COMPUTATIONAL INTELLIGENCE AND PATTERN ANALYSIS IN BIOLOGICAL INFORMATICS

Edited by

UJJWAL MAULIK
Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

SANGHAMITRA BANDYOPADHYAY
Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India

JASON T. L. WANG
Department of Computer Science, New Jersey Institute of Technology, Newark, New Jersey

WILEY
A JOHN WILEY & SONS, INC., PUBLICATION
COMPUTATIONAL INTELLIGENCE AND PATTERN ANALYSIS IN BIOLOGICAL INFORMATICS
Wiley Series on

**Bioinformatics: Computational Techniques and Engineering**

A complete list of the titles in this series appears at the end of this volume.
COMPUTATIONAL INTELLIGENCE AND PATTERN ANALYSIS IN BIOLOGICAL INFORMATICS

Edited by

UJJWAL MAULIK
Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

SANGHAMITRA BANDYOPADHYAY
Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India

JASON T. L. WANG
Department of Computer Science, New Jersey Institute of Technology, Newark, New Jersey
To Utsav, our students and parents  
—U. Maulik and 
S. Bandyopadhyay 

To my wife Lynn and 
daughter Tiffany 
—J. T. L. Wang
CONTENTS

Preface xi
Contributors xvii

PART 1 INTRODUCTION

1 Computational Intelligence: Foundations, Perspectives, and Recent Trends 3
Swagatam Das, Ajith Abraham, and B. K. Panigrahi

2 Fundamentals of Pattern Analysis: A Brief Overview 39
Basabi Chakraborty

3 Biological Informatics: Data, Tools, and Applications 59
Kevin Byron, Miguel Cervantes-Cervantes, and Jason T. L. Wang

PART II SEQUENCE ANALYSIS

4 Promoter Recognition Using Neural Network Approaches 73
T. Sobha Rani, S. Durga Bhavani, and S. Bapi Raja

5 Predicting microRNA Prostate Cancer Target Genes 99
Francesco Masulli, Stefano Rovetta, and Giuseppe Russo
PART III STRUCTURE ANALYSIS

6 Structural Search in RNA Motif Databases
   Dongrong Wen and Jason T. L. Wang

7 Kernels on Protein Structures
   Sourangshu Bhattacharya, Chiranjib Bhattacharyya,
   and Nagasuma R. Chandra

8 Characterization of Conformational Patterns in Active and
   Inactive Forms of Kinases using Protein Blocks Approach
   G. Agarwal, D. C. Dinesh, N. Srinivasan, and Alexandre G. de Brevern

9 Kernel Function Applications in Cheminformatics
   Aaron Smalter and Jun Huan

10 In Silico Drug Design Using a Computational
    Intelligence Technique
    Soumi Sengupta and Sanghamitra Bandyopadhyay

PART IV MICROARRAY DATA ANALYSIS

11 Integrated Differential Fuzzy Clustering for Analysis of
   Microarray Data
   Indrajit Saha and Ujjwal Maulik

12 Identifying Potential Gene Markers Using SVM
    Classifier Ensemble
   Anirban Mukhopadhyay, Ujjwal Maulik, and
   Sanghamitra Bandyopadhyay

13 Gene Microarray Data Analysis Using Parallel Point
    Symmetry-Based Clustering
   Ujjwal Maulik and Anasua Sarkar

PART V SYSTEMS BIOLOGY

14 Techniques for Prioritization of Candidate Disease Genes
   Jieun Jeong and Jake Y. Chen
CONTENTS

15 Prediction of Protein–Protein Interactions 325
  Angshuman Bagchi

16 Analyzing Topological Properties of Protein–Protein Interaction Networks: A Perspective Toward Systems Biology 349
  Malay Bhattacharyya and Sanghamitra Bandyopadhyay

