CHAPTER 41

Theoretical and Computational Models for Neuro, Genetic, and Neuro–Genetic Information Processing

Nik Kasabov, Lubica Benuskova

Knowledge Engineering and Discovery Research Institute, School of Information Technology, Auckland University of Technology, Auckland, New Zealand

CONTENTS

1. INTRODUCTION

Neuroscience, along with the information and mathematical sciences, has developed a variety of theoretical and computational models to model complex brain functions [1]. Along with this development, artificial neural networks—computational models that adopt principles

ISBN: 1-58883-042-X/$35.00

Handbook of Theoretical and Computational Nanotechnology
Edited by Michael Rieth and Wolfram Schommers
Volume X: Pages (1–38)

from the nervous system—have been developed into powerful tools for learning from data and generalization [2–6]. Artificial neural networks have been applied to a large amount of complex problems such as classification, prediction, diagnosis, and planning, not only for brain and molecular data modeling [7–9], but across all disciplines of science and engineering [10]. This chapter is a review of existing computational models of brain functions and of neural network models.

With the recent advancements in genetic research and with the successful sequencing of the human and other genomes, more information is becoming available about the interaction between brain functions and genes, about genes related to brain diseases (e.g., epilepsy [11], mental retardation [12], etc.), and about gene-based treatment of them [13]. It is well accepted now that brain functions are better understood and treated if information from molecular and neuronal level is integrated. For this reason, computational models that combine genetic and neuronal information are needed for modeling and prognosis. Such models are called neuro-genetic models. These models are further discussed in the section on future development with the anticipation that models of integrated molecular and neuronal information processing will be an important part of the theoretical and computational nanotechnology of the future.

2. NEURO-INFORMATION PROCESSING

2.1. Neuro-Information Processing in the Brain

Many functions are associated with neural cells (neurons) and with neural networks in the brain [14]. An ensemble of neurons operates in concert, thus defining the functions of a neural network that lead to such outcomes as perception of sound or brain disease (e.g., epilepsy [11] or mental retardation [12], etc.). At the level of the whole brain, complex dynamic interactions are observed, and cognitive functions are performed (e.g., learning, visual pattern recognition, speech and language processing, etc.).

It is estimated that there are from 10^{11} to 10^{12} of neurons in the human brain [14]. Three quarters of these neurons form a 4–6-mm-thick cerebral cortex that constitutes a heavily wrinkled brain surface. The cerebral cortex is thought to be a seat of cognitive functions, like perception, imagery, thinking, and so on. The cortex cooperates with evolutionary, older subcortical nuclei that are located in the middle of the brain, in and around the so-called brainstem (Fig. 1). Subcortical structures and nuclei comprise, for instance, thalamus, basal ganglia, hypothalamus, and dozens of other groups of neurons with more or less specific functions in modulating the state of the whole brain. For example, the input from all sensory organs comes to the cortex preprocessed in the thalamus. All the brain parts, either cortical or subcortical, are directly or indirectly heavily interconnected, thus forming a huge recurrent neural network.

Figure 1 shows a schematic functional division of the human cerebral cortex. One-third of the cortex is devoted to the processing of visual information in the primary visual cortex and higher-order visual areas. Association cortices take about one-half of the whole cortical surface. In the parietal-temporal-occipital association cortex, sensory, and language data are being associated. Memory and emotional data are associated in the limbic association cortex (internal and bottom portion of hemispheres). The prefrontal association cortex takes care of all associations, evaluation, planning ahead, and attention.

At the border between the frontal and parietal lobes, there is a somatic sensory cortex, which processes touch and other tactile signals (temperature, pain, etc.) from the body surface and interior. In front of this cortex, there is a primary motor cortex, which issues signals for voluntary muscle movements including speech. These signals are preceded by the preparation and anticipation of movements that takes place in the premotor cortex. The plan of actions and their consequences, as well as the inclusion and exclusion of motor actions into and from the overall goal of an organism, are performed within the prefrontal association cortex. Subcortical basal ganglia and cerebellum also participate in preparation and tuning of motor outputs in the sense of particular movements. For instance, the cerebellum executes routine automatized movements like walking, biking, driving, and so on. Language processing

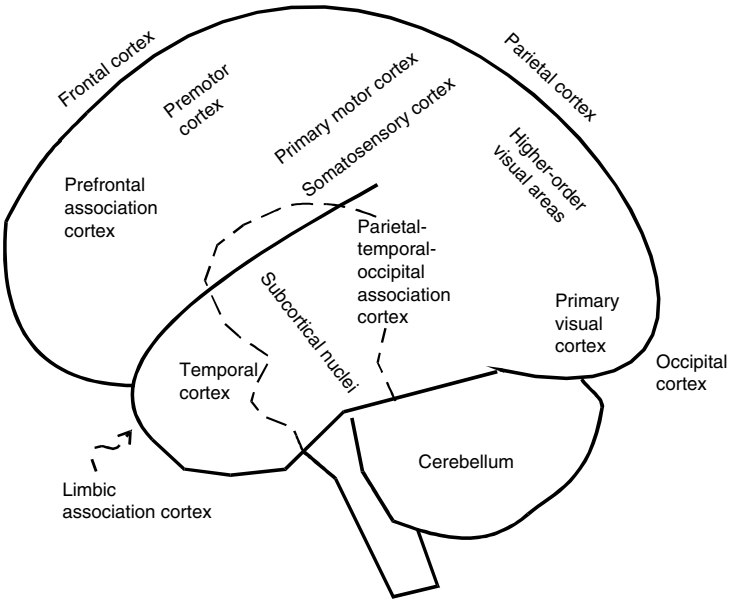

Figure 1. Gross anatomical and functional division of the human cerebral cortex. The same division applies for the other; in this case, the right hemisphere. A dashed curve marks the position of evolutionary older subcortical nuclei in the brainstem of the brain. Each of the depicted areas has far more subdivisions.

takes place within the temporal cortex, parietal-temporal-occipital association cortex, and frontal cortex. We want to point out that there are far more anatomical and functional subdivisions within each of the mentioned areas. Invaluable detailed information comes from the study of patients with mental deficits caused by injuries of particular brain areas. At present, noninvasive imaging techniques such as functional magnetic resonance (fMRI), positron emission tomography (PET), electroencephalogram (EEG), and others provide a rich source of information about the dynamics and organization of work within the healthy and diseased brain.

2.1.1. Processing of Signals by Neurons

Neuro-information processing in the brain depends not only on the properties of neural networks but also on the properties of processing units—neurons. A neuron (Fig. 2) receives and sends out electric and chemical signals. The place of signal transmission is a synapse. In the synapse, the signal can be nonlinearly strengthened or weakened. The strength or efficacy of synaptic transmission is also called a synaptic weight. One neuron receives and sends out signals through from 10^3 to 10^5 synapses. Dendrites (i.e., numerous bushy cell extensions) and soma constitute the input surface. Electrical signals transmitted by synapses can have a positive or negative electric sign. In the former case, we speak about excitatory synapses, and in the latter case about inhibitory synapses. Most of excitatory synapses are

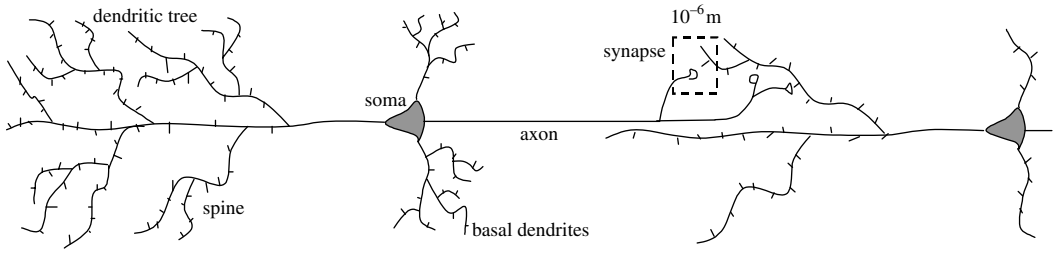

Figure 2. Schematic illustration of a neuron and its parts. There is a synapse at every dendritic spine. Synapses are also formed on the dendritic shafts and on the soma.

formed on spines, tiny extensions on dendrites. Spines are very important devices in relation to learning and memory.

When the sum of positive and negative contributions (signals) weighted by synaptic weights gets bigger than a particular value, called the excitatory threshold, a neuron fires; that is, it emits an output signal called a spike (Fig. 3). A spike is also called an action potential or a nerve impulse. The output frequency of a spike train (from 1 to 10^2 Hz) is proportional to the overall sum of positive and negative synaptic contributions. Spikes are produced at the initial segment of an axon (the only neuronal output extension). Then they quickly propagate along the axon toward other neurons within a network (propagation speed is 5–100 m/s). At its distant end, an axon makes thousands of branches, each of which is ended by a synaptic terminal (bouton).

2.1.2. Process of Synaptic Transmission

A synapse consists of a presynaptic terminal (bouton), synaptic cleft and postsynaptic membrane (Fig. 4). In the presynaptic terminal, there are dozens of vesicles filled with molecules of neurotransmitter (NT) ready to be released. When a presynaptic spike arrives into a terminal, calcium ions rush in and cause the fusion of vesicles with the presynaptic membrane. This process is also called exocytosis. Molecules of NT are released into the synaptic cleft (Fig. 4b) and diffuse toward the receptors within a postsynaptic membrane. Molecules of NT form a transient bond with the molecules of receptors. This causes the opening of ion channels associated with postsynaptic receptors. In the excitatory synapse, receptors are associated with sodium (Na^+) ion channels, and a positive excitatory postsynaptic potential (EPSP) is generated. In the inhibitory synapse, receptors are associated with chlorine (Cl^-) ion channels, and a negative inhibitory PSP is generated. Eventually, NT releases its bond with receptors and diffuses back to the presynaptic membrane and out of the synaptic cleft. Special molecular transporters within a presynaptic membrane take molecules of NT back inside the terminal, where they are recycled into new vesicles. This process is called a reuptake of NT. The whole synaptic transmission lasts for about 1 MS. Such a synapse is called a chemical synapse, because the transmission of an electric signal is performed in a chemical way.

The PSP, either excitatory or inhibitory, has some amplitude and duration. The amplitude and duration of the PSP depend on the number of activated receptor-ion channels and on how long they stay open, which may be miliseconds, tens of miliseconds, or hundreds of miliseconds. The duration of channel opening depends on the number of released NT molecules and on the type of receptors that are associated with ion channels. The amplitude of PSP also depends on the electric input resistance for ions, which in turn depends on the size and shape of a postsynaptic spine and dendrites and on the distance of synapse from soma. For instance, a short and stubby dendritic spine has a much smaller electric resistance than a long and thin spine. All these pre- and postsynaptic factors determine the weight (strength, efficacy) of a particular synapse.

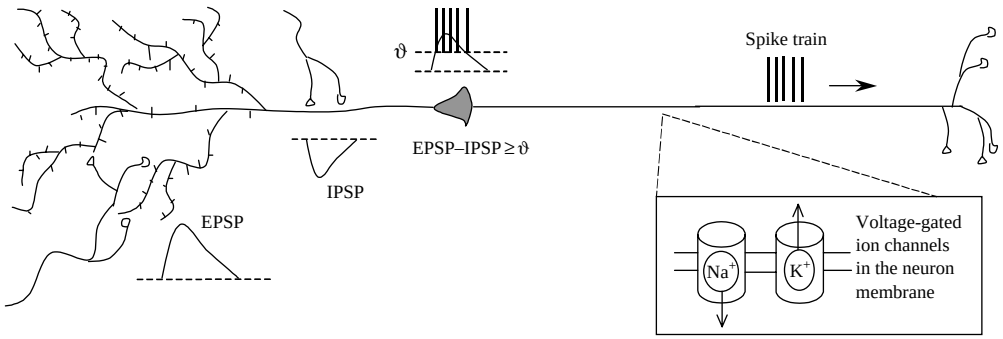

Figure 3. Electric synaptic potentials and axonal ion channels responsible for spike generation and propagation. EPSP = excitatory postsynaptic potential, IPSP = inhibitory postsynaptic potential, ϑ = excitatory threshold for an output spike generation.

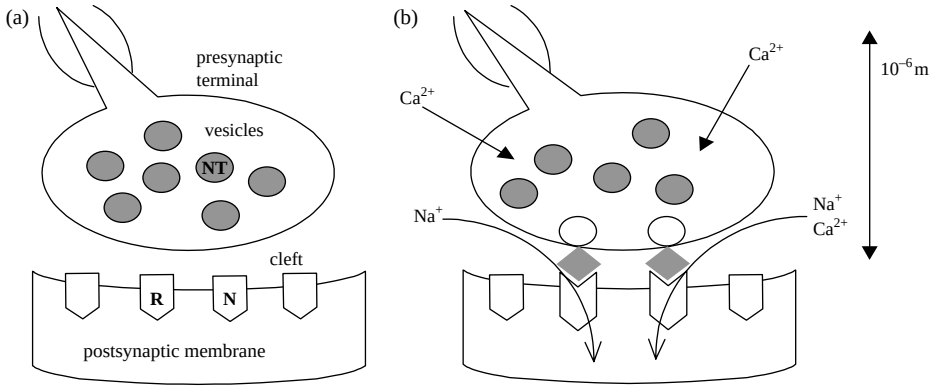

Figure 4. Scheme of synaptic transmission. (a) A synapse is ready to transmit a signal. (b) Transmission of electric signal in a chemical synapse. NT = neurotransmitter, R = AMPA-receptor-gated ion channel for sodium, N = NMDA-receptor-gated ion channel for sodium and calcium.

Within a postsynaptic membrane, there are also kinds of receptors that are not associated with an ion channel, but instead with an enzyme. When the overall amount of released NT reaches some critical concentration, these receptor–enzyme complexes activate particular cytoplasmatic enzymes, the so-called second messengers. Second messengers trigger chains of various biochemical reactions that may lead to a change in synaptic weight, or even to transient changes in gene expression leading to alteration in biomolecular synthesis of receptors, neurotransmitters, and enzymes. Thus, second messengers may act locally within a synapse itself, or they may activate further (third, and so on) messengers that carry the message to the genome of a neuron, thus causing a change in its biochemical machinery related to signal processing. Therefore, it is now widely accepted that the activity of a neuron itself influences its processing of information, and even its life itself, whether it survives or not.

2.1.3. Learning Takes Place in Synapses

For major discoveries in the field of synaptic mechanisms of learning, the 2000 Nobel prize for medicine went to the neuroscientists Eric R. Kandel and Paul Greengard. The third laureate, Arvid Carlsson, got the prize for discoveries of actions of the neurotransmitter dopamine. At present, it is widely accepted that learning is accompanied by changes in the synaptic weights in cortical neural networks [14]. Changes of synaptic weights are also called synaptic plasticity. In 1949, the Canadian psychologist Donald Hebb formulated a universal rule for these changes: "When an axon of cell A excites cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells so that A' s efficiency as one of the cells firing B is increased" [15].

In cerebral cortex and in hippocampus of humans and animals, learning takes place in excitatory synapses formed on dendritic spines that use glutamate as their neurotransmitter. In the regime of learning, glutamate acts on specific postsynaptic receptors, the so-called NMDA receptors (*N*-methyl-D-aspartate). NMDA receptors are associated with ion channels for sodium and calcium (see Fig. 5). The influx of these ions into spines is proportional to the frequency of incoming presynaptic spikes. Calcium acts as a second messenger, thus triggering a cascade of biochemical reactions that lead either to the long-term potentiation of synaptic weights (LTP) or to the long-term depression (weakening) of synaptic weights (LTD). In experimental animals, it has been recorded that these changes in synaptic weights can last for hours, days, or even weeks and months, up to a year. Induction of such long-term synaptic changes involves transient changes in gene expression [16].

