60. Brain, Gene, and Quantum Inspired Computational Intelligence

Nikola Kasabov

This chapter discusses opportunities and challenges for the creation of methods of computational intelligence (CI) and more specifically – artificial neural networks (ANN), inspired by principles at different levels of information processing in the brain: cognitive, neuronal, genetic, and quantum, and mainly, the issues related to the integration of these principles into more powerful and accurate CI methods. It is demonstrated how some of these methods can be applied to model biological processes and to improve our understanding in the subject area; generic CI methods being applicable to challenging generic AI problems. The chapter first offers a brief presentation of some principles of information processing at different levels of the brain and then presents brain inspired, gene inspired, and quantum inspired CI. The main contribution of the chapter, however, is the introduction of methods inspired by the integration of principles from several levels of information processing, namely:

1. A computational neurogenetic model that in one model combines gene information related to spiking neuronal activities.
2. A general framework of a quantum spiking neural network (SNN) model.
3. A general framework of a quantum computational neurogenetic model (CNGM).

Many open questions and challenges are discussed, along with directions for further research.

SPIN: 12742608 (Springer Handbook of Bio-/Neuroinformatics)
MS ID: hb23-060 Proof 1
Created on: 17 April 2013 10:50 CET

60.1 ~~Introduction~~

The[TS0] [TS1] brain is a dynamic information processing system that evolves its structure and functionality in time through information processing at different levels – Fig. 60.1: quantum, molecular (genetic), single neuron, ensemble of neurons, cognitive, evolutionary.

Principles from each of these levels have been already used as inspiration for CI methods, and more specifically – for methods of ANN. The chapter focuses on the interaction between these levels and mainly on how this interaction can be modeled and how it can be used in principle to improve existing CI methods and for a better understanding of brain, gene, and quantum processes.

At the quantum level, particles (atoms, ions, electrons, etc.), which make every molecule in the material world, move continuously, being in several states at the same time, and are characterized by probability, phase, frequency, and energy.

At a molecular level, RNA and protein molecules evolve in a cell and interact in a continuous way, based on the information stored in the DNA and on external factors, and affect the functioning of a cell (neuron) under certain conditions.

At the level of a neuron, the internal information processes and the external stimuli cause the neuron to produce a signal that carries information to be transferred to other neurons.

At the level of neuronal ensembles, all neurons operate in a *concert*, defining the function of the ensemble, for instance the perception of a spoken word.

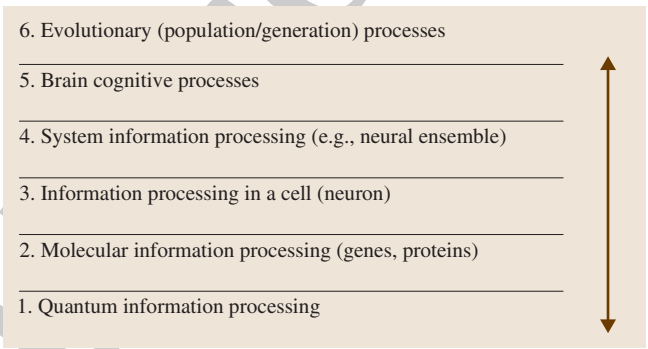

Fig. 60.1 Levels of information processing in the brain and the interaction between the levels

At the level of the whole brain, cognitive processes take place, such as language and reasoning, and global information processes are manifested, such as consciousness.

At the level of a population of individuals, species evolve through evolution, changing the genetic DNA code for a better adaptation.

The information processes at each level shown in Fig. 60.1 are very complex and difficult to understand, but much more difficult to understand is the interaction between the different levels. It may be that understanding the interaction through its modeling would be a key to understanding each level of information processing in the brain and perhaps the brain as a whole. Using principles from different levels in one ANN CI model and modeling their relationship can lead to a next generation of ANN as more powerful tools to understand the brain and to solve complex problems.

Some examples of CI models that combine principles from different levels shown in Fig. 60.1 are: computational neurogenetic models [60.1–3], quantum inspired CI and ANN [60.4, 5], and evolutionary models [60.6, 7]. Suggestions are made that modeling of higher cognitive functions and consciousness in particular can be achieved if principles from quantum information processing are considered [60.8, 9]. There are many issues and open questions to be addressed when creating CI methods that integrate principles from different levels; some of these are presented in this chapter.