Index 369
Computational biology is an interdisciplinary field devoted to the interpretation and analysis of biological data using computational techniques. It is an area of active research involving biology, computer science, statistics, and mathematics to analyze biological sequence data, genome content and arrangement, and to predict the function and structure of macromolecules. This field is a constantly emerging one, with new techniques and results being reported every day. Advancement of data collection techniques is also throwing up novel challenges for the algorithm designers to analyze the complex and voluminous data. It has already been established that traditional computing methods are limited in their scope for application to such complex, large, multidimensional, and inherently noisy data. Computational intelligence techniques, which combine elements of learning, adaptation, evolution, and logic, are found to be particularly well suited to many of the problems arising in biology as they have flexible information processing capabilities for handling huge volume of real-life data with noise, ambiguity, missing values, and so on. Solving problems in biological informatics often involves search for some useful regularities or patterns in large amounts of data that are typically characterized by high dimensionality and low sample size. This necessitates the development of advanced pattern analysis approaches since the traditional methods often become intractable in such situations.

In this book, we attempt to bring together research articles by active practitioners reporting recent advances in integrating computational intelligence and pattern analysis techniques, either individually or in a hybridized manner, for analyzing biological data in order to extract more and more meaningful information and insights from them. Biological data to be considered for analysis include sequence, structure, and microarray data. These data types are typically complex in nature, and require advanced methods to deal with them. Characteristics of the methods and algorithms
reported here include the use of domain-specific knowledge for reducing the search space, dealing with uncertainty, partial truth and imprecision, efficient linear and/or sublinear scalability, incremental approaches to knowledge discovery, and increased level and intelligence of interactivity with human experts and decision makers. The techniques can be sequential or parallel in nature.

Computational Intelligence (CI) is a successor of artificial intelligence that combines elements of learning, adaptation, evolution, and logic to create programs that are, in some sense, intelligent. Computational intelligence exhibits an ability to learn and/or to deal with new situations, such that the system is perceived to possess one or more attributes of reason, (e.g., generalization, discovery, association, and abstraction). The different methodologies in CI work synergistically and provide, in one form or another, flexible information processing capabilities. Many biological data are characterized by high dimensionality and low sample size. This poses grand challenges to the traditional pattern analysis techniques necessitating the development of sophisticated approaches.

This book has five parts. The first part contains chapters introducing the basic principles and methodologies of computational intelligence techniques along with a description of some of its important components, fundamental concepts in pattern analysis, and different issues in biological informatics, including a description of biological data and their sources. Detailed descriptions of the different applications of computational intelligence and pattern analysis techniques to biological informatics constitutes the remaining chapters of the book. These include tasks related to the analysis of sequences in the second part, structures in the third part, and microarray data in part four. Some topics in systems biology form the concluding part of this book.

In Chapter 1, Das et al. present a lucid overview of computational intelligence techniques. They introduce the fundamental aspects of the key components of modern computational intelligence. A comprehensive overview of the different tools of computational intelligence (e.g., fuzzy logic, neural network, genetic algorithm, belief network, chaos theory, computational learning theory, and artificial life) is presented. It is well known that the synergistic behavior of the above tools often far exceeds their individual performance. A description of the synergistic behaviors of neuro-fuzzy, neuro-GA, neuro-belief, and fuzzy-belief network models is also included in this chapter. It concludes with a detailed discussion on some emerging trends in computational intelligence like swarm intelligence, Type-2 fuzzy sets, rough sets, granular computing, artificial immune systems, differential evolution, bacterial foraging optimization algorithms, and the algorithms based on artificial bees foraging behavior.

Chakraborty provides an overview of the basic concepts and the fundamental techniques of pattern analysis with an emphasis on statistical methods in Chapter 2. Different approaches for designing a pattern recognition system are described. The pattern recognition tasks of feature selection, classification, and clustering are discussed in detail. The most popular statistical tools are explained. Recent approaches based on the soft computing paradigm are also introduced in this chapter, with a brief representation of the promising neural network classifiers as a new direction toward dealing with imprecise and uncertain patterns generated in newer fields.
In Chapter 3, Byron et al. deal with different aspects of biological informatics. In particular, the biological data types and their sources are mentioned, and two software tools used for analyzing the genomic data are discussed. A case study in biological informatics, focusing on locating noncoding RNAs in Drosophila genomes, is presented. The authors show how the widely used Infernal and RSmatch tools can be combined to mine roX1 genes in 12 species of Drosophila for which the entire genomic sequencing data is available.

