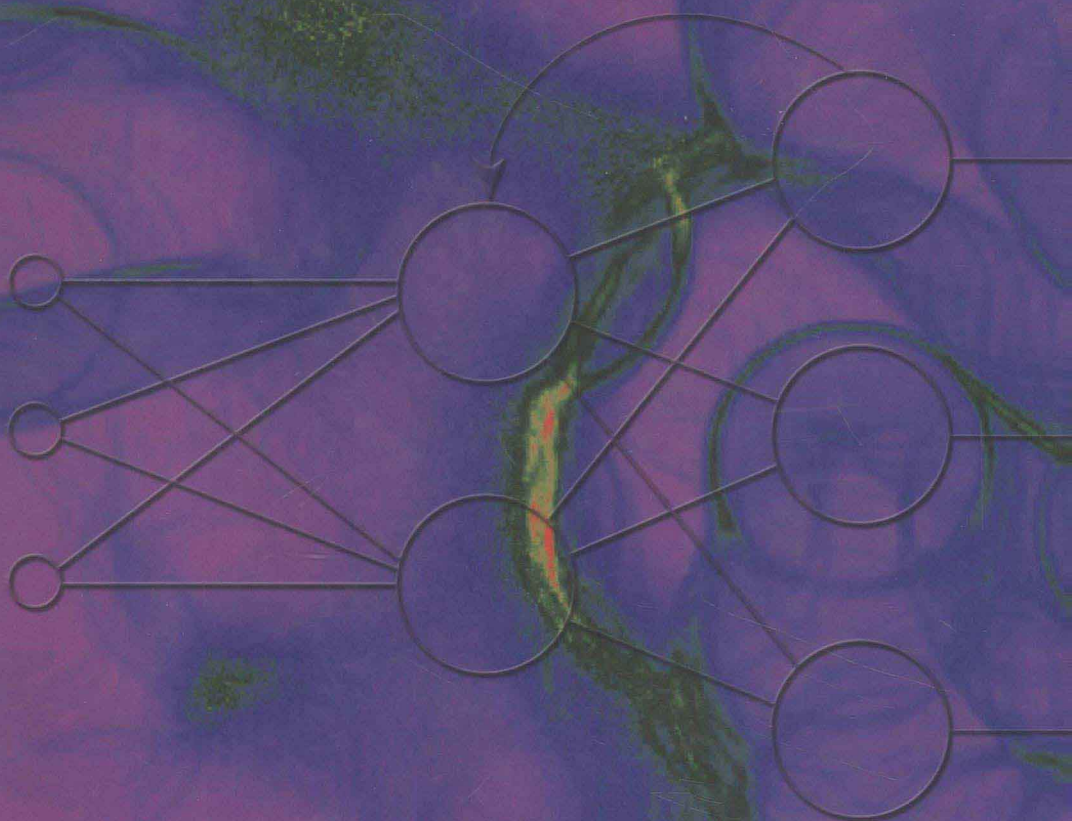


USING ARTIFICIAL INTELLIGENCE IN CHEMISTRY AND BIOLOGY

A Practical Guide



Hugh Cartwright



CRC Press
Taylor & Francis Group

USING ARTIFICIAL INTELLIGENCE IN CHEMISTRY AND BIOLOGY

A Practical Guide

Hugh Cartwright

Chapter 10, *Evolvable Developmental Systems*, contributed by

Nawwaf Kharma

Associate Professor, Department of Electrical and Computer Engineering,
Concordia University, Montreal, Canada



CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2008 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works
Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-13: 978-0-8493-8412-7 (Hardcover)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Cartwright, Hugh M., 1948-

Using artificial intelligence in chemistry and biology : a practical guide /
Hugh Cartwright ; chapter 10, Evolvable developmental systems, contributed by
Nawwaf Kharma.

p. cm.

Includes bibliographical references and index.

ISBN 978-0-8493-8412-7 (acid-free paper) 1. Chemistry--Data processing. 2.

Biology--Data processing. 3. Artificial intelligence. I. Kharma, Nawwaf. II.

Title.

