TAG XVIII
New Orleans
1996

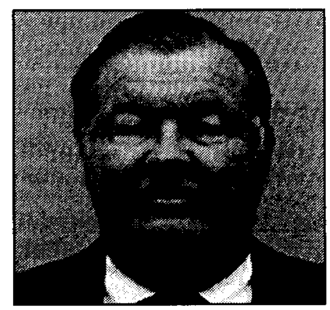
Designing Neural Networks for Financial Forecasting

New neural network users often follow a predictable pattern. They feed the network one year of data and a haphazard roster of technical studies including the RSI, MACD, ADX , three moving averages, Stochastic, and any other handy data. These new users then expect the network to generate accurate predictions thirty days into the future. They are dismayed and perplexed when they discover that the forecasts are worthless.

Ed Gately's workshop will cover the human biological brain and artificial neural networks. You will learn the proper steps you must take to generate meaningful results. He will show you how to select appropriate inputs and how to prepare, or preprocess, and otherwise manipulate your data prior to training your software. He will teach you how to extract the test data and how to train the neural net to the correct accuracy level. He will describe network architectures and activation functions, and will show you what to do if the network will not train to the desired accuracy level.

Finally, Ed will discuss some of the classic traps which await the beginning neural network user, such as colinearity, price shocks caused by such incidents as the Persian Gulf War, and the onset of option trading in a specific item. He will also describe the effects of other non-typical market events.

Anyone who has contemplated the use of neural net software or who wants to be current with this promising field will enjoy and benefit from this workshop.



Ed Gately

Ed Gately received his degree in physics and started his professional career as an electronics engineer. He worked his way up the corporate ladder to become the president of a successful electronics company. He began actively trading the stock market upon his retirement from that firm. At that time Ed began to research neural networks extensively as a complement to technical analysis.

Ed developed a neural network for forecasting the S&P 500 ten days into the future which, when tested with fresh data, generated amazingly accurate results. When forecasting ten days out, 38 percent of the predicted points were within one point of the actual S&P figure; 76 percent were within two points; and 93 percent within four points. When forecasting five days into the future the results were even better, with 56 percent of the predicted points coming within one point of the actual S&P; 77 percent within two points; and 99 percent within four points. Ed's book Neural Networks for Financial Forecasting *builds upon this research.*

Ed spends most of his time developing, writing about, and lecturing on neural networks. In addition to these efforts, he is currently working on a new book which will describe how to establish price and time targets in the financial markets.■

89 Hopkins
Manwah, NJ 07430
(201) 236-2307
FAX (201) 236-8979

A complete catalogue of all TAG XVIII tapes may be ordered from:

Telerate Seminars
701 Poydras, Suite 3900 • New Orleans, LA 70139-3901
(800) 535-7990 or (504) 592-4550
FAX (504) 592-4553

Copyright GATELY 1996.
All Rights Reserved.

BACKGROUND ON MECHANICAL-ELECTRONIC COMPUTING

Crude mechanical adding machines circa late 1700's

Electronic 4 function analog electronic computing
- Bell Labs 1930's

Electronic analog integration and differentiation
- Bell Labs 1940's

ENIAC - University of Pennsylvania - 1040's

BACKGROUND ON DEVELOPMENT OF NEURAL NETWORKS (ANN)

Theoretical basis - paper by McCullock and Pitts - 1943

Minsky - 1951 at MIT built a neural computer for learning a maze

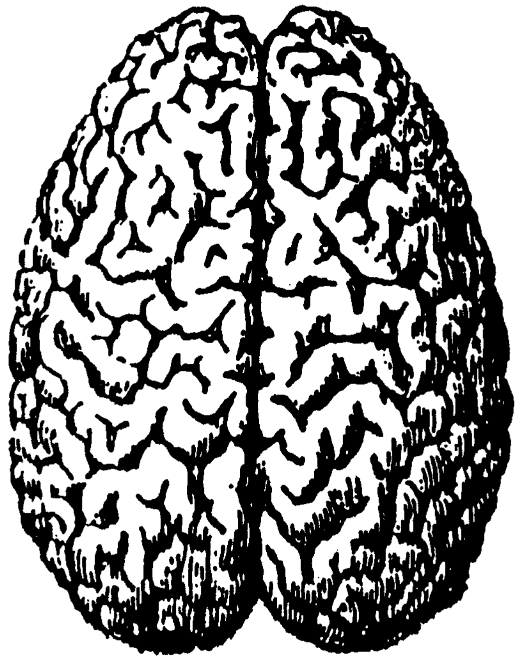
By late 1970's computers had sufficient power for practical research on ANN's

Development of Back Propagation in 1986 enabled solving of everyday problems

APPLICATIONS OF NEURAL NETWORKS

Credit card fraud
Bankruptcy prediction
Credit card applications
Mortgage applications
Product marketing
Corporate bond grading
Municipal bond grading
Stock market prediction
Bank failure prediction
Stock selection
Currency price prediction
Real estate appraisal
Crop forecasting
Commodity trading
Arbitrage pricing
Analysis of corporate financial health
Option pricing
Sales prospect selection
Capital markets analysis
Managerial decision making
Travel voucher screening
Security risk profiling
Economic indicator forecasts
Property tax analysis
Cash flow forecasting
Locating of tax evaders
Mutual fund selection
Predicting changes in market trend
Forecasting personnel requirements
Forecasting machine tool loading

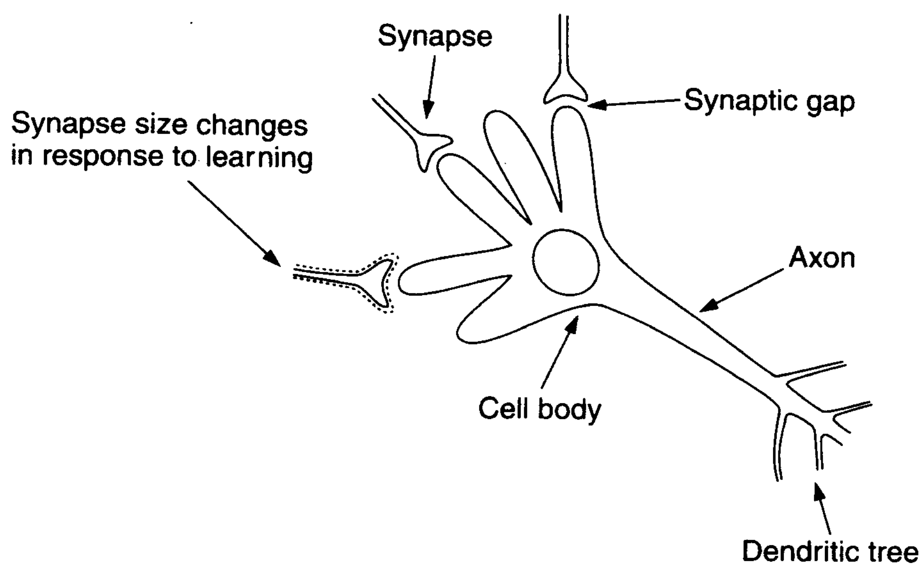
MACRO BRAIN STRUCTURE



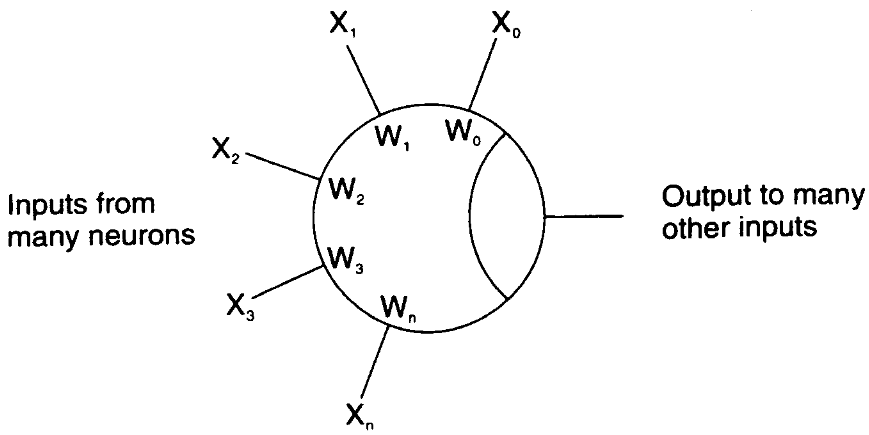
Top view of the brain showing division into left and right halves.

The two sides of the brain are connected by the *Corpus Callosum*, a thick bundle of nerve fibers.

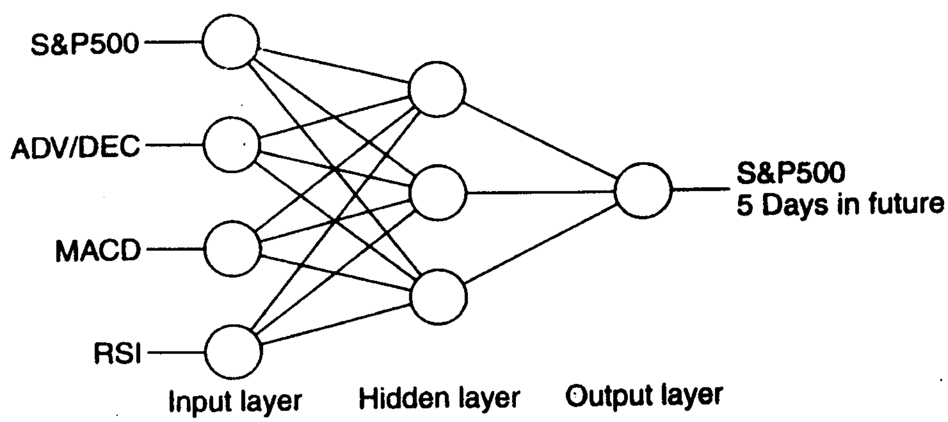
MICRO BRAIN STRUCTURE



Details of a biological neuron.



Schematic representation of a biological neuron showing how the neuron combines many inputs into an output that is in turn connected to many other neurons. For clarity, only one of many outputs is shown. The Xs represent inputs and the Ws are the interconnections between the inputs.



Simple three-layer artificial neural network for predicting the S&P 500 five days in advance using today's value of the S&P, the Advance/Decline Index, MACD, and RSI as inputs.

SUITABLE PROBLEMS FOR AN ANN

Rational problems, e.g. 2.000 + 2.000, best solved by conventional means. ANN would get answers 3.933 one time and 4.003 the next.

ANN best on intuitive problems.

DIFFERENCES BETWEEN HUMAN BRAIN AND ANN

Factor	Brain	Neural Network
Size	Very large, up to 100 billion neurons	Relatively small, rarely more than a few hundred neurons
Inputs	5 senses	Data sets, reduced to numbers, (digitized)
Work ethic	Good, but easily bored and often unfocused	Mechanical, solves only what is asked
Specialization	Yes, but with many interests and skills; can be easily distracted	1 task only
Construction	Biological	Mathematical
Interconnections per neuron	Up to 1000 not uncommon	Limited by state of art, rarely as many as 100
Brain side	Two, left and right (Analytic and intuitive)	Right side, only (intuitive)

Differences between human brain and neural network.

OVERVIEW OF DISCRETE STEPS FOR TRAINING AND USING ANN

- *Step 1. Determine What Is to Be Forecast.* Forecasting the snowfall in Vermont in November, while of some business interest, is irrelevant to your trading of the S&P 500.
- *Step 2. Collect Data Sets or Number Series That Have a Relationship to What Is to Be Forecast.* This information will be used to create the forecast. Your golf score on Sunday will have little relationship to the price of the S&P 500 a week from now, however, today's price of oil may have a large effect on next weeks S&P 500 price.
- *Step 3. Preprocess the Data to Combine Information or Make It More Useful.* Presenting the network with the price of gold today may help the network forecast the S&P 500 a week from now; however, presenting the network with the *change* in the price of gold over the past week should be much more meaningful to the network.

- *Step 4. Set Minimums and Maximums.* Determine the range of data and set minimum and maximum values to these levels. This makes processing more efficient by focusing the network's attention on the pertinent data—if the data goes from 100 to 200 then the minimums and maximums would be set at 100 and 200 not at −100 and 1000.

- *Step 5. Extract Test Set Data.* In order to be able to test the network with data that it has not seen before, that is, data used for *training* the network, it is common to set aside some of the data set for future testing. Therefore, the total data set is divided into two different sets, one for training and one for testing.

- *Step 6. Select Suitable Network Architecture.* Networks can have three, four, or even five layers; the number of neurons in the hidden layers can vary from one to several hundred. This construction of the network is called the *architecture of the network.* There are more than a dozen known architectures and some are better suited to solving a particular problem than others.

- *Step 7. Select a Suitable Learning (Training) Algorithm.* There are several algorithms (methods of training) that can be used with any particular network architecture. For specific problems, some work better than others.

- *Step 8. Train the Network.* Apply training data to a particular architecture and training algorithm. This adjusts the weights, which connect the neurons, in such a way that the network makes good forecasts when presented with that or similar data sets.

- *Step 9. Use the Network.* Apply new data sets or current data to the trained network to create a forecast.

CHOOSING WHAT IS TO BE PREDICTED AND SELECTING APPROPRIATE INPUTS

1. A studied selection of the target output,
2. Selection of inputs that have predictive relationship to the selected output, and
3. Selection of the optimum neural network architecture and training algorithms.

TIME SERIES FORECASTING OF ABSOLUTE VALUE DOES NOT WORK

Some of the variables that can be substituted for the absolute price level in time series prediction include:

1. The recent price change.
2. Whether this most recent data represents a market top or bottom.
3. Whether to buy, sell, or hold.
4. Whether forecast date represents the highest high or lowest low.
5. The change in value over the forecast period.
6. Whether the trend is up or down.
7. The rate of change.
8. The quality of making a trade at this time.
9. Whether the most recent data represents a change in trend.
10. Volatility.
11. Change in volatility.

