Precise Stock Price Prediction System Using Neural Networks Trained by Enhanced and Meticulous Learning

Yogesh Karunakar[1], Abhijeet Gole[2], Priti Jha[3]

[1]University of Mumbai – askyogi@gmail.com
[2]Ruia College,University of Mumbai – abhijeet.mumbai@gmail.com
[3]Ruia College,University of Mumbai

Abstract: In Most of the developing countries, investing in stocks, albeit the risk factor is the most lucrative way of earning quick bucks. This has lead to the development of various models for financial markets and investment. Black-Scholes model opened a new domain for research in the field of stock markets. The model develops partial differential equations whose solution, the Black–Scholes formula, is widely used in the pricing of European-style options. The Aim of "Neural Network Based Stock Price Forecasting Model" is to develop a Model which will be used to Forecast Future Stock Prices. It will be developed by using one of the Concepts in Artificial Intelligence [8], "Neural Networks". Network created for this Model trained and this Trained Network is tested for Prediction of the Future Stocks Prices. One Layer or Two Layer network is created and trained by using Backpropogation Algorithm in Neural Networks. Neural Networks are a class of Pattern Recognition Methods which have been successfully implemented in Data Mining and Prediction in a variety of fields

Keywords **–** Black-Scholes model, Neural Networks, Stock markets, Backpropogation, Pattern recognition**.**

1. Introduction

Neural Network Based Stock Price Forecasting Model: It is basically a model which will be used for predicting future stock prices. It has been developed using Neural Network Concepts. Neural Network is one of the areas in Artificial Intelligence. A Neural Network is a powerful data modeling tool that is able to capture and represent complex input/output relationships. The motivation for the development of neural network technology stemmed from the desire to develop an artificial system that could perform "intelligent" tasks similar to those performed by the human brain.

Neural Networks resemble the human brain in the following two ways:

i) A Neural Network acquires knowledge through learning.

ii) A Neural Network's knowledge is stored within inter-neuron connection strengths known as synaptic weights.

The Advantage of Neural Networks lies in their ability to represent both linear and non-linear relationships and in their ability to learn these relationships directly from the data being modeled. With the neural networks' ability to learn nonlinear, chaotic systems, it may be possible to outperform traditional analysis and other computer-based methods.

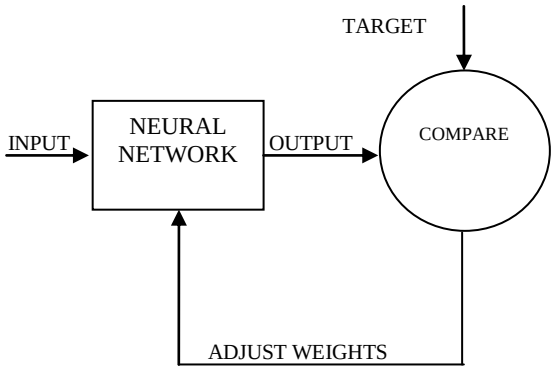


Figure1. Block representation of neural network

2. Neural Network

An Artificial Neural Network is a system based on the operation of biological neural networks, in other words, is an emulation of biological neural system. There are certain tasks that a program made for a common microprocessor is unable to perform. It's a system composed of a large number of basic elements arranged in layers and that are highly interconnected. The structure has several inputs and outputs, which may be trained to react (O's values) to the inputs stimulus (I's values) in the desired way. An artificial neuron is an element with inputs, output and memory that may be implemented with software or hardware. It has inputs (I)

that are weighted added and compared with an Activation Function.

The General Form of Equation:

$$S = I1 * w1 + I2 * w2 + \ldots + In * wn - b$$

$$O = f(S)$$

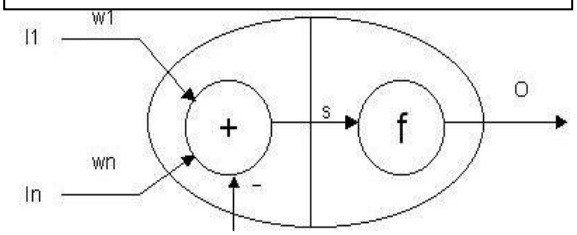
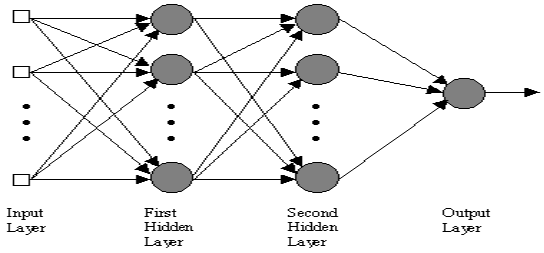