QD39.3.E46C375 2008

540.285'63--dc22

2007047462

Visit the Taylor & Francis Web site at
<http://www.taylorandfrancis.com>

and the CRC Press Web site at
<http://www.crcpress.com>

Preface

The use of Artificial Intelligence within science is growing at a remarkable rate. In the early 1990s, little was known about how Artificial Intelligence could be applied in a practical way to the physical and life sciences. At that stage, few experimental scientists showed any interest in the area. Now, hundreds of research papers are published each year and the numbers are rising rapidly. The change has been dramatic and yet, despite the growth, the field is still very young.

The upsurge of interest owes much to an increasing understanding of the scientific potential of Artificial Intelligence. This book explains in a lucid and straightforward way how these methods are used by scientists and what we can accomplish with them. Recognizing that not all experimental scientists are computer experts, the approach adopted here assumes no prior knowledge of Artificial Intelligence and no unusual skills in computer science or programming. It does, however, presume some scientific background. Each chapter is designed to take the reader quickly to the point at which meaningful scientific applications can be investigated.

Computer scientists may use this book to gain a clearer picture of how experimental scientists use Artificial Intelligence tools. Chemists, biochemists, physicists, and others in the experimental sciences who have data to analyze or simulations to run will find tools within these pages that may speed up their work or make it more effective. For both groups, the aim of this book is to encourage a broader application of these methods.

Many people have contributed to the production of this book. The final chapter was written by Nawwaf Kharma, following several months on sabbatical at Oxford University. His perceptive and challenging chapter gives a glimpse of the future of Artificial Intelligence in science. EJS, a software tool for the construction of simulations in Java, has been used to perform a number of calculations for this book. A complete version of EJS is included on the accompanying CD with the permission of Francisco Esquembre, to whom I am most grateful. Most of the figures in this text have been prepared by John Freeman, adding both clarity and humor to it. The editorial staff at Taylor & Francis Group, particularly Lance Wobus, Russ Heaps, and Pat Roberson, have been a regular source of encouragement and expert advice. Numerous members of my research group, past and present, have contributed to the ideas that are crystallized here, notably "the two Alexes," whose comments about the interface between Artificial Intelligence and science are often thought-provoking. Finally, my wife, Susie, tolerating my long and unsociable working hours, and my daughter, Jenny, a distant but engaging e-mail voice, have provided less tangible but invaluable contributions. I am grateful to them all.

Errors are almost unavoidable in a text of this sort and I shall be grateful to be notified of any that may be spotted by readers.

Hugh Cartwright

*Chemistry Department, Oxford University
St Anne's College and Oriel College, Oxford, United Kingdom*

The Author

Hugh M. Cartwright is a member of the Chemistry Department at Oxford University in the United Kingdom, where he is Laboratory Officer. He is also a Supernumerary Fellow at St. Anne's College, Oxford, and a Lecturer in Physical Chemistry at Oriel College, Oxford. His research interests lie in the broad field of the application of computational methods from Artificial Intelligence to problems within science. Past and current research, carried out by some seventy students, has covered a wide variety of areas. Typical of these are studies of biodegradation, the prediction of the properties of polymers, the analysis of oil spill data, automatic elucidation of scientific rules, the design of organic synthetic routes, the optimization of solid-state phosphor composition, quantitative structure-activity relationships (QSARs), unraveling mass spectrometric data from crude oils, and the analysis of large biomedical datasets. The work of his group focuses primarily on the use of genetic algorithms, neural networks, self-organizing maps, and growing cell structures.

He has written, edited, and provided invited chapters in a number of textbooks on the use of Artificial Intelligence in science and continues to run an active research group in the area. He has taught at universities in Canada, the United States, Ireland, and the United Kingdom, and is a board member of the CoLoS (Conceptual Learning of Science) International Consortium.