BENCHMARK

1. **S&P futures contract (SPY).**
2. **New York Stock Exchange 100 index (YXY).**
3. **Number of NYSE advances (ADV).**
4. **Number of NYSE declines (DEC).**
5. **Tick volume (TICK).**
6. **Number of up issues/number of down issues divided by volume of up issues/volume of down issues (TRIN).**

- **The network predicts the SPY five hours in the future.**
- **The network has 6 neurons in the hidden layer.**
- **The network has 1 output neuron.**
- **The network is of the back-propagation type.**
- **The network is trained using a momentum algorithm with a learn rate of 0.05 and a momentum factor of 0.5.**

- After every 200 runs during training, the network is applied to the test data (fresh data previously set aside) and the training stopped when the network can no longer improve its performance on the test data.

1

ABSOLUTE VALUE

	A	B	C
1	Actual(1)	Network(1)	Act-Net(1)
2	448.64	449.47	-0.83
3	448.23	449.78	-1.55
4	448.04	450.28	-2.24
5	447.94	449.82	-1.88
6	448.41	449.40	-0.99
7	448.66	449.76	-1.10
8	449.22	449.38	-0.16
9	448.38	449.01	-0.63
10	447.29	448.74	-1.45
11	447.57	448.38	-0.81
12	447.55	448.26	-0.71
13	446.92	448.75	-1.83
14	446.55	448.86	-2.31
15	445.75	449.20	-3.45
16	446.30	447.89	-1.59
17	445.31	447.08	-1.77
18	445.34	447.21	-1.87
19	445.18	447.22	-2.04
20	445.70	446.82	-1.12
21	445.85	446.51	-0.66
22	446.02	445.70	0.32
23	444.22	446.58	-2.36
24	445.79	446.00	-0.21
25	446.75	446.21	0.54
26	447.49	446.13	1.36
27	447.08	446.39	0.69
28	447.08	446.63	0.45
29	447.31	446.76	0.55
30	446.14	445.67	0.47
31	446.48	446.85	-0.37
32	446.26	447.68	-1.42
33	446.46	448.27	-1.81
34	446.23	447.87	-1.64
35	446.39	447.98	-1.59
36	447.18	447.98	-0.80
37	446.44	446.43	0.01
38	447.00	446.80	0.20
39	446.76	446.77	-0.01
40	446.52	447.08	-0.56
41	446.20	446.86	-0.66
42	444.11	446.92	-2.81
43	444.51	447.52	-3.01
44	446.58	446.76	-0.18
45	445.79	447.43	-1.64
46	445.26	446.97	-1.71
47	445.27	446.89	-1.62
48	446.21	446.56	-0.35
49	446.83	445.07	1.76
50	447.10	445.25	1.85
51	448.75	446.61	2.14
52	448.32	445.92	2.40
53	448.42	445.57	2.85
54	448.62	445.65	2.97

Test results of benchmark network. Printout of the forecast results of the benchmark system. Column A gives the actual value, column B gives the networks forecast value, and column C gives the difference between the actual value and the forecast value. Column C is also called the network error.

	A	B
1	Actual(1)	Network(1)
2	1.00	1.00
3	1.00	0.99
4	1.00	1.00
5	1.00	0.99
6	1.00	1.00
7	1.00	0.99
8	1.00	0.99
9	1.00	0.49
10	1.00	0.47
11	1.00	0.34
12	1.00	0.32
13	1.00	0.39
14	1.00	0.43
15	1.00	0.51
16	1.00	0.91
17	1.00	0.98
18	1.00	0.90
19	1.00	0.91
20	1.00	0.97
21	1.00	0.63
22	0.00	0.00
23	1.00	0.72
24	0.00	0.06
25	0.00	0.13
26	0.00	0.17
27	0.00	0.14
28	0.00	0.25
29	0.00	0.32
30	0.00	0.0000
31	0.00	0.51
32	1.00	0.96
33	1.00	0.91
34	1.00	0.87
35	1.00	0.89
36	1.00	0.92
37	0.00	0.05
38	0.00	0.19
39	0.00	0.15
40	0.00	0.28
41	1.00	0.26
42	1.00	0.32
43	1.00	0.58
44	0.00	0.19
45	1.00	0.63
46	1.00	0.42
47	1.00	0.39
48	0.00	0.25
49	0.00	0.000000
50	0.00	0.000
51	0.00	0.12
52	0.00	0.00
53	0.00	0.01
54	0.00	0.12

Figure 3–3 Up/down forecasts. Column B shows results of training the benchmark network to predict whether the SPY would be up or down 5 periods from now. Column A is the actual result where 1 is up and 0 is down.

	A	B	C	D	E
1	Actual(1)	Network(1)	Act-Net(1)	VALUE	ERROR
2	0.00	0.47	-0.47		
3	0.00	0.38	-0.38		
4	0.00	0.36	-0.36		
5	0.00	0.27	-0.27		
6	0.00	0.20	-0.20		
7	0.06	0.22	-0.16	0.99	-0.72
8	-0.04	0.19	-0.23	0.86	-1.04
9	-0.52	-0.14	-0.39	-0.63	-1.76
10	-0.55	-0.13	-0.42	-0.59	-1.89
11	-0.58	-0.16	-0.42	-0.72	-1.89
12	-0.57	-0.17	-0.40	-0.77	-1.80
13	-0.37	-0.12	-0.25	-0.54	-1.13
14	0.00	-0.07	0.07	-0.32	0.32
15	0.22	-0.02	0.25	-0.09	1.13
16	0.08	-0.14	0.21	-0.63	0.95
17	-0.15	-0.26	0.11	-1.17	0.50
18	-0.19	-0.24	0.05	-1.08	0.23
19	-0.25	-0.24	-0.00	-1.08	0.00
20	-0.51	-0.29	-0.23	-1.31	-1.04
21	-0.41	-0.24	-0.17	-1.08	-0.77
22	-0.34	-0.22	-0.12	-0.99	-0.54
23	-0.28	-0.12	-0.17	-0.54	-0.77
24	-0.50	-0.22	-0.28	-0.99	-1.26
25	-0.35	-0.15	-0.21	-0.67	-0.95
26	-0.31	-0.17	-0.14	-0.77	-0.63
27	-0.01	-0.12	0.11	-0.54	0.50
28	-0.10	-0.11	0.01	-0.50	0.05
29	0.16	-0.07	0.23	-0.32	1.04
30	-0.25	-0.22	-0.03	-0.99	-0.14
31	0.14	-0.04	0.18	-0.18	0.81
32	0.24	0.08	0.16	0.36	0.72

Figure 3–4 Percentage change forecast. Results of training network to forecast the percentage change expected rather than a value or change in value. Refer to the text for detailed explanation as to what each column represents.

SELECTING SUITABLE TIME HORIZON

4

	Type of Network	
Forecast Periods	GRNN	Backpro
1	0.9740	0.981
2	0.9695	0.965
3	0.9495	0.923
4	0.9564	0.912
5	0.9509	0.909
6	0.9766	0.900
8	0.9893	(0.887)
10	0.985	0.873
15	0.976	0.855
20	0.959	0.852
35	0.952	0.869
50	0.910	0.788
100	0.804	0.684

Table 3–1 Comparison of two architectures when period is varied. Resulting values of R-squared are shown, the correlation between actual and forecast values, when forecast period is varied for two different network architectures.

5

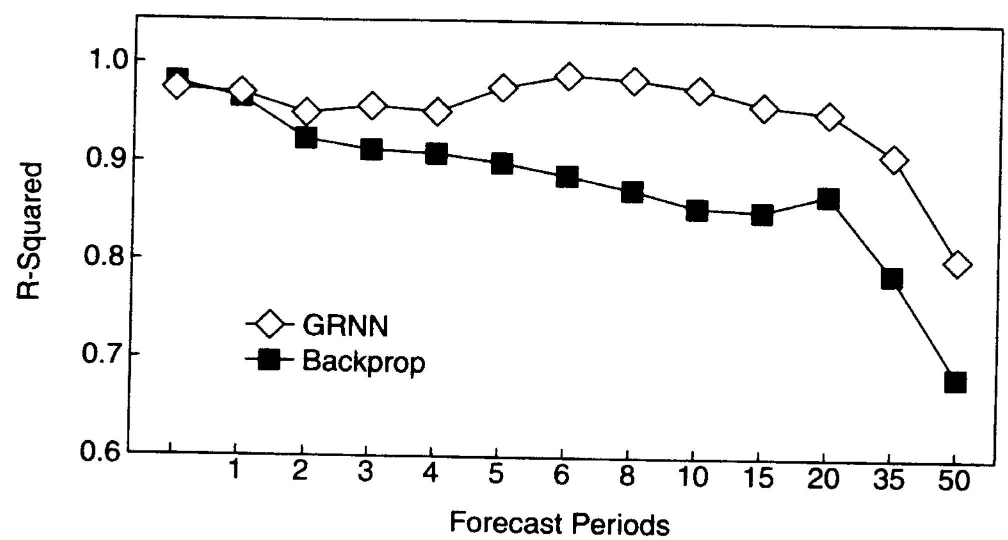
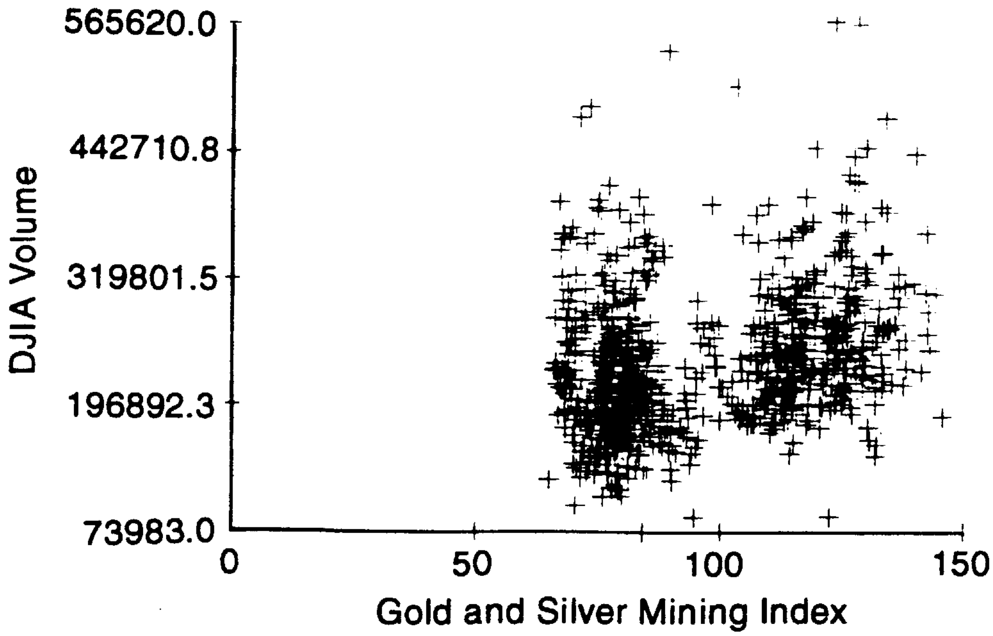


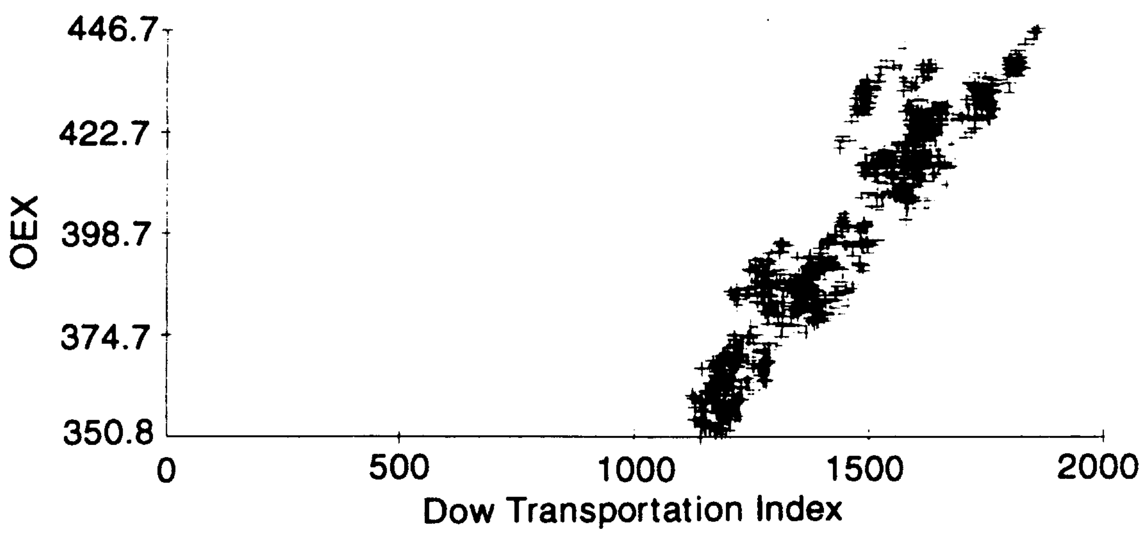
Figure 3–5 R-squared values for two different network architectures when forecasting for different periods. The backpropagation decays through period 6. Longer periods are difficult to forecast.

6



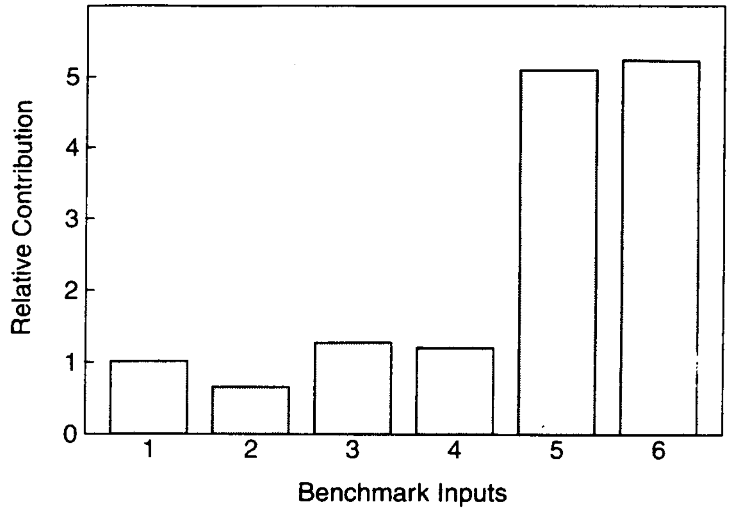
Scatter graph of DJIA volume plotted against the Gold and Silver Mining Index. The lack of form indicates that there is no relationship between the items being compared.