The second part of the book, Chapters 4 and 5, deals with the applications of computational intelligence and pattern analysis techniques for biological sequence analysis. In Chapter 4, Rani et al. extract features from the genomic sequences in order to predict promoter regions. Their work is based on global signal-based methods using a neural network classifier. For this purpose, they consider two global features: $n$-gram features and features based on signal processing techniques by mapping the sequence into a signal. It is shown that the $n$-gram features extracted for $n = 2, 3, 4$, and 5 efficiently discriminate promoters from nonpromoters.

In Chapter 5, Masulli et al. deal with the task of computational prediction of microRNA (miRNA) targets with focus on miRNAs’ influence in prostate cancer. The miRNAs are capable of base-pairing with imperfect complementarity to the transcripts of animal protein-coding genes (also termed targets) generally within the 3’ untranslated region (3’ UTR). The existing target prediction programs typically rely on a combination of specific base-pairing rules in the miRNA and target mRNA sequences, and conservational analysis to score possible 3’ UTR recognition sites and enumerate putative gene targets. These methods often produce a large number of false positive predictions. In this chapter, Masulli et al. improve the performance of an existing tool called miRanda by exploiting the updated information on biologically validated miRNA gene targets related to human prostate cancer only, and performing automatic parameter tuning using genetic algorithm.

Chapters 6–10 constitute the third part of the book dealing with structural analysis. Chapter 6 deals with the structural search in RNA motif databases. An RNA structural motif is a substructure of an RNA molecule that has a significant biological function. In this chapter, Wen and Wang present two recently developed structural search engines. These are useful to scientists and researchers who are interested in RNA secondary structure motifs. The first search engine is installed on a database, called RmotifDB, which contains secondary structures of the noncoding RNA sequences in Rfam. The second search engine is installed on a block database, which contains the 603 seed alignments, also called blocks, in Rfam. This search engine employs a novel tool, called BlockMatch, for comparing multiple sequence alignments. Some experimental results are reported to demonstrate the effectiveness of the BlockMatch tool.

In Chapter 7, Bhattacharya et al. explore the construction of neighborhood-based kernels on protein structures. Two types of neighborhoods, and two broad classes of kernels, namely, sequence and structure based, are defined. Ways of combining these kernels to get kernels on neighborhoods are discussed. Detailed experimental results are reported showing that some of the designed kernels perform competitively with the state of the art structure comparison algorithms, on the difficult task of classifying 40% sequence nonredundant proteins into SCOP superfamilies.
The use of protein blocks to characterize structural variations in enzymes is discussed in Chapter 8 using kinases as the case study. A protein block is a set of 16 local structural descriptors that has been derived using unsupervised machine learning algorithms and that can approximate the three-dimensional space of proteins. In this chapter, Agarwal et al. first apply their approach in distinguishing between conformation changes and rigid-body displacements between the structures of active and inactive forms of a kinase. Second, a comparison of the conformational patterns of active forms of a kinase with the active and inactive forms of a closely related kinase has been performed. Finally, structural differences in the active states of homologous kinases have been studied. Such studies might help in understanding the structural differences among these enzymes at a different level, as well as guide in making drug targets for a specific kinase.

In Chapter 9, Smalter and Huan address the problem of graph classification through the study of kernel functions and the application of graph classification in chemical quantitative structure–activity relationship (QSAR) study. Graphs, especially the connectivity maps, have been used for modeling chemical structures for decades. In connectivity maps, nodes represent atoms and edges represent chemical bonds between atoms. Support vector machines (SVMs) that have gained popularity in drug design and cheminformatics are used in this regard. Some graph kernel functions are explored that improve on existing methods with respect to both classification accuracy and kernel computation time. Experimental results are reported on five different biological activity data sets, in terms of the classifier prediction accuracy of the support vector machine for different feature generation methods.

