet, the nonlinear relationship learnt to high degrees of accuracy by the neural network. As can be seen from the two curves, the ANN does trace the true data to even a reasonable degree of accuracy. As has been labeled on the diagram, the estimated net profit and total cost are on its true value by more 99.9%. Thus, the ANN model would succeed.

The performance, training state, regression of ANN was as following in Fig. 7:

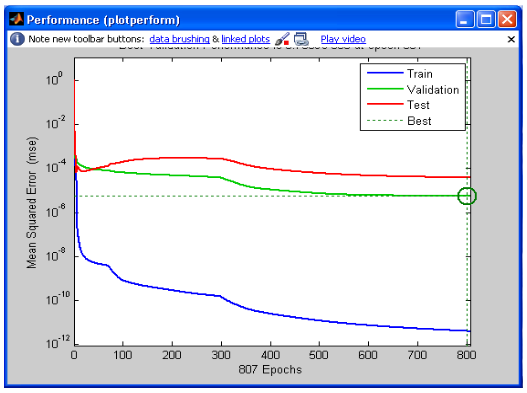


Fig. 6 Network means squared error Performance

The plot shows the mean squared error of the network starting at a large value and decreasing to a smaller value. In other words, it shows that the network is learning. The plot has three lines, because the 192 input and targets vectors are randomly divided into three sets. The 70% of the vectors are used to train the network.

The 15% of the vectors are used to validate how well the network generalized. Training on the training vectors continues as long that the training reduces the network's error on the validation vectors. After the network memorizes the training set (at the expense of generalizing more poorly), training is stopped. This technique automatically avoids the problem of over fitting, which plagues many optimization and learning algorithms.

Finally, the last 15% of the vectors provide an independent test of network generalization to data that the network has never seen.

In our case, the result is reasonable because of the following considerations:

1. The final mean-square error is small.
2. The test set error and the validation set error most likely has similar characteristics and behavior.
3. No significant over fitting has occurred by iteration 800 (where the best validation performance occurs)

Fig. 8 shows the results of linear regression between the network outputs and the corresponding targets.

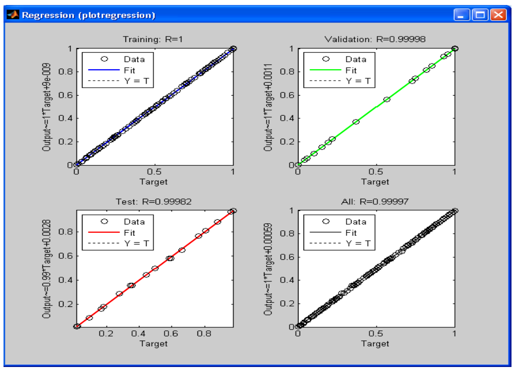


Fig. 7 Regression plots show the output with respect to training, validation, and test data

The network outputs are plotted versus the targets as open circles. A dashed line indicates the best linear fit. The solid line indicates the perfect fit (output equal to targets).

The correlation coefficient (R-value) between the outputs and targets is a measure of how well the variation in the output is explained by the targets. In our problem is closed to 1, that means, perfect correlation between targets and outputs, which indicates a good fit.

Fig. 9 shows that the output tracks the targets very well for training, testing, and validation, and the R-value is 0.99997 for the total response. In our case, it is difficult to distinguish the best linear fit line from the perfect fit line because the fit is so good. That means, our network response is satisfactory.

