

Chemical Design Automation News

Volume 8
Number 1
January 1993

Index

Allen and Kennard
outline new develop-
ments in the Cam-
bridge Structural
Database

1

New Briefs

2

Previews

3

New Products

4

People

8

Law Briefs

from Elizabeth F.
Enayati

10

Molecular Model- ling and Protein Engineering Series

Part 4: Snow
discusses protein
homology modelling

17

Turner, Weiner, Lupi,
Galloni, and Singh
describe new
molecular dynamics
algorithm, Part 2

16

Library Notes

from Vassilios
Galiatsatos

22

Lab Notes

from The European
Molecular Biology
Laboratory

24

Meeting Roundup

Garavelli and Abola
provide update on
protein database
developments; Hempel
highlights 1992
Albany Conference

28

Meetings

38

Chemical Design Automation News

FOUNDERS

Kureha Chemical
Polygen

EDITORIAL POLICY

It is the express editorial policy of *Chemical Design Automation News* to publish objective information on matters of technical interest relating to the use of computer automation techniques in chemical and engineered materials research. In accordance with this policy, we welcome participation from all individuals and institutions involved in the field.

Published monthly. ISSN 0886-6716.

Editor: Barbara F. Graham
Chief Science Editor: Frank A. Momany
Contributing Editors: Vassilios Galatsatos,
Kerwin D. Dobbs, Salvatore Profeta
Law Editor: Elizabeth F. Enayati
Book Editor: John J. Staszowski
Design and Production: James A. Wojno
Business Manager: Patricia J. Bolts

CDA News - USA:

16 New England Executive Park, Burlington, MA
01803-3297. Tel. (617) 229-9800, Fax (617) 229-9899,
e-mail: bfg@msl.com

Asia/Pacific Editor - Industrial:

Hiroshi Chuman, Intelligent Systems Department for
Research, Kureha Chemical Industry Co. Ltd.,
3-25-1, Hyakunin-cho, Shinjuku-ku, Tokyo 160,
Japan. Tel. 81-3-3362-7329, Fax 81-3-3362-7428

Asia/Pacific Editor - Academic:

Nobuhito Gô, Department of Chemistry, Faculty of
Science, Kyoto University, Kitashirakawa, Sakyo-ku,
606 Kyoto, Japan. Tel. 81-75-712-1497,
Fax 81-75-711-6083

European Editors - Industrial:

Frank Barney, SmithKline Beecham, Beecham
Pharmaceuticals Research Division, Medicinal
Research Centre, Coldharbour Road, The Pinnacles,
Harlow, Essex CM19 5AD UK. Tel. 44-279-622-000,
Fax 44-279-622-230.
Gerhard Klebe, BASF AG, ZHV/D-A30,
6700 Ludwigshafen, Germany
Tel. 49-621-604-1966, Fax 49-621-602-0440

European Editor - Academic:

Loes Kroon-Batenburg, Department of Crystallo-
graphy and Structural Chemistry, Bijvoet Center
for Biomolecular Research, Padualaan 8, 3584 CH
Utrecht, The Netherlands. Tel. 31-30-532-865 or 533-
601, Fax 31-30-533-940

SUBSCRIPTION RATES (12 issues, 1 year):

\$300 Corporate (USA, Canada, and Mexico)
\$345 Corporate (Overseas)
\$175 Non-profit (Worldwide)
\$55 Teaching Institution (Worldwide)
\$20 Student (Worldwide) Please enclose
photocopy of current student ID

© 1993 Molecular Simulations Incorporated

No part of this publication may be reproduced in any
form without the written permission of CDA News.

Chemical Design Automation News makes
reasonable efforts to assure the accuracy of infor-
mation reported herein. However, publication of
CDA News does not include any warranty of the
accuracy of such reported information.

CORPORATE SPONSORS

Autodesk	IBM
Biostructure	Molecular Simulations
Convex Computer	Tripos Associates
Cray Research	

branches and the interaction parameters. Analytical expressions are obtained for the mean-squared end-to-end distance of one and two different branches, the sizes of homopolymers and the star itself. The distance between the centers of mass of two different homopolymers is also calculated.