Computational ligand design is one of the promising recent approaches to address the problem of drug discovery. It aims to search the chemical space to find suitable drug molecules. In Chapter 10, genetic algorithms have been applied for this combinatorial problem of ligand design. The chapter proposes a variable length genetic algorithm for de novo ligand design. It finds the active site of the target protein from the input protein structure and computes the bond stretching, angle bending, angle rotation, van der Waals, and electrostatic energy components using the distance dependent dielectric constant for assigning the fitness score for every individual. It uses a library of 41 fragments for constructing ligands. Ligands have been designed for two different protein targets, namely, Thrombin and HIV-1 Protease. The ligands obtained, using the proposed algorithm, were found to be similar to the real known inhibitors of these proteins. The docking energies using the proposed methodology designed were found to be lower compared to three existing approaches.

Chapters 11–13 constitute the fourth part of the book dealing with microarray data analysis. In Chapter 11, Saha and Maulik develop a differential evolution-based fuzzy clustering algorithm (DEFC) and apply it on four publicly available benchmark microarray data sets, namely, yeast sporulation, yeast cell cycle, Arabidopsis Thaliana, and human fibroblasts serum. Detailed comparative results demonstrating the superiority of the proposed approach are provided. In a part of the investigation, an interesting study integrating the proposed clustering approach with an SVM classifier has been conducted. A fraction of the data points is selected from different clusters based on their proximity to the respective centers. This is used for training an SVM.
The clustering assignments of the remaining points are thereafter determined using the trained classifier. Finally, a biological significance test has been carried out on yeast sporulation microarray data to establish that the developed integrated technique produces functionally enriched clusters.

The classification capability of SVMs is again used in Chapter 12 for identifying potential gene markers that can distinguish between malignant and benign samples in different types of cancers. The proposed scheme consists of two phases. In the first, an ensemble of SVMs using different kernel functions is used for efficient classification. Thereafter, the signal-to-noise ratio statistic is used to select a number of gene markers, which is further reduced by using a multiobjective genetic algorithm-based feature selection method. Results are demonstrated on three publicly available data sets.

In Chapter 13, Maulik and Sarker develop a parallel algorithm for clustering gene expression data that exploits the property of symmetry of the clusters. It is based on a recently developed symmetry-based distance measure. The bottleneck for the application of such an approach for microarray data analysis is the large computational time. Consequently, Maulik and Sarker develop a parallel implementation of the symmetry-based clustering algorithm. Results are demonstrated for one artificial and four benchmark microarray data sets.

The last part of the book, dealing with topics related to systems biology, consists of Chapters 14–16. Jeong and Chen deal with the problem of gene prioritization in Chapter 14, which aims at achieving a better understanding of the disease process and to find therapy targets and diagnostic biomarkers. Gene prioritization is a new approach for extending our knowledge about diseases and potentially about other biological conditions. Jeong and Chen review the existing methods of gene prioritization and attempt to identify those that were most successful. They also discuss the remaining challenges and open problems in this area.

In Chapter 15, Bagchi discusses the various aspects of protein–protein interactions (PPI) that are one of the central players in many vital biochemical processes. Emphasis has been given to the properties of the PPI. A few basic definitions have been revisited. Several computational PPI prediction methods have been reviewed. The various software tools involved have also been reviewed.

Finally, in Chapter 16, Bhattacharyya and Bandyopadhyay study PPI networks in order to investigate the system level activities of the genotypes. Several topological properties and structures have been discussed and state-of-the-art knowledge on utilizing these characteristics in a system level study is included. A novel method of mining an integrated network, obtained by combining two types of topological properties, is designed to find dense subnetworks of proteins that are functionally coherent. Some theoretical analysis on the formation of dense subnetworks in a scale-free network is also provided. The results on PPI information of Homo Sapiens, obtained from the Human Protein Reference Database, show promise with such an integrative approach of topological analysis.