Contents

Preface xiii

The Author xv

1 Artificial Intelligence 1

1.1 What Is Artificial Intelligence?..... 2

1.2 Why Do We Need Artificial Intelligence and What Can We Do with It? 3

1.2.1 Classification 5

1.2.2 Prediction..... 5

1.2.3 Correlation..... 5

1.2.4 Model Creation 6

1.3 The Practical Use of Artificial Intelligence Methods 6

1.4 Organization of the Text 7

References 8

2 Artificial Neural Networks 9

2.1 Introduction 9

2.2 Human Learning..... 11

2.3 Computer Learning 12

2.4 The Components of an Artificial Neural Network 14

2.4.1 Nodes 14

2.4.2 Connections..... 14

2.4.3 Connection Weights 15

2.4.4 Determining the Output from a Node 15

2.4.4.1 The Input Signal..... 16

2.4.4.2 The Activation Function 17

2.5 Training 21

2.6 More Complex Problems..... 24

2.7 Layered Networks..... 26

2.7.1 More Flexible Activation Functions..... 28

2.7.1.1 Sigmoidal Activation Functions 29

2.8 Training a Layered Network: Backpropagation 30

2.8.1 The Mathematical Basis of Backpropagation 32

2.9 Learning Rate 35

2.10 Momentum..... 36

2.11 Practical Issues 37

2.11.1 Network Geometry and Overfitting..... 37

2.11.2 The Test Set and Early Stopping..... 38

2.11.3 Leave One Out 40

2.12 Traps and Remedies..... 40

2.12.1 Network Size 40

2.12.2	Sample Order	40
2.12.3	Ill-Conditioned Networks and the Normalization of Data.....	41
2.12.4	Random Noise.....	42
2.12.5	Weight Decay	43
2.12.6	Growing and Pruning Networks.....	43
2.12.7	Herd Effect.....	44
2.12.8	Batch or Online Learning.....	45
2.13	Applications.....	46
2.14	Where Do I Go Now?.....	47
2.15	Problems.....	47
	References.....	48
3	Self-Organizing Maps	51
3.1	Introduction.....	52
3.2	Measuring Similarity	54
3.3	Using a Self-Organizing Map	56
3.4	Components in a Self-Organizing Map.....	57
3.5	Network Architecture	57
3.6	Learning.....	59
3.6.1	Initialize the Weights.....	60
3.6.2	Select the Sample.....	62
3.6.3	Determine Similarity	62
3.6.4	Find the Winning Node.....	63
3.6.5	Update the Winning Node.....	64
3.6.6	Update the Neighborhood	65
3.6.7	Repeat.....	67
3.7	Adjustable Parameters in the SOM.....	71
3.7.1	Geometry	71
3.7.2	Neighborhood.....	73
3.7.3	Neighborhood Functions	73
3.8	Practical Issues	80
3.8.1	Choice of Parameters	80
3.8.2	Visualization	81
3.8.3	Wraparound Maps	85
3.8.4	Maps of Other Shapes.....	87
3.9	Drawbacks of the Self-Organizing Map.....	88
3.10	Applications.....	89
3.11	Where Do I Go Now?.....	93
3.12	Problems.....	93
	References.....	94
4	Growing Cell Structures	95
4.1	Introduction.....	96
4.2	Growing Cell Structures.....	98

4.3	Training and Evolving a Growing Cell Structure.....	99
4.3.1	Preliminary Steps.....	99
4.3.2	Local Success Measures.....	99
4.3.2.1	Signal Counter.....	100
4.3.2.2	Local Error.....	100
4.3.3	Best Matching Unit.....	102
4.3.4	Weights.....	102
4.3.5	Local Measures Decay.....	103
4.3.6	Repeat.....	103
4.4	Growing the Network.....	104
4.5	Removing Superfluous Cells.....	108
4.6	Advantages of the Growing Cell Structure.....	109
4.7	Applications.....	110
4.8	Where Do I Go Now?.....	111
4.9	Problems.....	111
	References.....	111
5	Evolutionary Algorithms.....	113
5.1	Introduction.....	113
5.2	The Evolution of Solutions.....	114
5.3	Components in a Genetic Algorithm.....	116
5.4	Representation of a Solution in the Genetic Algorithm.....	117
5.5	Operation of the Genetic Algorithm.....	119
5.5.1	Genetic Algorithm Parameters.....	120
5.5.2	Initial Population.....	120
5.5.3	Quality of Solution.....	121
5.5.4	Selection.....	124
5.5.5	Crossover.....	127
5.5.6	Mutation.....	128
5.6	Evolution.....	130
5.7	When Do We Stop?.....	133
5.8	Further Selection and Crossover Strategies.....	135
5.8.1	Selection Strategies.....	135
5.8.1.1	Many-String Tournaments.....	135
5.8.1.2	Roulette Wheel Selection.....	136
5.8.1.3	Stochastic Remainder Selection.....	136
5.8.1.4	Fitness Scaling.....	138
5.8.2	How Does the Genetic Algorithm Find Good Solutions?.....	140
5.8.3	Crossover Revisited.....	142
5.8.3.1	Uniform Crossover.....	143
5.8.3.2	Two-Point Crossover.....	144
5.8.3.3	Wraparound Crossover.....	144
5.8.3.4	Other Types of Crossover.....	147
5.8.4	Population Size.....	148