In Sect. 60.2 models inspired by information processes in the brain, which include local learning evolving connectionist systems (ECOS) and SNN are discussed briefly. Section 60.3 presents CI methods inspired by genetic information processes, mainly models of gene regulatory networks (GRN). In Sect. 60.4, the issue of combining neuronal with genetic information processing is discussed and the principles of CNGM are presented. Section 60.5 presents some ideas behind quantum inspired CI. Section 60.6 presents a model of a quantum inspired SNN and offers a theoretical framework for the integration of principles from quantum, -genetic, and neuronal information processing. Section 60.7 concludes the chapter with more open questions and challenges for the future.

SPIN: 12742608 (Springer Handbook of Bio-/Neuroinformatics)
MS ID: hb23-060 Proof 1
Created on: 17 April 2013 10:50 CET

[TS0] Please provide the index entry.
[TS1] The name "Introduction" is not allowed for the first section. Please supply a different name.

Editor's or typesetter's annotations (will be removed before the final TeX run)

60.2 CI and ANN Models Inspired by Neuronal and Cognitive Processes in the Brain

Many CI methods, in particular ANN, are brain inspired (using some principles from the brain), or brain-like (more biologically plausible models, usually developed to model a brain function) [60.1, 10–15]. Examples are: models of single neurons and neural network ensembles [60.16–22], cognitive ANN models [60.14, 15, 23, 24], etc.

These models have been created with the goals of:

- Modeling and understanding brain functions.
- Creating powerful methods and systems of CI for solving complex problems in all areas of science and the humanity.

In this section we present only two groups of models, namely ECOS and SNN ,as they will be used in other sections to create models that incorporate principles from other levels of information processing.

60.2.1 Local, Knowledge-Based Learning Evolving Connectionist Systems – Weakly Brain Inspired Models

ECOS are adaptive, incremental learning and knowledge representation systems that evolve their structure and functionality, where there is a connectionist architecture in the core of a system that consists of neurons (information processing units) and connections between them [60.25]. ECOS is a CI system based on neural networks, but using other techniques of CI, that operates continuously in time and adapts its structure and functionality through continuous interaction with the environment and with other systems. The adaptation is defined through:

1. A set of evolving rules.
2. A set of parameters (*genes*) that are subject to change during the system operation.
3. An incoming continuous flow of information, possibly with unknown distribution.
4. Goal (rationale) criteria (also subject to modification) that are applied to optimize the performance of the system over time.

ECOS learning algorithms are inspired by brain-like information processing principles, e.g.,

1. They evolve in an open space, where the dimensions of the space can change.
2. They learn via incremental learning, possibly in an on-line mode.
3. They learn continuously in a lifelong learning mode.
4. They learn both as individual systems and as an evolutionary population of such 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 evolve different types of knowledge representation from data, mostly a combination of memory-based and symbolic knowledge.

Many ECOS have been suggested so far, where the structure and the functionality of the models evolve through incremental, continuous learning from incoming data, sometimes in an on-line mode, and through interaction with other models and the environment. Examples are: growing SOMs [60.17], growing gas [60.26], RAN [60.27], growing RBF networks [60.28, 29], FuzzyARTMAP [60.14], EFuNN [60.25, 30, 31], DENFIS [60.32], and many more.

A block diagram of EFuNN is given in Fig. 60.2. It is used to model GRN in Sect. 60.5. At any time of the EFuNN continuous incremental learning, rules can be derived from the structure, which rules represent clusters of data and local functions associated with these clusters

IF < data is in cluster N_{cj},
defined by a cluster center N_j,
a cluster radius R_j
and a number of examples
N_{jexamp} in this cluster >
THEN < the output function is F_c > (60.1)

In the case of DENFIS, first-order local fuzzy rule models are derived incrementally from data, for example,

IF < the value of $\times 1$ is in the area defined by
a Gaussian membership function with a center
at 0.1 and a standard deviation of 0.05 > TS3,
AND < the value of $\times 2$ is in the area defined
by a Gaussian function with parameters
(0.25, 0.1) respectively >

TS2 Please provide a shorter running title.
TS3 > added. Please confirm.