The field of biological informatics is rapidly evolving with the availability of new methods of data collection that are not only capable of collecting huge amounts of data, but also produce new data types. In response, advanced methods of searching for
useful regularities or patterns in these data sets have been developed. Computational intelligence, comprising a wide array of classification, optimization, and representation methods, have found particular favor among the researchers in biological informatics. The chapters dealing with the applications of computational intelligence and pattern analysis techniques in biological informatics provide a representative view of the available methods and their evaluation in real domains. The volume will be useful to graduate students and researchers in computer science, bioinformatics, computational and molecular biology, biochemistry, systems science, and information technology both as a text and reference book for some parts of the curriculum. The researchers and practitioners in industry, including pharmaceutical companies, and R & D laboratories will also benefit from this book.

We take this opportunity to thank all the authors for contributing chapters related to their current research work that provide the state of the art in advanced computational intelligence and pattern analysis methods in biological informatics. Thanks are due to Indrajit Saha and Malay Bhattacharyya who provided technical support in preparing this volume, as well as to our students who have provided us the necessary academic stimulus to go on. Our special thanks goes to Anirban Mukhopadhyay for his contribution to the book and Christy Michael from Aptara Inc. for her constant help. We are also grateful to Michael Christian of John Wiley & Sons for his constant support.

U. Maulik, S. Bandyopadhyay, and J. T. L. Wang

November, 2009
CONTRIBUTORS

Ajith Abraham, Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, Auburn, Washington

G. Agarwal, Molecular Biophysics Unit, Indian Institute of Science, Bangalore, India

Angshuman Bagchi, Buck Institute for Age Research, 8001 Redwood Blvd., Novato, California

Sanghamitra Bandyopadhyay, Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India

Chiranjib Bhattacharyya, Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India

Malay Bhattacharyya, Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India

Sourangshu Bhattacharya, Department of Computer Science and Automation, Indian Institute of Science Bangalore, India

S. Durga Bhavani, Department of Computer and Information Sciences, University of Hyderabad, Hyderabad, India

Kevin Byron, Department of Computer Science, New Jersey Institute of Technology, Newark, New Jersey

Miguel Cervantes-Cervantes, Department of Biological Sciences, Rutgers University, Newark, New Jersey
CONTRIBUTORS

Basabi Chakraborty, Department of Software and Information Science, Iwate Prefectural University, Iwate, Japan

Nagasuma R. Chandra, Bioinformatics Center, Indian Institute of Science, Bangalore, India

Jake Y. Chen, School of Informatics, Indiana University-Purdue University, Indianapolis, Indiana

Swagatam Das, Department of Electronics and Telecommunication, Jadavpur University, Kolkata, India

Alexandre G. de Brevern, Université Paris Diderot-Paris, Institut National de Transfusion Sanguine (INTS), Paris, France

D. C. Dinesh, Molecular Biophysics Unit, Indian Institute of Science, Bangalore, India

Jun Huan, Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, Kansas

Jieun Jeong, School of Informatics, Indiana University-Purdue University, Indianapolis, Indiana

Francesco Masulli, Department of Computer and Information Sciences, University of Genova, Italy

Ujjwal Maulik, Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

Anirban Mukhopadhyay, Department of Theoretical Bioinformatics, German Cancer Research Center, Heidelberg, Germany, on leave from Department of Computer Science and Engineering, University of Kalyani, India

B. K. Panigrahi, Department of Electrical Engineering, Indian Institute of Technology (IIT), Delhi, India

S. Bapi Raju, Department of Computer and Information Sciences, University of Hyderabad, Hyderabad, India

T. Sobha Rani, Department of Computer and Information Sciences, University of Hyderabad, Hyderabad, India

Stefano Rovetta, Department of Computer and Information Sciences, University of Genova, Italy

Giuseppe Russo, Sbarro Institute for Cancer Research and Molecular Medicine, Temple University, Philadelphia, Pennsylvania

Indrajit Saha, Interdisciplinary Centre for Mathematical and Computational Modeling, University of Warsaw, Poland

Anasua Sarkar, LaBRI, University Bordeaux 1, France
CONTRIBUTORS

Soumi Sengupta, Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India

Aaron Smalter, Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, Kansas

N. Srinivasan, Molecular Biophysics Unit, Indian Institute of Science, Bangalore, India

Jason T. L. Wang, Department of Computer Science, New Jersey Institute of Technology, Newark, New Jersey