A subcellular switch between LTD and LTP is the concentration of calcium within spines [17]—an LTD/LTP threshold. In turn, the intraspine calcium concentration depends on the intensity of synaptic stimulation; that is, on the frequency of presynaptic spikes: more presynaptic spikes means more glutamate within synaptic cleft. Release of glutamate must coincide with a sufficient depolarization of the postsynaptic membrane to remove the magnesium

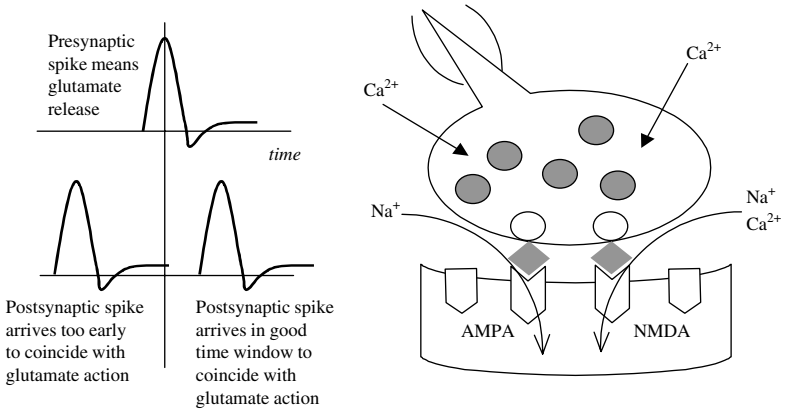

Figure 5. Spike timing dependent plasticity (STDP). Ions of sodium and calcium enter the postsynaptic spine through the NMDA-receptor gated ion channels. When postsynaptic neuron fires, the postsynaptic spikes back-propagate to spines. When their timing coincides with glutamate action, more calcium enters spines, and thus there is a bigger chance to achieve LTP.

block of the NMDA receptor. The greater the depolarization, the more ions of calcium enter the spine. Postsynaptic depolarization is primarily achieved via AMPA receptors; however, recently, a significant role of backpropagating postsynaptic spikes has been pointed out (see, e.g., Fig. 5) .[18]. Calcium concentrations below or above the LTD/LTP threshold switch on different enzymatic pathways that lead either to LTD or LTP, respectively. However, the current value of the LTD/LTP threshold (i.e., the properties of these two enzymatic pathways) can be influenced by levels of other neurotransmitters, by an average previous activity of a neuron, and possibly by other biochemical factors as well. This phenomenon is called metaplasticity, a plasticity of synaptic plasticity [19]. Dependence of the LTD/LTP threshold on different postsynaptic factors is the subject of the Bienenstock, Cooper, and Munro (BCM) theory of synaptic plasticity [20] (for a nice overview see, e.g., Ref. [21]). The BCM theory of synaptic plasticity has been successfully applied in computer simulations to explain experience-dependent changes in the normal and ultrastructrally altered brain cortex of experimental animals [22, 23].

The ease with which LTD and LTP can be evoked in the developing and the adult brain are not the same. One of the factors responsible for this difference may be the genetically programmed difference in NMDA receptor composition [24]. The NMDA receptor is made up of an NR1 subunit, which is obligatory for channel function, and a selection of developmentally and regionally regulated NR2 subunits (A–D). For example, the glutamate-evoked positive current has a longer duration in receptors containing NR2B subunits than in those containing NR2A subunits. The proportion of NR2B subunits is higher in young animals than in adults, which may account for the greater degree of synaptic plasticity seen in young animals.

2.1.4. Summary of Cortical Plasticity

Experimental developmental neuroscience brings abundant evidence that in the developing brain strong genetic programs determine the overall pattern of hierarchical organization and connections between brain areas. Nature provides an anatomical and physiological "scaffold" in the sense of a framework that outlines the structures to be formed it later. A newborn brain is not a "tabula rasa;" however, nurture and experience can shape it dramatically [14]. For instance, congenitally deaf kittens can develop almost normal auditory cortical maps and normal hearing after experiencing sounds through implanted artificial chochlea [25].

Information needed to specify precise subtle differentiation of neurons and subtle patterns of interneuronal connectivity far surpasses that which is contained in genetic programs. Instead of it, genetic programs provide for a vast overproduction of abundant and redundant synaptic connections in the developing brain. Individual differences in experience cause selective pruning of the majority of these synapses. Only synapses that mediate the genuine

individual experience of an individual remain. Process of experience-dependent synaptic pruning during early stages of brain development may constitute the basis of brain and mind individuality. Early developmental overproduction of redundant synapses lasts only for some time after birth. Time windows, that is, beginnings and durations, are different for different brain systems. They also differ for different animal species. In general, what lasts weeks and months in rats, cats, and monkeys usually lasts for years in humans.

Later in adolescence and adulthood, a new experience is "burned" into the brain not by a selective pruning of existing redundant synapses but, instead, by a selective creation of new connections and by changing the efficacies of the synaptic transmission of existing connections. This does not mean that synapses cannot be removed as a result of experience later in life, or that new connections cannot be created because of experience early in development. They can, but the prevailing process of experience-dependent cortical plasticity at different ages is not the same. Some cortical areas retain the capacity of synaptic plasticity for their whole course of life. These are association cortical areas, highest-order sensory areas, and premotor and emotional cortical areas of the brain. The capacity of the brain to be plastic and change its microstructure as a result of experience is of profound importance for discovering the rules of the mind/brain relation.

2.2. Coding and Representation of Information in the Brain

2.2.1. Principles of Information Representation in the Brain

The first principle of representation (and coding) of information in the brain is redundancy. Redundancy means that every piece of information (meant in any sense) is stored, transmitted, and processed by a redundant number of neurons and synapses so that it does not become lost when neural networks undergo damage; for instance, as a result of aging. When neural networks get damaged, their performance does not drop down to zero abruptly, like in a sequential computer, but instead it degrades gracefully. Computer models of neural networks also confirm the idea that a degradation of performance with the loss of neurons and synapses is neither total nor linear, but instead, neural networks can withstand quite substantial damage and still perform well.

Next, the contemporary view on the nature of neural representation is such that information (in the sense of content or meaning) is represented by place in the cortex (or, in general, in the brain). However, this placement is a result of both genetically determined anatomical framework and shaping by input through the process of experience-dependent plasticity. For instance, a sound pattern for the word "apple" is represented (coded) in the auditory areas of the temporal cortex. It is represented as a spatial pattern of active versus inactive neurons. This neural representation is associated (connected) through synaptic weights, with the neural representation of a visual image of apple in the parietal cortex, with the neural representation of an apple odour in the olfactory cortex, with memories on the grandma garden and facts about apples being represented in some other areas of the brain, and so forth. Neural representations (i.e., distributions or patterns of active neurons) within particular areas and their associations between areas appear as a result of learning (synaptic plasticity). Different objects are represented by means of different patterns or by distributions of active neurons within cortical areas. Thus, each object is represented by means of distributed networks of neurons, whereas the activity of neurons in these networks is also distributed.

Perception is an active process. Instead of a passive processing of all electrical signals that may or may not arrive from hierarchically lower processing levels, cortical neural networks should be able to use fragments of activity patterns to fill in the gaps, and thus quickly re-create the whole neural representation. The filling-in process can be nicely modeled by means of model neural networks; for instance, those of the Hopfield type [26] (Fig. 6).

Explicit neural representations (patterns of activities) are implicitly stored in the matrix of synaptic weights through which neurons in the network are interconnected. The weight distribution of storing a particular object representation is created as a result of an experience-dependent synaptic plasticity (learning). When a sufficiently large portion of this neural representation is activated from outside the network, few electric signals along all the

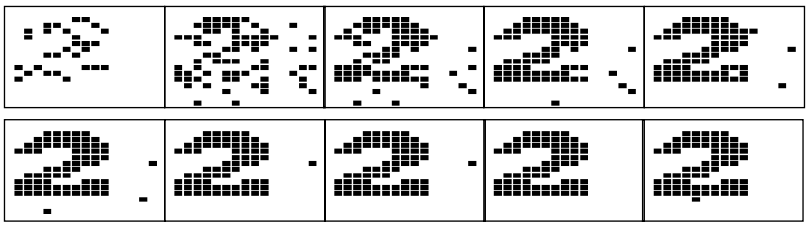

Figure 6. Illustration of spontaneous re-creation of neural representation after only few input impulses (figure in the uppermost left corner). Black pixel represents a firing neuron, whereas blank pixel represents a silent neuron. Between each pattern of activity from left to right (1 ms time frame), neurons in the network exchange only one impulse. Thus, basically, after exchanging only two to three spikes, the memory pattern is re-created. Network can reverberate the restored memory pattern until a different external input arrives.

synapses in the network quickly switch on the correct remaining neurons in the representation. Neural representations in the sense of patterns of activity have a holistic character. Patterns of activity are being recalled (restored) as a whole.

2.2.2. Problem of Neural Coding: The Brain is Fast, Neurons are Slow

Neurons within and between different brain areas send messages to each other by means of output spikes. Transmission time at one synapse takes about 1 ms. Neural representations of objects "communicate with each other;" that is, neurons within and between these representations send messages to each other. At present, the nature of these messages, that is, the nature of a neural code, is a mystery. It is because the recognition of sensory objects takes only about 200 ms [27], although it is performed by tens of different cortical areas in which the information processing involves billions of neurons. Even when taking into account the parallel processing of information, one hierarchical area of the cortex is left with only about 10–30 ms to perform its share of processing, the results of which are sent to hierarchically higher areas. Therefore, recently, a number of hypotheses have emerged based both on theoretical and experimental investigations, dealing with this problem.

These hypotheses can be divided into two categories: spike-timing hypotheses and rate code hypotheses [28].

2.2.2.1. Coding Based on Spike Timing

2.2.2.1.1. Reverse Correlation The first option is that the information about the salience of the object feature is encoded in the exact temporal structure of the output spike train. Let us say that two neurons fire three spikes within 20 ms. The first neuron fires a spike train with this temporal structure | ||, and the second neuron with this temporal structure |||. By means of the techniques of reverse correlation, it is possible to calculate which stimulus exclusively causes which temporal pattern of which neuron. The main proponents of this theory are Bialek and his coworkers, who have made its successful verification in the fly visual system [29].

2.2.2.1.2. Time to the First Spike Let, at time instant t_0, a stimulus arrive to the neural network. Neurons that fire the first (let us say in a window of 10 ms) carry the information about the stimulus features. The rest of neurons and the rest of impulses are ignored. This theory is favored by S. Thorpe [30].

2.2.2.1.3. Phase Information about the presence of the feature is encoded in the phase of neuron's impulses with respect to the reference background oscillation. Either they are in a phase lead or in a phase lag. The information can also depend on the magnitude of this phase lead (lag). This coding is preferred by people investigating hippocampus [31].

2.2.2.1.4. Synchronization Populations of neurons that represent features belonging to one object can be bound together by synchronous firing. Such a binding of features was discovered in the laboratory of W. Singer in the cat visual cortex [32]. It was also detected in the human cortex during perception [27].

2.2.2.2. The Rate Code

2.2.2.2.1. (*Temporal*) *Average Rate* This coding is still being considered for stationary stimuli that last up to around 500 ms or longer, so neurons have enough time to count (integrate) impulses over long time. Neurons that have the highest frequency signalize the presence of the relevant feature.

2.2.2.2.2. Rate as a Population Average An average frequency is not calculated as a temporal average but, rather, as a population average. One feature is represented by a population of many (10,000) neurons, for instance, in one cortical column. On the presence of a feature, most of the neurons are activated. When we calculate the number of spikes in a 10-ms window of all these neurons and divide this number by the number of neurons activated, we will get approximately the same average frequency as when calculating a temporal average rate of any of these neurons (provided they all fire with the same average rate). This idea has been thoroughly investigated in Ref. [33]. It has been also shown, that by means of population averaging we can get a reliable calculation of the neurons' average rates even in the case when they have a Poisson-like distribution of output spikes. Populations that relay the highest number of spikes signal the presence of the relevant feature.

On seeing all these options, one gets inevitably confused. Every hypothesis has some experimental support; thus, which one is correct, and which one is not? Is it possible that different brain areas use different codes? In one area it is the spike-timing code and in another area it is a rate code. Or does the brain switch between different codes according to the task to be performed? These possibilities remain to be explored.

Let us reason now, However, that all these hypotheses do not have to be, in fact, mutually exclusive. For instance, let us imagine that a population of neurons representing one feature of an object has a synchronized firing. Those neurons that have the highest output rates have also the shortest time to the first spike. Their synchronization guarantees that all of them have the same phase difference with respect to the background oscillation. Actually, the background oscillation might have helped to synchronize them. The information about the intensity of a feature in the stimulus (feature salience) is encoded in the average frequency of the whole population of neurons (within 10 ms) and is relayed on to the next population of neurons in the processing hierarchy. An actual temporal pattern of spikes relayed by one population of neurons indeed corresponds only to one stimulus. Thus, maybe the mentioned options are only different angles under which we can see the same process.

3. ARTIFICIAL NEURAL NETWORKS

Artificial neural networks (ANNs) are massivelly parallel computational systems inspired by biological neural networks. They can have different architectures and properties of their processing elements. Illustration of a general architecture of an ANN is shown in Fig. 7.

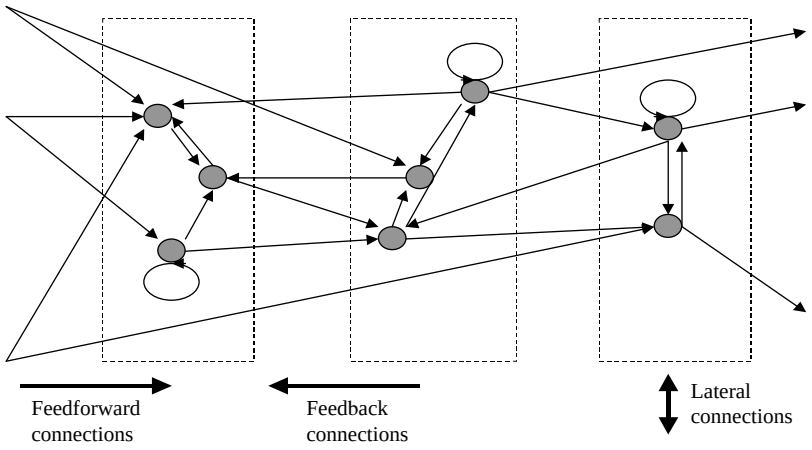

Figure 7. Formal neurons in ANNs are organized into layers (dashed rectangles). Connections between neurons within an ANN can be of the feed-forward, feedback, or lateral type.

There are different variations derived from this general architecture; for instance, there can be only feed forward connections between layers, or there can be only feedback connections in the one and only layer with the feed forward input, or there can be feedback connections allowed only between some layers, and so on.

In general, ANNs perform mappings of vectors from m-dimensional input space on vectors from n-dimensional output space. We can also say that ANNs learn how to associate vectors from one space to vectors from another space. Firts of all, there are many models of ANNs, each having different particular properties. However, they all are adaptible systems that can reorganize their internal structure based on their experience in the process called training or learning. ANNs are very often referred to as connectionist systems [34].

Basic processing elements (processing units) of ANNs are formal neurons based on the rate code hypothesis of a neural code according to which the information being sent from one neuron to another is encoded in the output rate of its spike train. Thus, the input–output function of an ith element of an ANN is

$$o_i(t) = f\left[\sum_{\forall k} w_{ik}(t)\, x_k(t)\right] \tag{1}$$

where $o_i, x_k, w_{ik} \in \Re$ are the output rate of the ith neuron, input rate of the kth input, and the synaptic weight between the kth and ith element, respectively. The function f is the so-called activation or transfer function of a neuron. It can be linear or nonlinear, continuous and differentiable, or binary, depending on which ANN model we are working with.

Different models of ANNs differ with respect to their architecture, transfer functions of their elements, and the rules used to modify the weights between neurons in the process of learning [5].

3.1. General Classification Scheme

Let us introduce a general classification scheme of ANNs that will lead to explanation of their properties, capabillities, and drawbacks [34].

3.1.1. General Notions

Most of the known ANN training algorithms are influenced by a concept introduced by Donald Hebb [15]. He proposed a model for unsupervised learning in which the synaptic strength (weight) is increased if both the source and the destination neurons become simultaneously activated. It is expressed as

$$w_{ij}(t+1) = w_{ij}(t) + c\, o_i(t)\, o_j(t) \tag{2}$$

where $w_{ij}(t)$ is the weight of the connection between the ith and jth neuron in the network at the moment t, and o_i and o_j are the output signals of the neurons i and j at the same moment t. The weight $w_{ij}(t+1)$ is the adjusted weight at the next time time moment $(t+1)$. Usually some kind of weights normalization is applied after each adjustment to prevent growth to infinity.

In general terms, a connectionist system $\{S, W, P, F, J, L\}$ that is defined by its structure S, its connection weights W, its parameter set P, its function F, its goal function J, and a learning procedure L, learns if the system optimizes at least part of its structure S and its function F when observing events $z1, z2, z3, \ldots$ from the problem space Z.

Through a learning process, the system improves its reaction to the observed events and captures useful information that may be later represented as knowledge.

The goal of a learning system is defined as to find the minimum of an objective function $J(S)$ named "the expected risk function" [2, 6]. The function $J(S)$ can be represented by a loss function $Q(Z, S)$ and an unknown probability distribution $\Pi(Z)$.

Most of the learning systems optimize a global goal function over a fixed part of the structure of the system. In NN, this part is a set of predefined and fixed number of connection weights (i.e., the set number of elements in the set W). As an optimization procedure, some known statistical methods for global optimization are applied [6]; for example, the gradient

descent method. Final structure S is expected to be globally optimal (i.e., optimal for data drawn from the whole-problem space Z).

In the case of a changing structure S and changing (e.g., growing) part of its connections W, where the input stream of data is continuous and its distribution is unknown, the goal function could be expressed as a sum of local goal functions J', each one optimized in a small subspace of $Z' \subset Z$ as data are drawn from this subspace. While the learning process is taking place, the number of dimensions of the problem space Z may also change over time. These scenarios would be reflected in different models of learning, as is explained next.