5.8.5	Focused Mutation.....	150
5.8.6	Local Search	151
5.9	Encoding.....	151
5.9.1	Real or Binary Coding?.....	151
5.10	Repairing String Damage	155
5.11	Fine Tuning.....	159
5.11.1	Elitism	159
5.11.2	String Fridge.....	160
5.12	Traps.....	161
5.13	Other Evolutionary Algorithms.....	162
5.13.1	Evolutionary Strategies.....	162
5.13.2	Genetic Programming	163
5.13.3	Particle Swarm Optimization.....	166
5.14	Applications.....	168
5.15	Where Do I Go Now?.....	169
5.16	Problems.....	170
	References.....	171
6	Cellular Automata	173
6.1	Introduction	174
6.2	Principles of Cellular Automata.....	175
6.3	Components of a Cellular Automata Model	177
6.3.1	The Cells and Their States.....	177
6.3.2	Transition Rules	179
6.3.3	Neighborhoods	180
6.3.3.1	The Neighborhood in a One-Dimensional Cellular Automata	180
6.3.3.2	The Neighborhood in a Two-Dimensional Cellular Automata	181
6.4	Theoretical Applications.....	182
6.4.1	Random Transition Rules.....	182
6.4.2	Deterministic Transition Rules.....	185
6.4.2.1	Voting Rules.....	185
6.5	Practical Considerations.....	191
6.5.1	Boundaries.....	191
6.5.2	Additional Parameters.....	195
6.6	Extensions to the Model: Excited States.....	195
6.7	Lattice Gases	197
6.8	Applications.....	198
6.9	Where Do I Go Now?.....	199
6.10	Problems.....	200
	References.....	201
7	Expert Systems.....	203
7.1	Introduction	204
7.2	An Interaction with a Simple Expert System	205

7.3	Applicability of Expert Systems.....	208
7.4	Goals of an Expert System.....	209
7.5	The Components of an Expert System.....	210
7.5.1	The Knowledge Base.....	211
7.5.1.1	Factual Knowledge.....	211
7.5.1.2	Heuristic Knowledge.....	213
7.5.1.3	Rules.....	214
7.5.1.4	Case-Based Library.....	214
7.5.2	Inference Engine and Scheduler.....	215
7.5.3	The User Interface.....	215
7.5.4	Knowledge-Base Editor.....	216
7.6	Inference Engine.....	216
7.6.1	Rule Chaining.....	218
7.6.1.1	Forward Chaining.....	218
7.6.1.2	Backward Chaining.....	219
7.7	Explanation and Limits to Knowledge.....	223
7.8	Case-Based Reasoning.....	225
7.9	An Expert System for All?.....	225
7.10	Expert System Shells.....	226
7.11	Can I Build an Expert System?.....	230
7.12	Applications.....	233
7.13	Where Do I Go Now?.....	234
7.14	Problems.....	234
	References.....	235
8	Fuzzy Logic.....	237
8.1	Introduction.....	237
8.2	Crisp Sets.....	239
8.3	Fuzzy Sets.....	240
8.4	Calculating Membership Values.....	244
8.5	Membership Functions.....	245
8.6	Is Membership the Same as Probability?.....	248
8.7	Hedges.....	249
8.8	How Does a Fuzzy Logic System Work?.....	250
8.8.1	Input Data.....	252
8.8.2	Fuzzification of the Input.....	252
8.9	Application of Fuzzy Rules.....	254
8.9.1	Aggregation.....	255
8.10	Applications.....	259
8.11	Where Do I Go Now?.....	260
8.12	Problems.....	260
	References.....	261
9	Learning Classifier Systems.....	263
9.1	Introduction.....	263