Dongrong Wen, Department of Computer Science, New Jersey Institute of Technology, Newark, New Jersey
PART I

INTRODUCTION
1

COMPUTATIONAL INTELLIGENCE: FOUNDATIONS, PERSPECTIVES, AND RECENT TRENDS

Swagatam Das, Ajith Abraham, and B. K. Panigrahi

The field of computational intelligence has evolved with the objective of developing machines that can think like humans. As evident, the ultimate achievement in this field would be to mimic or exceed human cognitive capabilities including reasoning, understanding, learning, and so on. Computational intelligence includes neural networks, fuzzy inference systems, global optimization algorithms, probabilistic computing, swarm intelligence, and so on. This chapter introduces the fundamental aspects of the key components of modern computational intelligence. It presents a comprehensive overview of various tools of computational intelligence (e.g., fuzzy logic, neural network, genetic algorithm, belief network, chaos theory, computational learning theory, and artificial life). The synergistic behavior of the above tools on many occasions far exceeds their individual performance. A discussion on the synergistic behavior of neuro-fuzzy, neuro-genetic algorithms (GA), neuro-belief, and fuzzy-belief network models is also included in the chapter.

1.1 WHAT IS COMPUTATIONAL INTELLIGENCE?

Machine Intelligence refers back to 1936, when Turing proposed the idea of a universal mathematics machine [1,2], a theoretical concept in the mathematical theory of computability. Turing and Post independently proved that determining the decidability of mathematical propositions is equivalent to asking what sorts of sequences of a
finite number of symbols can be recognized by an abstract machine with a finite set of instructions. Such a mechanism is now known as a Turing machine [3]. Turing’s research paper addresses the question of machine intelligence, assessing the arguments against the possibility of creating an intelligent computing machine and suggesting answers to those arguments, proposing the Turing test as an empirical test of intelligence [4]. The Turing test, called the imitation game by Turing, measures the performance of a machine against that of a human being. The machine and a human (A) are placed in two rooms. A third person, designated the interrogator, is in a room apart from both the machine and the human (A). The interrogator cannot see or speak directly to either (A) or the machine, communicating with them solely through some text messages or even a chat window. The task of the interrogator is to distinguish between the human and the computer on the basis of questions he/she may put to both of them over the terminals. If the interrogator cannot distinguish the machine from the human then, Turing argues, the machine may be assumed to be intelligent. In the 1960s, computers failed to pass the Turing test due to the low-processing speed of the computers.

The last few decades have seen a new era of artificial intelligence focusing on the principles, theoretical aspects, and design methodology of algorithms gleaned from nature. Examples are artificial neural networks inspired by mammalian neural systems, evolutionary computation inspired by natural selection in biology, simulated annealing inspired by thermodynamics principles and swarm intelligence inspired by collective behavior of insects or micro-organisms, and so on, interacting locally with their environment causing coherent functional global patterns to emerge. These techniques have found their way in solving real-world problems in science, business, technology, and commerce.

Computational Intelligence (CI) [5–8] is a well-established paradigm, where new theories with a sound biological understanding have been evolving. The current experimental systems have many of the characteristics of biological computers (brains in other words) and are beginning to be built to perform a variety of tasks that are difficult or impossible to do with conventional computers. To name a few, we have microwave ovens, washing machines, and digital cameras that can figure out on their own what settings to use to perform their tasks optimally with reasoning capability, make intelligent decisions, and learn from the experience. As usual, defining CI is not an easy task. Bezdek defined a computationally intelligent system [5] in the following way:

“A system is computationally intelligent when it: deals with only numerical (low-level) data, has pattern recognition components, does not use knowledge in the AI sense; and additionally when it (begins to) exhibit i) computational adaptivity, ii) computational fault tolerance, iii) speed approaching human-like turnaround and iv) error rates that approximate human performance.”