*Vassilios Gallatsatos, Ph.D., Institute of Polymer Science,
The University of Akron, Akron, OH 44325-3909.* ■

LAB NOTES

Neural Networks in Chemistry

By Burkhard Rost, Ph.D. and Gerrit Vriend, Ph.D.

The attempts to understand the functioning of the brain and to improve computers have profited from one another since the early days of electronic calculating. Over the last decade the application of artificial neural networks—implemented on computers—has become popular for various pattern recognition tasks. The basic procedure is that patterns are presented to a network that learns to extract intrinsic features and to group the patterns into classes. The networks not only perform arbitrarily complicated distinction tasks, but are able, as well, to generalize, *i.e.*, to perform the classification for new patterns. This means that a network, for example, for zip code recognition not only learns to distinguish between the hand-written zip codes on those envelopes presented to it for learning, but it learns as well to distinguish the codes for any future incoming envelope. This can be achieved, because the network learns to extract certain features from the letters and learns to relate these features to the required decision like “this is 6900 for Heidelberg” or not. Typical applications are the recognition of faces, speech, handwriting, QSAR analysis, particle detection in high energy physics, the prediction of developments in stock exchange markets, and the prediction of protein and gene structure. (For a survey of the properties of neural networks see for example the books of Hertz⁴ or Müller,⁶ or the article by Cowan.⁵)

The expression neural network stems from the description of the brain as an organism containing neurons which exchange information by their connections, the synapses (see Figure 1 (page 25), or for example, Amit,¹). It has been observed that learning somehow results in changes of these connections. The artificial networks translate neurons into simple units that have input and output. They are connected by the junctions J (similar to the synaptic connections). ▶

LAB NOTES

Figure 2 shows the simplest neural network called perceptron. The units on the left hand receive an external input. Typically, this input is a representation of the patterns to be classified. The generation of the output is done by a two step procedure. The first is that e.g. for unit i the input in_i is multiplied with the junction J_i that connects this unit to the output unit on the right hand side. The output unit adds up the products of all input units. For the three units shown in Figure 2, this sum is given by:

$$sum = J_1 * in_1 + J_2 * in_2 + J_3 * in_3,$$

which is known as the 'inner product' of the two vectors J and in (see Figure 2).

The second step is what distinguishes the network from linear techniques:

sum is not used as calculated, but it is changed to either e.g. 0 or 1 depending on its value.

The final output out of the network thus is given by:

$$out = \begin{cases} 1, & \text{if } sum > \text{threshold,} \\ 0, & \text{if } sum \leq \text{threshold.} \end{cases}$$

The basic perceptron can be extended by adding units, and/or by adding further layers of neurons. These are called 'hidden layers', as they receive as input the output of the previous layer and pass their output as input to the next layer. Figure 3 (page 26) shows the example of a more complicated network that is used for secondary structure prediction.

For the application of networks as a gadget to classify patterns, some aspects have to be taken care of:

1. The data (patterns) must be chosen such that it contains the maximal amount of information at hand. The way in which the patterns are coded is crucial.
2. The capacity of the network should fit to the amount of data. The capacity is determined by the number of connections and by the point at which the training is stopped. 'Training' refers to the typical procedure to divide the available data into two sets: one being used to train the network (i.e., to extract the features of the data), the other to evaluate the network's performance on new data (generalization). (This procedure is also known as cross-validation or jack-knife test.) If the capacity of the net is beyond the amount of information in the data, the network learns the details being specific for the training data instead of extracting the rules according to which the data was produced. Thus the net will be good for learned cases but bad for its typical task being the classification of yet unknown patterns.
3. Most often, it helps to put as much expertise about a problem as possible into the arrangement of the network's architecture. ▶