Editor's or typesetter's annotations (will be removed before the final TeX run)

SPIN: 12742608 (Springer Handbook of Bio-/Neuroinformatics)
MS ID: hb23-060 Proof 1
Created on: 17 April 2013 10:50 CET

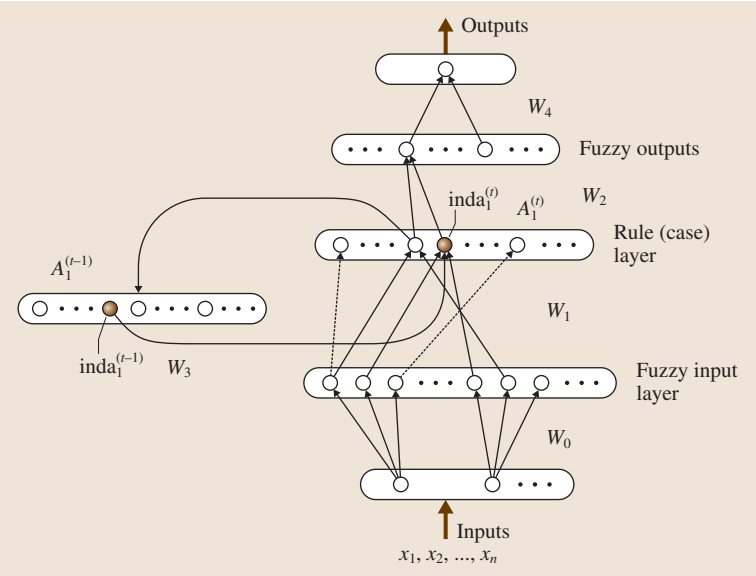

Fig. 60.2 An EFuNN architecture with a short term memory and feedback connections [60.33]. It is used in Sect. 60.5 to model GRN with inputs being the expression of genes at a time (t) and the outputs being the expression of genes/proteins at time ($t + \mathrm{d}t$)

THEN $<$ the output y is calculated by the formula

$$y = 0.01 + 0.7 \times 1 + 0.12 \times 2 > \quad (60.2)$$

In the case of EFuNN, local simple fuzzy rule models are derived, for example,

IF $\times 1$ is (Medium 0.8) and $\times 2$ is (Low 0.6)
THEN y is (High 0.7), radius $R = 0.24$;

$$N_{\text{examp}} = 6\,, \quad (60.3)$$

where: low, medium and high are fuzzy membership functions defined for the range of each of the variables $\times 1$, $\times 2$, and y; the number and the type of the membership functions can either be deduced from the data through learning algorithms, or can be predefined based on human knowledge [60.34, 35]; R is the radius of the cluster; and N_{examp} is the number of examples in the cluster.

A further development of the EFuNN and the DENFIS local ECOS models is the transductive weighted neuro-fuzzy inference engine (TWNFI) [60.30, 36]. In this approach, for every new vector (sample/example S) a *personalized* model is developed from existing nearest samples, where each of the variables is normalized in a different subrange of [0,1] so that they have a different influence on the Euclidean distance from (60.1), therefore they are ranked in terms of their importance to the output calculated for any new sample individually. Samples are also weighted in the model based on their distance to the new sample, where in the Euclidean distance formula variables are also weighted. Each personalized model can be represented as a rule (or a set of rules) that represents the personalized profile for the new input vector. The TWNFI model is evolving as new data samples, added to a data set, can be used in any further personalized model development. This includes using different sets of variables and features [60.30, 36].

ECOS have been applied to both model brain functions and as general CI tools [60.30]. In one application, an ECOS was trained to classify EEG data measured from a single person's brain, into four classes representing four perceptual states – hearing, seeing, both, and nothing [60.30]. In another application, ECOS were used to model emerging acoustic clusters, when multiple spoken languages are learned [60.30].

ECOS have been applied to a wide range of CI applications, such as adaptive classification of gene expression data, adaptive robot control, adaptive financial data modeling, adaptive environmental, and social data modeling [60.30].

ECOS are used in Sect. 60.3 for building GRN models.

60.2.2 Spiking Neural Networks – Strongly Brain Inspired Models

Spiking models of a neuron and of neural networks – SNN, have been inspired and developed to mimic more biologically the spiking activity of neurons in the brain when processing information.

SPIN: 12742608 (Springer Handbook of Bio-/Neuroinformatics)
MS ID: hb23-060 Proof 1
Created on: 17 April 2013 10:50 CET

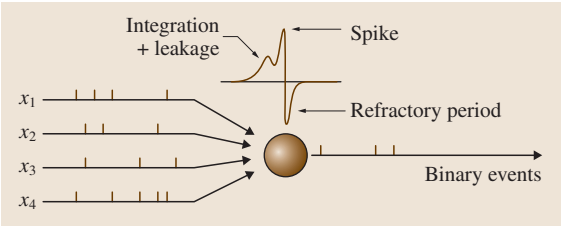

Fig. 60.3 A general representation of a spiking neuron model (after [60.13])

One model – the spike response model (SRM) of a neuron [60.31, 37] is described below and extended in Sect. 60.4 to a CNGM.