Fig 2: Structure of Artificial Neuron

The most common Neural Network Model is the Multilayer Perceptron (MLP). This type of Neural Network is known as a supervised network because it requires a desired output in order to learn. The goal of this type of network is to create a model that correctly maps the input to the output using historical data so that the model can then be used to produce the output when the desired output is unknown. A graphical representation of an MLP is shown below.



3. Training Phase or Learning Phase

A **Neural Network** can be trained in the following ways.

1. Supervised Learning
2. Unsupervised Learning

3.1 Supervised Learning:

Supervised learning is the type of learning that takes place when the training instances are labeled with the correct result, which gives feedback about how learning is progressing. This is akin to having a supervisor who can tell the agent whether or not it was correct.

The training data consist of pairs of input objects (typically vectors), and desired outputs.

Supervised Learning is a Stochastic approximation of an unknown average error, It attempts to map an unknown function F: X → Y from observed training samples (x1,y1)….(xn, yn) by minimizing an Least Mean Square error.

3.2 Unsupervised Learning:

In Unsupervised learning, we are given some data *x* and the cost function to be minimized, that can be any function of the data *x* and the network's output, *f*. The cost function is dependent on the task (what we are trying to model) and our *a priori* assumptions (the implicit properties of our model, its parameters and the observed variables).Unsupervised Learning use unlabelled Pattern samples i.e., It doesn't use Target data. Other neural network systems use similar types of input data. Simpler systems may use only past share prices[7] or chart information[8]. A system developed by Yoon[5] based its input on the types and frequencies of key phrases used in the president's report to shareholders. Bergerson[6] created a system that traded commodities by training it on human designed chart information rather than raw data. Such directed training has the advantage of focusing the neural network to learn specific features that are already known as well as reducing learning time. Finally, the self-organizing system built by Wilson[9] used a combination of technical, adaptive (based on limited support functions), and statistical indicators as inputs. Determining the proper input data is the first step in training the network. The second step is presenting the input data in a way that allows the network to learn properly without overtraining. Various training procedures have been developed to train these networks.

4. Algorithm and Strategies Developed

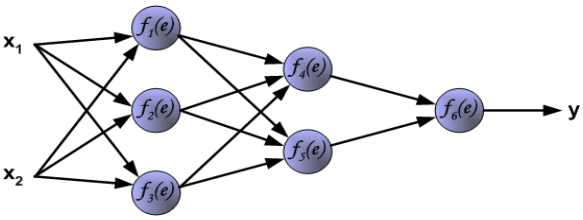
4.1 Training Algorithm

4.1.1 Back Propagation Algorithm:

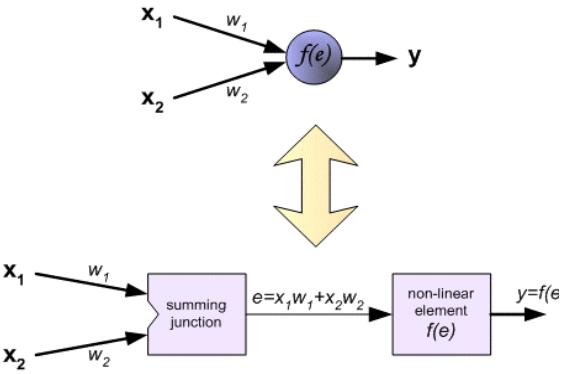
The back propagation algorithm trains a given feed-forward multilayer neural network for a given set of input patterns with known classifications. When each entry of the sample set is presented to the network, the network examines its output response to the sample input pattern. The output response is then compared to the known and desired output and the error value is calculated. Based on the error, the connection weights are adjusted. The back propagation algorithm is based on *Widrow-Hoff delta learning rule* in which the weight adjustment is done through *mean square error* [5]of the output response to the sample input. The set of these sample patterns are repeatedly presented to the network until the error value is minimized. Back Propagation Algorithm trains Multilayer Feed Forward Neural Network using Supervised Learning.