The above definition infers that a computationally intelligent system should be characterized by the capability of computational adaptation, fault tolerance, high computational speed, and be less prone to noisy information sources. It also implies high computational speed and less error rates than human beings. It is true that a high computational speed may sometimes yield a poor accuracy in the results. Fuzzy
WHAT IS COMPUTATIONAL INTELLIGENCE?

Logic and neural nets that support a high degree of parallelism usually have a fast response to input excitations. Further, unlike a conventional production (rule-based) system, where only a single rule is fired at a time, fuzzy logic allows firing of a large number of rules ensuring partial matching of the available facts with the antecedent clauses of those rules. Thus the reasoning capability of fuzzy logic is humanlike, and consequently it is less error prone. An artificial neural network (ANN) also allows firing of a number of neurons concurrently. Thus it has a high computational speed; it usually adapts its parameters by satisfying a set of constraints that minimizes the error rate. The parallel realization of GA and belief networks for the same reason have a good computational speed, and their inherent information filtering behavior maintain accuracy of their resulting outcome.

In an attempt to define CI [9], Marks clearly mentions the name of the constituent members of the family. According to him:

“... neural networks, genetic algorithms, fuzzy systems, evolutionary programming and artificial life are the building blocks of computational intelligence.”

At this point, it is worth mentioning that artificial life is also an emerging discipline based on the assumption that physical and chemical laws are good enough to explain the intelligence of the living organisms. Langton defines artificial life [10] as:

“... an inclusive paradigm that attempts to realize lifelike behavior by imitating the processes that occur in the development or mechanics of life.”

Now, let us summarize exactly what we understand by the phrase CI. Figure 1.1 outlines the topics that share some ideas of this new discipline.

The early definitions of CI were centered around the logic of fuzzy sets, neural networks, genetic algorithms, and probabilistic reasoning along with the study of their synergism. Currently, the CI family is greatly influenced by the biologically inspired models of machine intelligence. It deals with the models of fuzzy as well as granular computing, neural computing, and evolutionary computing along with their interactions with artificial life, swarm intelligence, chaos theory, and other emerging paradigms. Belief networks and probabilistic reasoning fall in the intersection of traditional AI and the CI. Note that artificial life is shared by the CI and the physicochemical laws (not shown in Fig. 1.1).

Note that Bezdek [5], Marks [9], Pedrycz [11–12], and others have defined computational intelligence in different ways depending on the then developments of this new discipline. An intersection of these definitions will surely focus to fuzzy logic, ANN, and GA, but a union (and generalization) of all these definitions includes many other subjects (e.g., rough set, chaos, and computational learning theory). Further, CI being an emerging discipline should not be pinpointed only to a limited number of topics. Rather it should have a scope to expand in diverse directions and to merge with other existing disciplines.

In a nutshell, which becomes quite apparent in light of the current research pursuits, the area is heterogeneous as being dwelled on such technologies as neural networks,
fuzzy systems, evolutionary computation, swarm intelligence, and probabilistic reasoning. The recent trend is to integrate different components to take advantage of complementary features and to develop a synergistic system. Hybrid architectures like neuro-fuzzy systems, evolutionary-fuzzy systems, evolutionary-neural networks, evolutionary neuro-fuzzy systems, and so on, are widely applied for real-world problem solving. In the following sections, the main functional components of CI are explained with their key advantages and application domains.

1.2 CLASSICAL COMPONENTS OF CI

This section will provide a conceptual overview of common CI models based on their fundamental characteristics.

1.2.1 Artificial Neural Networks

Artificial neural networks [13–15] have been developed as generalizations of mathematical models of biological nervous systems. In a simplified mathematical model of the neuron, the effects of the synapses are represented by connection weights that modulate the effect of the associated input signals, and the nonlinear characteristic exhibited by neurons is represented by a transfer function, which is usually the sigmoid, Gaussian, trigonometric function, and so on. The neuron impulse is then computed as the weighted sum of the input signals, transformed by the transfer function. The learning capability of an artificial neuron is achieved by adjusting the weights in
CLASSICAL COMPONENTS OF CI

FIGURE 1.2 Architecture of an artificial neuron and a multilayered neural network.

accordance to the chosen learning algorithm. Most applications of neural networks fall into the following categories:

*Prediction.* Use input values to predict some output.