3.1.2. Different Models of Learning in Connectionist Systems

There are many methods for learning that have been developed for connectionist architectures (for a review, see Ref. [1]). It is difficult and quite risky to try to put all the existing methods into a clear classification structure (which should also assume "slots" for new methods), but it is necessary to define the scope of the evolving connectionist systems paradigm.

A connectionist classification scheme is graphically represented in Table 1. and is explained below. On the one hand, this scheme is a general one, as it is valid not only for connectionist learning models but also for other learning paradigms; for example, evolutionary learning, case-based learning, analogy-based learning, and reasoning. On the other hand, the scheme is not a comprehensive one, as it does not present all existing connectionist learning models. It is only a working classification scheme needed for the purpose of this work.

A (connectionist) system that learns from observations $z1, z2, z3, \ldots$ from a problem space Z can be designed to perform learning in different ways. The classification scheme in Table 1 outlines the main questions and issues and their alternative solutions when constructing a connectionist learning system. Now, let us offer an explanation of individual issues and alternatives.

1. *What space has the learning system developed in?*
   a. *The learning system has developed in the original problem space Z*: The structural elements (nodes) of the connectionist learning system are points in the d-dimensional original data space Z. This is the case in some clustering and prototype learning systems. One of the problems here is that if the original space is high dimensional (e.g., a 30,000 gene expression space) it is difficult to visualize the structure of the system and observe some important patterns. For this purpose, special visualization techniques, such as principle component analysis (PCA), or Sammon mapping, are used to project the system structure S into a visualization space V.
   b. *The learning system has developed in its own machine space M*: The structural elements (nodes) of the connectionist learning system are created in a system (machine) space M, different from the d-dimensional original data space Z. An example is the self-organizing map (SOM) neural network [35]. SOMs develop in a two-, three-, or more dimensional topological spaces (maps) from the original data.
2. *Is the space open?*
   a. An open problem space is characterized by unknown probability distribution $P(Z)$ of the incoming data and a possible change in its dimensionality. Sometimes the dimensionality of the data space may change over time, involving more, or fewer, dimensions; for example, adding new modalities to a person identification system.
   b. A closed problem space has a fixed dimensionality and either a known distribution of the data or a distribution that can be approximated in advance through statistical procedures.
3. *Is learning online?*
   a. *Batch-mode, offline learning*: In this case, a predefined learning (training) set of data $P = \{z1, z2, \ldots, zp\}$ is learned by the system through propagating this data set several times through the system. Each time, the system optimizes its structure W, based on the average value of the goal function over the whole data set P. Many traditional algorithms, such as the back-propagation algorithm, use this type of learning [36, 37].

Table 1. A general classification scheme of ANN models.

Issue	Alternatives
What space has the learning system developed in?	(a) The learning system has developed in the original data space Z (b) The learning system has developed in its own machine space M
Is the space open?	(a) An open data space (b) A closed space that has a fixed dimensionality
Is learning online?	(a) Batch-mode, offline learning (b) Online, pattern mode, incremental learning (c) Combined online and offline learning
Is learning lifelong?	(a) Single-session learning (b) Lifelong learning
Are there desired output data and in what form?	(a) Unsupervised learning (b) Supervised learning (c) Reinforcement learning (d) Combined learning
Is evolution of populations of individuals over generations involved in the learning process?	(a) Individual, development-based learning (b) Evolutionary learning, (population-based learning over generations) (c) Combined
Is the structure of the learning system of a fixed size, or it is evolving?	(a) Fixed-size structure (b) Dynamically changing structure
How structural modifications in the learning system partition the problem space?	(a) Global partitioning (global learning) (b) Local partitioning (local learning)
What knowledge representation is facilitated in the learning system?	(a) No explicit knowledge representation is facilitated in the system (b) Memory-based knowledge (c) Statistical knowledge (d) Analytical knowledge (e) Symbolic knowledge (f) Combined knowledge (g) Meta-knowledge (h) "Consciousness" (the system knows about itself) (i) "Creativity" (e.g., generating new knowledge)
If symbolic knowledge is represented in the system, of what type is it?	(a) Propositional rules (b) First-order logic rules (c) Fuzzy rules (d) Semantic maps (e) Schemata (f) Meta-rules (g) Finite automata (h) Higher-order logic
If the systems' knowledge can be represented as fuzzy rules, what type of fuzzy rules are they?	(a) Zadeh-Mamdani fuzzy rules (b) Takagi-Sugeno fuzzy rules (c) Other types of fuzzy rules (e.g., type 2 fuzzy rules)
Is learning active?	(a) Active learning in terms of data selection, filtering and searching for relevant data (b) Passive learning—the system accepts all incoming data

b. *Online, pattern mode, incremental learning*: Online learning is concerned with learning each data example separately as the system operates (usually in a real time), and the data might exist only for a short time. After observing each data example, the system makes changes in its structure (the W parameters) to optimize the goal function J. A typical scenario for online learning is when data examples are drawn randomly from a problem space and fed into the system one by one for training. Although there are chances of drawing the same examples twice or several times, this

is considered as a special case, in contrast to the offline learning, when one example is presented to the system many times as part of the training procedure. Methods for online learning in NN are studied in Refs. [38–40]. In Ref. [41], a review of some statistical methods for online learning, mainly gradient descent methods applied on fixed-size connectionist structures, is presented.

Some other types of learning, such as incremental learning and lifelong learning, are closely related to online learning.

Incremental learning is the ability of a NN to learn new data without fully destroying the patterns learned from old data and without the need to be trained on either old or new data. According to Schaal and Atkeson [42], incremental learning is characterized by the following features:

- Input and output distributions of data are not known, and these distributions may change over time
- The structure of the learning system W is updated incrementally
- Only limited memory is available, so that data have to be discarded after they have been used.

Online learning, incremental learning, and lifelong learning are typical adaptive learning methods. Adaptive learning is aiming at solving the well-known stability/plasticity dilemma, which means that the system is stable enough to retain patterns learned from previously observed data while being flexible enough to learn new patterns from new incoming data.

Adaptive learning is typical for many biological systems and is also useful in engineering applications, such as robotic systems and process control. Significant progress in adaptive learning has been achieved as a result of the adaptive resonance theory (ART) [4] and its various models, which include unsupervised models (ART1–3, FuzzyART) and supervised versions (ARTMAP, FuzzyARTMAP-FAM) [43].

c. *Combined online and offline learning*: In this mode the system may work for some of the time in an online mode, after which it switches to offline mode, and so forth. This is often used for optimization purposes, when a small "window" of data from the continuous input stream can be kept aside, and the learning system that works in an online mode can be locally or globally optimized through offline learning on this window of data through "window-based" optimization of the goal function $J(W)$.

4. *Is the learning process lifelong?*
   a. *Single-session learning*: The learning process happens only once over the whole set P of available data (and it may even take many iterations during training). After that, the system is set in operation and never trained again. This is the most common learning mode in many existing connectionist methods.
   b. *Lifelong learning*: Lifelong learning is concerned with the ability of a system to learn from continuously incoming data in a changing environment during its entire existence. Growing as well as pruning may be involved in the lifelong learning process, as the system needs to restrict its growth while always maintaining a good learning and generalization ability.

5. *Are there desired output data, and in what form are they available?* The availability of examples with desired output data (labels) that can be used for comparison with what the learning system produces on its outputs defines four types of learning.
   a. *Unsupervised learning*: There are no desired output data attached to the examples $z1, z2, z3, \ldots$. The data are considered as coming from an input space Z only.
   b. *Supervised learning*: There are desired output data attached to the examples $z1, z2, z3, \ldots$. The data are considered as coming in (x, y) pairs from both an input space X and an output space Y that collectively define the problem space Z (Fig. 8). The connectionist learning system associates data from the input space X with data from the output space Y.

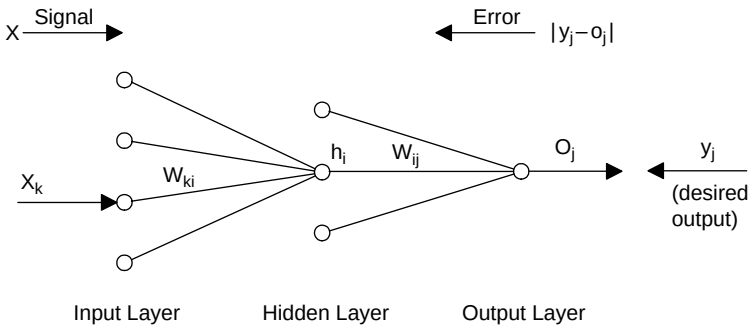

Figure 8. Illustration of a multilayer perceptron and supervised learning by means of the error back-propagation algorithm. Back-propagated error is proportional to the difference between the desired and actual output.

c. *Reinforcement learning*: In this case, there are no exact desired output data, but some hints about the "goodness" of the system reaction are available. The system learns and adjusts its structural parameters from these hints. In many robotic systems, a robot learns from the feedback from the environment that may be used as a qualitative indication of the correct movement of the robot.
d. *Combined learning*: This is the case in which a connectionist system can operate in more than one of the above learning modes.

6. *Is evolution of populations of individuals over generations involved in the learning process?*
   a. *Individual, development-based learning*: A system is developed independently and is not part of a development process of a population of individual systems.
   b. *Evolutionary learning population-based learning over generations*: Here, learning is concerned with the performance of not only an individual system but also, a population of systems that improve their performance through generations. The best individual system is expected to emerge—to evolve from such populations.

      Evolutionary computation (EC) methods, such as genetic algorithms (GA), have been widely used for optimizing ANN structures [44–46]. Such ANNs are called evolutionary neural networks. They use ideas from Darwinism. Most of the evolutionary computation methods developed so far assume that the problem space is fixed; that is, the evolution takes place within a predefined problem space, and this space does not change dynamically. Therefore, these methods do not allow for modeling real, online adaptation. In addition, they are very time consuming, which also prevents them from being used in real-world applications.

7. *Is the structure of the learning system of a fixed size, or it is evolving?*

   Here we will refer again to the bias/variance dilemma (see, e.g., Refs. [4, 47, 48]). With respect to an ANN structure, the dilemma states that if the structure is too small, the ANN is biased to certain patterns, and if the NN structure is too large, there are too many variances that may result in overtraining, poor generalization, and so forth. To avoid this problem, an ANN structure should change dynamically during the learning process, thus better representing the patterns in the data and the changes in the environment.
   a. *Fixed-size structure*: This type of learning assumes that the size of the structure S is fixed (e.g., number of neurons, number of connections) and that through learning, the system changes some structural parameters (e.g., W—the values of connection weights). This is the case in many multilayer perceptron ANNs trained with the back-propagation algorithm [2, 6, 36, 37, 49–51].
   b. *Dynamically changing structures*: According to Heskes and Kappen [52], there are three different approaches to dynamically changing structures: constructivism, selectivism, and a hybrid approach.

      Connectionist constructivism is about developing ANNs that have a simple initial structure and that grow during its operation through inserting new nodes. This theory is supported by biological facts (see Ref. [40]). The insertion can be controlled

either by a similarity measure of input vectors or by the output error measure, or by both, depending on whether the system performs an unsupervised or supervised mode of learning. A measure of difference between an input pattern and already stored patterns is used for deciding whether to insert new nodes in the adaptive resonance theory models ART1 and ART2 [4] for unsupervised learning. There are other methods that insert nodes based on the evaluation of the local error. Such methods are growing cell structure and growing neural gas [39]. Other methods insert nodes based on a global error to evaluate the performance of the whole ANN. One such method is the Cascade–Correlation Method [53]. Methods that use both similarity and output error for node insertion are used in Fuzzy ARTMAP [43] and also in EfuNN (Evolving Fuzzy NN) [34].

Connectionist selectivism is concerned with pruning unnecessary connections in a NN, which starts its learning with many, in most cases redundant, connections [54, 55]. Pruning connections that do not contribute to the performance of the system can be done by using several methods: Optimal-Brain Damage [56], Optimal Brain Surgeon [41], and Structural Learning with Forgetting [57].

8. *How do structural modifications affect the partitioning of the problem space?* When a connectionist model is created, either in a supervised or in an unsupervised mode, the nodes and the connections partition the problem space Z into segments. Each segment of the input sub-space is mapped onto a segment from the output sub-space in case of a supervised learning. The partitioning in the input sub-space imposed by the model can be one of the following two types:
   a. *Global partitioning (global learning)*: Learning causes global partitioning of the space. Partitioning hyperplanes can be modified either after every example is presented (in the case of online learning) or after all examples are presented for one iteration (in the case of a batch-mode learning).

      Through the gradient descent learning algorithm, the problem space is partitioned globally. This is one of the reasons why global learning in multilayer perceptrons suffers from the catastrophic forgetting phenomenon [58, 59]. Catastrophic forgetting (also called unlearning) is the inability of the system to learn new patterns without forgetting previously learned patterns. Methods to deal with this problem include rehearsing of the NN on a selection of past data or on generated new data points from the problem space [58].

      Other techniques that use global partitioning are support vector machines (SVMs) (see Ref. [60]) for a comparative study of ANN, fuzzy systems, and SVM). Through learning, the SVMs optimize the positioning of the hyperplanes to achieve maximum distance from all data items on both sides of the plane.
   b. *Local partitioning (local learning)*: In the case of a local learning, the structural modifications of the system affect the partitioning of only a small part of the space in which the current data example is drawn from. Examples are given in Fig. 9a and 9b, where the space is partitioned by circles and squares into a two-dimensional space. Each circle or square is the subspace defined by a neuron. The activation of each neuron is defined by local functions imposed on its subspace. An example of such local functions is kernels, as shown in Fig. 9a. Kernels K are defined by formulas as given below

$$K(x) = \frac{\exp(-x^2/2)}{\sqrt{2\pi}} \quad \text{while } \sum K(x) = 1 \quad \text{for all } x \in Z \tag{3}$$

      Other examples of local partitioning are when the space is partitioned by hypercubes and fractals in a three-dimensional space.

      Before creating a model, it is important to choose which type of partitioning will be more suitable for the task in hand. In the evolving connectionist systems presented later, the partitioning is local. Local partitioning is easier to adapt in an online mode, faster to calculate, and does not cause catastrophic forgetting.

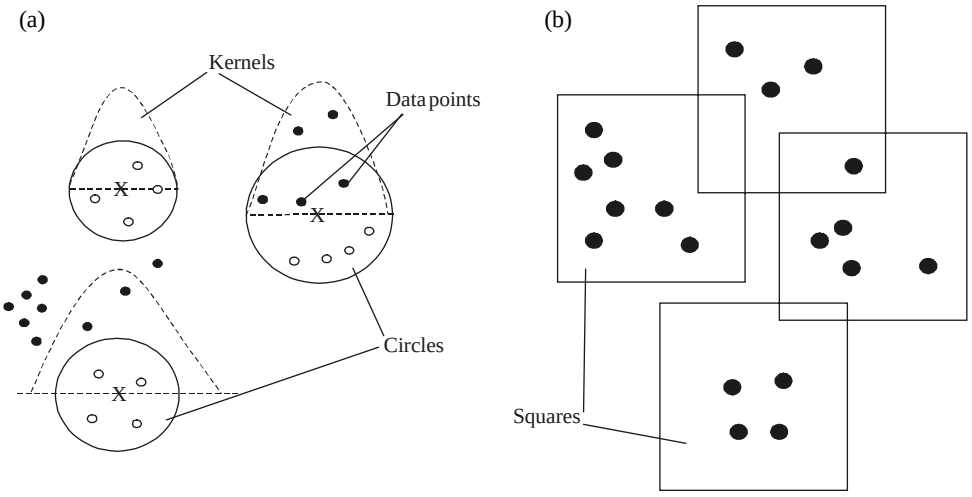

Figure 9. Partitioning of the problem space Z by (a) circles and (b) squares into a two-dimensional space. Reprinted with permission from [34], N. Kasabov, "Evolving Connectionist Systems. Methods and Applications in Bioinformatics, Brain Study and Intelligent Machines." Springer, London, 2003. © 2003, Springer-Verlag.

9. *What knowledge representation is facilitated in the learning system?* It is a well-known fact that one of the most important characteristics of the brain is that it can retain and build knowledge. However, it is not known yet how exactly the activities of the neurons in the brain are transferred into knowledge.

   For the purpose of the discussion in this chapter, knowledge can be defined as the information learned by a system that the system can interpret in different ways and can use in inference procedures to obtain new facts and new knowledge.

   Traditional ANNs and connectionist systems have been known to be poor facilitators of representing and processing knowledge, despite some early investigations [61, 62]. However, some of the issues of knowledge representation in connectionist systems have already been addressed in the so-called knowledge-based neural networks (KBNNs) [63–65]. KBNNs are ANNs that are prestructured in a way that allows for data and knowledge manipulation, which includes learning, knowledge insertion, knowledge extraction, adaptation, and reasoning. KBNNs have been developed either as a combination of symbolic artificial intelligence (AI) systems and NN [66], or as a combination of fuzzy logic systems and NN [67–71]. Rule insertion and rule extraction operations are examples of how a KBNN can accommodate existing knowledge along with data, and how it can "explain" what it has learned. There are different methods for rule extraction that are applied to practical problems [72–77].