4.1.2 Graphical Representation of Steps of BackPropagation Algorithm:

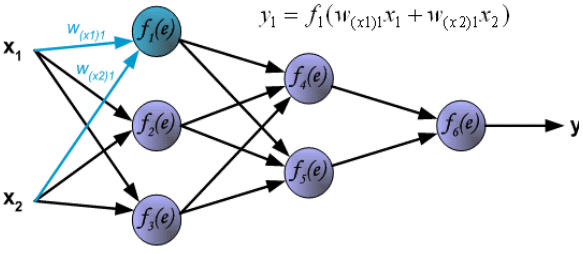
Step1:
It describes teaching process of multi-layer neural network employing *Back propagation* Algorithm. It is three layer neural network with two inputs and one Output.



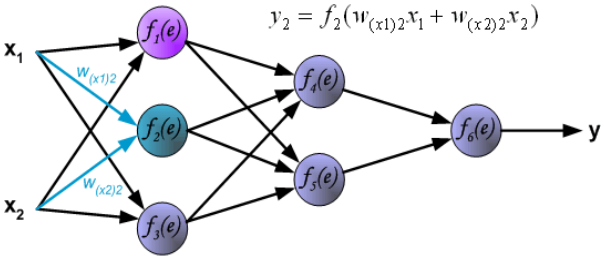
Each neuron is composed of two units. First unit adds products of weights coefficients and input signals. The second unit realizes nonlinear function, called neuron activation function. Signal *e* is adder output signal, and *y = f(e)* is output signal of nonlinear element. Signal *y* is also output signal of neuron.



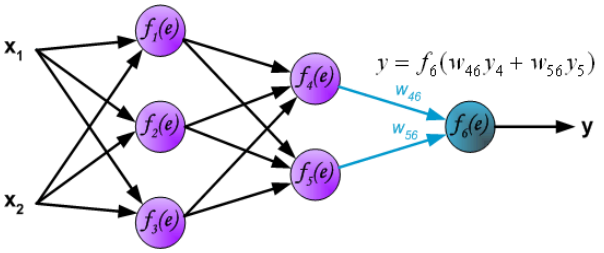
Step2:
To teach the Neural Network we need training data set. The training data set consists of input signals (x_1 and x_2) assigned with corresponding target (desired output) *z*. The network training is an iterative process. In each iteration weights coefficients of nodes are modified using new data from training data set.



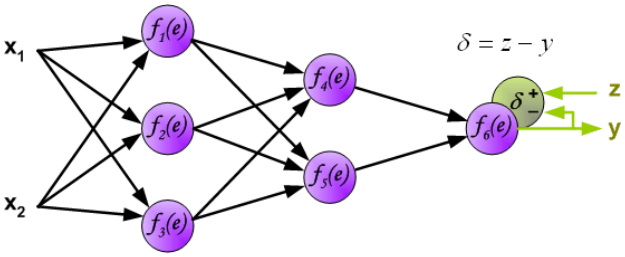
Similarly, for 2nd Neuron, It will be calculated as shown below. Each Neuron's Output will propagate through the Next Layer.



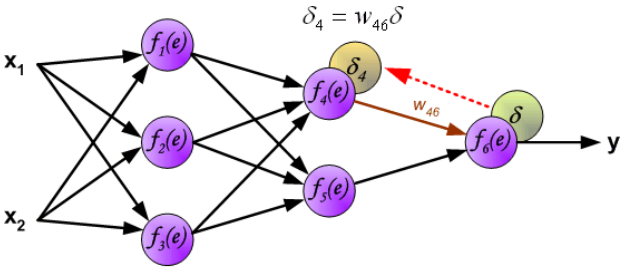
And for the Last Neuron, The final output is calculated as follows:

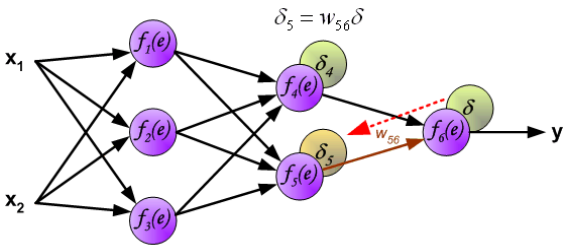


Step3:
In the next algorithm step the output signal of the network *y* is compared with the desired output value (the target), which is found in training data set. The difference is called error signal *d(delta)* of output layer neuron.