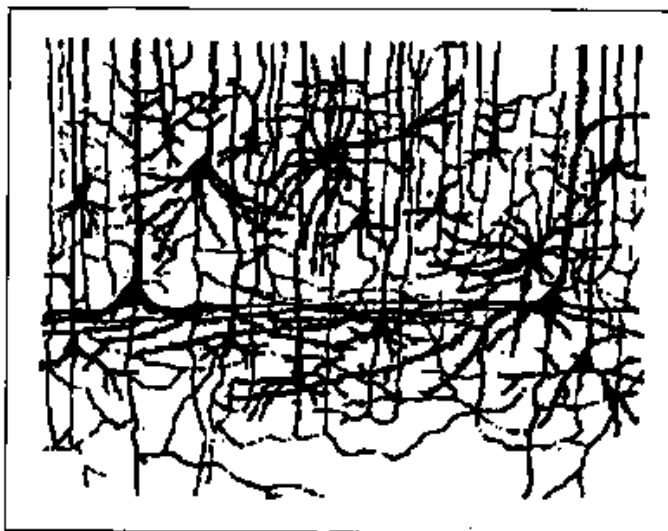


Figure 1. Neurons and their interconnections in the brain. The figure shows a slice of the neocortex. The cells (neurons) can be seen as black nodes. They are interconnected by the synapses. (Taken from Kahle, 79, p. 31.)

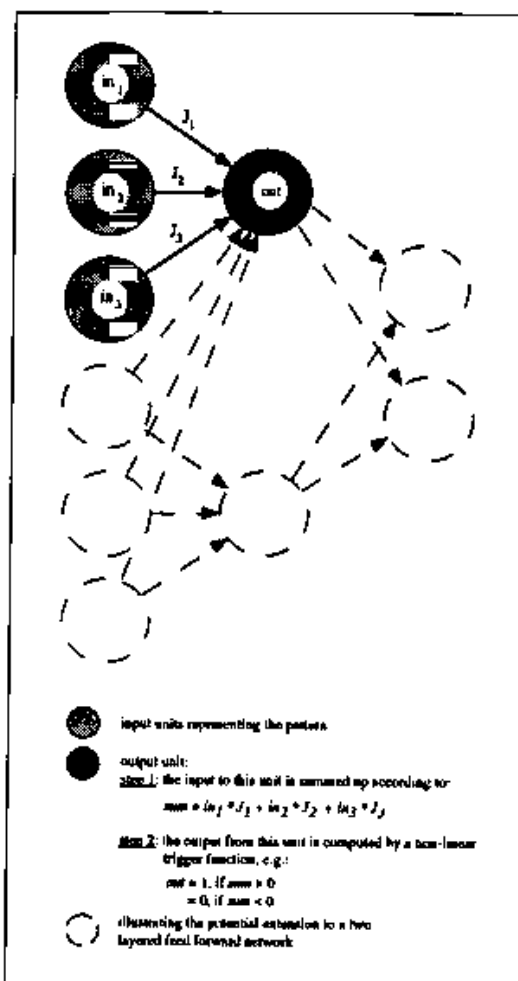
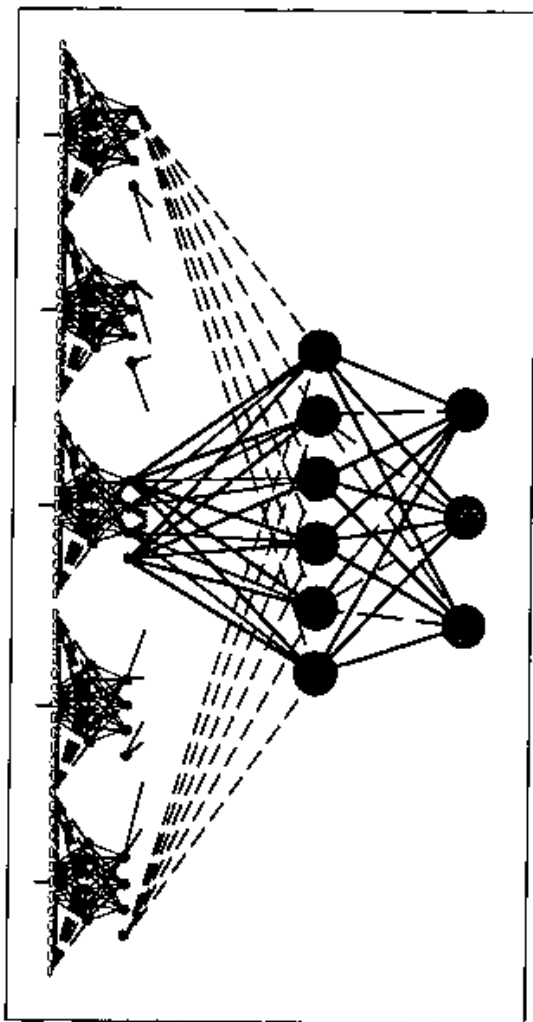


Figure 2. The perceptron and its potential extension. This figure shows the simplest architecture of a neural network, the perceptron.