A neuron i receives input spikes from presynaptic neurons $j \in \Gamma_i$, where Γ_i is a pool of all neurons presynaptic to neuron i. The state of the neuron i is described by the state variable $u_i(t)$ that can be interpreted as a total postsynaptic potential (PSP) at the membrane of soma (Fig. 60.3). When $u_i(t)$ reaches a firing threshold $\vartheta_i(t)$ TS4, neuron i fires, i. e., emits a spike. The value of the state variable $u_i(t)$ is the sum of all postsynaptic potentials, i. e.,

$$u_i(t) = \sum_{j \in \Gamma_i} \sum_{t_j \in F_j} J_{ij} \left(t - t_j - \Delta_{ij}^{ax} \right) . \quad (60.4)$$

The weight of the 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 $\varepsilon_{ij}(t - t_j - \Delta_{ij}^{ax})$. Δ_{ij}^{ax} is an axonal delay between neurons i and j which increases with Euclidean distance between neurons.

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

$$\varepsilon_{ij}^{\text{synapse}}(S) A^{\text{synapse}} \left(\exp\left(\frac{s}{\tau_{\text{decay}}^{\text{synapse}}} \right) - \exp\left(-\frac{s}{\tau_{\text{rise}}^{\text{synapse}}} \right) \right) . \quad (60.5)$$

The following notations are used above: $\tau_{\text{decay/rise}}^{\text{synapse}}$ are time constants of the rise and fall of an individual PS, A is the PSP's amplitude, and *synapse* represents the type of the activity of the synapse from the neuron j to neuron i that can be measured and modeled separately for *fast_excitation, fast_inhibition, slow_excitation, and slow_inhibition*, all integrated in formula [60.13]. These types of PSPs are based on neurobiology [60.38] and will be the basis for the development of the computational neurogenetic model in Sect. 60.4, where the different synaptic activities are represented as functions of different proteins (neurotransmitters and neuroreceptors).

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

$$u_i^{\text{ext\_input}}(t) = J_{ik}^{\text{ext\_input}} \varepsilon_{ik}(t). \quad (60.6)$$

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

SNN models can be built with the use of the above spiking neuron model. Spiking neurons within an SNN can be either excitatory or inhibitory. Lateral connections between neurons in an SNN may have weights that decrease in value with distance from neuron i for instance, according to a Gaussian formula, while the connections between neurons themselves can be established at random.

SNN can be used to build biologically plausible models of brain functions. Examples are given in [60.13, 31, 37, 38]. Figure 60.4 graphically shows an application of an SNN to model brain functions that connect signals from the thalamus to the temporal cortex (from [60.13]).

Other applications of SNN include image recognition. In [60.39] an adaptive SNN model is developed where new SNN submodules (maps) are created incrementally to accommodate new data samples over time. For example, a new submodule of several spiking neurons and connections evolves when a new class of objects (e.g., a new face in the case of a face recognition problem) is presented to the system for learning at any time of this process. When there are no active inputs presented to the system, the system merges close spiking neuronal maps depending on their similarity.

Developing new methods for learning in evolving SNN is a challenging direction for future research with a potential for applications in both computational neuroscience and pattern recognition, e.g., multimodal information processing – speech, image, odor, gestures, etc.

SNN are extended to CNGM in Sect. 60.4.

TS4 Please confirm.

Editor's or typesetter's annotations (will be removed before the final TeX run)

SPIN: 12742608 (Springer Handbook of Bio-/Neuroinformatics)
MS ID: hb23-060 Proof 1
Created on: 17 April 2013 10:50 CET