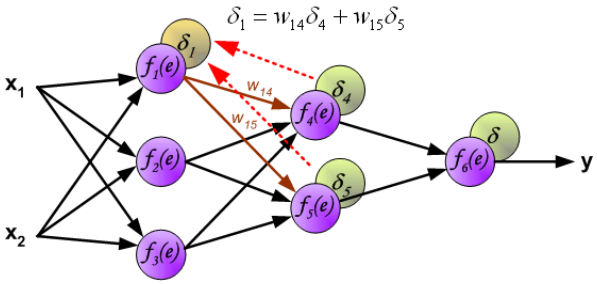


Step4:
The idea is to propagate error signal *d* (computed in single teaching step) back to all neurons, which output signals were input for discussed neuron.





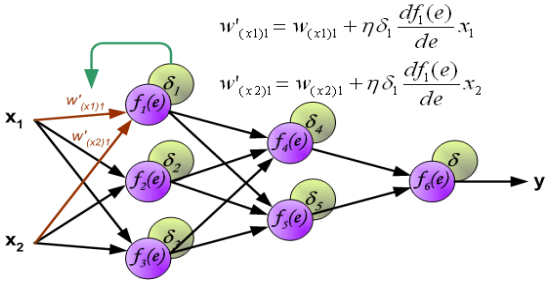
The weights' coefficients w_{mn} used to propagate errors back are equal to this used during computing output value. Only the direction of data flow is changed (signals are propagated from output to inputs one after the other).



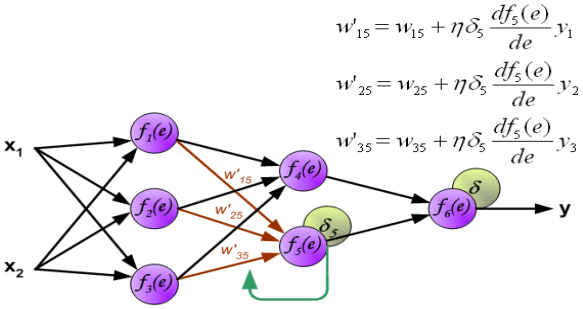
This technique is used for all network layers. If propagated errors came from few neurons they are added.

Step5:

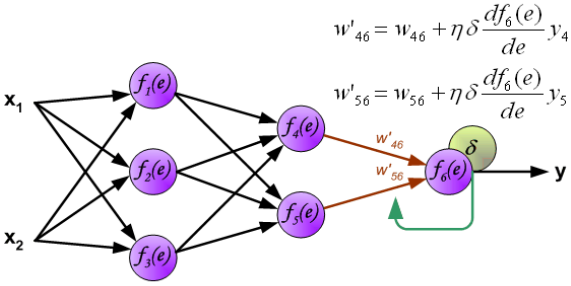
The Modified weight value is calculated for the first neuron as shown in the figure. When the error signal for each neuron is computed, the weights coefficients of each neuron input node may be modified. In Calculation shown below *df(e)/de* represents derivative of neuron activation function (which weights are modified).



Similarly, the weight values will be modified for the successive Neurons as Shown in the following figure.



After, all the weight values are modified, the whole procedure from step3 is repeated**.**



4.2 Formal Background

CreateNetwork: In this Module, An Artificial Neural Network of different layers is created. One Layer/Two Layer Network is created in Neural Network Toolbox.
Then a feed-forward Backpropagation network is created.
Load Data: In this Module, Input Data is loaded from the Database. A Preprocessing is performed on the input data. The inputs and targets are scaled so that they fall in the range [-1, 1].
Training: In this module, One Layer or Two Layer Network Created is passed through Training Phase. In Training Phase or Learning Phase, a given network is trained so as to make it learn different kinds of input which may appear in the future when it will be tested.
Finally, when the network is trained, it is simulated or tested with the inputs for which it will predict an output that will have minimum squared error (mse).

5. Implementation of the Model:

The Input Data Consists of 865 Tuples Containing Information such as Date, Open Price, High Price, Low Price, Close Price, Volume of Stocks.