Continued from page 25

One example for a successful application of a multi-layered network is the prediction of secondary structure of proteins which is an important, yet unsolved, task in molecular biology. The information about the three-dimensional (tertiary) structure of a protein is coded in its one-dimensional (primary) structure (see Anfinsen²). The secondary structure is a reduced description of the very complicated three-dimensional (tertiary) structure. A segment of amino acids can be translated into a binary vector used as input to a layered network (as shown in Figure 3). The output of the net consists, for example, of three units representing the three most important secondary structure types (helix, strand, rest). Thus, the output becomes a prediction of the secondary structure for the sequence segment fed in. Shifting the segment through various

Figure 3. Prediction of the secondary structure of a protein by a layered network. Such a network architecture was used by the author to predict secondary structure of proteins. In total the system uses more than 1,000 units with more than 20,000 connections.



proteins of known structure generates a set of samples, which can be learned iteratively by the network (see e.g., Qian and Sejnowski⁷ or Zhang *et al.*¹⁰). The performance accuracy of the network reaches up to 69% (Rost and Sander⁸). Figure 4 (page 27) shows typical prediction results for a protein (ubiquitin).

Drug design is to date most often initiated by a four step process:

1. A set of compounds is obtained by either serendipity or trial and error. These compounds typically have a common frame, and are often all derived from the same lead compound.
2. Biological data like log (LC50) values or binding constants is determined.
3. QSAR techniques are applied.
4. New compounds are predicted.

In this lead optimization process the steps 2-4 are repeated till an adequate compound is obtained.

Classically QSAR analysis uses a series of compounds with an important structure element in common. The aim is to parameterize the physical characteristics of substituents that differ between the compounds, and to correlate these parameters quantitatively to the activity. The largest problem is to parameterize hydrophobic, electrostatic or steric properties of individual substituents. However, even if these problems were to be solved perfectly, there is no guarantee for a successful QSAR analysis. The structure activity relation might be non-linear. On top of that the parameters might be correlated in a non-linear way, e.g. changes in one substituent influence the physical properties of another substitute. The natural ability of neural networks to detect non-linear data correlation makes them perfectly suited for analysis of QSAR problems that suffer from hidden non-linearity.

We will show one example where a non-linearity has been observed, and we will show how a very simple neural network could have been used to find this non-linearity. Tipker (Tipker⁹) analyzed a set of 21 DDT analogs (Tipker⁹) with activity against *Anopheles Albinanus*. The results of this study made beautifully clear that the initial QSAR studies were inadequate because the parameterization has been kept linear. Tipker assumed that the bulkiness of one of the groups had to be described by two parameters (one linear, one quadratic) to obtain good results. The original (linear) QSAR analysis gave a standard deviation between observed and calculated log (LC50) values of 0.618 (Tipker⁹). Tipker obtained a standard deviation of 0.374 by adding the non-linear term.

We used a novel type of neural network. The architecture of this extremely simple network is shown in Figure 5. Rather than using bit patterns as input, all

LAB NOTES

input is numerical. Training the network on the full 21 analogs gave a standard deviation of -0.2. The large discrepancy between 0.618 and 0.2 is a clear indicator for trouble in the original QSAR analysis.

The question remains however, whether the network simply learned the data by heart, or really extracted the hidden rules. To check this a jack-knife test has been performed. The network shown in Figure 5 has been

trained 21 times using 20 of the analogs. After every training the 21-st point was predicted. The standard deviation now became -0.4, clearly worse than 0.2, but still comparable with 0.374. Our results indicate that also Tipkers improved analysis of this dataset is not yet completely taking all effects into account. However, the limited number of analogs (21) probably precludes further improvement of the data analysis.