7



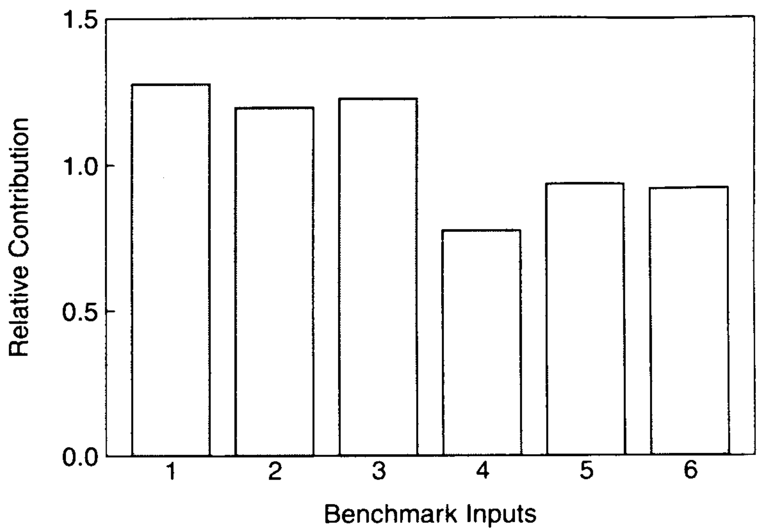
OEX and Dow Jones Transportation Index plotted as a scatter graph. The elongated form shows that there is a relationship between these two variables. A straight line would indicate a perfect relationship. The width of the band is a measure of noice and real differences between the two indices.

8



Contribution of the benchmark inputs. Relative contribution of benchmark inputs when forecasting value of SPY 5 periods in the future. The number of the inputs corresponds to the numbers in Table 4-1.

9



Benchmark inputs forecasting the percentage change in the SPY. Relative contribution of benchmark inputs when forecasting the percentage change of the SPY 5 periods in advance. Compare to Figure 4-3.

Input Selection - Time Series Forecasting

- Fundamental Factors
  - Price of Gold
  - $/Yen
  - 3 Month Treasury Bill
  - GNP
  - Inflation, etc.

Technical Factors

Volatility

Relative Strength Index (RSI)

Average Directional Movement (ADX)

Momentum

Stochastic, etc.

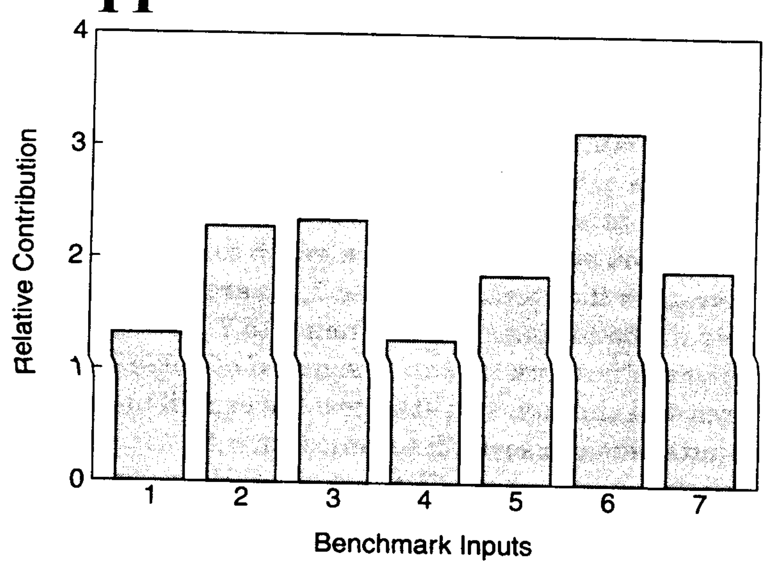
Effects of Adding Various Inputs

10

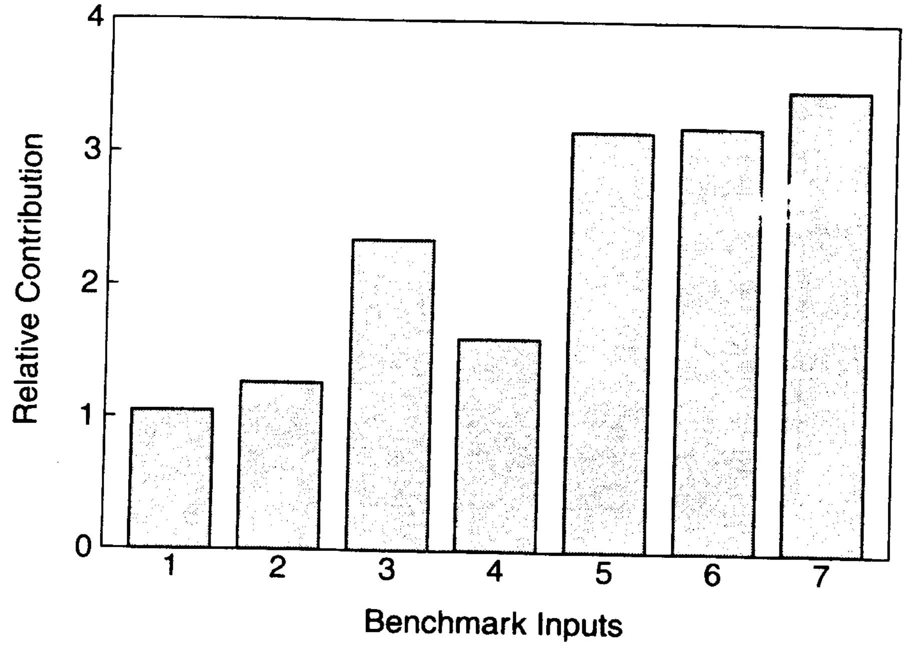
Variation of R^2 (error) with different inputs. Each input was added as a seventh input to the benchmark network.

Test #	7th Input	R-Squared
1	5-Period lag of SPY	0.9327
2	10-Period moving average less 5-period moving average	0.9302
3	Momentum	0.9302
4	% Width of Bollinger band	0.9284
5	Stochastic	0.9283
6	10-Period moving average	0.9292
7	Moving average convergence/divergence	0.9278
8	10-Period lag of SPY	0.9278
9	5-Period moving average	0.9240
10	12-Period moving average of RSI	0.9169
11	12-Period relative strength index (RSI)	0.9167

11



First of a series of graphs showing the effect of adding various technical indicators to the benchmark network. In each case, the 7th column is the added indicator. The other 6 inputs are those of the benchmark network. R-squared values are given for each network. For reference, the benchmark network had an R-squared of 0.9262. For this graph, a 5-period lag of the SPY was added as the seventh input. The R-squared value is 0.9327.



Benchmark with 10 day moving average of SPY. In this case, a 10-period moving average of SPY was used as the 7th input. The R-squared value was 0.9292. Compare the relative contributions of the 7th inputs in this and the previous figures. They could not be more different. Yet the R-squared values differ by only one number in the third place. Also notice the difference between inputs 3 and 5.

PREPARING DATA

1. Collect data via modem from Dow Jones, Dial Data, CSI, etc.

2. Transfer Data from Technical Analysis program to Spreadsheet.

3. If data is in text mode (most likely) use *Parse*function of spreadsheet to separate columns. Parsing will change data from month-day-year to day of the century e.g. 33,333.

4. Use *File Combine* function to bring various data, e.g. Tick, Trin, Spy, etc., to single spreadsheet. Do not earse date information. Check that data on each row has the same date. Move data up/down to get all data in time sequence. In the case of missing data use average of data on each sie of missing data. More amateur Neural Networks fail from lack of date integrity than any other reason. Garbage in - garbage out.

5. At this point the data can be preprocessed.

	A	B	C	D	E	F	G	H
1	ADV	DEC	TICK	TRIN	SPY	YXY	Lead(5) of SPY	Lag(5) of SPY
2	994.00	540.00	426.00	0.86	450.23	249.42	450.50	
3	1070.00	625.00	88.00	0.79	450.28	249.41	450.08	
4	1144.00	663.00	248.00	0.81	451.00	249.71	448.64	
5	1122.00	687.00	153.00	0.95	450.70	249.58	448.23	
6	1201.00	703.00	252.00	1.06	450.65	249.57	448.04	
7	1213.00	740.00	112.00	0.88	450.50	249.49	447.94	450.23
8	1214.00	781.00	31.00	0.91	450.08	249.26	448.41	450.28
9	668.00	809.00	-39.00	0.91	448.64	248.69	448.66	451.00
10	729.00	924.00	-12.00	0.91	448.23	248.50	449.22	450.70
11	777.00	965.00	12.00	1.01	448.04	248.44	448.38	450.65
12	790.00	1012.00	-2.00	1.02	447.94	248.31	447.29	450.50
13	838.00	1048.00	28.00	0.98	448.41	248.52	447.57	450.08
14	892.00	1008.00	85.00	0.98	448.66	248.70	447.55	448.64
15	938.00	1002.00	109.00	0.97	449.22	248.83	446.92	448.23
16	688.00	764.00	-74.00	1.31	448.38	248.62	446.55	448.04
17	718.00	919.00	-125.00	1.52	447.29	248.10	445.75	447.94
18	774.00	961.00	38.00	1.45	447.57	248.20	446.30	448.41
19	807.00	976.00	-36.00	1.52	447.55	248.21	445.31	448.66
20	804.00	1043.00	-94.00	1.60	446.92	247.90	445.34	449.22
21	812.00	1071.00	84.00	1.71	446.55	247.76	445.18	448.38
22	904.00	1005.00	115.00	1.80	445.75	247.42	445.70	447.29
23	724.00	763.00	40.00	1.00	446.30	247.59	445.85	447.57
24	724.00	926.00	-110.00	1.08	445.31	247.15	446.02	447.55
25	760.00	989.00	-32.00	0.96	445.34	247.11	444.22	446.92
26	786.00	1038.00	-153.00	0.97	445.18	246.99	445.79	446.55
27	809.00	1046.00	61.00	0.98	445.70	247.20	446.75	445.75
28	834.00	1070.00	-63.00	0.93	445.85	247.23	447.49	446.30
29	855.00	1060.00	-10.00	0.90	446.02	247.32	447.08	445.31
30	544.00	985.00	-235.00	0.90	444.22	246.41	447.08	445.34
31	675.00	1031.00	0.00	0.78	445.79	247.04	447.31	445.18
32	722.00	1010.00	104.00	0.69	446.75	247.47	446.14	445.70
33	777.00	1025.00	28.00	0.70	447.49	247.78	446.48	445.85
34	821.00	1049.00	7.00	0.71	447.08	247.58	446.26	446.02
35	830.00	1075.00	-54.00	0.68	447.08	247.57	446.46	444.22
36	871.00	1056.00	88.00	0.71	447.31	247.69	446.23	445.79
37	648.00	831.00	-119.00	1.03	446.14	247.20	446.39	446.75
38	729.00	923.00	-31.00	0.97	446.48	247.34	447.18	447.49
39	740.00	995.00	-80.00	0.96	446.26	247.24	446.44	447.08
40	787.00	1044.00	-91.00	0.88	446.46	247.29	447.00	447.08
41	802.00	1085.00	-182.00	0.93	446.23	247.16	446.76	447.31
42	842.00	1078.00	-103.00	0.91	446.39	247.22	446.52	446.14
43	903.00	1054.00	40.00	0.86	447.18	247.61	446.20	446.48
44	672.00	808.00	-45.00	0.90	446.44	247.27	444.11	446.26
45	750.00	878.00	8.00	0.79	447.00	247.53	444.51	446.46
46	813.00	918.00	-114.00	0.92	446.76	247.37	446.58	446.23
47	811.00	974.00	-171.00	0.92	446.52	247.26	445.79	446.39
48	820.00	1035.00	-165.00	0.99	446.20	247.08	445.26	447.18
49	757.00	1199.00	-535.00	1.19	444.11	246.12	445.27	446.44
50	770.00	1207.00	-265.00	1.17	444.51	246.17	446.21	447.00

Small section of typical neural network spreadsheet.

Preprocessing extracts the information that has real value to the network

Two variables can be reduced into one input which has more value to the network than the orgional variables, e.g. ADV/DEC.

1. Create Lags (Telescoping) by copying data to new column and moving it down the appropiate number of units.

2. Create item to be forecast by copying corresponding input to new column and moving data up the appropiate number of units.