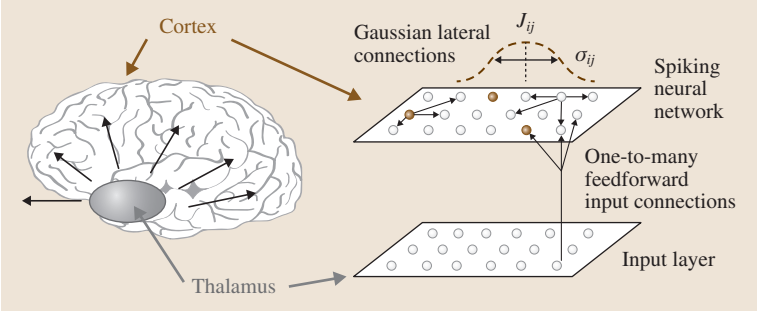

Fig. 60.4 An example of a SNN to model a function of the cortex with internal inputs from the thalamus and external input stimuli. About 20% of $N = 120$ neurons are inhibitory neurons that are randomly positioned on the grid (*filled circles*). External input is random with a defined average frequency (e.g., between 10–20 Hz) (after [60.13])

60.2.3 Open Questions

Further development of brain-like or brain inspired ANN requires some that some questions be addressed:

- How much should an ANN mimic the brain in order to become an efficient CI model?
- How is a balance between structure definition and learning achieved in ANN?
- How can ANN evolve and optimize their parameters and input features over time in an efficient way?
- How can incremental learning in ANN be applied without the presentation of an input signal (e.g., *sleep* learning)?

60.3 Gene Inspired Methods of Computational Intelligence

60.3.1 The Central Dogma in Molecular Biology and GRN

The central dogma of molecular biology states that DNA, which resides in the nucleus of a cell or a neuron, transcribes into RNA and then translates into proteins, which process is continuous, evolving, so that proteins, called transcription factors, cause genes to transcribe, etc. [60.40, 41] (Fig. 60.5).

The DNA is a long, double stranded sequence (a double helix) of millions or billions of 4 base molecules (nucleotides) denoted as A, C, T, and G, which are chemically and physically connected to each other through other molecules. In the double helix, they make pairs such that every A from one strand is connected to a corresponding T on the opposite strand and every C is connected to a G. A gene is a sequence of hundreds and thousands of bases as part of the DNA that is translated into protein. Only less than 5% of the DNA of the human genome constitutes genes, the other part is a noncoding region that contains useful information as well.

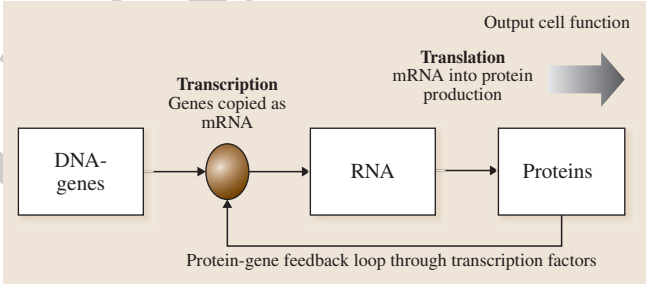

Fig. 60.5 The genes in the DNA transcribe into RNA and then translate into proteins that define the function of a cell. (The central dogma of molecular biology)

The DNA of each organism is unique and resides in the nucleus of each of its cells. But it is the proteins that are expressed from the genes and define the function of the cell that make a cell alive. The genes and proteins in each cell are connected in a dynamic *GRN* consisting of regulatory pathways.

Normally, only a few hundreds of genes are expressed as proteins in a particular cell. At the transcription phase, one gene is transcribed in many RNA copies and their number defines the expression level of this gene [60.40, 41]. Some genes may be over-expressed, resulting in too much protein in the cell, some genes may be under-expressed resulting in too little protein; in both cases the cell may be functioning in a wrong way, which may be causing a disease. Abnormal expression of a gene can be caused by a gene mutation – a random change in the code of the gene, where a base molecule is either inserted or deleted, or altered into another base molecule. Drugs can be used to stimulate or suppress the expression of certain genes

SPIN: 12742608 (Springer Handbook of Bio-/Neuroinformatics)
MS ID: hb23-060 Proof 1
Created on: 17 April 2013 10:50 CET

and proteins, but how that will affect indirectly the other genes related to the targeted one must be evaluated and this is where computational modeling of GRN can help.

It is always difficult to establish the interaction between genes and proteins. The question *What will happen with a cell or the whole organism if one gene is under-expressed or missing?* is now being attempted by the use of a technology called *knock-out gene technology*. This technology is based on the removal of a gene sequence from the DNA and letting the cell/organism to develop, where parameters are measured and compared with the parameters when the gene was not missing.

