Bioorganic & Medicinal Chemistry Letters, Vol.2, No.3, pp. 213-218, 1992
Printed in Great Britain

0960-894X/92 $3.00 + .00

REGRESSION ANALYSIS FOR QSAR USING NEURAL NETWORKS

David J. Livingstone[1*] and David W. Salt[2]

[1]SmithKline Beecham Research, The Frythe, Welwyn, Herts, AL6 9AR, UK
[2]School of Mathematical Studies, Portsmouth Polytechnic, Portsmouth, Hants, PO1 2EG, UK.

(*Received* 21 *November* 1991)

Abstract: Neural networks have been used to analyse QSAR data giving promising results. However, there is the danger of chance "correlations" and "over-fitting". We have examined a reported analysis and shown that the size of the hidden layer can be reduced giving more efficient training while maintaining predictive performance .

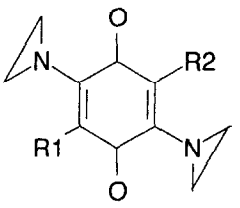
There have been reports over the last few years of the use of neural networks to predict protein structure[1-3]. In some cases these methods have performed better than more conventional techniques and it seems that they might offer advantages in such applications. More recently, studies have been published in the Journal of Medicinal Chemistry in which networks have been used to analyze QSAR data sets performing the equivalent of discriminant[4] and regression analysis[5,6]. These have shown some very promising results, particularly one of the latter reports in which it was demonstrated that regression analysis on three physicochemical properties using the network performed much better than conventional regression on the same data set using two indicator variables[6]. The inclusion of indicator variables in the original treatment was necessary in order to improve the squared correlation coefficient from 0.39 to 0.77. The network analysis yielded a correlation coefficient of 0.82 supported by cross-validation.

Although these results appear very encouraging it should be appreciated that, by analogy with regression, there is the possibility that chance effects may occur as demonstrated by Topliss and Edwards[7]. This may be a particular danger if care is not taken over the choice of the number of units in the hidden layer(s) of a network. The number of connections, and hence adjustable parameters, in a network is determined by the number of units. We have begun a systematic study of the properties of neural networks as applied to QSAR and report here some preliminary results.

We would like to stress that this is not intended as a criticism of any of the earlier reports cited. Indeed, these authors should be congratulated for the imagination to apply such new

techniques since we feel that there is considerable potential in the use of these methods in QSAR. It is important that this is stated at the outset since it would seem that other analytical methods have suffered due to unnecessary adverse comment. Early reports of the use of pattern recognition techniques in QSAR[8,9] were criticized partly on the basis of the choice of parameters employed[10-13]. While the remarks concerning the descriptors may be justified, it appears that these comments have prevented the wider application of pattern recognition methods in QSAR. We believe that this is unfortunate since pattern recognition techniques offer a number of advantages over more conventional methods.

Neural networks were constructed using a commercial software package, ANSIM (Science Applications International Corporation, San Diego, California) running on a 33 MHz 386 based PC clone. The time taken to train a network is clearly problem dependent but also depends on its architecture and the values chosen for the learning rates, parameters which affect the speed of convergence. In practice it was found that network training for typical QSAR data sets (< 50 compounds, < 10 descriptors) could be achieved in a few hours. The data set examined in this study involved the anticarcinogenic activity of a series of substituted carboquinones[14] (**1**). The compounds were described by six physicochemical parameters, $MR_{1,2}$, $\pi_{1,2}$, π_2, MR_1, F and R , as shown in Table 1, where the subscript refers to the position of substitution. The biological data reported by Nakao and co-workers[14] included the minimum effective dose (MED) on a chronic treatment schedule. This is the dose to give a 40% increase in lifespan compared with controls and is shown in Table 1 as the logarithm of a reciprocal concentration. MED values fitted by neural network models are also shown in the Table.