Neural Network Based Forecasting Model is implemented as shown:

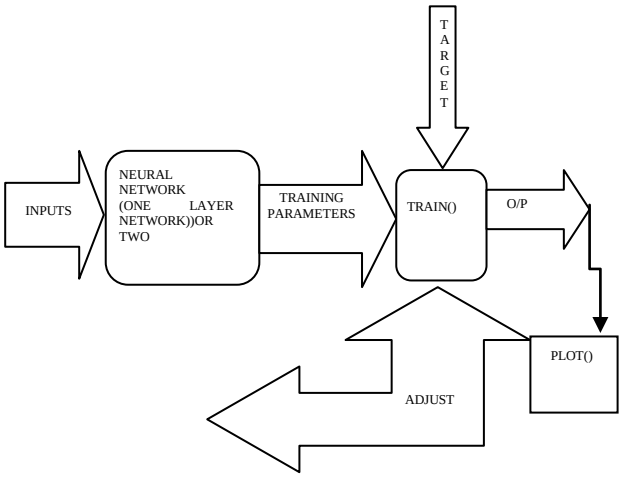
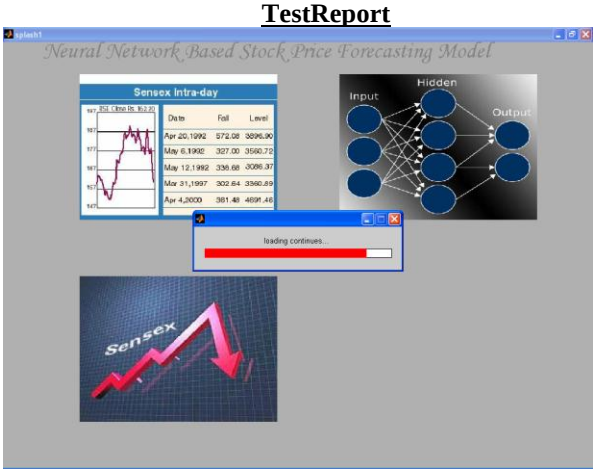
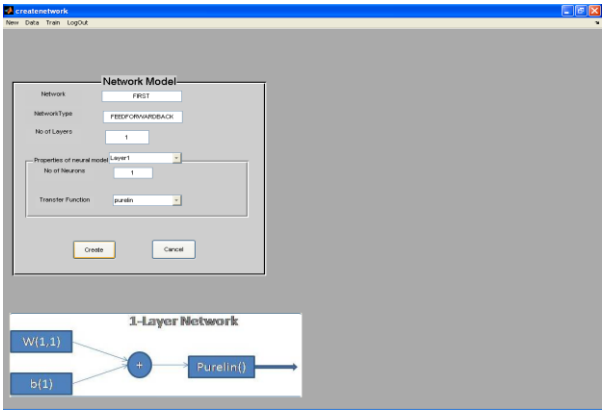


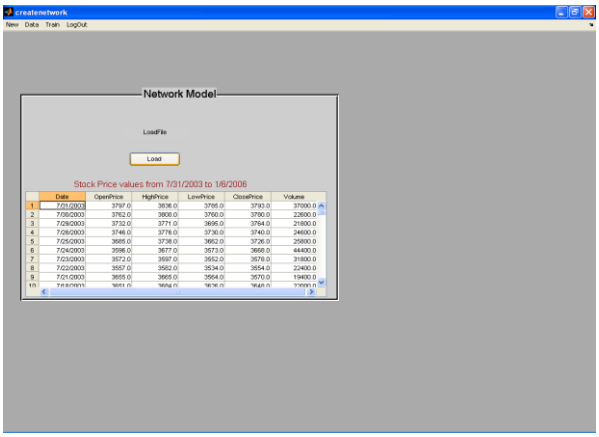
Figure: Block diagram for system implementation