60.3.2 GRN-ANN Models

Modeling GRN is the task of creating a dynamic interaction network between genes that defines the next time expression of genes based on their previous time expression. A detailed discussion of the methods for GRN modeling can be found in [60.41, 43, 44]. Models of GRN, derived from gene expression RNA data, have been developed with the use of different mathematical and computational methods, such as: statistical correlation techniques; evolutionary computation; ANN; differential equations, both ordinary and partial; Boolean models; kinetic models; state-based models; and others [60.41].

A model of GRN, trained on time-course data is presented in [60.42] where the human response to fibroblast serum data is used (Fig. 60.6) and a GRN is extracted from it (Fig. 60.7). The method uses a genetic algorithm to select the initial cluster centers of the time course clustered gene expression values and then applies a Kalman filter to derive the gene connecting equations.

In [60.44] a GRN-ECOS is proposed and applied on small-scale cell line gene expression data. An ECOS is evolved with inputs being the expression level of a certain number of selected genes (e.g., 4) at a time moment (t) and the outputs being the expression level of the same or other genes/proteins at the next time moment ($t + \mathrm{d}t$). After an ECOS is trained on time course gene expression data, rules are extracted from the ECOS and linked between each other in terms of ~~time-arrows~~ CE[5] of their creation in the model, thus representing the GRN. The rule nodes in an ECOS capture clusters of input genes that are related to the output genes/proteins at the next time moment. Figure 60.7 shows an example of EFuNN used for modeling GRN [60.33, 44].

The rules extracted from an EFuNN model, for example, represent the relationship between the gene expression of a group of genes $G(t)$ at a time moment t

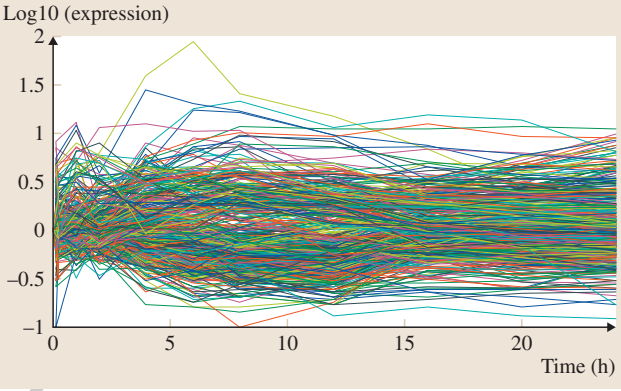

Fig. 60.6 Time-course gene expression data representing the response of thousands of genes of fibroblast to serum (after [60.42])

and the expression of the genes at the next time moment $G(t + \mathrm{d}t)$, e.g.,

$$\begin{aligned} &\text{IF } g_{13}(t) \text{ is High (0.87) and } g_{23}(t) \text{ is Low (0.9)} \\ &\text{THEN } g_{87}(t + \mathrm{d}t) \text{ is High (0.6) and} \\ &\qquad g_{103}(t + \mathrm{d}t) \text{ is Low.} \end{aligned} \tag{60.7}$$

Through modifying a threshold for rule extraction one can extract stronger or weaker patterns of a dynamic relationship.

Adaptive training of an ECOS makes incremental learning of a GRN possible, as well as adding ~~new inputs/outputs~~ (new genes~~)~~ to ~~the~~ GRN CE[6].

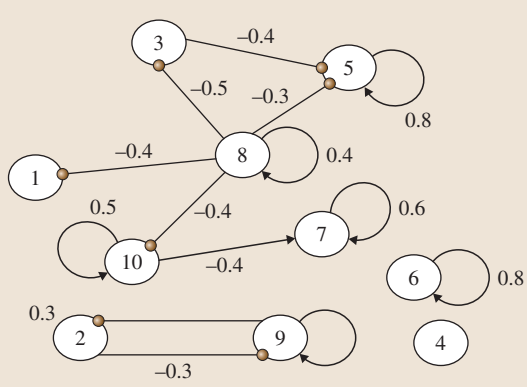

Fig. 60.7 A GRN obtained with the use of the method from [60.42] on the data from Fig. 60.5 after the time gene expression series are clustered into 10 clusters. The nodes represent gene clusters while the arcs represent the dynamic relation (interaction) between these gene groups over consecutive time moments

CE[5] Please check terminology.

CE[6] Please check that this is the intended meaning.