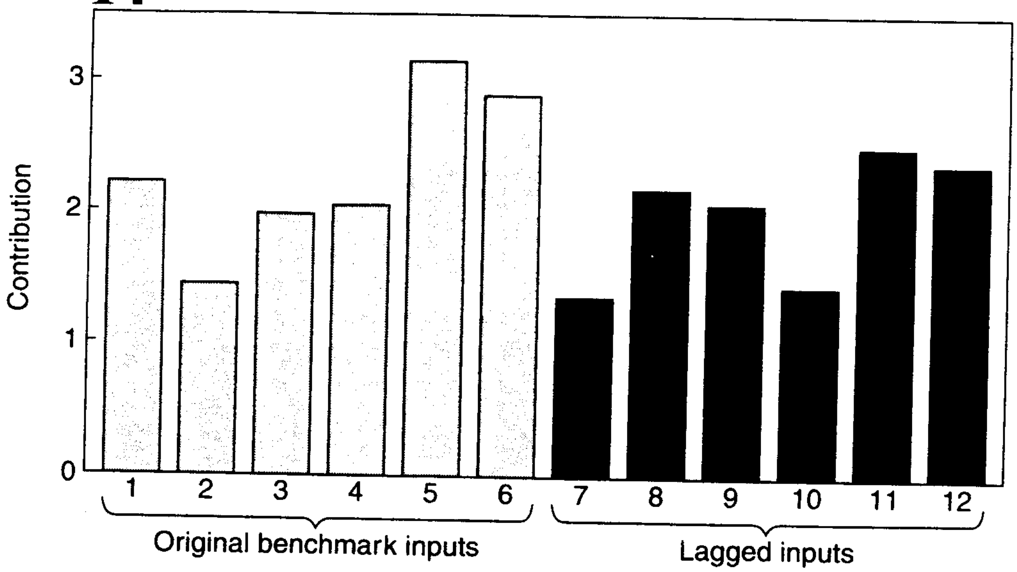
3. Create moving averages

4. Add technical indicators

5. Create differences by subtracting present value from value X units ago

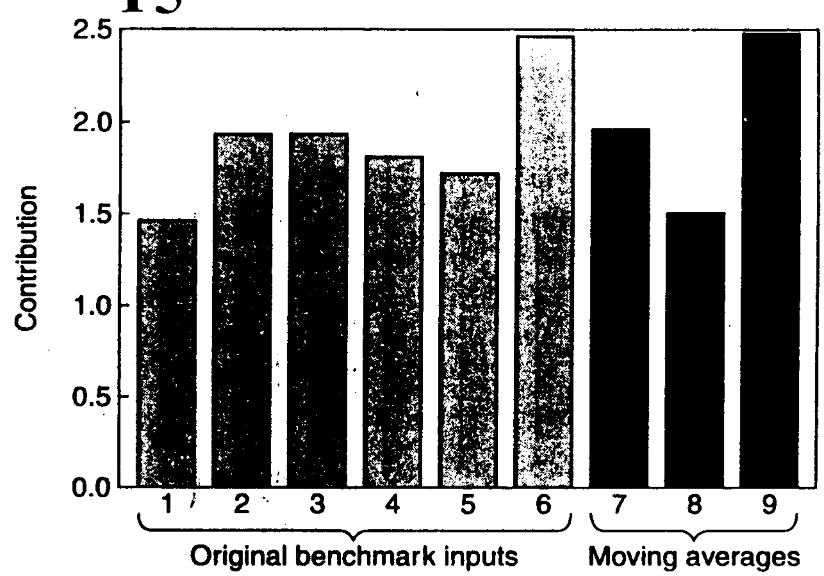
6. Create Quotients

14



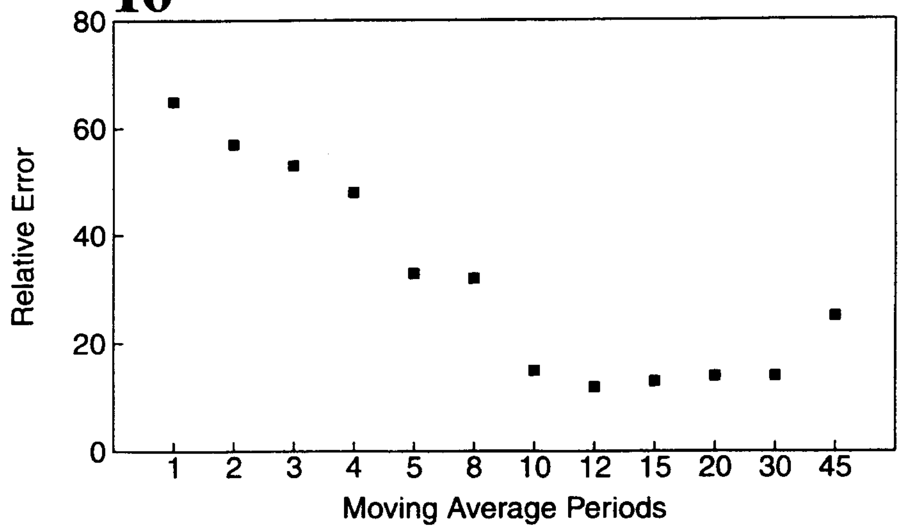
Adding lagged inputs to benchmark system. Contribution graph showing the effect of adding 5-period lagged inputs to original benchmark inputs (numbered 1–6). Note that lagged inputs (inputs 7–12) contribute to the solution almost as much as the original inputs. Input 7 is a lagged version of input 1, and so forth.

15



Adding moving averages. Contribution of original benchmark inputs (1–6) and moving averages of SPY (inputs 7–9). Moving average periods were 4, 9, 18 periods. Note that moving averages contribute as much to the solution as the original inputs.

16



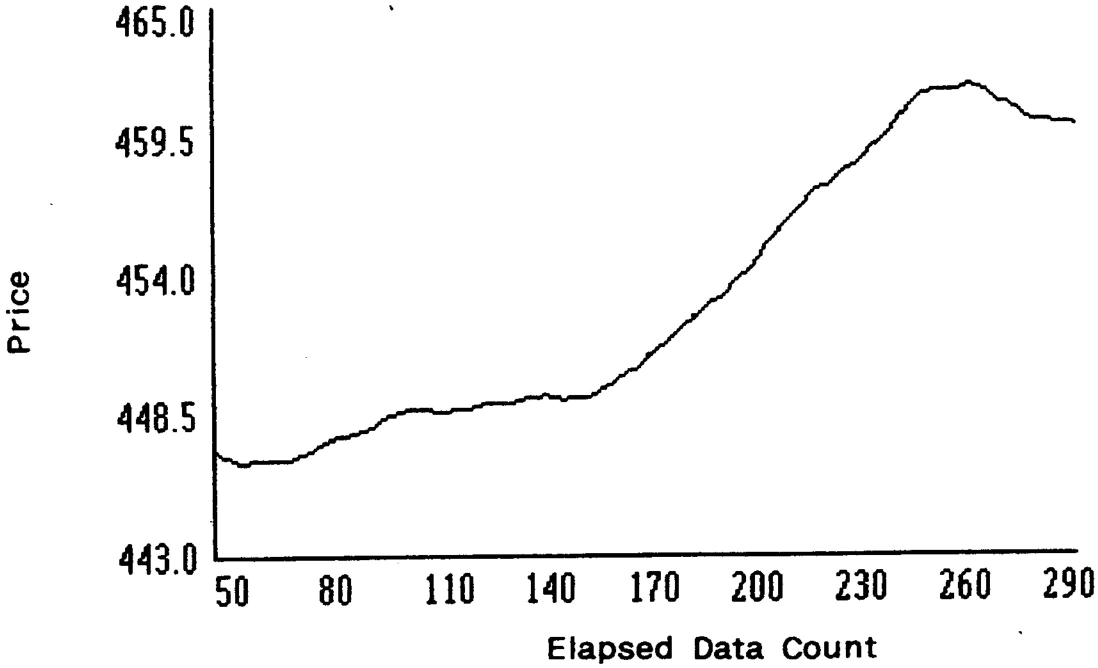
Locating optimum moving average. Effect of adding various moving averages to the network described in Appendix A. The sudden change in error between the 8 and 10-period moving average is real. The data was tested several times to establish that the discontinuity existed.

Normalizing Data

17

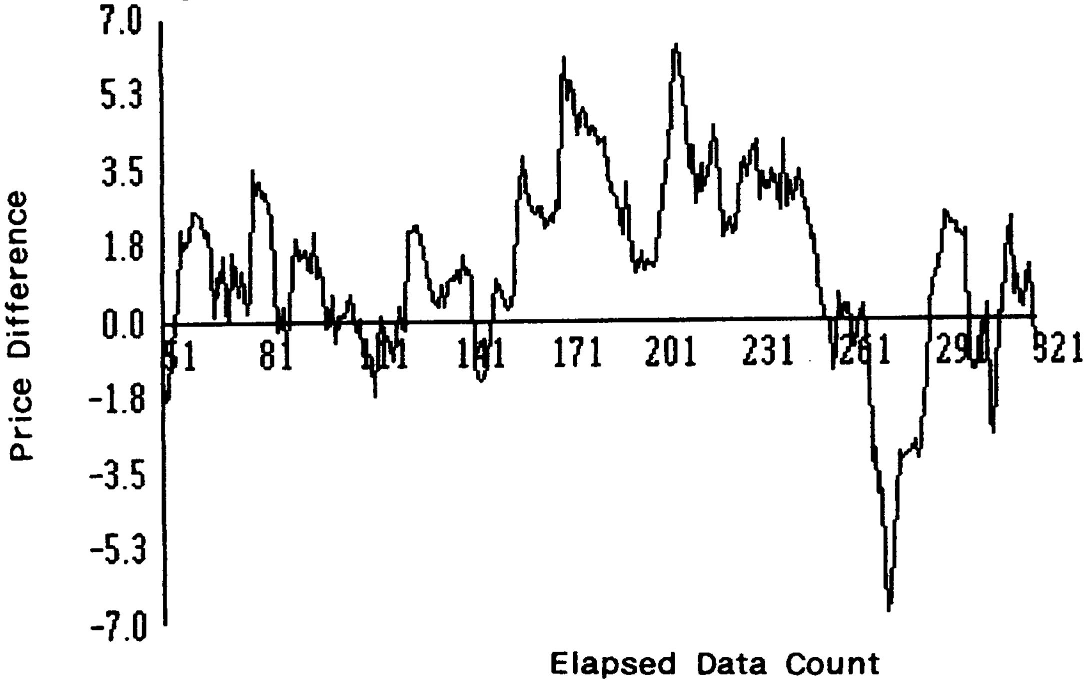
Unsmoothed data. Unsmoothed, original benchmark SPY data. Compare to Figure 5–7 which shows 50-day moving average of data.

18



Smoothed data original benchmark SPY data smoothed with a 50-day moving average. Compare to Figure 4–7 which is the unsmoothed data.

19



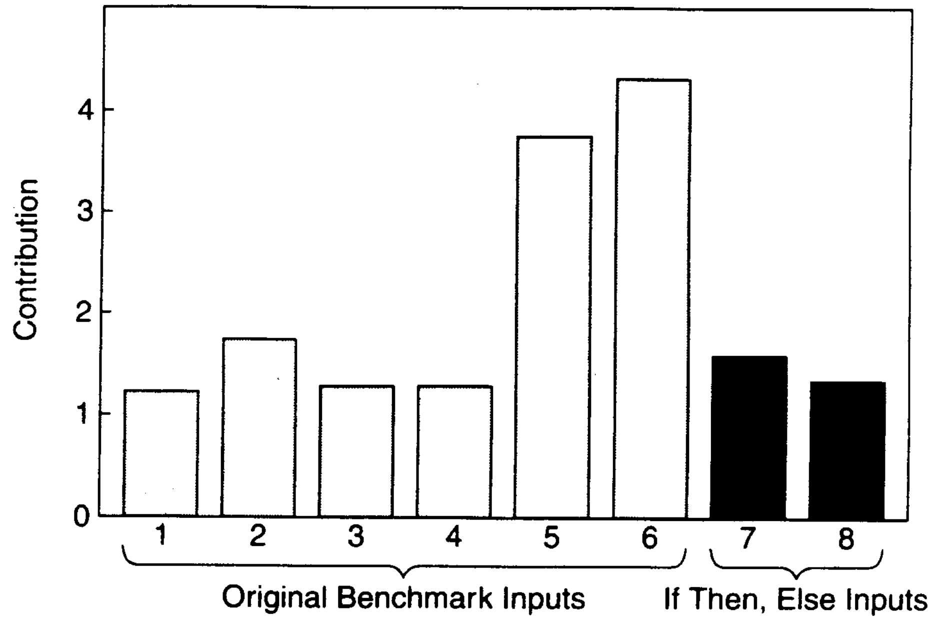
Effect of removing long-term trend. Original SPY data less a 50-day moving average (Figure 5–6 less Figure 5–7). This technique eliminates the longer term trend, and emphasizes the short time swings in the data.

IF the 4-period moving average of SPY *is equal to or greater than* the 18-period moving average of SPY

THEN the column equals +1

ELSE the column equals 0

20



Effect of adding If, Then, Else inputs. Inputs 7 and 8 are the inputs resulting from using the If, Then, Else function. Notice how the addition of inputs 7 and 8 has reduced the contributions of the first four inputs. Compare this to Figure 5–10.

OTHER DATA MANIPULATION

Seasonality

Corn in the winter has low volatility - enter 0 in separate column in spread sheet. In the summer when the volatility is high enter a +1. This allow network to recognize times of high or low volatility.

Days of the week - months of the year, etc.

To determine how many bank tellers are needed each day of the week, set up a column for each day of the week. Put a zero in each column, except for the one representing the day of the week which receives a +1.

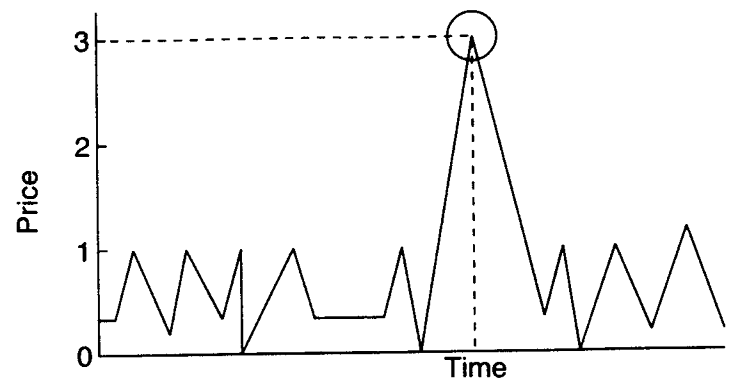
Minimums and Maximums

Most programs have a method of setting minimum and maximums for the data. If the data goes from 460 to 540, those are the values to set not 0 to 1,000.