1

The original report of the use of a network to analyze these data involved a network of 7 input units, one hidden layer of 12 units and a single output unit[5]. Training was carried out so as to reproduce values of a dose to give a standard effect at the output unit and it was suggested that this network gave better results than conventional multiple linear regression. We found that a re-construction of this network using ANSIM gave comparable results with training being achieved after 66,000 iterations (approx 1.8 hours). Although at this point the ratio of compounds in the training set to input parameters was acceptable (6.2) in terms of commonly stated guidelines for the avoidance of chance effects, the use of 12 units in the hidden layer gave cause for concern that the "fit" may have too few degrees of freedom.

Table 1. Physicochemical and Biological Data for 37 Carboquinones[a]

No.[a]	$MR_{1,2}$	$\pi_{1,2}$	π_2	MR_1	F	R	MED[b]	MED[c]	MED[d]
2	5.08	3.92	1.96	2.54	0.16	-0.16	4.33	4.26	4.35
3	4.5	3.66	3.16	0.57	-0.081	-0.26	4.47	4.63	4.47
4	4.86	5.0	2.5	2.43	-0.08	-0.26	4.63	4.33	4.65
5	3.0	2.6	1.3	1.5	-0.08	-0.26	4.77	5.10	4.83
6	3.57	2.51	2.01	0.57	-0.12	-0.14	4.85	5.11	4.86
7	3.0	3.0	1.5	1.5	-0.08	-0.26	4.92	4.98	4.86
8	3.79	2.16	1.66	0.57	-0.04	-0.13	5.15	5.13	5.16
9	6.14	0.72	0.36	3.07	-0.08	-0.26	5.16	5.19	5.15
10	2.06	2.0	1.0	1.03	-0.08	-0.26	5.46	5.46	5.47
11	2.28	1.03	0.53	0.57	-0.08	-0.26	5.57	5.98	5.61
12	1.58	-0.04	-0.02	0.79	0.52	-1.02	5.59	5.71	5.59
13	2.07	1.8	1.3	0.57	-0.08	-0.26	5.60	5.54	5.55
14	4.24	0.98	-0.52	1.5	-0.04	-0.13	5.63	5.80	5.62
16	1.14	1.0	0.5	0.57	-0.08	-0.26	5.66	6.07	5.67
17	1.6	1.3	1.3	0.1	-0.04	-0.13	5.68	5.66	5.68
18	2.75	1.53	1.03	0.57	-0.04	-0.13	5.68	5.50	5.68
19	3.56	1.45	-0.05	1.5	-0.08	-0.26	5.68	5.72	5.70
20	3.42	1.03	0.53	1.71	-0.08	-0.26	5.69	5.56	5.69
21	4.23	0.98	-0.02	1.03	-0.04	-0.13	5.76	5.76	5.75
22	2.78	1.23	0.73	0.57	-0.08	-0.26	5.78	5.82	5.73
23	1.96	2.0	1.5	0.57	-0.08	-0.26	5.82	5.43	5.86
24	1.6	1.5	1.0	0.57	-0.08	-0.26	5.86	5.73	5.84
25	4.45	0.01	-0.49	0.57	-0.04	-0.13	6.03	6.32	6.01
26	3.09	0.75	0.25	0.57	-0.08	-0.26	6.14	6.10	6.24
27	3.77	0.48	-0.52	1.03	-0.04	-0.13	6.16	6.10	6.22
28	3.55	1.25	0.75	0.57	-0.08	-0.26	6.18	5.77	6.22
29	3.77	0.48	-0.02	0.57	-0.04	-0.13	6.18	6.06	6.16
30	3.09	0.95	0.45	0.57	-0.08	-0.26	6.18	5.98	6.19
31	2.63	0.45	-0.05	0.57	-0.08	-0.26	6.21	6.33	6.32
32	3.09	0.95	-0.05	1.03	-0.08	-0.26	6.25	6.02	6.20
33	1.78	0.34	-0.26	0.57	-0.08	-0.26	6.39	6.47	6.38
34	3.09	0.75	0.25	0.57	-0.08	-0.26	6.41	6.10	6.24
35	3.31	-0.02	-0.52	0.57	-0.04	-0.13	6.41	6.42	6.44
36	1.66	0.18	0.18	0.1	0.1	-0.92	6.45	6.62	6.46
37	2.42	-0.32	-0.16	1.21	-0.08	-0.26	6.54	6.38	6.53
38	2.13	0.68	0.18	0.57	0.06	-1.05	6.77	6.47	6.75
39	2.47	-0.13	-0.63	0.57	-0.04	-0.13	6.90	6.56	6.77

[a]Data and compound numbers from reference 5.