We have seen two examples of applications of neural networks. A very elaborate study has led to the best secondary structure prediction method available to date; whereas an extremely simple network has been used to detect 'trouble' in a QSAR analysis.

Neural networks are not the 'good-for-all' ultimate solution to all problems of mankind. However, when used carefully, and when the results are analyzed critically, they can be used in fields as remote as postal services, protein secondary structure prediction, and QSAR analysis.

By Burkhard Rost, Ph.D. and Gerrit Vriend, Ph.D.,
EMBL, Heidelberg, FRG.

References

1. Amit, Daniel J. (1989) *Modelling Brain Function: The world of Attractor Networks*, Cambridge University Press, Cambridge, UK.
2. Anfinsen, C. B.; Epstein, C. J.; Goldberger, R. F. (1963) "The genetic control of tertiary protein structure: studies with model systems," *Cold Spring Harbor Symp. Quant. Biol.*, 28, 439-49.
3. Cowan, J. D. (1990) "Neural networks: the early days," In Touretzky, David S. (ed.) *Neural Information Processing Systems 2*, 828-842, Morgan Kaufmann, San Mateo, CA.
4. Hertz, J. A.; Krogh, A.; Palmer, R. G. (1991) *Introduction to the Theory of Neural Computation*, Addison-Wesley, Redwood City, CA.

sequence	QIFVKTLTGKTITLEVEPSDTIENVKAKIQDKEGIPFDQQ			
observed	EEEEEE	EEEE	HHHHHHHHHH	HHEE
predicted	EEEE	EEEE	HHHHHHH	
sequence	RLIFAGKQLEDGRITLSDYNIQKESTLHLVLRGG			
observed	EE	EE	HHH	EEEE
predicted	EEHE	HH		HHHHHHH

Figure 4. Prediction of the secondary structure of a protein by a layered network. This figure compares the secondary structure being observed and predicted for the human chromosomal protein ubiquitin (1ubq). The first line gives the sequence of amino acids. The prediction was received by a two layered feed-forward network. H stand for helix, E for extended structure or strand, the blanks for non of these two.

5. Kahle, Werner (1979) *Nervensystem und Sinnesorgane*, Thieme, Stuttgart, FRG
6. Müller, B. and Reinhardt, J. (1990) *Neural Networks*, Springer, Berlin asf.
7. Qian, Ning and Sejnowski, Terrence J. (1988) "Predicting the Secondary Structure of Globular Proteins Using Neural Network Models," *J. Mol. Biol.*, 202: 865-84.
8. Rost, Burkhard and Sander, Christian (1992) *Improved prediction of protein secondary structure by use of sequence family profiles in neural networks*, Preprint EMBL Heidelberg, FRG.
9. Tipker, J. (1988) "Stereo-chemical problems in QSAR analysis," in: Ariens, E. J.; Rensen, J. J. S. and van Welling, W. (eds.), *Stereoselectivity of Pesticides*, Vol. 1, pp. 501-18 Elsevier, Amsterdam asf.
10. Zhang, Xiru; Mesirov, Jill P. and Waltz, David L. (1992) "Hybrid System for Protein Secondary Structure Prediction," *J. Mol. Biol.*, 225: 1049-63. ■

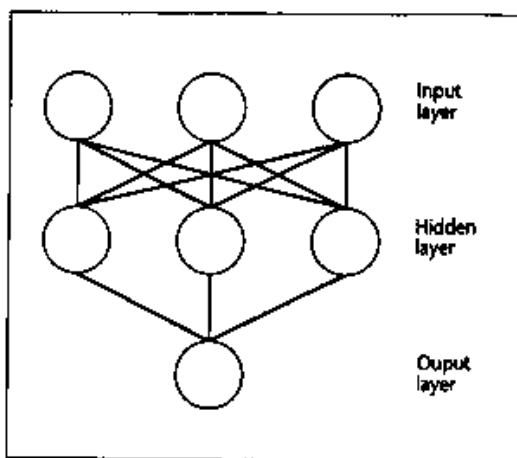


Figure 5. Architecture of QSAR neural network