   Generally speaking, learning systems can be distinguished on the basis of the type of knowledge they represent.

   a. *No explicit knowledge representation is facilitated in the system*: An example for such connectionist system is the traditional multilayer perceptron network trained with the backpropagation algorithm [36, 37, 49, 51, 78].
   b. *Memory-based knowledge*: The system retains examples, patterns, prototypes, and cases; for example instance-based learning [79], case-based reasoning systems [80], and exemplar-based reasoning systems [81].
   c. *Statistical knowledge*: The system captures conditional probabilities, probability distribution, clusters, correlation, principal components, and other statistical parameters [5].
   d. *Analytical knowledge*: The system learns an analytical function f: X? Y, that represents the mapping of the input space X into the output space Y. Regression techniques and kernel regressions in particular are well established [5, 82].
   e. *Symbolic knowledge*: Through learning, the system associates information with predefined symbols. Different types of symbolic knowledge can be facilitated in a learning system, as discussed further below.

f. *Combined knowledge*: The system facilitates learning of several types of knowledge.
g. *Metaknowledge*: The system learns hierarchical level of knowledge representation in which metaknowledge is also learned; for example, which piece of knowledge is applicable at what time.
h. *"Consciousness" of a system*: The system becomes "aware" of what it is, what it can do, and where its position among the rest of the systems in the problem space is.
i. *"Creativity" of a system*: An ultimate type of knowledge would be such knowledge that allows the system to act creatively, to create scenarios, and possibly to reproduce itself; for example, a system that generates other systems (programs) and improves in time based on its performance in the past.

10. *What type of symbolic knowledge is facilitated by the system?* If we can represent the knowledge learned in a learning system as symbols, different types of symbolic knowledge can be distinguished.

    - Propositional rules
    - First-order logic rules
    - Fuzzy rules
    - Semantic maps
    - Schemata
    - Meta-rules
    - Finite automata
    - Higher-order logic.

11. *If the systems' knowledge can be represented as fuzzy rules, what types of fuzzy rules are facilitated by the system?* Different types of fuzzy rules can be used; for example, Zadeh-Mamdani fuzzy rules [83, 84], Takagi-Sugeno fuzzy rules [85], and other types of fuzzy rules (e.g., type-2 fuzzy rules; for a comprehensive reading, see Ref. [86]).

    Generally speaking, different types of knowledge can be learned from a process or from an object in different ways, all involving the human participation. These ways include direct learning by humans, simple problem representation as graphs, analytical formulas, using ANN for learning and rule extraction, and so forth these forms can be viewed as alternative and possibly equivalent forms in terms of final results obtained after a reasoning mechanism is applied on them. Elaborating analytical knowledge in a changing environment is a very difficult process involving changing parameters and formulas with the changing data. If evolving processes are to be learned in a system and also understood by humans, neural networks that are trained in an online mode and their structure are interpreted as knowledge are the most promising models at present.

12. *Is the learning process active?* Humans and animals are selective in terms of processing only important information. They are searching actively for new information [87, 88]. Similarly, we can have two types of learning in an intelligent system: active learning in terms of data selection and filtering, and searching for relevant data, and passive learning, when the system accepts all incoming data.

3.1.3. Brief Overview of Major Problems With Existing Learning Systems

Despite the successfully developed and used ANNs, fuzzy systems, GAs, hybrid systems, and other methods and techniques for adaptive training, there are a number of problems with them. The main problems are listed below.

a. *Difficulty in preselecting the system architecture*: Usually an ANN model has a fixed architecture (e.g., a fixed number and organization of neurons and connections). This makes it difficult for the system to adapt to new data of unknown distribution. A fixed ANN architecture would definitely prevent the ANN from learning in a lifelong learning mode.
b. *Catastrophic forgetting:* When an ANN learns a new item, it forgets the old ones. This phenomenon was explained in the global learning paradigm above.

c. *Excessive training time required*: Training an ANN in a batch mode usually requires many iterations of propagating data through the ANN structure. This may not be acceptable for an adaptive online system that would require a fast adaptation.
d. *Lack of knowledge representation facilities*: Many of the existing ANN architectures capture statistical parameters during training but do not have linguistic meaning. This problem is called the "black box" problem. It occurs when only limited information is learned from the data, and essential aspects, which may be more appropriate and more useful for the future work of the system, are missed forever.

To overcome these problems, improved and new connectionist and hybrid methods and techniques are required both in terms of learning algorithms and system development.

3.2. Evolving Connectionist Systems

3.2.1. Evolutionary Computation

Evolutionary computation (EC) is concerned with population-based search and optimization of individual systems through generations of populations [89–92]. In other words, some property (or properties) of an individual will be improved not only through an individual development but also through natural selection. Methods of EC in principle include two stages: a stage of creating a new population of individuals, and a stage of development of the individual systems, so that a system develops and evolves through interaction with the environment that is also based on the genetic material embodied in the system.

The most popular among the EC techniques are the GAs. They are computational models for the optimization of complex combinatorial and organizational problems with many variants, by employing analogy with nature's evolution. GAs were introduced for the first time in the work of John Holland [91]. They were further developed by him and other researchers [89–92].

The terms used in the GA are analogous to the terms used to explain the evolution processes. They are:

- *gene*: a basic unit, which defines a certain characteristic (property) of an individual;
- *chromosome*: a string of genes, used to represent an individual, or a possible solution to a problem in the solution space;
- *population*: a collection of individuals;
- *crossover (mating) operation*: substrings of different individuals are taken and new strings (offsprings) are produced;
- *mutation*: random change of a gene in a chromosome;
- *fitness (goodness) function*: a criterion that evaluates how good each individual is;
- *selection*: a procedure of choosing a part of the population that will continue the process of searching for the best solution while the other parts of the individuals "die."

The main steps of a GA are outlined here:

1. Initialize a population of n individuals P
2. REPEAT
   2a {apply a crossover operation between the individuals from P to create an offspring set of individuals R}
   2b {apply a fitness function to evaluate the fitness of the individuals in R}
   2c {apply a selection criteria to select the fittest individuals from R in a new set P}
   2d {apply a mutation operator on the individuals from P}

UNTIL {an individual from P has reached a desired fitness or end of the procedure is reached}
When using the GA method for a complex multioptional optimization problem, there is no need for in-depth problem knowledge, or a need for many data examples stored beforehand. What is needed here is merely a "fitness" or "goodness" criterion for the selection of the most promising individuals (they are partial solutions to the problem). This criterion may require a mutation as well, which is a heuristic approach of a "trial-error" type. This implies keeping (recording) the best solutions at each of the stages.

The simple genetic algorithms introduced by John Holland are characterized by: simple, binary genes (i.e., the genes take values of 0 and 1 only); simple, fixed single-point crossover operation, in which the crossover operation is done by choosing a point where a chromosome is divided into two parts swapped with the two parts taken from another individual (see Fig. 10); and fixed-length encoding (i.e., the chromosomes had a fixed length of six genes).

The main issues in using genetic algorithms relate to the choice of genetic operations (crossover, selection, mutation).

Genetic algorithms comprise a great deal of parallelism. Thus, each of the branches of the search tree for best individuals can be used in parallel with the others. This allows for an easy realization of the genetic algorithms on parallel architectures.

Genetic algorithms are search heuristics for the "best" instance in the space of all possible instances. The following issues are important for any genetic algorithm.

- *The encoding scheme*, that is, how to encode the problem in terms of genetic algorithms (what variables to choose as genes, how to construct the chromosomes, etc).
- *The population size*—how many possible solutions should be kept for further development;
- *The crossover operations*—how to combine old individuals and produce new, more prospective ones;
- *The selection criteria;*
- *The mutation operator*—when and how to apply mutation.

In short, the major characteristics of the genetic algorithms are the following.

First, they are heuristic methods for search and optimization. As opposed to the exhaustive search algorithms, the GAs do not produce all variants to select the best one. Therefore, they may not lead to the perfect solution but, rather, to one, which is closest to it taking into account the time limits. But nature itself is imperfect too (partly because the criteria for perfection keep changing), and what seems to be close to perfection according to one "goodness" criterion may be far from it according to another.

Second, they are adaptable, which means that they have the ability to learn, to accumulate facts and knowledge without having any previous knowledge. They begin only with a "fitness" criterion for selecting and storing individuals (partial solutions) that are "good" and dismissing those which are "not good."

GAs can be incorporated in learning modules as part of an expert system or of other information processing systems.

Other EC techniques include evolutionary strategies (these techniques use only one chromosome and a mutation operation, along with a fitness criterion, to navigate in the solution (chromosomal) space); evolutionary programming—These are EC techniques applied to the automated creation and optimization of sequence of commands (operators) that constitute a program (or an algorithm) to solve a given problem [90].

The theory of GA and the other EC techniques includes different methods for selection of individuals from a population, different cross-over techniques, and different mutation techniques.

Selection is based on fitness. A common approach is proportional fitness (i.e., "if I am twice as fit as you, I have twice the probability of being selected.") Roulette wheel selection gives chances to individuals according to their fitness evaluation. Other selection techniques include tournament selection (every time of selection, the roulette wheel is turned twice,

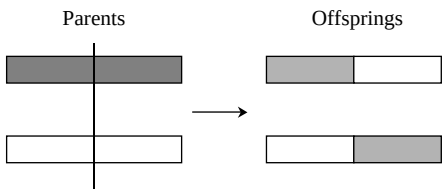

Figure 10. Schematic illustration of the operation of single-point crossover in GA. Parenting individuals exchange parts (in this case one half) of their chromosomes to create exactly two offsprings.

and the individual with the highest fitness is selected), rank ordering, and so on [93]. An important feature of the selection procedure is that fitter individuals are more likely to be selected. The selection procedure can also involve keeping the best individuals from the previous generation (if this principle was used by the nature, Michelangelo would have been alive today, he is one of the greatest artists ever, with the best genes in this respect). This operation is called elitism.

After the best individuals are selected from a population, a cross-over operation is applied between these individuals. Different single or multiple-point crossover operations can be used. Then the selected individuals undergo mutation.

Mutation can be performed in the following ways: For a binary string, just randomly "flip" a bit; for a more complex structure, randomly select a site, delete the structure associated with this site, and randomly create a new substructure

Some EC methods just use mutation (no crossover; e.g., evolutionary strategies). Normally, however, mutation is used to search in a "local search space," by allowing small changes in the genotype (and, therefore, hopefully in the phenotype). In the field of ANNs, optimal values of parameters (weights, architecture, etc.) can be sought not only through learning but also through evolution; that is, through the process of selection and crossover of the best individual neural networks. This process can be combined with individual learning to lead to the Baldwin effect [94]. The genotypes after Baldwinian learning remain unchanged; however, learning can influence indirectly the selection process by altering the fitness of individuals, and thus eventually evolution is affected [95].

3.2.2. Artificial Intelligence Versus Emerging Intelligence

Many authors see intelligence as a set of features or fixed properties of mind that are stable and static. According to this approach, intelligence is genetically defined, given, rather than developed.

Intelligence is also seen as a constant and continuous adaptation. Darwin's contemporary H. Spencer proposed in 1855 the law of intelligence, stating that "the fundamental condition of vitality is that the internal state shall be continually adjusted to the external order" (see Ref. [96], p. 14). Intelligence is "the faculty of adapting oneself to circumstances" according to Henri Simon and Francis Binet, the authors of the first IQ test (see Ref. [97]).

In Ref. [98], intelligence is defined as "the human capacity to acquire knowledge, to acquire a set of adaptations and to achieve adaptation."

Knowledge representation, concept formation, reasoning, and adaptation are obviously the main characteristics of intelligence, on which all authors agree [99, 100]. How these features can be implemented in a computer model is the main objective of the area of AI.

AI develops methods, tools, techniques, and systems that make possible the implementation of intelligence in our computer models. This is a "soft" definition of AI, which is in contrast with the first definition of AI (the "hard" one) given by Alan Turing in 1950. According to the Alan Turing's test for AI, if a person communicates in natural language with another person or an artificial system behind a bar without being able to distinguish between the two, and even more, without being able to identify whether it is a male or a female, as the system should be able to fool the human in this respect, then the system behind the bar can be considered an AI system. The described test points to an ultimate goal of AI that is understanding concepts and a language on the one hand, and to a silly task of fooling people on the other hand, rather than toward showing how one can achieve the goal.

Many techniques of AI prove to be useful to a point but do not cope very well with the dynamic nature and the combinatorial complexity of many problems from life sciences and engineering. Table 2 shows graphically different levels of evolving processes in the human brain that cannot be modeled by the use of existing AI methods.

In a general sense, information systems should help trace and understand the dynamics of the processes, automatically evolve "rules" that would change over time, "take a shortcut" in the complex problem spaces, and improve all the time. These requirements define a subset of AI, which can be called emerging intelligence (EI). The emphasis here is not on the achievement of the ultimate goal of AI as defined by Turing but, rather, on creating systems

Table 2. Different levels of evolving processes in the human brain that cannot be modeled by the use of existing AI methods.

1. Evolutionary development Function examples: genome evolution, creation of new individuals and species
2. Brain level Function examples: cognition, speech and language, consciousness
3. Neural network level Function examples: sound perception, visual image processing
4. Whole cell, neuronal level Function examples: neuronal processes of activation and growth
5. Molecular level Function examples: DNA translation into RNA, RNA transcription into proteins

Reprinted with permission from [34], N. Kasabov, "Evolving Connectionist Systems. Methods and Applications in Bioinformatics, Brain Study and Intelligent Machines." Springer, London, 2003. © 2003, Springer-Verlag.

that learn all the time, improve their performance, and become more and more intelligent. A constructivist working definition of EI is given below. It emphasizes the dynamic and knowledge-based structural and functional self-development of a system.

EI is a feature of an information system that develops its structure and functionality in a continuous, self-organized, adaptive, and interactive way from incoming information, possibly from many sources, and performs intelligent tasks typical for humans (e.g., adaptive pattern recognition, concept formation, languages learning, and intelligent control).

David Fogel [101], in his highly entertaining and highly sophisticated book *Blondie* 24—*Playing at the Edge of AI, describes a case of EI as a system* that learns to play checkers online without using any instructions and improves after every game. The system uses connectionist structure and evolutionary algorithms, along with statistical analysis methods.

3.2.3. Framework for Evolving Connectionist Systems

Evolving connectionist systems (ECOS) are multimodular connectionist architectures that facilitate modeling of evolving processes and knowledge discovery [34].

An evolving connectionist system is a neural network or a collection of such networks that operate continuously in time and adapt their structure and functionality through a continuous interaction with the environment and with other systems.

The adaptation is defined through a set of parameters that is subject to change during the system operation, an incoming continuous flow of information with unknown distribution, and a goal (rationale) criteria (also subject to modification) that is applied to optimize the performance of the system over time.

If we refer to the general framework of a learning system $\{S, W, P, F, L, J\}$, as discussed previously, in ECOS, the set of parameters P can be regarded as a chromosome of "genes," and both developmental learning and evolutionary computation can be applied for the system's optimization.

ECOS comprises a small subset of all the connectionist models and systems that follow the working classification scheme discussed above. They evolve either in the problem data space or in their own system space. They learn in any of the learning modes: unsupervised, supervised, reinforced or combined with the following specific characteristics:

1. They evolve in an open space.
2. They learn in an online, pattern mode, with incremental learning, and possibly through one pass of the incoming data through the system.
3. They learn in a lifelong learning mode.
4. They learn both as individual systems and as evolutionary population systems.
5. They use constructive learning and have evolving structures.
6. They learn and partition the problem space locally, thus allowing for a fast adaptation and tracing the evolving processes over time.
7. They facilitate different types of knowledge; mostly a combination of memory-based, statistical and symbolic knowledge. The evolving connectionist models presented in

the first part of this book facilitate Zadeh–Mamdani fuzzy rules (EFuNN, Chapter 3; HyFIS, Chapter 5), Takagi–Sugeno fuzzy rules (DENFIS, Chapter 5), and type 2 fuzzy rules (Chapter 5) in Ref. [34].

Each evolving connectionist system consists of four main parts: data acquisition, preprocessing and feature evaluation, connectionist modeling part, and knowledge acquisition.

Table 3 is a generalized algorithm of the functioning of an ECOS [34]. An online processing of all this information makes it possible for the ECOS to interact with users in an intelligent way. If man–system interaction can be achieved in this way, this processing can be used to extend system–system interactions as well.

Modeling evolving processes is a difficult task, as in many cases it is not well defined in terms of global optimization and goal function. However, it is vital that a variety of methods be developed that can be applied to the number of challenging real-life applications.