9.2	A Basic Classifier System	266
9.2.1	Components of a Classifier System.....	268
9.2.1.1	The Environment and Messages from It	269
9.2.1.2	Input Message List.....	271
9.2.1.3	The Classifier List	271
9.2.1.4	The Output Interface	272
9.3	How a Classifier System Works	272
9.3.1	Gather the Input.....	272
9.3.2	Compare Input Messages with the Classifiers.....	273
9.3.3	Generate Output Messages	274
9.3.4	Update the Environment.....	276
9.4	Properties of Classifiers.....	276
9.4.1	Generalist and Specialist Classifiers.....	276
9.4.2	Classifier Specificity	277
9.4.3	Classifier Strength	278
9.5	Learning Classifier System.....	279
9.5.1	Bids	279
9.5.2	Rewards	281
9.5.3	Learning through Rule Culling.....	281
9.5.4	Learning through Evolution	283
9.5.5	Coupled Classifiers.....	284
9.5.6	Taxation.....	285
9.6	Applications.....	286
9.7	Where Do I Go Now?.....	286
9.8	Problems.....	287
	References.....	287
10	Evolvable Developmental Systems	289
	<i>Nawwaf Khurma</i>	
10.1	Introduction and Motivation.....	289
10.2	Relationship between Evolution and Development.....	290
10.3	Production Rules and the Evolution of Digital Circuits.....	293
10.3.1	Description of Sample Problem.....	293
10.3.2	Representation of Potential Solution	294
10.3.3	Developmental Procedure.....	295
10.3.4	Fitness Evaluation.....	297
10.3.5	Overall Evolution.....	297
10.3.6	Experimental Results	298
10.3.7	Summary and Challenges.....	298
10.4	Cellular Automata-Like Systems and Evolution of Plane Trusses	299
10.4.1	Developmental Procedure.....	301
10.4.2	Representation of Potential Solution	302
10.4.3	Overall Evolution.....	304
10.4.4	Fitness Evaluation.....	305

10.4.5	Experimental Results	306
10.4.6	Summary and Challenges.....	306
10.5	Cellular Growth and the Evolution of Functions	308
10.5.1	Description of Sample Problem.....	308
10.5.2	Representation of Potential Solution	309
10.5.3	Developmental Procedure.....	309
10.5.4	Fitness Evaluation.....	312
10.5.5	Overall Evolution.....	312
10.5.6	Experimental Results	312
10.5.7	Summary and Challenges.....	313
10.6	Genomic Regulatory Networks and Modeling Development ...	314
10.6.1	Description of Sample Problem.....	314
10.6.2	Representations of Potential Solutions	314
10.6.2.1	Complex Model of Development.....	314
10.6.2.2	Simple Model of Development.....	315
10.6.3	Developmental Procedures.....	316
10.6.4	Fitness Evaluation.....	317
10.6.5	Overall Evolution.....	319
10.6.6	Extended Example.....	319
10.6.7	Summary and Challenges.....	322
10.7	Summary	323
10.8	Future Challenges and Epilogue	324
10.8.1	Epilogue	326
	Acknowledgments.....	327
	References	327
	Index.....	329

1

Artificial Intelligence

Computers don't understand. At least, they don't understand in the way that we do. Of course, there are some things that are almost beyond comprehension: Why do women collect shoes? What are the rules of cricket? Why do Americans enjoy baseball? And why are lawyers paid so much?

These are tough questions. As humans, we can attempt to answer them, but (at least for the next few years) computers cannot help us out. Humans possess far more "intelligence" than computers, so it is to be expected that if a human struggles to understand shoe collecting, a computer will be totally baffled by it. This makes it all the more surprising that computers that use "artificial intelligence" can solve a variety of scientific tasks much more effectively than a human.

Artificial Intelligence (AI) tools are problem-solving algorithms. This book provides an introduction to the wide range of methods within this area that are being developed as scientific tools. The application of AI methods in science is a young field, hardly out of diapers in fact. However, its potential is huge. As a result, the popularity of these methods among physical and life scientists is increasing rapidly.