[b]Observed MED (log 1/C) values from reference 5.

[c]Fitted MED (log 1/C) values from a 7:12:1 neural network from reference 5.

[d]Fitted MED (log 1/C) values from a 7:5:1 network as shown in table 2.

There was also a perceived risk that, with so many units in the hidden layer, the network would "over fit" the data, thus making it difficult to generalize the results to new data sets. Networks with large numbers of connection weights relative to the number of data points tend to memorise the data during training rather than 'identifying' or 'learning' any rules associating the independent variables with the dependent variable.

The number of units required in the hidden layer will vary according to the application and undoubtedly relates to the structure of the data. Consequently, networks with differing numbers of hidden units will need to be investigated before a satisfactory architecture is achieved. It has been recommended that a reasonable number to start with can be obtained by taking the square root of the sum of the number of input and output units, and then adding a few[15]. We have constructed networks with smaller numbers of units in the hidden layer and have found that the network was still able to train satisfactorily. The largest number of units employed was 12 and the smallest number was 4. Results of the network training are shown in Table 2 where the mean square error (MSE) is calculated as

$$MSE = \frac{\Sigma (Y_{observed} - Y_{predicted})^2}{\text{(No. of compounds x No. of output units)}}$$

and the residual variance (RV) is calculated as Aoyama *et al*[5] and is given by:

$$RV = \frac{\Sigma (Y_{observed} - Y_{predicted})^2}{\text{(No. of compounds - 1)}}$$

Table 2 Fitting errors with different numbers of units in the hidden layer.

Units	MSE ($x10^3$)	RV ($x10^3$)	Units	MSE	RV
4	6.09	6.25	9	2.64	2.71
5	2.64	2.71	10	2.64	2.71
6	2.63	2.71	11	2.64	2.72
7	2.63	2.71	12	2.64	2.71
8	2.63	2.71	12[a]	42.9	44.1

[a] results from Aoyama *et al*[5].

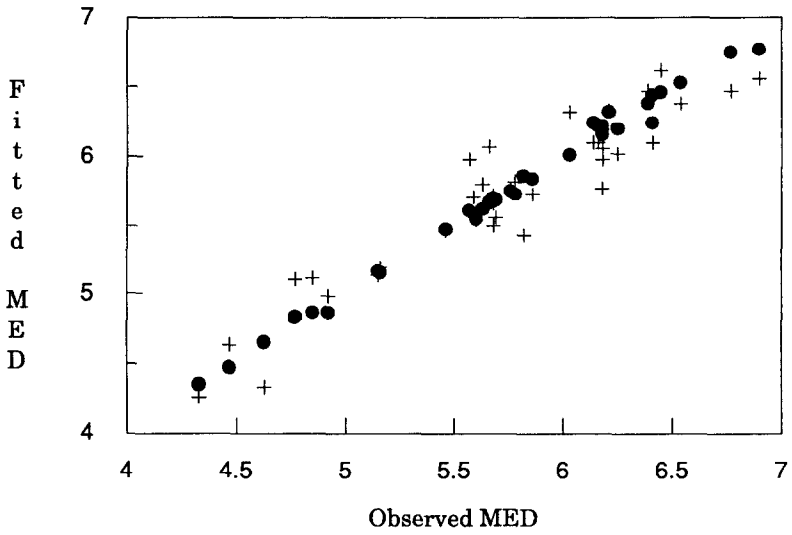


Figure 1. Network Fitting. MED values from this work (•) and from reference 5 (+) *vs* Observed MED