4. GENE INFORMATION PROCESSING

In living systems, many dynamic, adaptive, evolving processes are observed at different levels and different stages of the development that are involved in a complex interaction. At a molecular level and a cell level, the DNA, the RNA, and the protein molecules evolve and interact in a continuous way. The genes form dynamic gene networks (GNs) that define the complexity of the living organism [7]. It is not just the number of the genes in a genome, but the interaction between them that makes one organism more complex than another. The confirmation that there might be only about 30,000 protein-coding genes in the human genome is one of the key results of the monumental work of the human genome project [102]. There is a mere one-third increase in gene numbers from a rather unsophisticated nematode (*Caenorhabditis elegans*, with about 20,000 genes) [103] to humans (and other mammals). In fact, the genomes of all mammals are so similar that it is hard to understand how they can produce such different animals. If their genes are alike, it is probably changes in when, where, and how active they are that drives the differences between species.

Table 3. Generalized ECOS algorithm, introduced in [34].

```
Set some preliminary parameter values for the ECOS parameters (chromosomes)
REPEAT {in a lifelong learning mode}
  IF input, or input-output data is available DO
   Read input data (or input- output data pairs if such are available)
   Evaluate the input-output features:
         (a) add new ones if necessary;
         (b) select the current most appropriate ones for the task
   Propagate input data through the NN modules and evaluate the similarity of the input data to
the modules
   If there is not sufficient similarity—create new modules, or create new connections in an
existing module
   Calculate the output of the system
   Calculate a feedback from the output to the system through:
        A supervised mode of learning, if output error values are calculated, or
        A reinforcement mode of learning—if just hints about the correctness of the output values
are available, or
        Report the output values if the system is in a recall mode
  Modify the structure of ECOS based on the feedback
  Extract and report the current knowledge learned by the ECOS, e.g. through rule extraction
techniques
  Optimize the ECOS structure based on some accumulated statistical and ECOS parameters
(possibly EC methods)
ELSE
  Apply inner structural and functional learning for structure improvement (e.g., sleep learning)
UNTIL {the system is stopped, or interrupted}
```

4.1. Genes and Cellular Processes

DNA is a chemical chain that is present in the nucleus of each cell of an eukaryotic organism, and it consists of ordered double-helix pairs of small chemical molecules (bases): adenine (A), cytosine (C), guanidine (G), and thymidine (T), linked together by a deoxyribose sugar phosphate nucleic acid backbone.

The central dogma of molecular biology (see Fig. 11) states that the DNA is transcribed into RNA, which is translated into proteins [104].

DNA contains millions of base pairs, but only 5% or so is used for the production of proteins, and these are the segments from the DNA that contain genes. Each gene is a sequence of base pairs that is used in the cell to produce proteins. Genes have length of hundreds to thousands of bases.

RNA has a similar structure to DNA, but here thymidine is substituted by uridine (U). In pre-RNA, only segments that contain genes are extracted from the DNA. Each gene consists of two types of segments: exons, which are segments translated into proteins, and introns, which are segments that are considered redundant and do not take part in the protein production. Removing the introns and ordering only the exon parts of the genes in a sequence is called splicing, and this process results in the production of messenger RNA (or mRNA) sequences.

mRNAs are directly translated into proteins. Each protein consists of a sequence of amino acids, each of them defined as a base triplet, called a codon. From one DNA sequence many copies of mRNA are produced; the presence of certain gene in all of them defines the level of the gene expression in the cell and can indicate what and how much of the corresponding protein will be produced in the cell.

The above description of the central dogma of the molecular biology is very much a simplified one, but it will in understanding the rationale behind using connectionist and other information models in bioinformatics.

Genes are complex chemical structures, and they cause dynamic transformation of one substance into another during the whole life of an individual, as well as throughout the life of the human population over many generations. When genes are "in action," the dynamics of the processes in which a single gene is involved are very complex, as this gene interacts with many other genes and proteins and is influenced by many environmental and developmental factors.

Modeling these interactions, learning about them, and extracting knowledge is a major goal for bioinformatics. Bioinformatics is concerned with the application of the methods of information sciences for the analysis, modeling, and knowledge discovery of biological processes in living organisms.

The whole process of DNA transcription, gene translation, and protein production is continuous, and it evolves over time. Proteins have 3D structures that unfold over time, governed by physical and chemical laws. Proteins make some genes to express and may

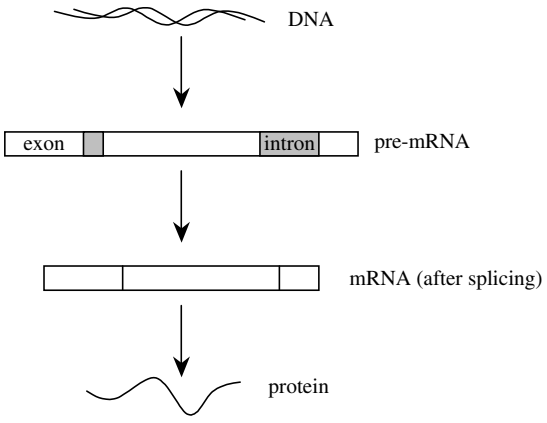

Figure 11. DNA is transcribed into RNA, which is translated into proteins—the central dogma of molecular biology.

suppress the expression of other genes. The genes in an individual may mutate, change slightly their code, and therefore express differently at a next time. Thus, genes may change, mutate, and evolve in a life time of a living organism.

4.2. Computational Models of Gene Information Processing

In a single cell, the DNA, the RNA and the protein molecules interact in a continuous way during the process of the RNA transcription from DNA (genotype), and the subsequent RNA to protein (phenotype) translation [105–109]. A single gene interacts with many other genes in this process, inhibiting, directly or indirectly, the expression of some of them, and promoting others at the same time. This interaction can be represented as a gene regulatory network (GRN) [7, 8, 110–115]. Our challenge is to create computational models of GRN from both dynamic data (e.g., gene expression data of thousands of genes over time, and also from protein data) and from static data (e.g., DNA), under different external inputs (diseases, drugs, etc.). A large amount of data on gene interactions for specific genomes, as well as on partial models, is available from public domain databases such as GenBank and PubMed (http://www.ncbi.nlm.nih.gov/), KEGG (http://www.genome.ad.jp/kegg/), Stanford Microarray Database, and many more [116]. Collecting both static and time course gene expression data from up to 30,000 genes is now a common practice in many biological, medical, and pharmaceutical laboratories in the world through the introduction of microarray technologies (see, e.g, www.ebi.ac.uk/microarray). Sophisticated information and mathematical methods are needed for the analysis, modeling, and discovery of GRN from this data.

Several generic information methods for modeling and for the discovery of variable interaction networks from time course data have been proposed and used in the domain of GRN modeling. Among them are statistical methods that include correlation techniques, linear regression, Bayesian networks, and hidden Markov models [9, 115, 117–119]; neural networks [120, 121]; evolutionary computation, and genetic algorithms in particular [116, 122, 123]; directed graphs [105, 116, 117]; Petri nets [116]; and ordinary and partial differential equations [8]. There have also been specific methods developed for the purpose of cell modeling [8, 124, 125]. A detailed survey of the elements and the pathways in the control of gene expression and the principles of their computational modeling can be found, for instance, in the books [8, 110, 113, 116] and in the review papers [126, 127]. With respect to the taxonomy of GRN models, their principles and descriptions, exhaustive reviews can be found for instance in Refs. [127, 128].

In Ref. [129], an evolving connectionist system (ECOS) is incrementally evolved from incoming data $\mathbf{X}(t_0)$, $\mathbf{X}(t_1)$, $\mathbf{X}(t_2)$, ..., representing the values of all, or some, of the gene expression variables or their clusters. Consecutive vectors $\mathbf{X}(t)$ and $\mathbf{X}(t+k)$ are used as input and output vectors, respectively, in an ECOS model, as shown in Fig. 12a. After training an ECOS on the data, rules are extracted through IF–THEN representation of the rule nodes [e.g., IF x1(t) is High (0.87) and x2(t) is Low (0.3) THEN x3(t + k) is High (0.6) and x5(t + k) is Low (0.2)]. Each rule represents a transition between the current and next state of the variables, as shown in Fig. 12b, where each rule is shown as an arrow. All rules together form a representation of the GRN. Figure 12b shows two trajectories, N_1 and N_2, that represent two GRNs, derived under different conditions, in the 2D PCA (principal component analysis) coordinate space of all n variables. By modifying a threshold for rule extraction, one can extract in an incremental way stronger or weaker patterns of relationships between the variables [77].

Using another ECOS model [130], other types of variable relationship rules can be extracted [e.g.: IF x1(t) is (0.63 0.70 0.76) and x2(t) is (0.71 0.77 0.84) and x3(t) is (0.71 0.77 0.84) and x4(t) is (0.59 0.66 0.72) THEN $x5(t + k) = 1.84 - 1.26x1(t) - 1.22x2(t) + 0.58x3(t) - 0.3x4(t)$], where the cluster for which the value of x_5, as defined in the rule above, is a fuzzy cluster represented through triangular membership functions defined as triplets of values for the left, center, and right points of the triangle on a normalization range of Ref. [131]. The fuzzy representation allows for models to deal with imprecise data [10]. The rules extracted from the ECOS form a representation of the GRN (see Fig. 12c).

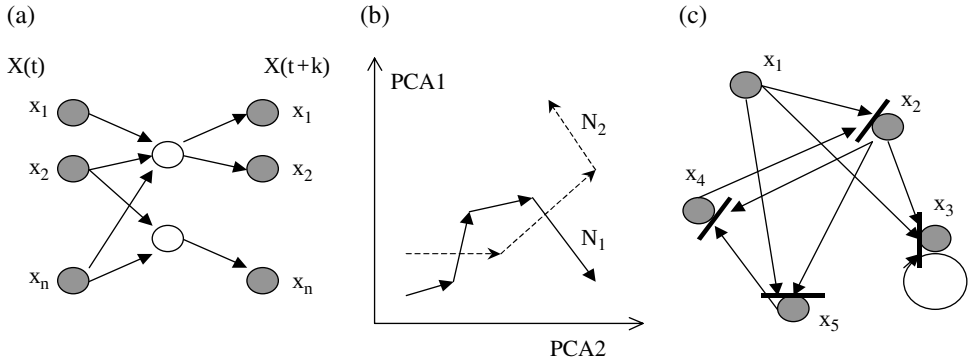

Figure 12. (a) A hypothetical ECOS for GRN modeling; (b) state transitions (rules, represented as arrows) in the two PCA dimensional space of the *n* variables; (c) part of a GRN extracted from an ECOS model.

Rules may change with the addition of new data, thus making it possible to identify stable versus dynamic parts of the GRN. The relationship between variables from Fig. 12c can be represented by a simple linear, or a complex nonlinear, relationship function.

5. NEURO-GENETIC INFORMATION PROCESSING

Let us consider the differences between humans and chimpanzees. After sequencing ~3 million letters of the chimp genome and comparing them with the human draft, Svante Pääbo of the Max Planck Institute for Evolutionary Anthropology in Leipzig, Germany, and his group reasoned that DNA sequence cannot be the cause of such species variation only 1.3% of letters are different [132]. These letters belong mainly to the genes expressed in the brain. Then the researchers and the team of U.S. scientists measured the levels of gene activity in the brains of humans and chimps. Chimp and human brain transcription patterns are very different. The human brain has accelerated usage of genes, with ~90% of the studied genes being more highly expressed than in the chimp brain [133], and mainly the up-regulation of genes involved in synaptic transmission and plasticity (learning and memory), energy metabolism, and growth. Higher levels of such a neuronal genetic activity are likely to have important consequences in cognitive and behavioral capacities of humans.

However, to understand these genetic differences, we also need to know what is going on in a chimp's mind; after all, we know that chimpanzees and bonobo chimpanzees have quite impressive language and learning skills [134] (see also http://www.gsu.edu/~wwwlrc/biographies/kanzi.html). Therefore, it is important to study the neuro-genetic interactions in all species to compare them with humans.

In September 2003, the news spread that the Microsoft cofounder Paul G. Allen has donated $100 million to launch a private research organization in Seattle devoted to deciphering the links between our genes and our brain (Allen Brain Atlas Project, http://www.nba.com/blazers/news/Allen_Institute_for_Brain_Scie-84505-41.html). It is rather a heroic task, considering that approximately 6000 genes are thought to be expressed only in the brain, with many more that are expressed in the brain and also in other parts of the body as well.

Therefore, it will be crucial to study, both theoretically and experimentally, the consequences of mutated genes on the activity of neural networks and on the neurological and mental deficits that can follow. It is how the links can be established, determining which gene(s) and which interactions between genes are responsible for which neuronal, and eventually which mental, function.

5.1. Neuro-Genetic Processes in the Brain

Complex faculties of brains, such as, for example, intelligence, are under the influence of both genetic and environmental factors, but what does it actually mean, for instance, for intelligence to be under the influence of genes? What are the nature and rules of this

dependence? There is vast scientific evidence that intelligence (like other mental faculties, processes, states, etc.) depends on the normal functioning of brain neural networks, and it is the set-up of brain neural networks that is under the influence of genes. Whole-brain development from conception is guided by the complex sequence of switching on and off many different genes operating in their own complex and intricate networks (GRNs). Through protein synthesis, genes determine the structure and connectivity of the brain, including all the biochemical processes involved in information processing and the mere survival of brain cells.

For the correct brain function, there is interplay between genetic and epigenetic factors, like signals from outside of neurons, either from other neurons or other cells in the brain (i.e., glia) or from somewhere else in the body (e.g., hormones). Proteins, molecules, and ions acting on neurons from outside can and do act on the genome to influence its activity. It is how the environment exerts its influence on the structure of brain neural networks.

What is the nature of the questions we can ask?

- Can we perform a reverse engineering of brain genetic networks? That is, can we identify causal regulatory interactions between genes from time-dependent multigene expression measurements?
- What are the ways that genes can determine the function of brain neural networks?

Thus, in the first research area on genes and the CNS (central nervous system), researchers have applied GRN inference techniques to an extensive survey of gene expression in CNS development. Detailed cluster analysis has uncovered waves of expression for about 112 genes that characterize distinct phases of the development of spinal cord and hippocampus [114, 135]. Thus, the first area of research covers the monitoring of the activity of many genes in parallel and the use of the models of GRN to help and guide the inference of regulatory connections between genes, resulting in a gene-interaction diagram of gene-interaction pathways. There is still a long way to go before all 6000 genes are processed in a similar way but with better techniques and better theoretical and computational models.

The second area of study is to reveal the consequences of genes mutation on neural and mental functions (see Table 4) [136]. It is crucial to study, both theoretically and experimentally, the consequences of mutated genes on the activity of neural networks and on the neurological and mental deficits that can follow. It is how the links can be established; that is, which genes and which interactions between genes are responsible for which neuronal, and eventually which mental, function. We can see that the detailed pathogenesis of these diseases is at this time unknown. This is the area into which computational models can bring new insights.

5.2. Computational Modeling of Neuro-Genetic Processes

The genes, encoded in the DNA, that are transcribed into RNA and then translated into proteins in each cell contain important information related to the brain activity. A specific gene from the genome relates to the activity of a neuronal cell in terms of a specific function. However, the functioning of the brain is much more complex than that. The interaction between the genes is what defines the functioning of a neuron. Even in the presence of a mutated gene in the genome that is known to cause a brain disease, the neurons can still function normally provided a certain pattern of interaction between the genes is maintained—a certain state of the GN [137]. In contrast, if there is no mutated gene in the genome, certain abnormalities in brain functioning can be observed as defined by a certain state of the interaction between the genes [11]. The above-cited and many other observations point to the significance of modeling a neuron and a neuronal ensemble at the gene level to predict the state of the ensemble. The process of modeling the gene interaction with the goal of brain understanding is a significant challenge to biologists, mathematicians, information and computer scientists, brain scientists, and researchers from many other areas.

5.2.1. Principles of a General Neuro-Genetic Model

In Ref. [127], a computational model of GRN of early neurogenesis in *Drosophila* is developed with the use of artificial neural networks of a Hopfield type, where each gene (gene product) is a node in a recurrent network and the values of connections express the

Table 4. Single and multiple genes related to some brain functions and abnormalities [136].