Outliers

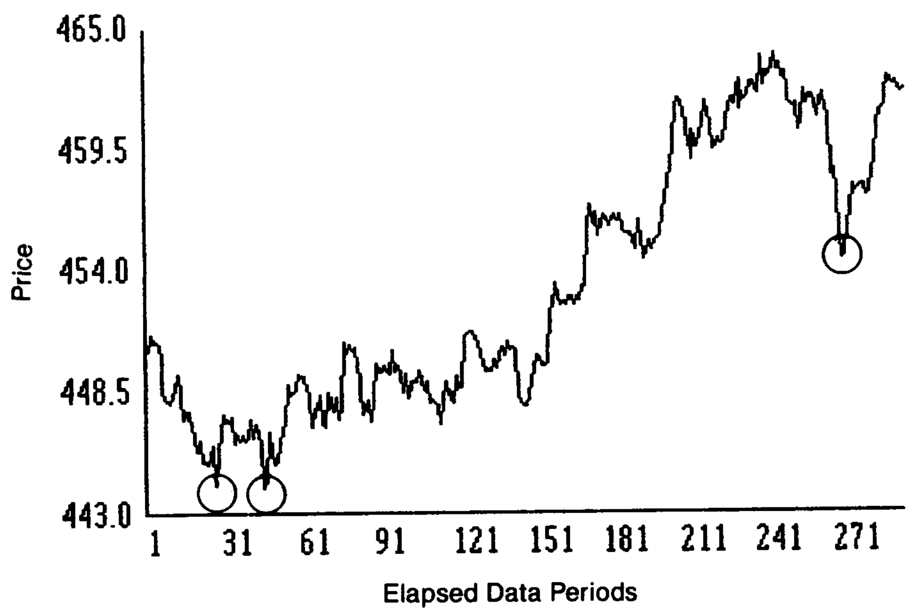
An Outlier is a value which is out of the normal range of data.

21



A data outlier where most of the data is between 0 and +1. Note outlier at time T and value 3. Copyright © 1990 Ward Systems Group.

22



Outliers in the benchmark series. Note possible outliers circled at periods 27, 46, and 265.

The question is, do we want to include the outliers inside the min/max values or to exclude them. Each case is different and can be resolved only by testing.

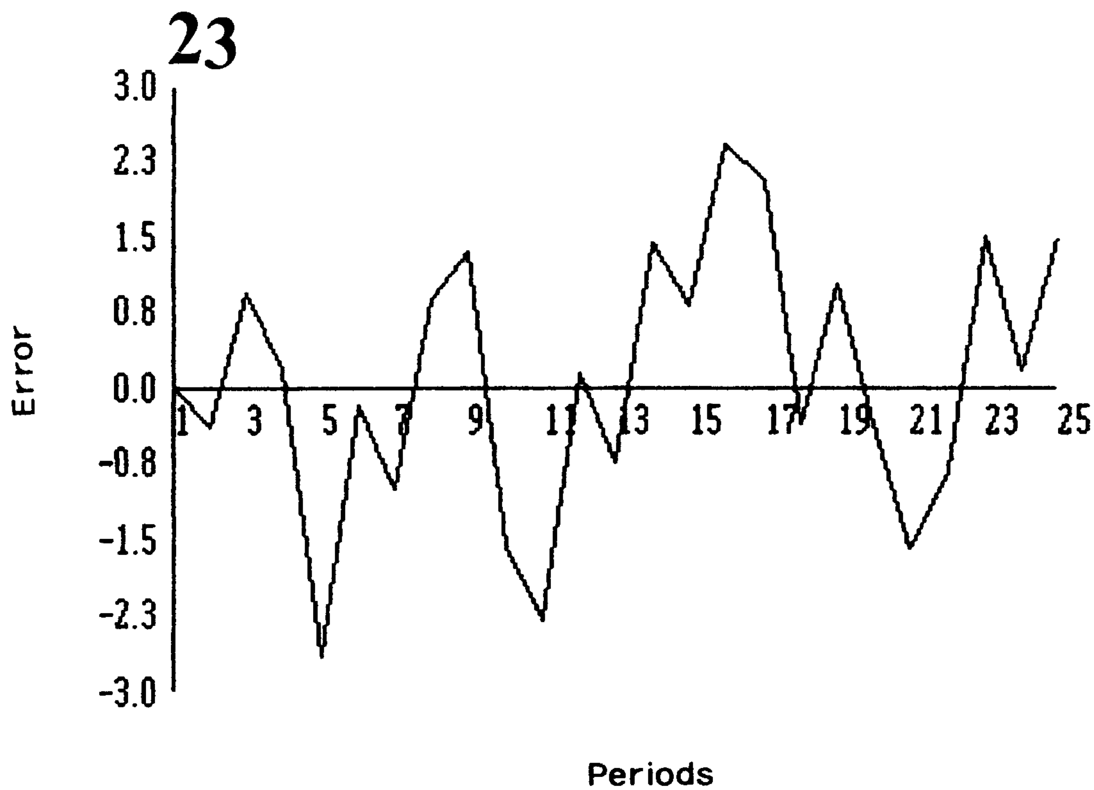


Figure 6–4 Error curve of benchmark system with MIN/MAX set to actual values of the minimums and maximums.

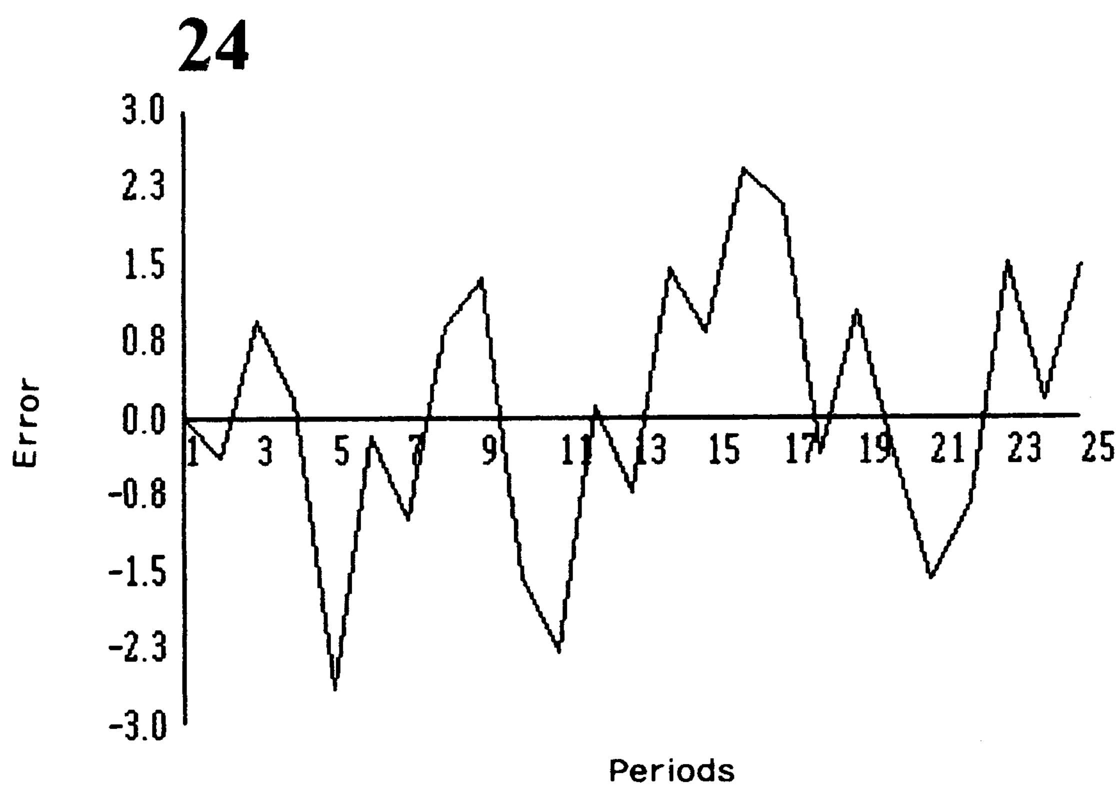


Figure 6–5 MIN/MAX set to value 40 percent greater than actual minimums and maximums.

Extracting test data

Part of the data is normally set aside in advance of training the network for "out of sample testing".

Questions?

How Much? At least 10%

Where from?

Randomly selected?

Every 10th fact?

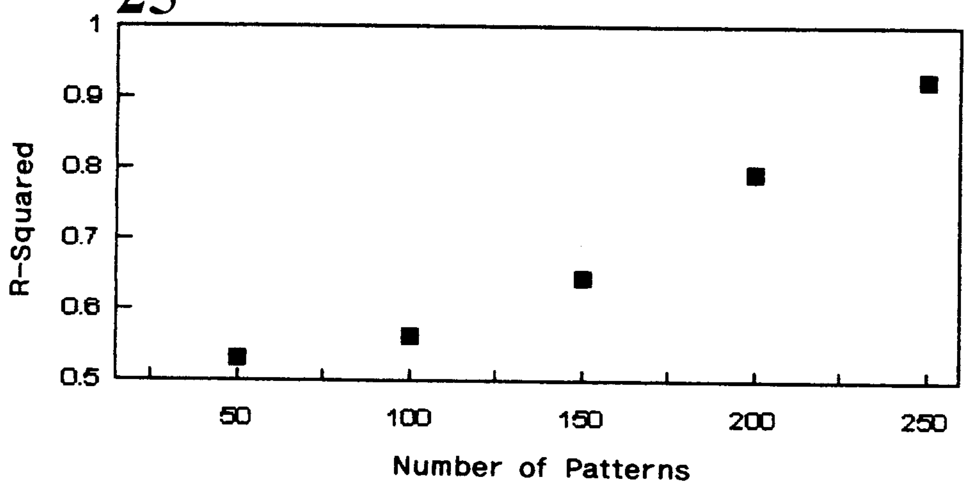
First 10%, Middle 10%, Last 10%?

How much data is enough?

Rule of thumb used to be 10 facts for every input. *NOT TRUE*!

Using the benchmark 6 input network the following accuracy was obtain with various amounts of data.

25



Shows improvement of R-squared as number of patterns is increased. Network had six inputs.

Effects of Non Typical Events

1987 crash

Persian Gulf War

Early 1970's data - option trading

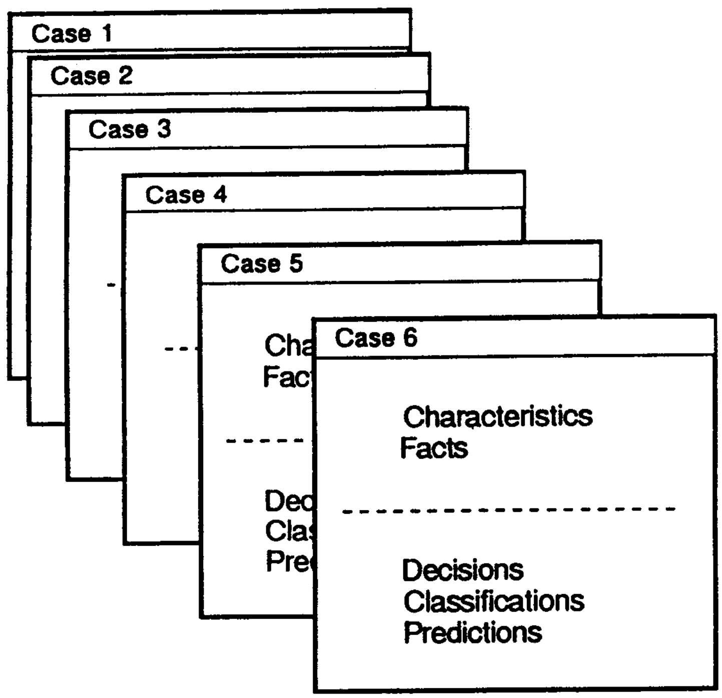
TRAINING

Training is the process of teaching the network what we want it to learn.

Each cycle through all the data is called an epoch. Training usually involves hundreds if not thousands of epochs.

Training is accomplished by varying the strength (weights) of the connections between the neurons, and seeing if a better network is obtained.

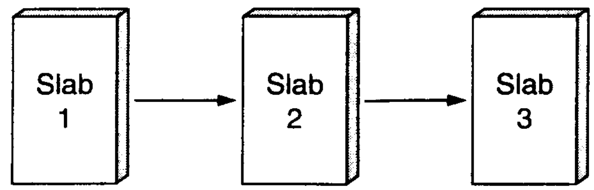
26



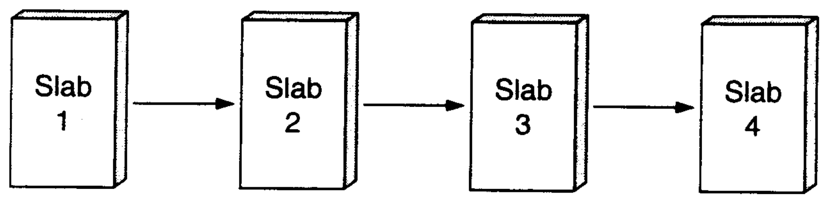
Learning facts—in training, we present one case after another until learning is accomplished. Used with permission of Ward Systems Group.

Effect of training algorithm on accuracy.

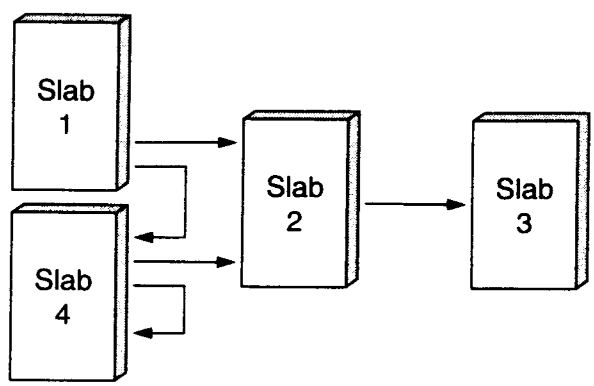
Training Algorithm	Presentation	Minimum Average Error
Momentum (Learn rate 0.05, Momentum 0.5)		
	Random	0.00435
	Rotation	0.00417
Momentum (Learn rate 0.1, Momentum 0.1)		
	Random	0.00434
	Rotation	0.00403
Proprietary #1	Random	0.00434
	Rotation	0.00424
Proprietary #2	Random	NA
	Rotation	0.00456



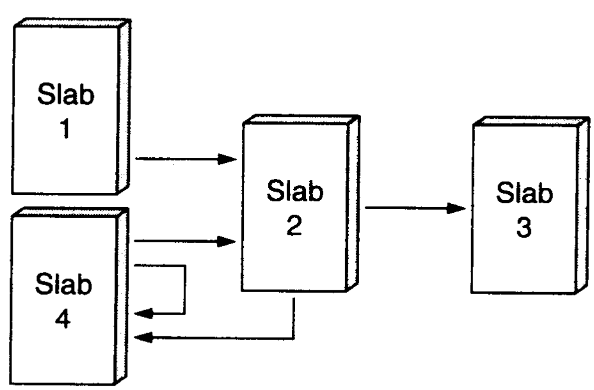
Schematic of 3-layer network. Slab 1 represents the input layer of neurons, slab 2 the hidden layer, and slab 3 the output layer.