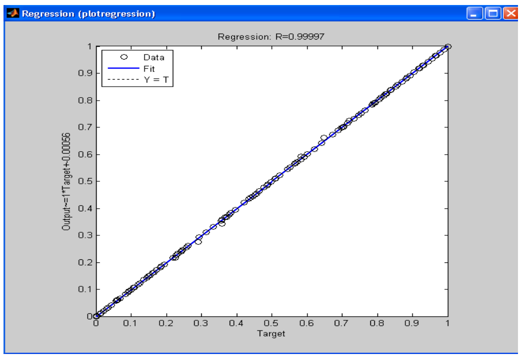


Fig. 8 A regression plot shows the output with respect to the target

TABLE III
OUTPUT TRACKS

	Samples	MSE	R
Training	42	4.04298e^{-12}	0.99999
Validation	9	5.70585e^{-6}	0.999978
Testing	9	3.77055e^{-5}	0.999795
All		6.02163e^{-6}	0.999966

ANN forms a useful tool in predicting total cost and net profit for enterprise. With sufficiently large amounts of data, ANN has strong potential for prediction and can learn with high degree of accuracy. As can be concluded from the performance of the trading strategy, ANN tends to perform well. ANN predicted the cost and net profit with high accuracy and the hypothesis of convergence to the relationship was acceptable. Furthermore, we were successfully able to get a reasonable prediction of our target (15% testing data vectors). ANN performs well high desired level of precision and a feasible. Further testing with intra-day real time data would be important to certify the profitability of this strategy. Work may also be done using ANN with inputs spread over the entire market instead of concentrating on a particular sector. Finally, good prediction carried out on the optimal design of the ANN model. This means that our model may not be the optimal design model and may find another model strong than ours.

VI. Recommendation

To improve the results, as a first approach, we have reinitialize the network and the training many times to reach sufficiently and accurate network. In each time of initialize a feed-forward network, the network parameters are changed many times until receive the desired solutions.

As a second approach, increasing the number of hidden neurons and used larger numbers of neurons in the hidden layer give the network more flexibility because the network has more parameters it can optimize. On other hand, using too large number of hidden layer might cause the problem to be under-characterized and the network must optimize more parameters than there are data vectors to constrain these parameters.

Finally, we have used additional training data. Providing additional data for the network is more likely to produce a network that generalizes well to new data.

References

[1] Hiew, M. and G. Green, "Beyond Statistics. A Forecasting System That Learns," The Forum, 1992, Vol. 5, pp. 1 and 6.

[2] Hawley, D.D., Johnson, J.D. and Raina, D. (1990), "Artificial Neural Systems: A New Tod for Financial Decision-Making," Financial Analysts Journal, (November-December) 63- 72.

[3] Huntley, D.G. (1991), "Neural Nets: An Approach to the Forecasting of Time Series," Social Science Computing Review, 9 (1), 27-38.

[4] Rumelhart, D. and J. McClelland, Parallel Distributed Processing, Cambridge: MIT Press, 1986.

[5] Wasserman, P.D., Neural Computing: Theory and Practice, Van Nostrand Reinhold: New York, 1989.

[6] Hornik, K., M. Stinchcombe, and H. White, "Multilayer Feed forward Networks are Universal Approximators," Neural Networks, 1989, 2(5), 359-366.

[7] Connor, 1988; Donaldson, Kamstra and Kim, 1993, "Artificial neural network models for forecasting and decision making".

[8] Kang, S., An Investigation of the Use of Feed forward Neural Networks for Forecasting, Ph.D. Dissertation, Kent State, 1991.

[9] Marquez, L., Function Approximation Using Neural Networks: A Simulation Study, Ph.D. Dissertation, University of Hawaii, 1992.

[10] B. Scholz-Reiter, T. Hamann, H. Höhns, and G. Middelberg, 'Decentral closed loop control of production systems by means of artificial neural networks', in Proceedings of the 37th CIRP-International Seminar on Manufacturing Systems, pp. 199 – 203, (2004).

[11] S.F. Crone, 'Forecasting in inventory management using artificial neuronal networks - a novel approach through asymmetric cost functions', in Einsatz von Fuzzy-Sets, Neuronalen Netzen und Künstlicher Intelligenz in industrieller Produktion und Umweltforschung, 59 – 69, VDI - Verlag, Düsseldorf, (2003).

[12] N. Wang and J. Yu, 'Neuron based nonlinear pid control', PRICAI 2006: Trends in Artificial Intelligence, 4099, 1089–1093, (2006).