Editor's or typesetter's annotations (will be removed before the final TeX run)

SPIN: 12742608 (Springer Handbook of Bio-/Neuroinformatics)
MS ID: hb23-060 Proof 1
Created on: 17 April 2013 10:50 CET

A set of DENFIS models can be trained, one for each gene g_i, so that an input vector is the expression vector $G(t)$ and the output is a single variable $g_i(t + dt)$. DENFIS allows for a dynamic partitioning of the input space. Takagi–Sugeno fuzzy rules, which represent the relationship between gene g_i with the rest of the genes, are extracted from each DENFIS model, e.g.,

$$\begin{aligned}&\text{IF } g_1 \text{ is } (0.63, 0.70, 0.76) \text{ and}\\ &g_2 \text{ is } (0.71, 0.77, 0.84) \text{ and}\\ &g_3 \text{ is } (0.71, 0.77, 0.84) \text{ and}\\ &g_4 \text{ is } (0.59, 0.66, 0.72)\\ &\text{THEN } g_5 = 1.84 - 1.26g_1 - 1.22g_2\\ &\qquad + 0.58g_3 - 0.03g_4. \end{aligned} \quad (60.8)$$

The ECOS structure from Fig. 60.2 can be used in a multilevel, hierarchical way, where the transcription process is represented in one ECOS and translation in another ECOS, which inputs are connected to the outputs of the first one, using feedback connections to represent transcription factors.

Despite the variety of different methods used so far for modeling GRN and for systems biology in general, there is no single method that will suit all requirements to model a complex biological system, especially to meet the requirements for adaptation, robustness, and information integration.

In the next section GRN modeling is integrated with SNN to model the interaction between genes/proteins in relation to activity of a spiking neuron and an SNN as a whole.

60.4 Computational Neurogenetic Models

60.4.1 General Notions

With the advancement of molecular and brain research technologies more and more data and information are being made available about the genetic basis of some neuronal functions (see, for example, the brain-gene map of a mouse [60.45] and the brain-gene ontology BGO in [60.46]).

This information can be utilized to create biologically plausible ANN models of brain functions and diseases that include models of gene interaction. This area integrates knowledge from computer and information science, brain science, and molecular genetics and it is here called CNGM [60.2].

A CNGM integrates genetic, proteomic, and brain activity data and performs data analysis, modeling, prognosis, and knowledge extraction that reveals the relationship between brain functions and genetic information. Let us look at this process as a process of building mathematical function or a computational algorithm as follows.

A future state of a molecule M' or a group of molecules (e.g., genes and proteins) depends on its current state M and on an external signal Em

$$M' = Fm(M, Em). \quad (60.9)$$

A future state N' of a neuron or an ensemble of neurons will depend on its current state N and on the state of the molecules M (e.g., genes) and on external signals En

$$N' = Fn(N, M, En). \quad (60.10)$$

Finally, a future neuronal state C' of the brain will depend on its current state C and also on the neuronal N and the molecular M state, and on the external stimuli Ec

$$C' = Fc(C, N, M, Ec). \quad (60.11)$$

The above set of equations (or algorithms) is a general one and in different cases it can be implemented differently, e.g., one gene – one neuron/brain function; multiple genes – one neuron/brain function, no interaction between genes; multiple genes – multiple neuron/brain functions, where genes interact in a GRN and neurons also interact in a neural network architecture; multiple genes – complex brain/cognitive function/s, where genes interact within GRN and neurons interact in several hierarchical neural networks.

Several CNGM models have been developed so far, varying from modeling a single gene in a biologically realistic ANN model [60.3] to modeling a set of genes forming an interaction GRN [60.13,43]. In the next section we give an example of a CNGM that combines SNN and GRN into one model [60.13].

60.4.2 A Computational Neurogenetic Model that Integrates GRN Within an SNN Model

The main idea behind the model proposed in [60.2] is that interaction of genes in neurons affect the dynam-

SPIN: 12742608 (Springer Handbook of Bio-/Neuroinformatics)
MS ID: hb23-060 Proof 1
Created on: 17 April 2013 10:50 CET