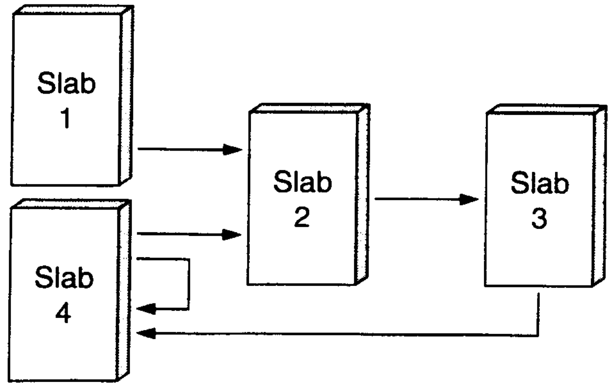
Schematic of a 4-layer network. Slabs 2 and 3 are the hidden layers.



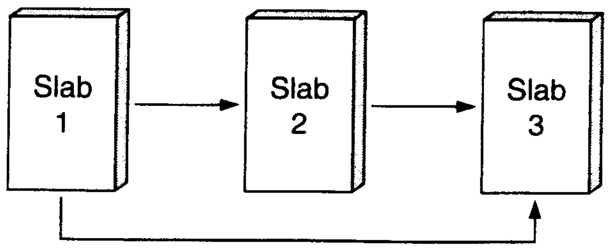
Recurrent network with feedback from the input layer (slab 1) to the recurrent layer (slab 4).



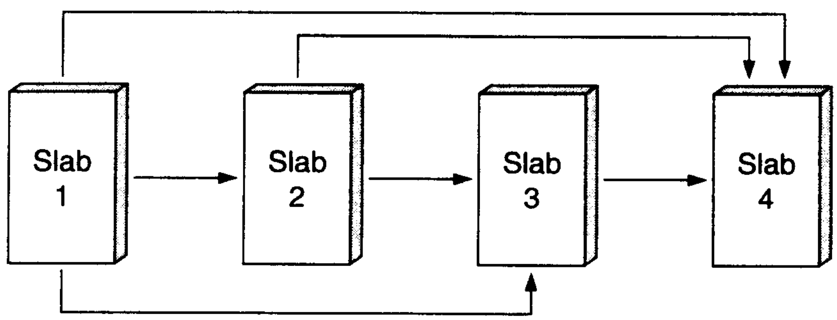
Recurrent network with feedback from hidden layer (slab 2) to the recurrent layer (slab 4).



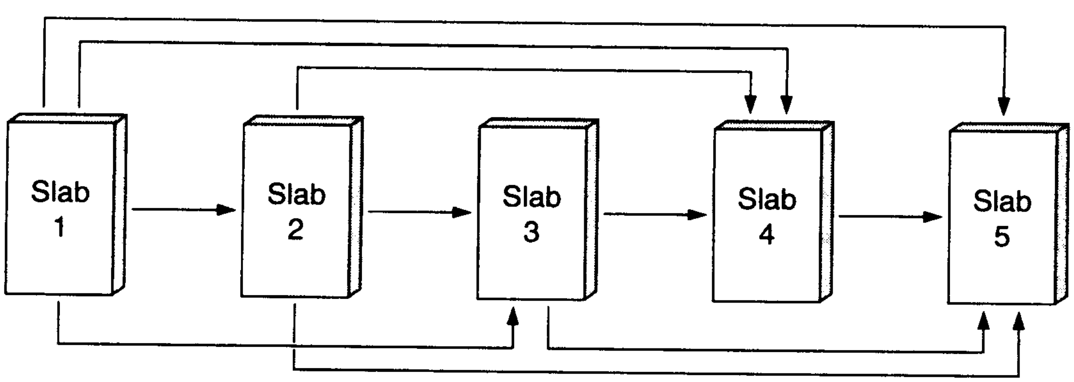
Recurrent network with feedback from output layer (slab 3) to recurrent layer (slab 4).



A 3-layer jump connection network.

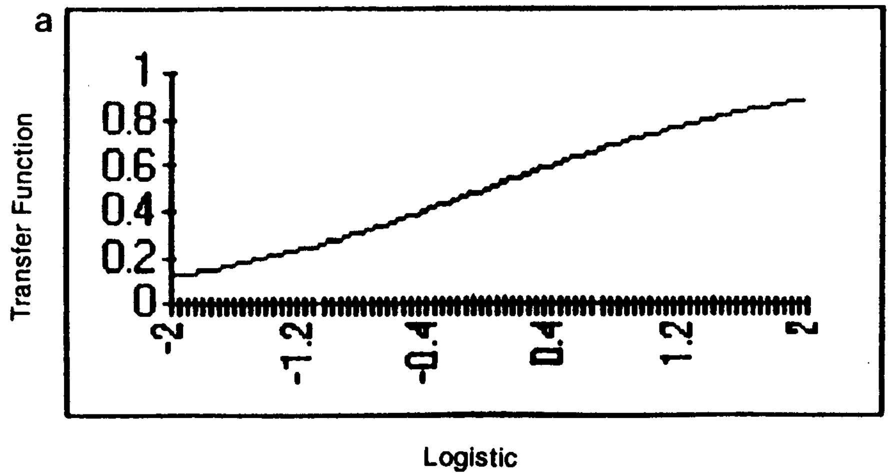


A 4-layer jump connection network.

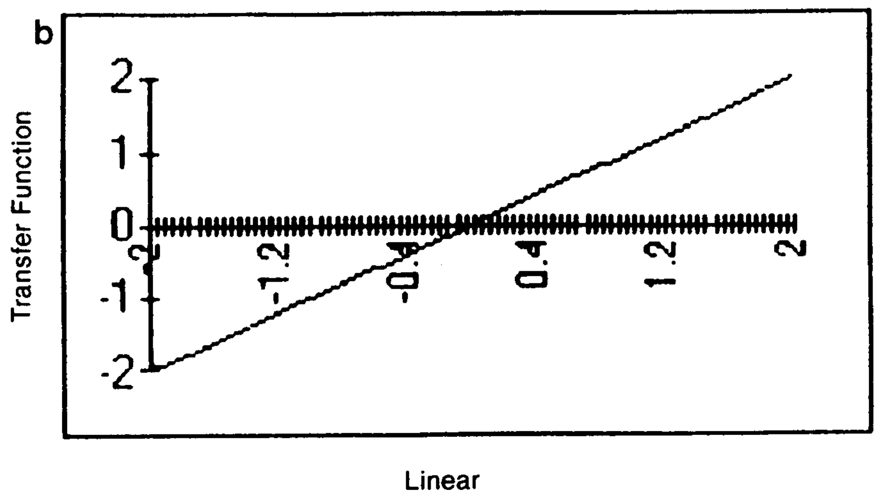


A 5-layer network with jump connections. This network performed very poorly when tested with the benchmark data.

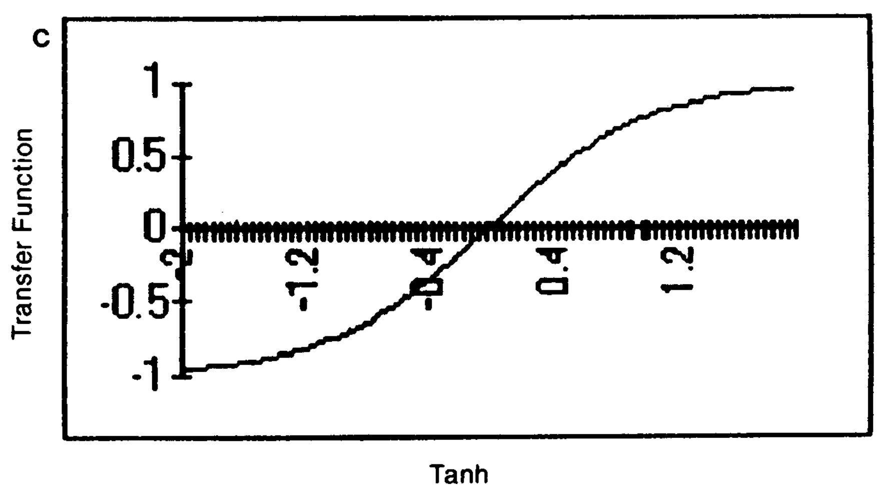
29



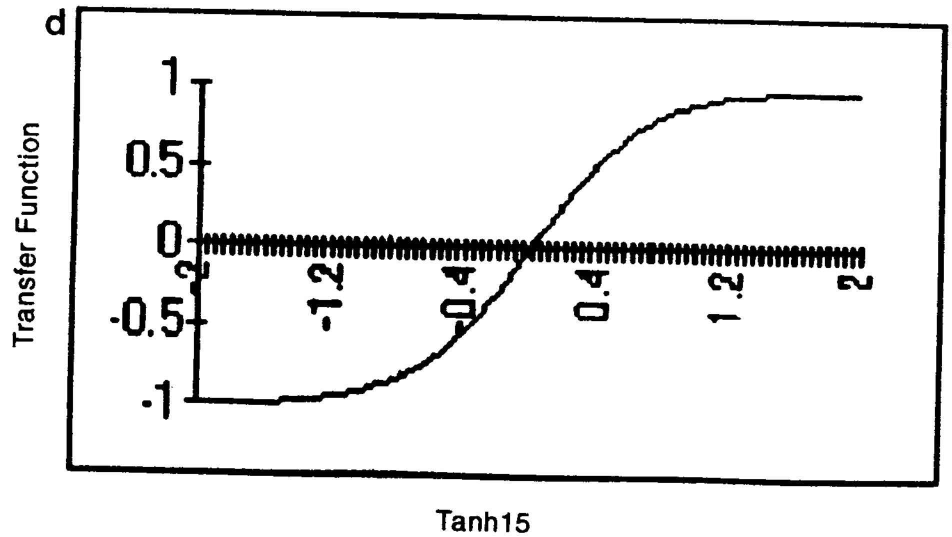
Logistic activation function.



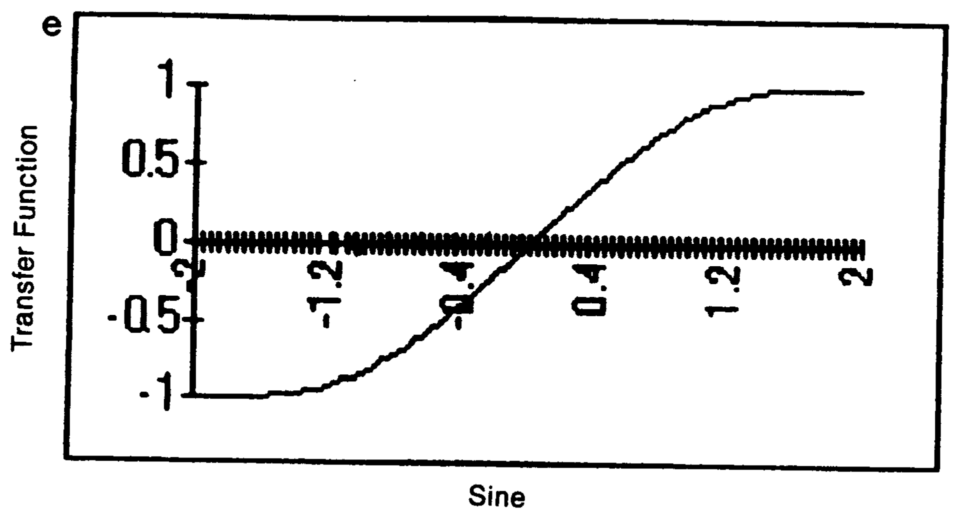
Linear activation function.



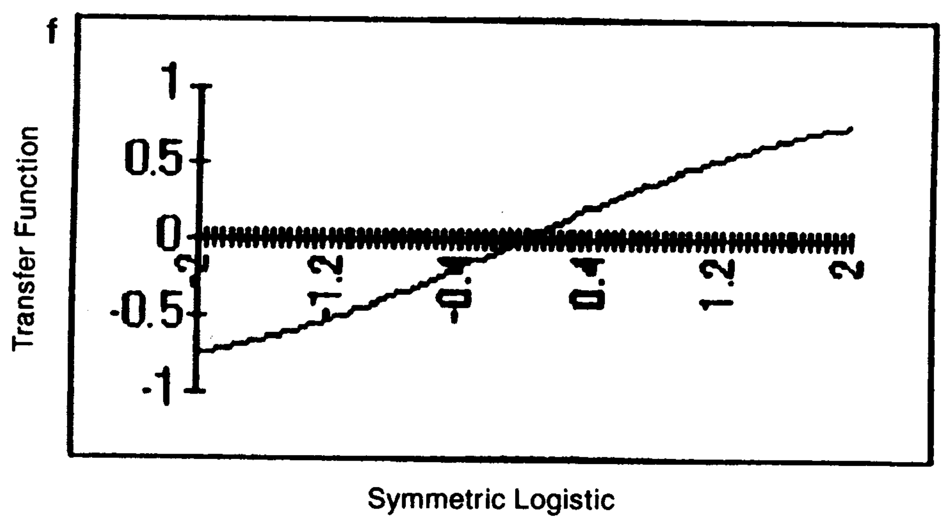
Tanh activation function.