Disease	Mutations of genes identified so far	Location of genes on chromosomes	Brain abnormality	Symptoms
Alzheimer disease (AD)	PS2 (AD4) PS1 (AD3) ? ?	1 14 19 21	plaques made of fragmented brain cells surrounded by amyloid-family proteins, tangles of cytoskeleton filaments	progressive inability to remember facts and events and later to recognize friends and family
Amyotrophic lateral sclerosis (ALS)	SOD1	21	progressive degeneration of motor neuron cells in the spinal cord and brain	loss of motor control which ultimately results in paralysis and death
Angelman syndrome (AS)	UBE3A	maternally derived chromosome 15 (segment 15q11–13)	mutations in UBE3A disrupt protein break down during brain development	mental retardation, abnormal gait, speech impairment, seizures, frequent laughing, smiling, and excitability
Epilepsy (many forms)	Multiple	Multiple	abnormal cell firing in the brain	recurring seizures
Fragile X syndrome	FMR1	X	impaired synaptic function of glutamatergic synapses	the most common inherited form of mental retardation
Huntington disease (HD)	HD gene	4	dilatation of ventricles and atrophy of caudate nucleus	degenerative neurological disease that leads to dementia
Williams syndrome	LIM kinase and elastin coding sequences	7	?	high competence in language, music and interpersonal relations, with low IQ

sign and strength of their interactions. Here a model is presented [138] that is based on two sets of essential genes—a set of genes $\mathbf{G}_{\text{gen}}$ that defines generic neuronal functions, and a set of genes $\mathbf{G}_{\text{spec}}$ that defines specific neuronal functions (e.g., epileptic behavior). Consider a function that describes measurement of GABA (e.g., GABA_{A}) in the synaptic cleft. The measurement of cross-membrane potential using a patch-clamp procedure provides a function of time during an event that probes spiking properties of the neuron when a certain voltage is delivered to the neuron.

The two sets of genes together form a set $\mathbf{G} = \{g_1, g_2, \ldots, g_n\}$ that will be used, and a GN of this set will be defined in the model. An example of a GN of four genes is given in Fig. 13. The expression level of each gene $g_j(t + \Delta t)$ is a function of the gene expression levels of the rest of the genes $\mathbf{G}(t)$. As a simple model, we will assume that this function is a linear function

$$g_j(t + \Delta t) = w_{j,0} + w_{j,1} g_1(t) + \cdots + w_{j,n} g_n(t) \tag{4}$$

The square matrix of gene connection weights $\mathbf{W}$ represents the GN. This model can be run for consecutive time moments.

Here we use the gene instead of proteins, using a standard assumption that if a gene is up-regulated, more proteins defined by this gene will be produced in the neuronal cell.

A set of neuronal functions (parameters) $\mathbf{P} = \{p_1, p_2, \ldots, p_m\}$ from a neural network model is related to particular genes, so that each parameter p_j is a function of the expression of several (or in the partial case, one) genes. For simplicity in our model, we will assume that one parameter p_j depends on one gene g_k through a liner function

$$p_j(t + \Delta t) = z_{j,0} + z_{j,k} g_k(t) \tag{5}$$

The parameter vector $\mathbf{Z}$ defines the relationship between the selected genes $\mathbf{G}$ and the spiking characteristics of a neuron.

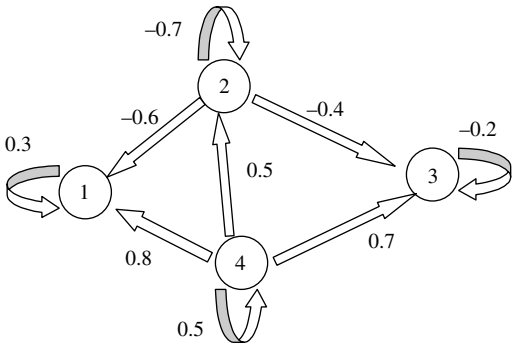

Figure 13. A hypothetical example of a GN comprising of four genes related to both generic and specific functions of a neuron. The genes are connected with arcs that represent the relationship (both sign and strength) between the level of expression of this gene at time moment (t) and the next time moment ($t + \Delta t$).

This generic neuro-genetic (NGM) model can be run continuously over time in the following way:

1. Define the initial expression values of the genes **G**, $\mathbf{G}(t = 0)$ in the neuron and the matrix **W** of the GN if that is possible.
2. Run the GN and define the next moment state of the gene set $\mathbf{G}(t + \Delta t)$ using Eq. (4).
3. Define the values of the parameters **P** from the gene state **G** using Eq. (5).
4. Define the spiking activity of neuron(s) (taking into account all external inputs to the neural network).
5. Go to step 2.

We assume that the same matrix **W** defines the GN of each neuron in a SNN. The spiking activity of all neurons are defined using the algorithm in the section below for a time interval T, thus allowing us to calculate the probability distribution function (PDF) of the spiking activity of neurons in the spiking neural network (SNN).

5.2.2. Using a Genetic Algorithm to Find the Values of Parameters of a Neuro-Genetic Model

We assume that the functioning of a neural network is evaluated as its PDF of neural activity, thus making it possible to observe and model different brain functions such as epilepsy; alpha, beta, and gamma states; learning; memorizing; sleeping; and so forth. Usually, EEG data are available to test these models. In general, the functioning of a neural network can be expressed and evaluated in other terms; for instance, metabolic or other activity terms.

The task for us is to define a set of parameter values for **W** and **Z** so that the SNN has a desired PDF of neural activity, denoted here as PDF^. To solve this problem with the use of the above-described model, we can apply the methods of EC and, in particular, a GA method [116, 139, 140] with a fitness function of PDF = PDF^ within a margin of tolerance.

Parameter estimation is a very difficult task in inferring GN models, mainly because of the lack of observation data relative to the number of genes involved. In this respect, EC that are robust, global optimization methods become important tools to accurately inference a GN.

EC, inspired by the Darwin theory of evolution, searches with a swarm of points based on the objective function (say, the output error) feedback of these points. It has been used for parameter estimation or optimization in many engineering applications. Unlike classical derivative-based (like Newton) optimization methods, EC is more robust against noise and multimodality in the search space. In addition, EC does not require the derivative information of the objective function and is thus applicable to complex, black box problems.

These characteristics make EC highly suitable for identifying parameters of GN models for three reasons. First, the derivative information of the underlying model (e.g., Petri Net) is usually not available. Second, data are scarce or missing (causing multimodality) and noisy, requiring a robust, global optimization algorithm that is not easily misled by suboptima

and noise. Third, qualitative inference of parameters is difficult with such small number of observations relative to the large number of genes involved.

In the GA implementation here, two chromosomes of parameters will be used—**W** and **Z**, so that for every generation (a set of values) of these chromosomes, the SNN is run for the time period of T and the PDF is evaluated. Then it is compared with the desired PDF^, and if the fitness function is not satisfied, the process continuous with modified values for the parameters **W** and **Z** according to the selected GA strategy.

5.2.3. Example of a Neural Network Model With Parameters Related to Gene Net

A spiking model of a neuron—an element of the SNN—can be, for instance, inspired by the spike-response model (SRM) of a neuron [28, 141]. Neuron i receives input spikes from presynaptic neurons $i \in \Gamma_i$, where Γ_i is a pool of all neurons presynaptic to neuron i (see Fig. 14). The state of neuron i is described by the state variable $u_i(t)$ that can be interpreted as a total PSP at the membrane of soma. When $u_i(t)$ reaches the firing threshold $\vartheta_i(t)$, neuron i fires (i.e., emits a spike; see Fig. 15a). The moment of $\vartheta_i(t)$ crossing defines a firing time t_i of an output spike. The value of the state variable $u_i(t)$ is the sum of all postsynaptic potentials, i.e. (see Fig. 15b)

$$u_i(t) = \sum_{j \in \Gamma_i} \sum_{t_j \in F_j} J_{ij} \varepsilon_{ij}(t - t_j - \Delta_{ij}^{ax}) \tag{6}$$

The weight of synaptic connection from neuron j to neuron i is denoted by J_{ij}. It takes positive (negative) values for excitatory (inhibitory) connections, respectively. Depending on the sign of J_{ij}, a presynaptic spike generated at time t_j increases (or decreases) $u_i(t)$ by an amount $\varepsilon_{ij}(t - t_j - \Delta_{ij}^{ax})$. The term Δ_{ij}^{ax} is an axonal delay between neurons i and j that increases with Euclidean distance between neurons.

The positive kernel $\varepsilon_{ij}(t - t_j - \Delta_{ij}^{ax}) = \varepsilon_{ij}(s)$ expresses an individual PSP evoked by a presynaptic neuron j on neuron i. A double exponential formula can be used (see Fig. 15c)

$$\varepsilon_{ij}^{\text{synapse}}(s) = A^{\text{synapse}} \left[\exp\left(-\frac{s}{\tau_{\text{decay}}^{\text{synapse}}} \right) - \exp\left(-\frac{s}{\tau_{\text{rise}}^{\text{synapse}}} \right) \right] \tag{7}$$

where $\tau_{\text{decay/rise}}^{\text{synapse}}$ are time constants of the rise and fall of an individual PSP, A is the PSP's amplitude, and synapse = fast_excitation, fast_inhibition, slow_excitation, and slow_inhibition, respectively. These types of PSPs are based on neurobiological data [142, 143].

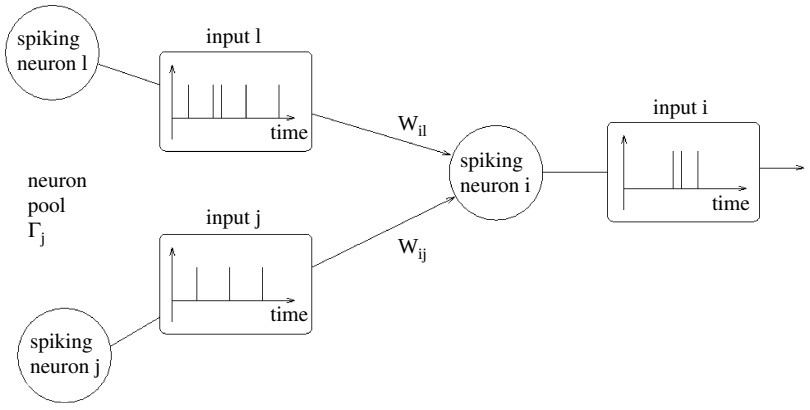

Figure 14. In response to input series of spikes from the pool of presynaptic neurons Γ_i, a neuron i generates its own series of output spikes.

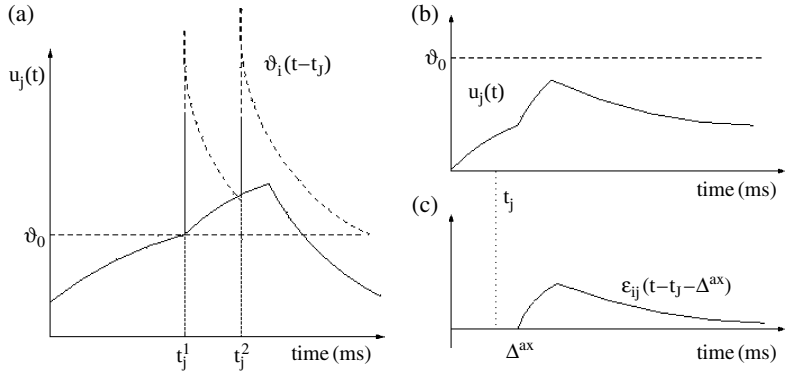

Figure 15. Spiking neuron model. (a) When the state variable $u_i(t)$ of a spiking neuron reaches the firing threshold $\vartheta_i(t)$ at time t_i, a neuron fires an output spike. However, an actual firing threshold rises after each output spike and decays back to the initial value. (b) Subthreshold temporal summation of individual postsynaptic potentials. (c) The kernel $\varepsilon_{ij}(t - t_j - \Delta_{ij}^{ax})$ describes an individual PSP evoked by the presynaptic spike fired at time t_j after some axonal delay Δ_{ij}^{ax}.

Immediately after firing an output spike at t_i, neuron's firing threshold $\vartheta_i(t)$ increases m times and then returns to its initial value ϑ_0 in an exponential fashion (see Fig. 15a).

$$\vartheta_i(t - t_i) = m \times \vartheta_0 \exp\left(-\frac{t - t_i}{\tau_{\text{decay}}^{\tau}}\right) \tag{8}$$

where $\vartheta_{\text{decay}}^{\vartheta}$ is the time constant of the threshold decay. In such a way, absolute and relative refractory periods are modeled.

External inputs from the input layer are added to the right-hand side of Eq. (5) at each time step, thus incorporating the background noise or the background oscillations. Each external input has its own weight $J_{ik}^{\text{ext\_input}}$ and $\varepsilon_k(t)$, such that

$$u_i^{\text{ext\_inpu}}(t) = J_{ik}^{\text{ext\_input}} \varepsilon_{ik}(t) \tag{9}$$

It is optional to add some degree of Gaussian noise to the right-hand side of Eq. (6) to obtain a stochastic neuron model instead of a deterministic one.

Figure 16 illustrates the basic architecture of a SNN. Spiking neurons within the network can be either excitatory or inhibitory. There can be as many as 10–20% of inhibitory neurons

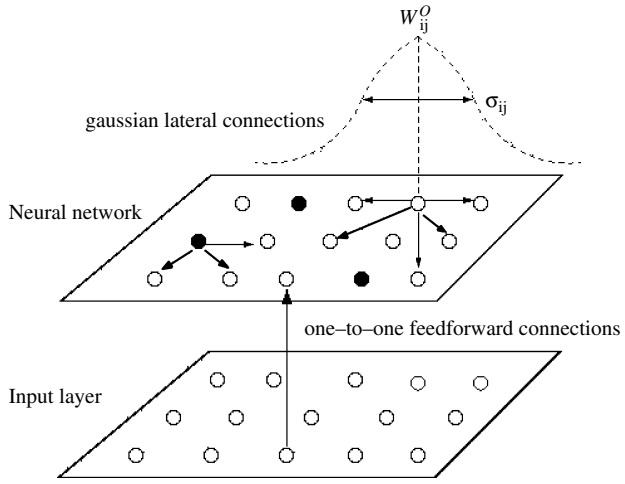

Figure 16. Architecture of the spiking neural network (SNN). About as many as 10–20% of neurons are inhibitory neurons that are randomly positioned on the grid (filled circles). Excitatory and inhibitory lateral connections decrease in strength with distance according to the Gaussian distribution. There are one-to-one feed-forward connections from the input layer.

positioned randomly on the rectangular grid of N neurons. Lateral connections between neurons have weights that decrease in value with distance from neuron i, for instance, according to a Gaussian formula, whereas the connections between neurons themselves can be established at random.

Figure 17 illustrates a record of activity of the introduced SNN. It is useful to keep a record of spiking activities of all neurons individually and in total, as well as the record of the total membrane potential, which is in fact proportional to EEG [144]. Various analytical tools are developed, for instance, for evaluation of the degree of synchrony between neurons [141] and for the evaluation of frequency spectra, like the fast Fourier transform and others [141, 145]. Presented SNN belongs among the simplest tools. There are much more detailed models of spiking neurons than SRM or the so-called integrate-and-fire (I&F) model neurons. These more detailed models include, for instance, the various ion receptor and channel kinetics [142, 146], and also include multicompartmental neuron models in which the effect of spatial—not only temporal—summation of PSPs on the neuron input surface is taken into account [147, 148].

5.2.4. Example of Relation of Genes to Particular Neural Network Parameters

Let us take as an example of the set of genes $\mathbf{G}_{\text{spec}}$ that define specific neuronal functions (e.g., epileptic behavior)—the set of genes that are presumably mutated in individuals suffering from the Childhood Absence Epilepsy (CAE). CAE is an idiopathic (i.e., arising from an unknown cause), generalized, nonconvulsive epilepsy. The main features are absence seizures. A typical absence is a nonconvulsive epileptic seizure, characterized by a brief (4–20 s) impairment of consciousness. This may happen up to ~200 times a day. Absence seizures occur spontaneously (i.e., they are not evoked by sensory or other stimuli [11, 149]). Absence is accompanied by a generalized, synchronous, bilateral, 2.5–4-Hz spike and slow-wave discharge (SWD) in the EEG. SWDs can start anywhere in the cortex, and from there they quickly spread to the entire cortex and thalamus [150].

Table 5 the genes that are most probably mutated in CAE, their coded proteins, the neuronal function or functions these proteins are responsible for, related parameters in the SNN, and a putative alterations in these functions, as well as changes in SNN parameters. Putative changes in the neural function can be derived from numerous studies performed on humans, rats, and mice [11, 142, 143, 149, 151–155]. It should be pointed out that other types of idiopathic epilepsies like the frontal lobe epilepsy (ADNFLE), temporal lobe epilepsy (TLE), juvenile myoclonic epilepsy (JME), adult myoclonic epilepsy (AME), and so forth

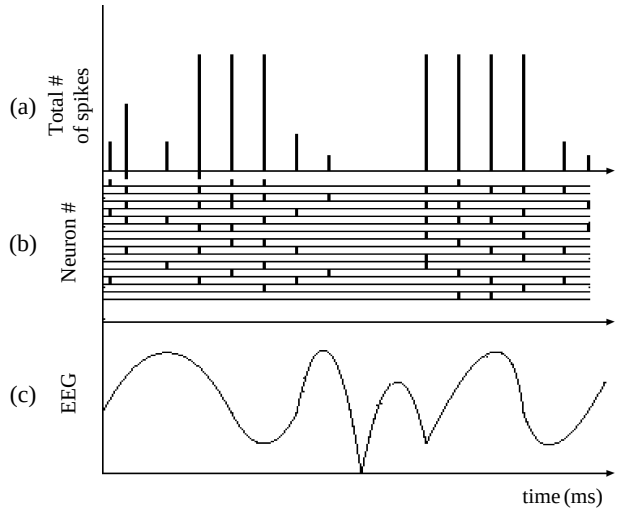

Figure 17. Temporal evolution of the network activity. (a) Total number of spikes generated by all neurons at each time step. (b) Traces of each neuron spiking activities. (c) Total sum of individual membrane potentials as a measure proportional to EEG.