It can be seen from this table that the ANSIM networks give almost identical results in terms of MSE and RV for all hidden layer sizes between 5 and 12. The performance of these networks begins to decline as the hidden layer size is reduced to 4 but even this network gives better fits than those reported by Aoyama *et al*[5]. This is shown in Figure 1 where the two sets of fitted MED values from Table 1 are plotted against the observed MED values. The improved performance of the ANSIM network over the network of Aoyama and co-workers may result from a different number of training iterations and software implementation of the back propagation algorithm. However, we believe that the lack of performance of the network employed by these authors is a direct consequence of the functionality of their single output unit. Although non-linear transfer functions (modified logistic) are used for the hidden units, the output unit merely sums the input to it from the hidden units. In contrast, the summed input into the output unit of the ANSIM networks is processed through the same type of non-linear transfer function as that used on the hidden units. The results shown in Table 2 and the Figure demonstrate that the use of a non-linear transfer function on the output unit provides a far more efficient network in terms of mean square error and, more importantly, in terms of the number of weight parameters that need to be estimated from the data.

Aoyama and colleagues have also examined the effect of non-linear relationships by adding the squares of the descriptors in table 1 to the data set[5]. The network used to analyse these data required an extra six units in the input layer and employed an increased size hidden layer of twenty six units. It was suggested that this second network provides better fitting but this is not borne out by the published results where the mean square error

is identical for both networks[5]. Experiments have been carried out in our own laboratory in which both real and artificially constructed data sets, of known non-linear structural form, have been processed through similarly constructed ANSIM networks. These have been successfully trained and have provided acceptable fitting capability with only the linear terms being provided as input. It is clear from this work and others[6] that the intrinsic non-linear capability of networks with non-linear processing units automatically handles non-linear associations in the data.

In conclusion, it has been shown that neural networks with quite a small number of units in the hidden layer can be used to carry out regression analysis on QSAR data sets. The use of such small hidden layers results in improvements in the efficiency of network training without any apparent degradation of performance in fitting. We believe that the use of networks with such an architecture will help to ensure that chance effects are avoided and that the resultant equivalent of regression models will not simply be trivial solutions due to "over-fitting". The computational overhead involved in performing regression analysis with a network should not be regarded as an insurmountable problem. These calculations were carried out on a relatively slow computer and yet were achieved in a reasonable time scale; much more powerful computers are now routinely available. The advantages of carrying out regression in this way are not clear as yet but it would seem that there is considerable potential in the approach.

References

1 Qian, N; Sejnowski, T.J. *J. Mol. Biol.* **1988**, *202*, 865.
2 McGregor, M.J.; Flores, T.P.; Sternberg, M.J.E. *Protein Eng.* **1989**, *2*, 521.
3 Muskal, S.M.; Holbrook, S.R.; Kim, S.-H. *Protein Eng.* **1990**, *3*, 667.
4 Aoyama, T.; Suzuki, Y.; Ichikawa, H. *J. Med. Chem.* **1990**, *33*,905.
5 Aoyama, T.; Suzuki, Y.; Ichikawa, H. *J. Med. Chem.* **1990**, *33*, 2583.
6 Andrea, T.A.; Kalayeh, H. *J. Med. Chem.* **1991**, *34*, 2824.
7 Topliss, J.G.; Edwards, R.P. *J. Med. Chem.* **1979**, *22*, 1238.
8 Ting, Kai-Li H.; Lee, R.C.T.; Milne, G.W.A.; Shapiro, M.; Guarino, A.M. *Science* **1973**, *180*, 417.
9 Kowalski, B.R.; Bender, C.F. *J. Am. Chem. Soc.* **1974**, *96*, 916.
10 Clerc, J.T.; Naegeli, P.; Seibl, J. *Chimia* **1973**, *27*, 639.
11 Perrin, C.L. *Science* **1974**, *186*, 551.
12 Unger, S.H. *Cancer Chemother. Rep. part 2* **1974**, *4*, 45.
13 Mathews, R.J. *J. Am. Chem. Soc.* **1975**, *97*, 935.
14 Nakoa, H.; Arakawa, M.; Nakamura, T.; Fukushima, M. *Pharm. Bull.* **1972**, *20*, 1968.
15 Eberhart, R.C.; Dobbins, R.W. *Neural Network PC Tools, A practical Guide*; Academic Press: San Diego, 1990; p 42.