*Classification.* Use input values to determine the classification.

*Data Association.* Like classification, but it also recognizes data that contains errors.

*Data Conceptualization.* Analyze the inputs so that grouping relationships can be inferred.

A typical multilayered neural network and an artificial neuron are illustrated in Figure 1.2. Each neuron is characterized by an activity level (representing the state of polarization of a neuron), an output value (representing the firing rate of the neuron), a set of input connections, (representing synapses on the cell and its dendrite), a bias value (representing an internal resting level of the neuron), and a set of output connections (representing a neuron’s axonal projections). Each of these aspects of the unit is represented mathematically by real numbers. Thus each connection has an associated weight (synaptic strength), which determines the effect of the incoming input on the activation level of the unit. The weights may be positive or negative. Referring to Figure 1.2, the signal flow from inputs $x_1 \cdots x_n$ is considered to be unidirectional indicated by arrows, as is a neuron’s output signal flow ($O$). The neuron output signal $O$ is given by the following relationship:

$$O = f(\text{net}) = f \left( \sum_{j=1}^{n} w_j x_j \right)$$  \hspace{1cm} (1.1)

where $w_j$ is the weight vector and the function $f(\text{net})$ is referred to as an activation (transfer) function and is defined as a scalar product of the weight and input vectors

$$\text{net} = w^T x = w_1 x_1 + \cdots + w_n x_n$$ \hspace{1cm} (1.2)
COMPUTATIONAL INTELLIGENCE

where $T$ is the transpose of a matrix and in the simplest case the output value $O$ is computed as

$$O = f(\text{net}) = \begin{cases} 1 & \text{if } w^T x \geq \theta \\ 0 & \text{otherwise} \end{cases}$$ (1.3)

where $\theta$ is called the threshold level and this type of node is called a linear threshold unit.

The behavior of the neural network depends largely on the interaction between the different neurons. The basic architecture consists of three types of neuron layers: input, hidden and output layers. In feedforward networks, the signal flow is from input to output units strictly in a feedforward direction. The data processing can extend over multiple (layers of) units, but no feedback connections are present, that is, connections extending from outputs to inputs of units in the same or previous layers.

Recurrent networks contain feedback connections. Contrary to feedforward networks, the dynamical properties of the network are important. In some cases, the activation values of the units undergo a relaxation process such that the network will evolve to a stable state in which these activations do not change anymore. In other applications, the changes of the activation values of the output neurons are significant, such that the dynamical behavior constitutes the output of the network. There are several other neural network architectures (Elman network, adaptive resonance theory maps, competitive networks, etc.) depending on the properties and requirement of the application. The reader may refer to [16–18] for an extensive overview of the different neural network architectures and learning algorithms.

A neural network has to be configured such that the application of a set of inputs produces the desired set of outputs. Various methods to set the strengths of the connections exist. One way is to set the weights explicitly, using a priori knowledge. Another way is to train the neural network by feeding its teaching patterns and letting it change its weights according to some learning rule. The learning situations in neural networks may be classified into three distinct types. These are supervised, unsupervised, and reinforcement learning. In supervised learning, an input vector is presented at the inputs together with a set of desired responses, one for each node, at the output layer. A forward pass is done and the errors or discrepancies, between the desired and actual response for each node in the output layer, are found. These are then used to determine weight changes in the net according to the prevailing learning rule. The term ‘supervised’ originates from the fact that the desired signals on individual output nodes are provided by an external teacher. The best-known examples of this technique occur in the back-propagation algorithm, the delta rule, and perceptron rule. In unsupervised learning (or self-organization) an (output) unit is trained to respond to clusters of patterns within the input. In this paradigm, the system is supposed to discover statistically salient features of the input population [19]. Unlike the supervised learning paradigm, there is no a priori set of categories into which the patterns are to be classified; rather the system must develop its own representation of the input stimuli. Reinforcement learning is learning what to do—how to map