Table 5. List of genes putatively mutated in CAE, their coded proteins, neuronal function these proteins are responsible for, related parameters in the SNN, and a putative alteration in this function: ? means increase, ? means decrease.

Mutated Gene	Protein	Neuronal Function and its putative Alteration	SNN Parameters
GRIK1	Ionotropic GluR5 (Kainate receptor 2 for glutamate)	Fast excitation ? or ? (depending on the place in the brain)	$A^{\text{fast\_exc}}$, $\tau^{\text{fast\_exc}}_{\text{decay/rise}}$? or ?
GABRB3	$GABA_A$ receptor $\beta 3$ subunit	Fast inhibition?	$A^{\text{fast\_inh}}$, $\tau^{\text{fast\_inh}}_{\text{decay/rise}}$?
GPHN	Gephyrin	Fast inhibition?	$A^{\text{fast\_inh}}$, $\tau^{\text{fast\_inh}}_{\text{decay/rise}}$?
CHRNA4	nAChR $\alpha 4$ subunit	Both fast and slow inhibition?	J^{inh}_{ij} ?
OPRM1	μ-Opioid receptor type 1	Firing threshold?	ϑ_0, $\tau^{\vartheta}_{\text{decay}}$?

are connected to different channelopathies and receptoropathies [153, 156]; that is, to gene mutations different from CAE, so the presented table applies only to CAE.

A more detailed model of a neuron including the receptor and channel kinetics will enable us to link mutated genes to the parameters at a more detailed level. Still, it will be a hard task to determine quantitatively the values of coefficients of mutual interactions between genes themselves and between genes and their coded proteins [i.e., the values of coefficients in Eqs. (2) and (3)]. Here we have proposed GA as a means of such optimization; however, these values also can be obtained from experimental data if they are available. In any case, experimental data will serve as an ultimate test of neuro-genetic theoretical and computational models when applied to particular problems, such as, for instance, epilepsy. Once a functional neuro-genetic model of a particular neurological disease is developed, then the effects of genetic and other parameter changes can be simulated and predicted by means of the theoretical computational model. Before any neuro-genetic causal cures are going to be administered, mutated genes should be identified and their interactions with other genes and neuronal functions known to avoid any unwanted consequences of the perspective gene therapy. The gene therapy proper is another nontrivial issue. Once it is known what should be added or removed from the cell genome, reliable nanotechnology must be developed for carrying out this operation. This applies not only for possible genetic cures of epilepsy but also for any other neurological or mental disorder.

6. CONCLUSIONS AND FUTURE DEVELOPMENT

Nanotechnology has a tremendous potential for the cure of brain diseases. This approach requires a deep understanding of chemical and information processes in the brain, in single neurons and in the nuclei of these neurons, and especially in how these processes relate to each other. In this respect, there is a need for theories and computational models to model and predict the outcome of brain abnormalities and their treatment. Neuro-genetic modeling is a promissing approach that will be further developed and applied. A new field of nanomedicine can be developed in the future to deal with nanotechnology in medicine and health care [157, 158].

In between, many nanotechnical problems must be solved. For instance, the brain, similar like other tissues, responds to alien substances with a healing process. For instance, various neural probes (usually composed of silicon) become encapsulated with glial scar tissue, which can impede with normal neuron function. From a nanomaterial point of view, nanophase materials can influence interaction with proteins and other molecules that take part in cell processes in many unwanted ways [159]. Another area of future study and possible application will be the development of nanoscale logic networks and nanochips [160, 161]. They will lead not only to an unprecedented miniaturization of conventional computers but also to miniaturization of neurocomputers and neurochips. It is not only about implementation of classical ANNs in hardware but also about the development of the so-called neuromorphic systems. Neuromorphic systems are implementations in silicon of sensory and neural systems whose architecture and design are based on neurobiology that can compete with

human senses and pattern-recognition systems and run in real time [162, 163]. Researchers in this area also work on developing communication between living vertebrate neurons and electronic systems [162].

Although a lot is known about the brain, issues about its functioning, representation, and processing of information are still subjects of an intense research. Applicability of nanotechnology will depend not only on the nanotechnological progress itself but also on the progress in understanding the brain, its dynamical behavior, and how normal and disturbed neural functions affect the brain dynamics. The nature of brain dynamics is still unknown. Some researchers find evidence of chaos, whereas some are doubtfull [164]. The main proponents of a chaotic dynamic, Freeman [165] and Tsuda [166], argue in favor of chaotic itinerancy based on EEG and other neurophysiological data. According to the picture of chaotic itinerancy, a complex system such as the (human) brain evolves by steps along a trajectory in the state space. Each step corresponds to a shift from one basin of attraction to another. Attractors represent classes for abstraction and generalization. Thus, the brain states evolve aperiodically through sequences of attractors. In a closed system, the next attractor would be chosen solely by internal dynamics. In an open system, such as the brain, external inputs interfere with internal dynamics. Moreover, because of the changes induced by learning, trajectories continually change. Chaotic itinerancy occurs in sequence of cortical states marked by state transitions that appear in temporal discontinuities in neural activity patterns [165].

Experimental EEG data show that the entire cerebral cortex is constantly wandering in the fractal distributions of phase transitions that give the $1/f^{\alpha}$ form of the temporal and spatial frequency spectra (with $\alpha \in (1, 3)$, [165]). From this type of frequency spectrum, it appears that the brain maintains a state of self-organized criticality (SOC) [167]. The SOC state can form the basis of the brain is capacity to rapidly adjust to new external and internal stimuli. State changes resembling phase transitions occur continually everywhere in cortex at scales ranging from milimeters to $\sim$0.1 m. Local neural activity can trigger a massive state change.

However, several issues of caution should be pointed out. In spite of the compelling evidence for SOC in the brain, the nature of the critical state is still unknown in neurobiological interpretation. The spatial and temporal power spectral densities (PSDs) often show the $1/f^{\alpha}$ form; however, more often this form is broken down as a result of distortions by clinically defined peaks. Therefore, the measurements of α vary widely. Aperiodic oscillations giving the $1/f^{\alpha}$ PSD are commonly referred to as chaotic. However, the brain activity is not at all consistent with low-dimensional deterministic chaos [164, 168]. It is high dimensional, noisy, non-Gaussian, and nonstationary [165]. Therefore, the conditions for the assessment of this type of dynamics are difficult to be met. Moreover, brains are open systems driven by stochastic input. Thus, it seems that the brain activity can hardly conform to the mathematical definitions of chaos. Whether the term chaotic itinerancy (or any other term from the chaotic vocabulary) is appropriate to describe state transitions in brain and cortex in particular remains open to challenge. Thus, the complex spatio-temporal activity data from the brain still await explanation.

Another issue that is unresolved at present and that will be highly relevant when it will be possible to nanotechnologically enter the brain is the issue of consciousness and other mental phenomena. Neurobiologists are trying to identify the so-called neural correlates of consciousness and to gather experimental data in support of the stream of transient semiglobal coherencies in brain electrical activity being (somehow) the basis for consciousness [169–171]. In his influential book *Shadows of the Mind*, physicist Roger Penrose brings the problem of explaining consciousness to the domain of physics [172]. His critics question the competence of physics ever having anything of importance to say about mental phenomena in general, and consciousness in particular (http://psyche.cs.monash.edu.au/psyche-index-v2.html#som). The grounds for this criticism vary, ranging from computational to neurobiological arguments. In his response to this critique [173], Penrose states that he certainly does not expect to find any answers in (contemporary) subatomic physics. Instead, he has been arguing for a new physics, for a radical upheaval in the very basis of physical theory. According to him, at present, any scientific (including physical) theory does not help

us to come to terms with the puzzle of mentality, including consciousness within such a physically determined universe. Even at this point, the point of mystery of mentality, there is not a general agreement among scientists. Some think that there is no mystery at all and that the consciousness and other mental phenomena emerge from a particular underlying basis, be it the specific computations [174] or specific properties of the brain processes [175]. In another influential book written on consciousness [176], philosopher and mathematician, David J. Chalmers clearly argues that consciousness and mentality are indeed genuinly puzzling and are not explainable by present theories. If one takes consciousness seriously, Chalmers says, one has to go beyond a strict materialist framework. The fundamental laws linking the physical and the experiential are yet to be discovered, although he attempts to define and search for them. Penrose also tries to link together the physical and experiential, and he sees the link in a new physical theory based on the union of Einstein's general relativity with quantum theory. However, he writes [173], we do not yet know the very form this new theory must take. It may have a character very different from that of a traditional physical theory. Penrose does not believe that any real progress will be achieved toward solving the mysteries of how mental phenomena fit in with the physical universe until there are some important changes in our picture of physical reality. However, already at this point, Chalmers asks why quantum processes (or any other specific physical processes) in microtubules (or any other brain substructures) should give rise to consciousness, any more than specific computational processes [177]. What all scientists agree on is that mentality is causally linked to the brain, whether its basis is particular computations or particular biological or physical processes. Brain nanotechnology will interfere with all these processes; therefore, a potential danger lies here for unforeseen consequences. However, nanotechnology by targeted and controlled manipulation of selected molecules might help in the search for those very crucial material phenomena in the brain that might be causally linked to consciousness and subjective experience. In any case, we expect that serious ethical issues will have to be dealt with in the future brain nanotechnology.

REFERENCES

1. M. Arbib, Ed., "The Handbook of Brain Theory and Neural Networks," 2nd edn. MIT Press, Cambridge, MA, 2003.
2. S. Amari, *Proc. IEEE* 78, 1143 (1990).
3. T. Kohonen, *Proc. IEEE* 78, 1464 (1990).
4. G. Carpenter and S. Grossberg, "Pattern Recognition by Self-Organizing Neural Networks." MIT Press, Cambridge, MA, 1991.
5. C. M. Bishop, "Neural Networks for Pattern Recognition." Oxford Univ. Press, Oxford, 1995.
6. S. Amari and N. Kasabov, Eds., "Brain-Like Computing and Intelligent Information Systems." Springer-Verlag, Singapore, 1998.
7. P. Baldi and S. Brunak, "Bioinformatics—A Machine Learning Approach," 2nd edn. MIT Press, Cambridge, MA, 2001.
8. J. Bower and H. Bolouri, Eds., "Computational Modelling of Genetic and Biochemical Networks," MIT Press, Cambridge, MA, 2001.
9. H. de Jong, *J. Comput. Biol.* 9, 67 (2002).
10. N. Kasabov, "Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering," MIT Press, Cambridge, MA, 1996.
11. V. Crunelli and N. Leresche, *Nat. Rev. Neurosci.* 3, 371 (2002).
12. H.-H. Ropers, M. Hoeltzenbein, V. Kalscheuer, H. Yntema, B. Hamel, J.-P. Fryns, J. Chelly, M. Partington, J. Gecz, and C. Moraine, *Trends Genet.* 19, 316 (2003).
13. Y. Zhang, F. Schlachetzki, and W. M. Pardridge, *Mol. Ther.* 7, 11 (2003).
14. E. R. Kandel, J. H. Schwartz, and T. M. Jessell, "Principles of Neural Science," 4th edn. McGraw-Hill, New York, 2000.
15. D. Hebb, "The Organization of Behavior." Wiley, New York, 1949.
16. W. C. Abraham, B. Logan, J. M. Greenwood, and M. Dragunow, *J. Neurosci.* 22, 9626 (2002).
17. H. Z. Shouval, M. F. Bear, and L. N Cooper, *Proc. Natl. Acad. Sci. USA* 99, 10831 (2002).
18. H. Markram, J. Lübke, M. Frotscher, and B. Sakmann, *Science* 275, 213 (1997).
19. W. C. Abraham and M. F. Bear, *Trends Neurosci.* 19, 126 (1996).
20. E. Bienenstock, L. N. Cooper, and P. Munro, *J. Neurosci.* 2, 32 (1982).
21. P. Jedlicka, *Bratislava Med. Lett.* 103, 137 (2002).
22. L. Benuskova, M. E. Diamond, and F. F. Ebner, *Proc. Natl. Acad. Sci. USA* 91, 4791 (1994).
23. L. Benuskova, V. Rema, M. Armstrong-James, and F. F. Ebner, *Proc. Natl. Acad. Sci. USA* 98, 2797 (2001).

24. T. V. P. Bliss, *Nature* 401, 25 (1999).
25. A. Kral, R. Hartmann, J. Tillein, S. Heid, and R. Klinke, *Cerebral Cortex* 12, 797 (2002).
26. J. J. Hopfield, *Proc. Natl. Acad. Sci. USA* 79, 2554 (1982).
27. E. Rodriguez, N. George, J.-P. Lachaux, J. Martinerie, B. Renault, and F. J. Varela, *Nature* 397, 434 (1999).
28. W. Maass and C. M. Bishop, Eds., "Pulsed Neural Networks." MIT Press, Cambridge, MA, 1999.
29. F. Rieke, D. Warland, R. de Ruyter van Steveninck, and W. Bialek, "Spikes—Exploring the Neural Code," MIT Press, Cambridge, MA, 1996.
30. M. Fabre-Thorpe, A. Delorme, C. Marlot, and S. Thorpe, *J. Cogn. Neurosci.* 13, 171 (2001).
31. J. Huxter, N. Burgess, and J. O'Keefe, *Nature* 425, 828 (2003).
32. P. Fries, P. R. Roelfsema, A. K. Engel, P. Koenig, and W. Singer, *Proc. Natl. Acad. Sci. USA* 94, 12699 (1997).
33. M. N. Shadlen and W. T. Newsome, *J. Neurosci.* 18, 3870 (1998).
34. N. Kasabov, "Evolving Connectionist Systems. Methods and Applications in Bioinformatics, Brain Study and Intelligent Machines." Springer-Verlag, London, 2003.
35. T. Kohonen, "Self-Organizing Maps," 2nd edn. Springer-Verlag, Berlin, 1997.
36. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Eds., Learning internal representations by error propagation, "Parallel Distributed Processing: Explorations in the Microstructure of Cognition" (D. E. Rumelhart and J. L. McClelland, Eds.). MIT Press/Bradford Books, Cambridge, MA, 1986.
37. P. Werbos, *Proc. IEEE* 87, 10 (1990).
38. J. S. Albus, *Trans. ASME J. Dynam. Syst. Meas. Control* 27, 220 (1975).
39. B. Fritzke, *Adv. Neural Information Processing Syst.* 7, 625 (1995).
40. D. Saad, Ed., "On-Line Learning In Neural Networks." Cambridge Univ. Press, Cambridge, 1999.
41. B. Hassibi and D. G. Stork, in "Advances in Neural Information Processing Systems" (D. S. Touretzky and Morgan Kaufmann, Eds.), Vol. 4, pp. 164–171. San Francisco, CA, 1992.
42. S. Schaal and C. Atkeson, *Neural Comput.* 10, 2047 (1998).
43. G. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, *IEEE Trans. Neural Networks* 3, 698 (1991).
44. X. Yao, *Intl. J. Neural Syst.* 4, 203 (1993).
45. D. B. Fogel, "Evolutionary Computation—Toward a New Philosophy of Machine Intelligence." IEEE Press, New York, 1995.
46. M. Watts and N. Kasabov, in "Proceedings of the 5th International Conference on Neural Information Processing" (S. Usui and T. Omori, Eds.), Vol. 2, pp. 793–796. IOS Press, Kitakyushu, 1998.
47. S. Grossberg, *J. Stat. Phys.* 1, 319 (1969).
48. S. Grossberg, "Studies of Mind and Brain." Reidel, Boston, 1982.
49. F. Rosenblatt, "Principles of Neurodynamics," Spartan Books, New York, 1962.
50. M. Arbib, "The Metaphorical Brain—An Introduction to Cybernetics as Artificial Intelligence and Brain Theory," Wiley Interscience, New York, 1972.
51. M. Arbib, "Brains, Machines and Mathematics." Springer, Berlin, 1987.
52. T. M. Heskes and B. Kappen, in "Mathematic Foundations of Neural Networks," pp. 199–233. Elsevier, Amsterdam, 1993.
53. C. Fahlman and C. Lebiere, "The Cascade-Corelation Learning Architecture, in Advances in Neural Information Processing Systems," (D. Turetzky and M. Kaufmann, Eds.), Vol. 2, 1990.
54. G. A. Rummery and M. Niranjan, in Cambridge University, Engineering Department, 1994.
55. A. Sankar and R. J. Manmone, *IEEE Trans. Comput.* 42, 291 (1993).
56. Y. LeCun, J. S. Denker, and S. A. Solla, in "Advances in Neural Information Processing Systems" (D. S. Touretzky and Morgan Kaufmann, Eds.), pp. 598–605. San Francisco, CA, 1990.
57. M. Ishikawa, *Neural Networks* 9, 501 (1996).
58. A. Robins, *Connection Sci.* 8, 259 (1996).
59. D. Miller, J. Zurada, and J. H. Lilly, in "Proceedings of the IEEE International Confference on Neural Networks," Vol. 1, pp. 448–454. 1996.
60. V. Kecman, "Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models (Complex Adaptive Systems)." MIT Press, Cambridge, MA, 2001.
61. G. E. Hinton, *Artificial Intelligence* 40, 185 (1989).
62. G. E. Hinton, *Artificial Intelligence* 46, 1 (1990).
63. G. G. Towell and J. W. Shavlik, *Machine Learning* 13, 71 (1993).
64. G. G. Towell and J. W. Shawlik, *Artificial Intelligence* 70, 119 (1994).
65. I. Cloete and J. Zurada, Eds., "Knowledge-Based Neurocomputing." MIT Press, Cambridge, MA, 2000.
66. G. G. Towell, J. W. Shawlik, and M. Noordewier, "Refinement of Approximate Domain Theories by Knowledge-Based Neural Networks," in Proceedings of 8th National Conference AI AAA'90, pp. 861–866, 1990.
67. T. Furuhashi, K. Nakaoka, and Y. Uchikawa, "A new Approach to Genetic Based Machine Learning and an Efficient Finding of Fuzzy Rules," Proceedings of WWW'94 Workshop, pp. 114–122, 1994.
68. R. Jang, *IEEE Trans. Syst. Man Cybernetics* 23, 665 (1993).
69. T. Yamakawa, H. Kusanagi, E. Uchino, and T. Miki, "A new Effective Algorithm for Neo Fuzzy neuron Model," Proceedings of the 5th IFSA World Congress, pp. 1017–1020, 1993.
70. W. Hauptmann and K. Heesche, "A Neural Network Topology for Bidirectional Fuzzy-Neuro Transformation," Proceedings of FUZZ-IEEE/IFES, pp. 1511–1518, 1995.
71. N. Kasabov, *Neurocomputing* 13, 95 (1996).