In the conventional image of AI, we might expect to see scientists building laboratory robots or developing programs to translate scientific papers from Russian to English or from English to Russian, but this does not accurately reflect the current use of AI in science. The creation of autonomous robots and the perfection of automatic translators are among the more important areas of research among computer scientists, but the areas in which most experimental scientists are involved are very different.

Science is dominated by problems; indeed, without them there would be hardly any science. When tackling a new problem, scientists are generally less concerned about *how* they solve it, provided that they get a solution, than they are about the quality of the solution that is obtained. Thus, the pool of mathematical and logical algorithms that scientists use for problem solving continues to grow. That pool is now being augmented by AI methods.

Until the second half of the 1990s, experimental scientists, by and large, knew little about AI, suspecting (without much evidence) that it might be largely irrelevant in practical science. This turned out to be an unduly pessimistic view. AI algorithms are in reality widely applicable and the power of these methods, allied to their simplicity, makes them an attractive proposition in the analysis of scientific data and for scientific simulation. The purpose of this book is to introduce these methods to those who need to solve scientific problems, but do not (yet) know enough about AI to take advantage

of it. The text includes sufficient detail that those who have some modest programming skills, but no prior contact with AI, can rapidly set about using the techniques.

1.1 What Is Artificial Intelligence?

Computer scientists usually define AI in terms of what it can accomplish. They are especially interested in the creation of intelligent software that can reproduce the kind of behavior that humans would recognize as intelligent, such as understanding language or conducting a conversation.

Some of this work on human-like behavior spills over into the scientific laboratory where work progresses on producing intelligent robotic systems for chemical or biological laboratories. Robotic sample handlers are commonplace in science, and already incorporate enough rudimentary intelligence to be able to cope with the demands of the job. However, a robotic sampler is really pretty dumb and, in truth, that is how it should be. A sampler whose role is to feed samples into a mass spectrometer should do no more than unquestioningly follow predefined rules of priority. If it had ideas above its station and started making its own decisions about whose sample would be processed next, based on the color of the users' hair or the number of syllables in their name perhaps, it would soon be headed for the great sampler scrap heap in the sky.

Experimental scientists tend not to be interested in robotic samplers, even those with attitude; instead they want software that will solve problems. For them, AI tools may be of considerable value as problem solvers. In this book, we explore such uses.

For most scientific purposes, a reasonable working definition of AI is: "A computer program that can learn."

Although computer scientists might quibble at this definition, it describes most AI methods rather well. Methods that depend on learning, such as artificial neural networks and genetic algorithms, are already proving their worth in science, and there are others, such as growing cell structure networks, whose potential is just starting to be recognized.

The learning that AI algorithms depend on can be achieved in several different ways. An expert system (Chapter 7) is tutored by a human expert and starts off resembling an empty box awaiting information. This requires that a scientist spend long periods feeding relevant information into the system. By contrast, artificial neural networks (Chapter 2) inspect examples of what they are required to understand and learn by extracting examples from a database or by interacting with a simulation. Classifier Systems (Chapter 9) talk directly with their environment, attempting to control it, receiving feedback from it, and learning from any mistakes. The self-organizing map (Chapter 3) makes its deductions by just looking at data without having to be told what it is expected to learn.

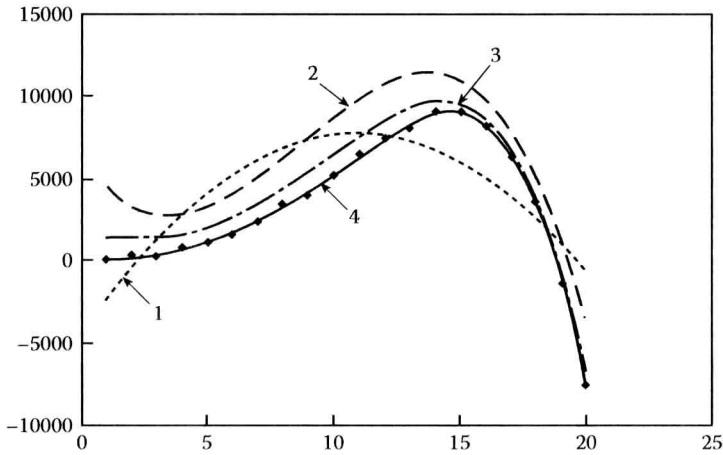


FIGURE 1.1
Progressive improvements in the attempts by a genetic algorithm to find a curve that fits a set of data points well (lines 1 to 3). The solid line is fitted using standard least squares.