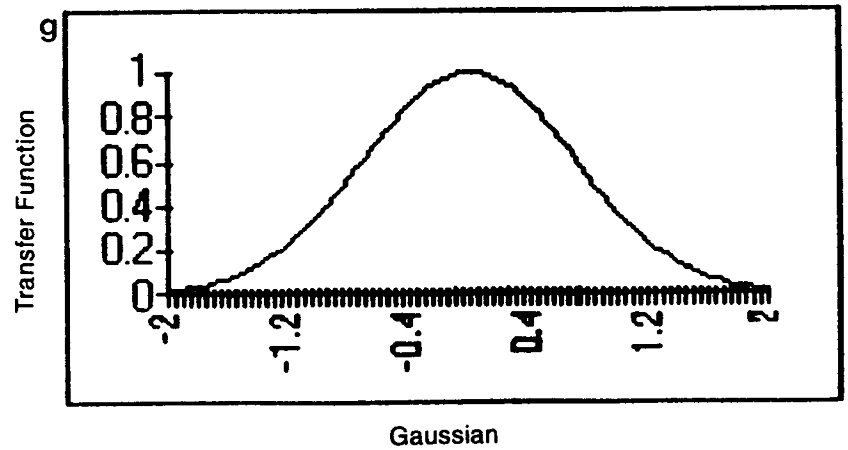
Tanh15 activation function.



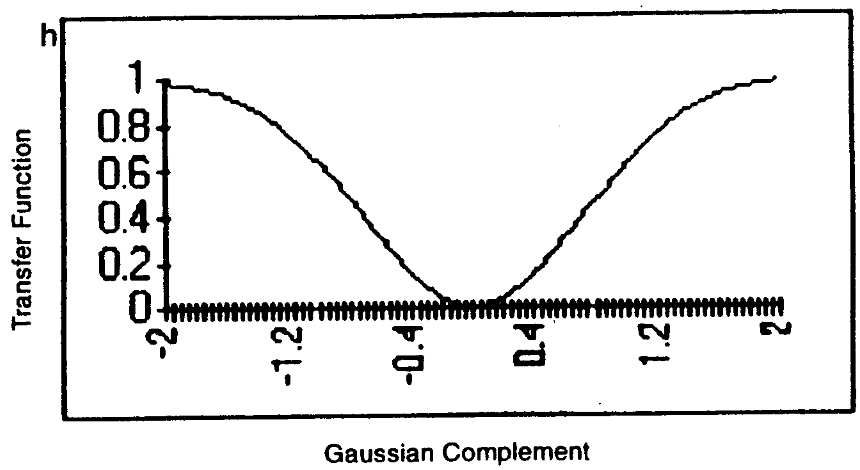
Sine activation function.



Symmetric logistic activation function.



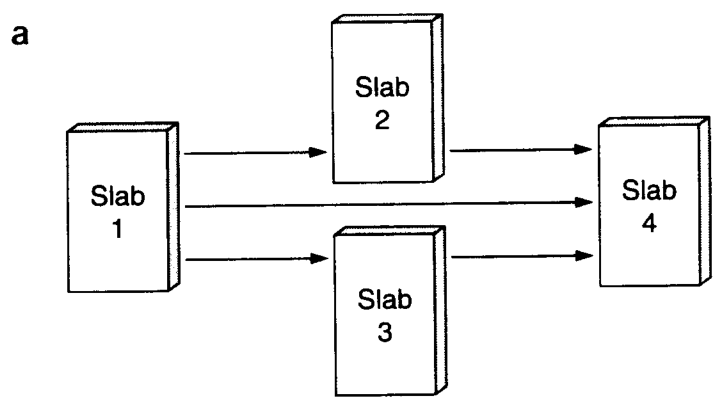
Gaussian activation function.



Gaussian-complement activation function.

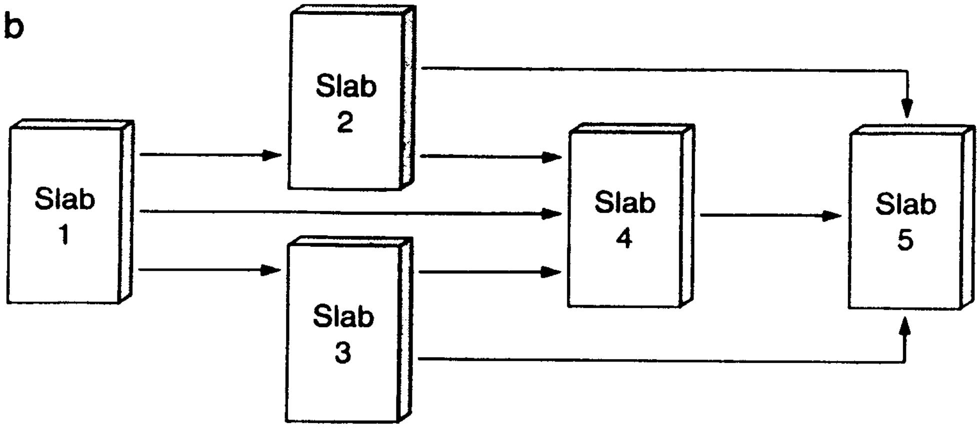
WARD NETWORKS

30



Modified Ward network. Two hidden slabs each with a different activation function and a jump connection, allowing slab 1 to go directly to slab 4.

b

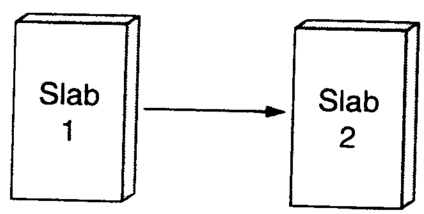


Ward network having three hidden slabs, each with different activation functions.

KOHONEN SELF ORGANIZING MAP NETWORKS

Purpose to separate outputs into categories - e.g. classify real estate data into high-medium-low cost homes.

c



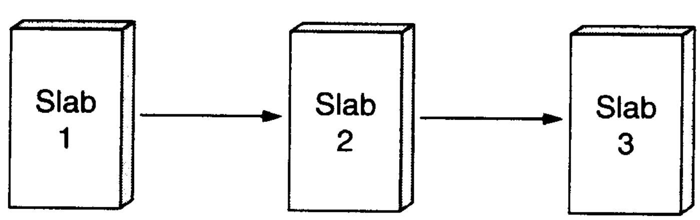
Schematic representation of and unsupervised (Kohonen) neural network.

GENERAL REGRESSION NEURAL NETWORKS (GRNN)

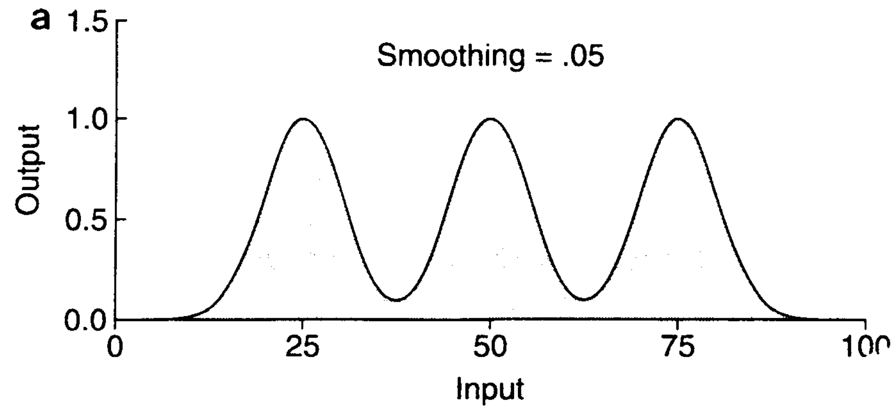
3-layer networks with the number of neurons equal to the number of training facts.

Much *FASTER* and *ACCURATE* than back-propagation networks for financial problems.

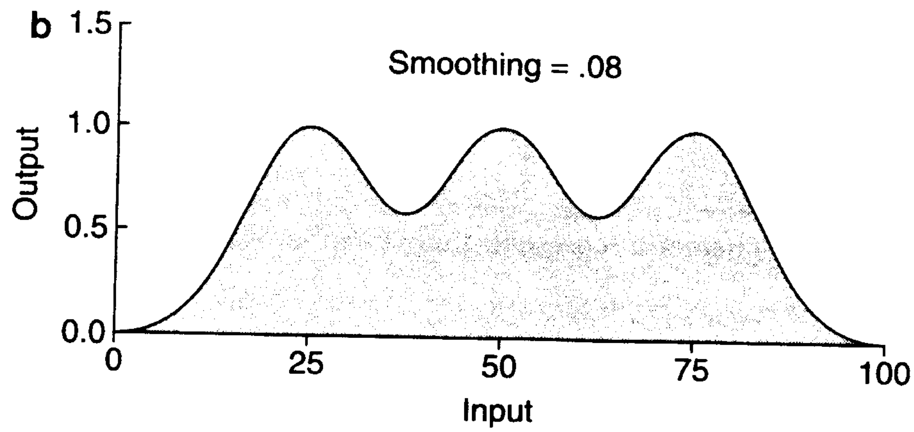
d



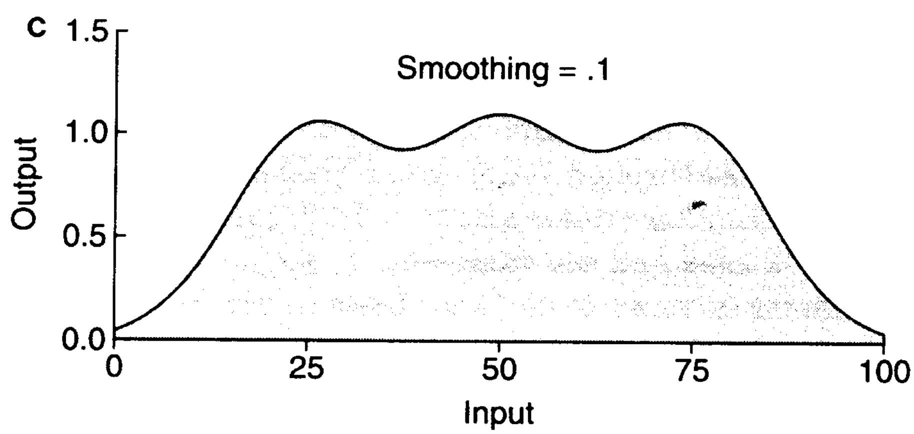
Schematic representation of a General Regression Neural Network (GRNN).



Output verses input with smoothing factor of 0.05.

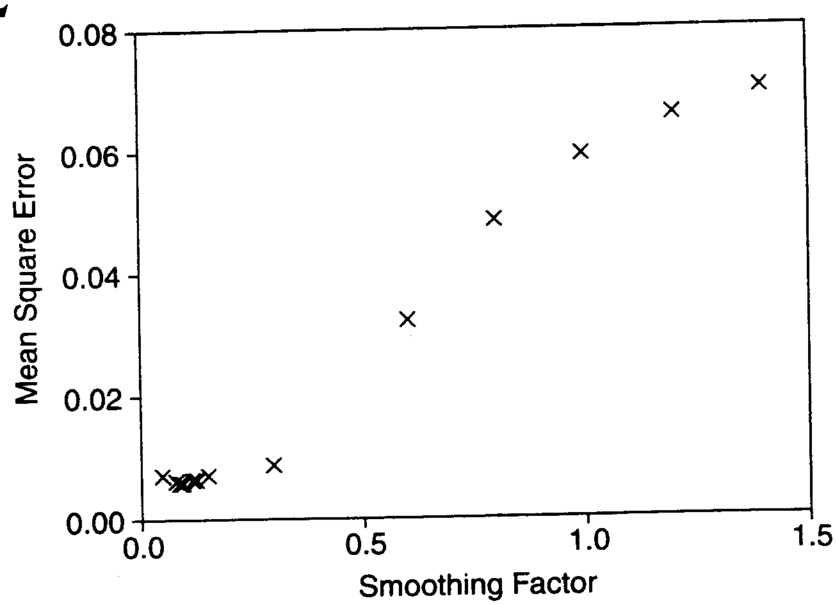


Output verses input with a smoothing factor of 0.08.



Output verses input with a smoothing factor of 0.1.

32

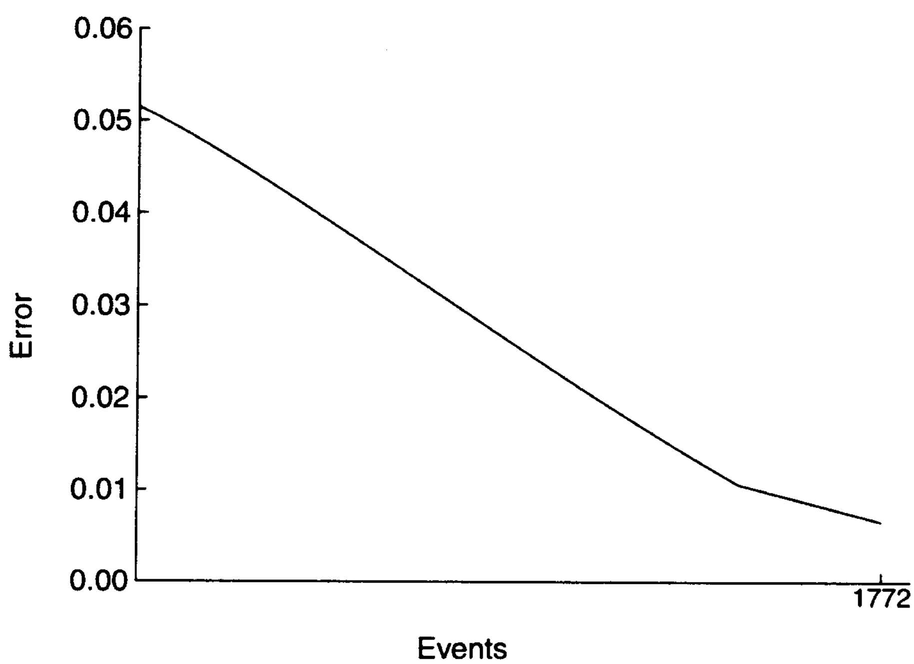


Plot of error of a network verses the smoothing factor.