72. Y. Hayashi, in "Advances in Neural Information Processing Systems" (R. P. Lippman, J. E. Moody, and D. S. Touretzky, Eds.), Vol. 3, pp. 578–584. Morgan Kaufmann, San Mateo, CA, 1991.
73. N. Kasabov, *Fuzzy Sets Syst.* 82, 2 (1996).
74. W. Duch, R. Adamczak, and K. Grabczewski, *Neural. Proc. Lett.* 7, 211 (1998).
75. N. Kasabov, in "Methodologies for the Conception, Design and Application of Soft Computing" (T. Yamakawa and G. Matsumoto, Eds.), pp. 271–274. World Scientific, Singapore, 1998.
76. S. Mitra and Y. Hayashi, *IEEE Trans. Neural Networks* 11, 748 (2000).
77. N. Kasabov, *IEEE Trans. Syst. Man Cybernetics B. Cybernetics* 31, 902 (2001).
78. S. Amari, *IEEE Trans. Electronic Comput.* 16, 299 (1967).
79. D. W. Aha, D. Kibler, and M. K. Albert, *Machine Learning* 6, 37 (1991).
80. M. T. Mitchell, R. Keller, and S. Kedar-Cabelli, *Machine Learning* 1, 47 (1997).
81. S. L. Salzberg, "Learning with Nested Generalized Exemplars." Kluwer, Boston, MA, 1990.
82. S. Haykin, "Neural Networks—A Comprehensive Foundation." Prentice Hall, Engelwood Cliffs, NJ, 1994.
83. L. A. Zadeh, *Information Control* 8, 338 (1965).
84. E. Mamdani, *IEEE Trans. Comput.* 26, 1182 (1997).
85. T. Takagi and M. Sugeno, *IEEE Trans. Syst. Man Cybernetics* 15, 116 (1985).
86. J. M. Mendel, "Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions." Prentice Hall, New York, 2001.
87. J. G. Taylor, "The Race for Consciousness." MIT Press, Cambridge, MA, 1999.
88. W. Freeman, "Neurodynamics." Springer-Verlag, London, 2000.
89. D. E. Goldberg, "Genetic Algorithms in Search, Optimisation and Machine Learning." Addison-Wesley, Reading, PA, 1989.
90. J. Koza, "Genetic Programming." MIT Press, Cambridge, MA, 1992.
91. J. H. Holland, "Adaptation in Natural and Artificial Systems." Univ. of Michigan Press, Ann Arbor, 1975.
92. J. H. Holland, "Emergence." Oxford Univ. Press, Oxford, 1998.
93. D. Fogel, L. Fogel, and V. Porto, *Biol. Cybernetics* 63, 487 (1990).
94. J. M. Baldwin, *Am. Nat.* 30, 441 (1896).
95. V. Kvasnicka and J. Pospichal, in "Advances in Soft Computing—Engineering Design and Manufacturing" (R. Roy, T. Furuhashi, and P. K. Chawdhry, Eds.), pp. 481–496. Springer-Verlag, London, 1999.
96. K. Richardson, "The Making of Intelligence." Phoenix, London, 1999.
97. A. Newell and H. A. Simon, "Human Problem Solving." Prentice Hall, Engelwood Cliffs, NJ, 1972.
98. H. C. Plotkyn, "The Nature of Knowledge." Penguin, London, 1994.
99. E. Rosch and B. B. Lloyd, Eds., "Cognition and Categorization." Lawrence Erlbaum, Mahweh, NJ, 1978.
100. E. E. Smith and D. L. Medin, "Categories and Concepts." Harvard Univ. Press, Cambridge, MA, 1981.
101. D. Fogel, "Blondie 24 Playing at the Edge of AI." Morgan Kaufmann, San Diego, CA, 2002.
102. J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, and G. G. Sutton, *Science* 291, 1304 (2001).
103. C. Elegans, Sequencing Consortium, *Science* 282, 2012 (1998).
104. F. Crick, *Nature* 227, 561 (1970).
105. S. Kauffman, *J. Theoret. Biol.* 44, 167 (1974).
106. J. L. deRisi, V. R. Iyer, and P. O. Brown, *Science* 275, 680 (1997).
107. B. Sobral, in "From Jay Lush to Genomics: Visions for Animal Breeding and Genetics" (J. M. Dekkers, S. J. Lamont, M. F. Rothschild, Eds.). Iowa State Univ. Press, Ames, 1999.
108. P. A. Pevzner, "Computational Molecular Biology: An Algorithmic Approach." MIT Press, Cambridge, MA, 2000.
109. J. R. Koza, W. Mydlowec, G. Lanza, J. Yu, and M. A. Keane, "Reverse Engineering of Metabolic Pathways from Observed Data using Genetic Programming," in Proceedings of the Pacific Symposium on Biocomputing, Vol. 6, pp. 434–445. 2001.
110. J. Collado-Vides and R. Hofestadt, Eds., "Gene Regulation and Metabolism. Post-Genomic Computational Approaches." MIT Press, Cambridge, MA, 2002.
111. L. Hunter, *Can. Artificial Intelligence* 35, 10 (1994).
112. J. Dow, G. Lindsay, and J. Morrison, "Biochemistry Molecules, Cells and the Body." Addison-Wesley, Boston, MA, 1995.
113. J. Collado-Vides, B. Magasanik, and T. F. Smith, Eds., "Integrative Approaches to Molecular Biology." MIT Press, Cambridge, MA, 1996.
114. P. D'Haeseleer, X. Wen, S. Fuhrman, and R. Somogyi, "Linear Modeling of mRNA Expression Levels during CNS Development and Injury," Proceedings of the Pacific Symposium on Biocomputing, Vol. 4, pp. 41–52. Hawaii, 1999.
115. P. D'Haeseleer, S. Liang, and R. Somogyi, *Bioinformatics* 16, 707 (2000).
116. G. Fogel and D. Corne, "Evolutionary Computation for Bioinformatics." Morgan Kaufmann, San Francisco, CA, 2003.
117. T. Akutsu, S. Miyano, and S. Kuhara, "Identification of Genetic Networks from a Small Number of Gene Expression Patterns under the Boolean Network Model," Proceedings of the Pacific Symposium on Biocomputing, Vol. 4, pp. 17–28. Hawaii, 1999.
118. S. Gomez, S. Lo, and A. Rzhetsky, *Genetics* 159, 1291 (2001).
119. A. Lindlof and B. Olsson, *Information Sci.* 146, 103 (2002).
120. J. Vohradsky, *J. Biol. Chem.* 276, 36168 (2001).
121. J. Vohradsky, *FASEB J.* 15, 846 (2001).

122. S. Ando, E. Sakamoto, and H. Iba, "Evolutionary Modelling and Inference of Genetic Networks," Proceedings of the 6th Joint Conference on Information Sciences, pp. 1249–1256. Cary, NC, 2002.
123. Mimura and H. Iba, "Inference of a Gene Regulatory Network by Means of Interactive Evolutionary Computing," Proceedings of the 6th Joint Conference on Information Sciences, pp. 1243–1248. Cary, NC, 2002.
124. K. W. Kohn and D. S. Dimitrov, in "Computer Modeling and Simulation of Complex Biological Systems" (S. S. Iyengar, Ed.), pp. 101–123. CRC Press, Boca Raton, FL, 1998.
125. J. Vides, B. Magasanik, and T. Smith, "Integrated Approaches to Molecular Biology." MIT Press, Cambridge, MA, 1996.
126. M. A. Savageau, *Chaos* 11, 142 (2001).
127. G. Marnellos and E. D. Mjolsness, in "Modeling Neural Development" (A. vanOoyen, Ed.), pp. 27–48. MIT Press, Cambridge, MA, 2003.
128. L. F. A. Wessels, E. P. vanSomeren and M. J. T. Reinders, in "Proceedings of the Pacific Symposium on Biocomputing," p. 508–519. Hawaii, 2001.
129. N. Kasabov and D. Dimitrov, in "Proceedings of the ICONIP'2002—International Conference on Neuro-Information Processing." IEEE Press, Singapore, 2002.
130. N. Kasabov and Q. Song, DENFIS: *IEEE Trans. Fuzzy Syst.* 10, 144 (2002).
131. T. MathWorks, "Neural Network Toolbox User's Guide," Vol. 4. The Math Works Inc., 2001.
132. W. Enard, P. Khaitovich, J. Klose, S. Zoelner, F. Heisig, and S. Paabo, *Science* 296, 340 (2002).
133. M. Caceres, J. Lachuer, M. A. Zapala, J. C. Redmond, L. Kudo, D. H. Geschwind, D. J. Lockhart, T. M. Preuss, and C. Barlow, *Proc. Natl. Acad. Sci. USA* 100, 13030 (2003).
134. S. Savage-Rumbaugh and R. Lewin, "Kanzi: The Ape at the Brink of the Human Mind." Wiley, New York, 1994.
135. R. Somogyi, S. Fuhrman, and X. Wen, in "Computational Modeling of Genetic and Biochemical Networks" (J. M. Bower and H. Bolouri, Eds.), pp. 119–157. MIT Press, Cambridge, MA, 2001.
136. In "Genes and Disease." National Centre for Biotechnology Information (NCBI), 2003; http://www.ncbi.nlm.nih.gov/books/bv.fcgi?call=bv.View..ShowSection&rid=gnd.chapter.75.
137. R. Morita, E. Miyazaki, C. G. Fong, X.-N. Chen, J. R. Korenberg, A. V. Delgado-Escueta, and K. Yamakawa, *JH8, Biochem. Biophys. Res. Comm.* 248, 307 (1998).
138. L. Benuskova, S. G. Wysoski, and N. Kasabov, "Neuro-Genetic Model of Spiking Neural Networks," in preparation (2004).
139. J. H. Holland, "Adaptation in Natural and Artificial Systems." Univ. of Michigan Press, Ann Arbor, MI, 1975.
140. D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning." Addison-Wesley, Reading, MA, 1989.
141. W. Gerstner and W. M. Kistler, "Spiking Neuron Models." Cambridge Univ. Press, Cambridge, 2002.
142. A. Destexhe, *J. Neurosci.* 18, 9099 (1998).
143. A. V. Semyanov, *Neurophysiology* 34, 71 (2002).
144. J. W. Freeman, "Mass Action in the Nervous System." Academic Press, New York, 1975.
145. W. J. Freeman, M. D. Holmes, B. V. Burke, and S. Vanhatalo, *Clin. Neurophysiol.* 114, 1053 (2003).
146. P. Kudela, P. J. Franaszcuk, and G. K. Bergey, *Biol. Cybernetics* 88, 276 (2003).
147. K.-H. Yang, P. J. Franaszczuk, and G. K. Bergey, *Biol. Cybernetics* 89, 242 (2003).
148. J. M. Bower and D. Beeman, "The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System," 2nd ed. TELOS/Springer-Verlag, New York, 1998.
149. C. Marini, L. A. Harkin, R. H. Wallace, J. C. Mulley, I. E. Scheffer, and S. F. Berkovic, *Brain* 126, 230 (2003).
150. H. K. M. Meeren, J. P. M. Pijn, E. L. J. M. VanLuijtelaar, A. M. L. Coenen, and F. H. LopesdaSilva, *J. Neurosci.* 22, 1480 (2002).
151. A. Contractor, G. T. Swanson, A. Sailer, S. O'Gorman, and S. F. Heineman, *J. Neurosci.* 20, 8269 (2000).
152. R. A. Deisz, *Neuropharmacology* 38, 1755 (1999).
153. R. M. Gardiner, *Epilepsy Res.* 36, 91 (1999).
154. G. E. Homanics, T. DeLorey, L. L. Firestone, J. J. Quinlan, and A. Handforth, *Proc. Natl. Acad. Sci. USA* 94, 4143 (1997).
155. S. Jones, S. Sudweeks, and J. L. Yakel, *Trends Neurosci.* 22, 555 (1999).
156. O. K. Steinlein and J. L. Noebeles, *Curr. Opin. Genet. Dev.* 10, 286 (2000).
157. R. A. Freitas, "Nanomedicine, Volume I: Basic Capabilities." Landes Bioscience, Georgetown, TX, 1999.
158. R. A. Freitas, "Nanomedicine, Volume IIA: Biocompatibility." Landes Bioscience, Georgetown, TX, 2003.
159. T. J. Webster, M. C. Waid, J. L. McKenzie, R. L. Price, and J. U. Ejiofor, *Nanotechnology* 15, 48 (2004).
160. A. S. Sadek, K. Nikolic, and M. Forshaw, *Nanotechnology* 15, 192 (2004).
161. G. Bauer, J. Hassmann, H. Walter, J. Haglmueller, C. Mayer, and T. Schalkhammer, *Nanotechnology* 15, 1289 (2004).
162. L. S. Smith and A. Hamilton, Eds., "Neuromorphic Systems: Engineering Silicon from Neurobiology," Progress in Neural Processing. World Scientific Publishing, London, 1998.
163. P. Tikovic, M. Voros and D. Durackova, *J. Elec. Eng.* 52, 68 (2001).
164. J. Theiler, *Phys. Lett. A* 196, 335 (1995).
165. W. J. Freeman, *Chaos* 13, 1 (2003).
166. I. Tsuda, *Behav. Brain Sci.* 24, 793 (2001).
167. P. Bak, C. Tang, and K. Wiesenfeld, *Phys. Rev. Lett.* 59, 381 (1987).
168. L. Benuskova, M. Kanich, and A. Krakovska, in "Proceedings of the World Congress on Neuroinformatics" (F. Rattay, Ed.). ARGESIM/ASIM-Verlag, Vienna, 2001.

169. C. Koch and F. Crick, in "Large-Scale Neuronal Theories of the Brain" (C. Koch and J. L. Davis, Eds.). MIT Press, Cambridge, MA, 1994.
170. W. Singer, in "Understanding Representation in the Cognitive Sciences" (A. Riegler, M. Peschl, and A. vonStein, Eds.). Kluwer Academic/Plenum Publishers, New York, 1999.
171. G. M. Edelman and G. Tononi, "Consciousness. How Matter Becomes Imagination." Penguin Books, London, 2000.
172. R. Penrose, "Shadows of the Mind: A Search for the Missing Science of Consciousness." Oxford Univ. Press, Oxford, 1994.
173. R. Penrose, Beyond the Doubting of a Shadow, *PSYCHE:* 2 (1996); http://psyche.cs.monash.edu.au/v2/psyche-2-23-penrose.html.
174. D. C. Dennett, "Consciousness Explained." Penguin Books, New York, 1991.
175. J. Searle, "Consciousness and Language." Cambridge Univ. Press, Cambridge, MA, 2002.
176. D. J. Chalmers, "The Conscious Mind: In Search of a Fundamental Theory." Oxford Univ. Press, Oxford, 1996.
177. D. J. Chalmers, Minds, Machines, and Mathematics, *PSYCHE:* 2 (1996); http://psyche.cs.monash.edu.au/v2/psyche-2-09-chalmers.html.