But, you may have spotted a problem here. If an AI program has to learn, does that not suggest that its performance might be fairly hopeless when we first ask it to solve a problem? Indeed it does, as Figure 1.1 indicates.

The figure shows a polynomial fit to a dataset calculated according to a standard least squares algorithm (solid line); this is compared with a series of attempts to find a fit to the same data using a genetic algorithm.

The genetic algorithm is unimpressive, at least to begin with (line 1). The quality of the fit that it finds does improve (lines 2 and 3) and eventually it will reach a solution that matches the least squares fit, but the algorithm takes far longer to find the solution than standard methods do, and the fit is no better.

1.2 Why Do We Need Artificial Intelligence and What Can We Do with It?

So why bother? If established and reliable methods exist to solve scientific problems, surely there is no point in using methods from AI that may be slower or less effective? The key here is the phrase: “if established and reliable methods exist.” In many situations, there is no need to turn to AI methods, but not every scientific problem is as straightforward as finding the line of best fit. When more conventional methods are unavailable, or are of insufficient power, an AI method may be just what is needed.

There are several reasons why it may be difficult to find an adequate solution to a scientific problem; the two most common are scale and complexity.

In many types of problems, the number of possible solutions can be huge. The folding of a protein is a typical example. Proteins fold to take up a conformation of minimum energy, although the conformation that the protein adopts in a living organism may not be that of the global minimum energy. The number of ways in which a large molecule, such as a protein, can be folded so that its energy is at least locally minimized and all bond lengths and angles are reasonable is so large that it is impossible to investigate all of them computationally. As it is not feasible to check every conformation, some search method is needed that makes intelligent decisions about which conformations to assess and which to ignore.

A problem of similar scale is the identification of useful drugs from among the huge number that could, in principle, be prepared. The number of molecules that might potentially act as drugs can only be estimated, but it is probably of the order of 10^{40} . Of this huge number of possibilities only a tiny fraction will ever be synthesized, and of those only a handful will make it through the process of high throughput screening to human trials. Any computational procedure that is devised to screen molecules for possible value as drugs must be able, somehow, to search through this huge number of candidates to identify promising structures without actually inspecting every molecule (Figure 1.2).

The chemical flowshop, which we shall meet in Chapter 5, is similarly challenging to solve. Many different chemicals are synthesized within a single line of processing units in a defined sequence. The efficiency of the flowshop, which is determined by the length of time required to synthesize the complete group of chemicals, is strongly correlated with the order in which chemicals are made. Therefore, determining a near-optimum order is important in the operation of flowshops if the process is to be financially viable. For n chemicals, the number of possible sequences is $n!$, a number

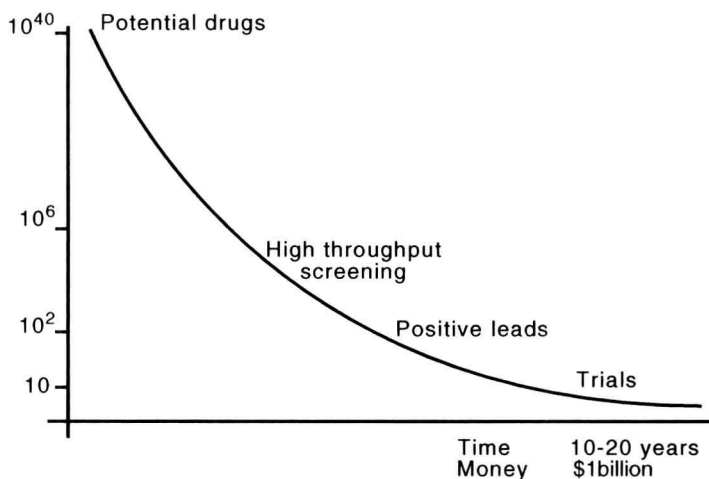


FIGURE 1.2
The drug development process.