TRAINING AND USING THE TRAINED NETWORK

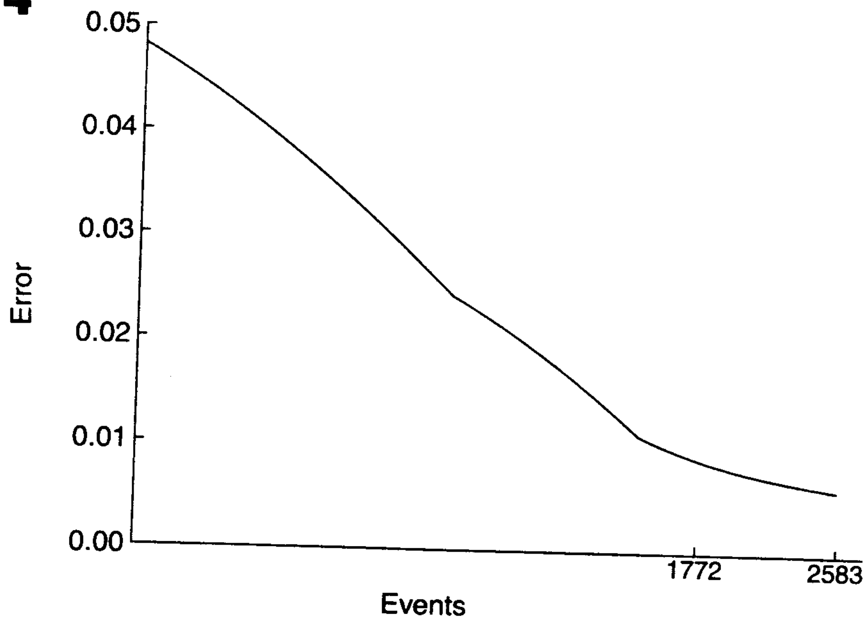
Starting training is usually as easy and clicking the left button on the mouse.

33



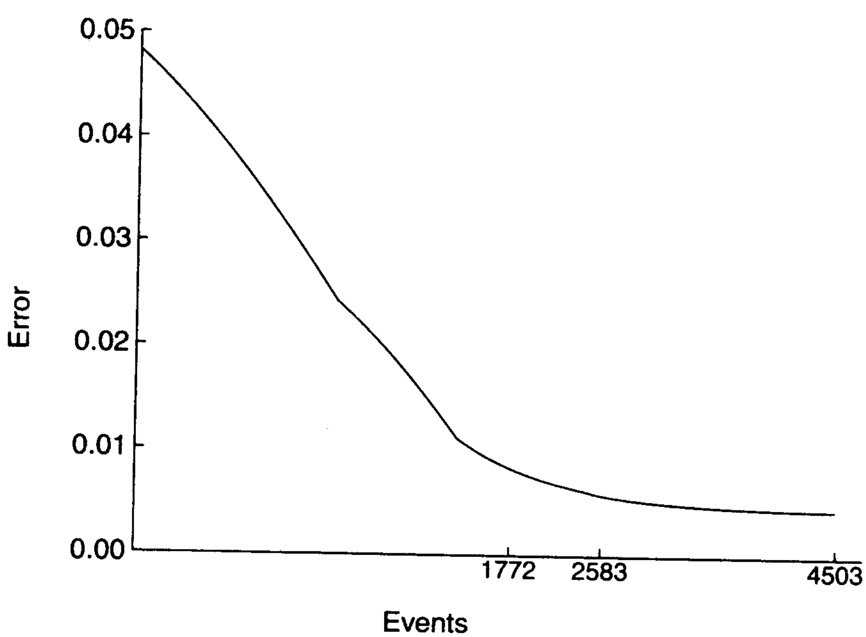
Error verses training time. This graph shows the error of the test set as training progresses. 1,772 learning events have taken place and the rate of training is beginning to slow down.

34



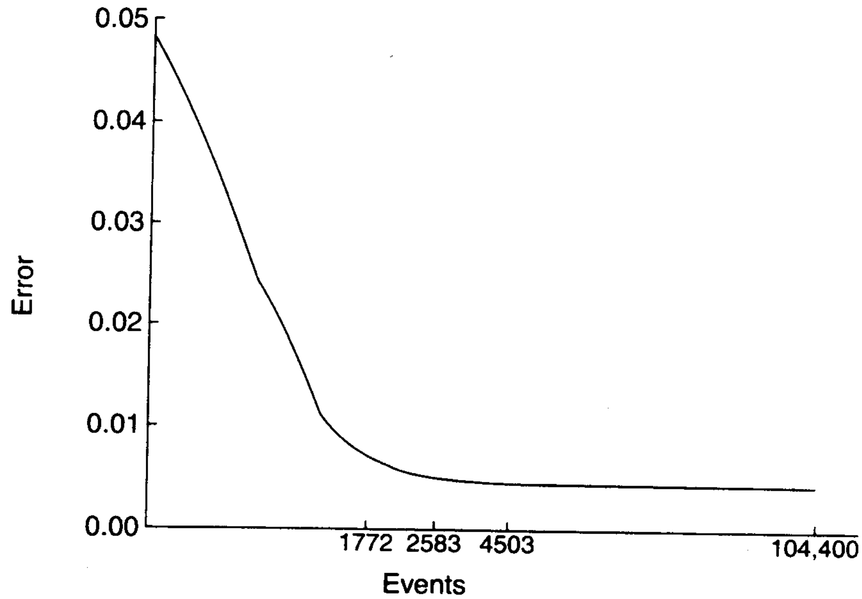
Error verses training time. This graph is the same as that shown in [illegible] except the training has been extended to 2,583 events. Notice that the rate of training is slowed even further.

35



Error verses training time. Here the training has been extended to 4,503 training events. Training rate is further slowed.

36

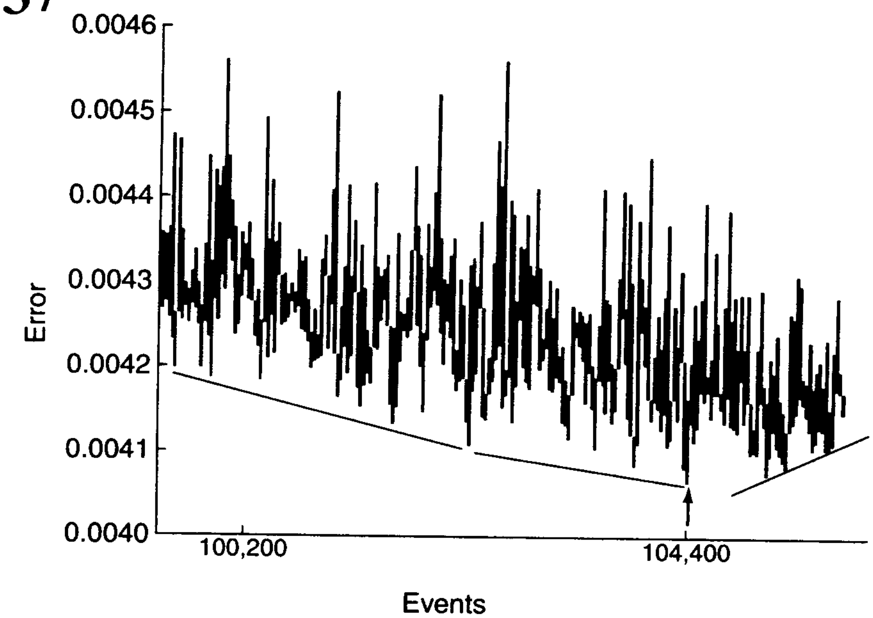


The training has been extended to 104,400 training events. Learning is very slow, but the network appears to be still learning after all this time.

It is very important that the training is stopped at the optimum moment. If the network is over trained, it begins to memorize the data rather than knowing the data. An over trained network does not perform as well on out of sample data.

Most NN packages have a way of periodically testing the network with out of sample data, and if the result is better this test, then the network is saved; if the results are worse than the previous test then training is stopped.

37



This is a magnification of the end of the curve shown [illegible]. Notice that the network is not able to improve on the performance that occurred at the arrow. Normally, if the network cannot improve after 10,000 events, it is considered trained.

	A	B	C
1	Actual(1)	Network(1)	Act-Net(1)
2	450.50	450.05	0.45
3	450.08	449.90	0.18
4	448.64	450.69	-2.05
5	448.23	450.10	-1.87
6	448.04	449.97	-1.93
7	447.94	450.05	-2.11
8	448.41	449.58	-1.17
9	448.66	448.74	-0.08
10	449.22	448.59	0.63
11	448.38	448.37	0.01
12	447.29	448.26	-0.97
13	447.57	448.72	-1.15
14	447.55	448.93	-1.38
15	446.92	449.28	-2.36
16	446.55	447.90	-1.35
17	445.75	447.08	-1.33
18	446.30	447.45	-1.15
19	445.31	447.31	-2.00
20	445.34	446.88	-1.54
21	445.18	446.73	-1.55
22	445.70	446.21	-0.51
23	445.85	447.06	-1.21
24	446.02	446.53	-0.51
25	444.22	446.74	-2.52
26	445.79	446.58	-0.79
27	446.75	446.96	-0.21
28	447.49	447.01	0.48
29	447.08	447.17	-0.09
30	447.08	446.15	0.93
31	447.31	447.13	0.18
32	446.14	447.86	-1.72
33	446.48	448.25	-1.77
34	446.26	447.94	-1.68
35	446.46	447.95	-1.49
36	446.23	448.16	-1.93
37	446.39	446.76	-0.37
38	447.18	447.12	0.06

Output and error of original benchmark network. Column A gives the actual data before training. Column B gives the network's forecast, and column C is the error or the difference between columns A and B. Mean squared error = 2.447, mean error = 1.265%, and time to train = 0:05:42.

39

	A	B	C
1	Actual(1)	Network(1)	Act-Net(1)
2			
3			
4			
5			
20			
21	445.18	445.70	-0.52
22	445.70	445.70	0.00
23	445.85	445.85	0.00
24	446.02	445.99	0.03
25	444.22	444.29	-0.07
26	445.79	445.77	0.02
27	446.75	446.80	-0.05
28	447.49	447.34	0.15
29	447.08	447.19	-0.11
30	447.08	447.08	0.00
31	447.31	446.17	1.14
32	446.14	446.19	-0.05
33	446.48	446.39	0.09
34	446.26	446.35	-0.09
35	446.46	446.40	0.06
36	446.23	446.24	-0.01
37	446.39	446.40	-0.01
38	447.18	447.07	0.11
39	446.44	446.59	-0.15
40	447.00	446.94	0.06
41	446.76	446.65	0.11
42	446.52	446.37	0.15
43	446.20	446.20	-0.00
44	444.11	444.12	-0.01
45	444.51	444.53	-0.02
46	446.58	446.37	0.21
47	445.79	445.86	-0.07
48	445.26	445.48	-0.22

Output and error of modified benchmark network. The inputs have been doubled by adding 20-period moving averages of each of the input, and the network has been trained on a General Regression Neural Network (GRNN). The mean error has been reduced from 1.265 percent to 0.092 percent and the mean squared error from 2.447 to 0.52, while the time to train has been reduced from over 5 minutes to 32 seconds. Mean squared error = 0.052, mean error = 0.092%, and time to train = 0:00:32.

To use the network, new data is normally entered into a file or tacked on the bottom of the spread sheet. Then the network is run.

Recurrent networks require data from the last 20 to 50 periods in order to recognize patterns like double bottoms or head and shoulder tops.

COLINEARITY

Inputs that are similar but not identical are said to be *colinear*, e.g. OEX and S & P-500.

Colinear inputs confuse the network, the network does not know which one to believe and goes back and forth between them. More data is not always better. Use of colinear inputs is a common beginners mistake.

TRAINING TO THE CORRECT ACCURACY LEVEL

Occasionally a set of data will not train to the accuracy level needed by the user. Some options that can be tried are:

1. Rethink the output target. Are you trying to be too accurate?
2. Add or subtract neurons in the hidden layer.
3. Rethink the inputs. Is this the proper data to forecast your objective? Is the preprocessing sufficient to allow the program to train properly?
4. Try a different network architecture such as recurrent or GRNN.
5. Rethink the output target.
6. Try a different learning algorithm.
7. Rethink the inputs.
8. Add or subtract hidden layers.
9. Rethink the problem.
10. Determine which inputs are contributing to the solution and which might be holding training back.
11. Tighten or loosen the min/max limits.
12. Change the forecast goal. For example, forecast the change in S&P rather than the absolute value.
13. In forecasting only one output, multiply that output by −1 and then forecast both outputs, the real one and the one multiplied by −1. Sometimes, when the network has to work harder, it comes up with more accurate answers. If these two outputs do not have very similar absolute values, something is wrong with the network.
14. Rethink the output, inputs, problem, architecture, learning algorithm, minimums and maximums, and preprocessing.

Using the lessons taught earlier in the book and using the same data, a new network was created which far surpasses the original benchmark net work. The network was expanded to 12 inputs by adding the 20-period moving average of each of the original six inputs. Next, rather than using the ever-popular back-propagation network, the 12 inputs were trained in a General Regression Neural Network (GRNN).

Comparison of original and new network inputs.

	Original Benchmark Network	New 12-Input Network
Inputs	ADV, DEC, TICK, TRIN, SPY, YXY	ADV, DEC, TICK, TRIN, SPY, YXY, MA(20)-ADV, MA(20)-DEC, MA(20)-TICK, MA(20)-TRIN, MA(20)-SPY, MA(20)-YXY

Comparision of statistics of back-propagation architecture vs. GRNN architecture.

Network Training Algorithm	Back-Propagation Momentum (0.05, 0.5)	General Regression Neural Network
Training time	5:42	00:32
R-Squared	0.9262	0.9984
Mean squared error	2.447	0.052
Mean error	1.265	0.092