TestReport





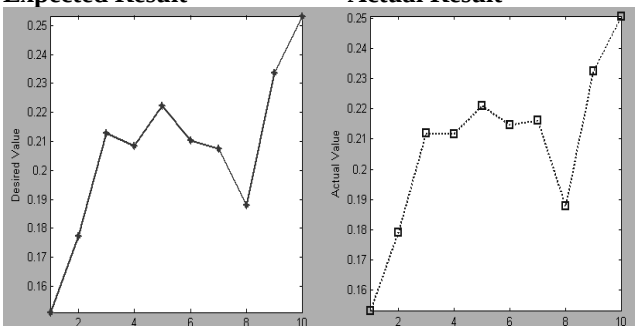
ONE LAYER NETWORK



INPUTDATA

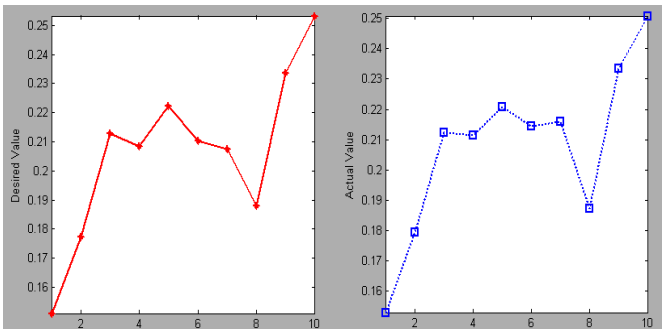
TestCaseNo: 1
TestUnit: Training Module (One Layer Network)
TestData:
Learning Rate=0.11,
Goal=0,
Momentum Unit=0.12,
TransferFunction='Purelin'.
Epochs=300.

Expected Result **Actual Result**



TestPerformance: Good

TestCaseNo: 2
TestUnit: Training Module(Two Layer Network)
TestData:
Learning Rate=0.10,
Goal=0,
Momentum Unit=0.12,
TransferFunction for 1st layer='Purelin'.
TransferFunction for 2nd layer='Purelin'.
Epochs=300.



Expected Result **Actual Result**

TestPerformance: Very Good.

6. Analysis and Conclusion:

With trainlm(), as one of the training functions , the Actual Output shown by the network is closer to the Desired Output. If one transfer function is 'tansig' and other transfer function is 'logsig' and traning function is 'trainlm', the Actual Output shown by the network shows much variations and thus not appropriate. In one layer Network, with transfer function as purelin and training function as 'trainlm' , the Output given by the Network approximates the Desired Output i.e, Minimum squared Error (MSE) is considerably low. In the Two Layer Network , with one of the transfer functions as 'tansig'/'logsig' and second transfer function as 'purelin', the output by the network falls within the range of the Desired Output , thus considerable.In the Two Layer Network , with one of the transfer functions as 'tansig'/'logsig' and second transfer function as 'purelin',learning rate=0.11 and momentum unit=0.12 ,the output by the network falls within the range of the Desired Output.

In the Two Layer Network , with one of the transfer functions as 'purelin' and second transfer function also as 'purelin', the output by the network falls exactly within the range of the Desired Output , thus making it very appropriate and (MSE) very low. With trainlm(), as one of the training functions, the Output shown by the network is appropriate, it nears the Desired Output, MSE is considerably low.

With traingd(),as one of the training functions ,the Output shows large variation , MSE high , thus not applicable for the problem under consideration.

References

[1]Addison Wesley, Eugene Charniak, "Introduction to Neural Networks".

[2]Bart Kosko , "Neural Networks and Fuzzy Systesm" A Dynamic System Approach to Machine Intelligence, Prentice-Hall of India Pvt Ltd.,1992

[3]Margaret H.Dunham, "Data mining introductory and advanced topics, Dorling Kindersley(India) Pvt.Ltd.,2006.

[4]James Garson, "Intro to Connectionism" Introduction to science and theory of connectionism, 1997.

[5] Y. Yoon and G. Swales. Predicting stock price performance: A neural network approach. In *Neural Networks in Finance and Investing*, chapter 19, pages 329–342. Probus Publishing Company, 1993

[6] K. Bergerson and D. Wunsch. A commodity trading model based on a neural network-expert system hybrid. In *Neural Networks in Finance and Investing*, chapter 23, pages 403–410. Probus Publishing Company, 1993.

[7] G. Tsibouris and M. Zeidenberg. Testing the Efficient Markets Hypothesis with gradient descent algorithms. In *Neural Networks in the Capital Markets*, chapter 8, pages 127–136. John Wiley andSons, 1995.

[8] K. Kamijo and T. Tanigawa. Stock price pattern recognition: A recurrent neural network approach. In *Neural Networks in Finance and Investing*, chapter 21, pages 357–370. Probus Publishing Company,1993.

[9] C. L. Wilson. Self-organizing neural network system for trading common stocks. In *Proc. ICNN'94,Int. Conf. on Neural Networks*, pages 3651–3654, Piscataway, NJ, 1994. IEEE Service Center.