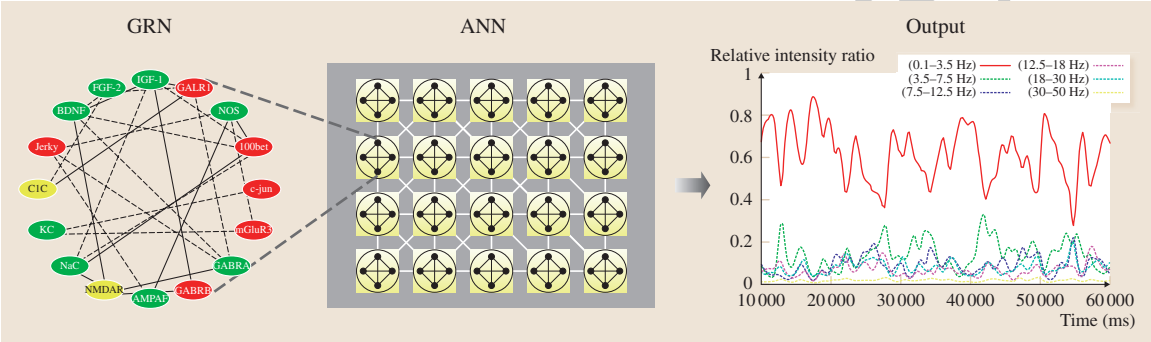

Fig. 60.8 A CNGM, where a GRN is used to represent the interaction of genes, and a SNN is employed to model a brain function. The model output is compared against real brain data for validation of the model and for verifying the derived gene interaction GRN after model optimization is applied [60.13]

ics of the whole ANN through neuronal parameters, which are no longer constant but change as a function of gene/protein expression. Through optimization of the GRN, the initial gene/protein expression values, and the ANN parameters, particular target states of the ANN can be achieved, so that the ANN can be tuned to model real brain data in particular.

This idea is illustrated in Fig. 60.8. The behavior of the SNN is evaluated by means of the local field potential (LFP), thus making it possible to attempt modeling the role of genes in different brain states, where EEG data is available to test the model. A standard FFT signal processing technique is used to evaluate the SNN output and to compare it with real human EEG data. A broader theoretical and biological background of CNGM construction is given in [60.13].

In general, we consider two sets of genes – a set G_{gen} that relates to general cell functions and a set G_{spec} that defines specific neuronal information-processing functions (receptors, ion channels, etc.). The two sets together form a set $\boldsymbol{G} = \{G_1, G_2, \ldots, G_n\}$. We assume that the expression level of each gene is a nonlinear function of expression levels of all the genes in $\boldsymbol{G}$

$$g_j(t + \Delta t') = \sigma\left(\sum_{k=1}^{n} w_{jk} g_k(t)\right). \qquad (60.12)$$

In [60.13] it is assumed that:

1. One protein is coded by one gene.
2. The relationship between the protein level and the gene expression level is nonlinear.
3. Protein levels lie between the minimal and maximal values. Thus, the protein level is expressed by

$$p_j(t + \Delta t) = \left(p_j^{\max} - p_j^{\min}\right) \times \sigma\left(\sum_{k=1}^{n} w_{jk}\, g_k(t)\right) + p_j^{\min}. \qquad (60.13)$$

The delay constant introduced in the formula corresponds to the delay caused by the gene transcription, mRNA translation into proteins and posttranslational protein modifications, and also the delay caused by gene transcription regulation by transcription factors.

Some proteins and genes are known to affect the spiking activity of a neuron represented in an SNN model by neuronal parameters, such as *fast_excitation, fast_inhibition, slow_excitation, and slow_inhibition* (Sect. 60.2). Some neuronal parameters and their correspondence to particular proteins are summarized in Table 60.1.

Besides ~~the gene coding for the proteins mentioned above and those~~ directly affecting the spiking dynamics of a neuron CE7, a GRN model can include other genes relevant to a problem in hand, e.g., modeling a brain function or a brain disease. In [60.13] these genes/proteins are c-jun, mGLuR3, Jerky, BDNF, FGF-2, IGF-I, GALR1, NOS, and S100beta [60.13].

The goal of the CNGM in Fig. 60.8 is to achieve a desired SNN output through optimization of the model parameters. The LFP of the SNN, defined as $\text{LFP} = (1/N)\Sigma u_i(t)$, by means of FFT is evaluated in order to compare the SNN output with the EEG signal analyzed in the same way. It has been shown that brain LFPs in principle have the same spectral characteristics as EEG [60.47].

In order to find an optimal GRN within the SNN model, so that the frequency characteristics of the LFP of the SNN model are similar to the brain EEG

CE7 Please check that this is the intended meaning.

Editor's or typesetter's annotations (will be removed before the final TeX run)

SPIN: 12742608 (Springer Handbook of Bio-/Neuroinformatics)
MS ID: hb23-060 Proof 1
Created on: 17 April 2013 10:50 CET

Table 60.1 Neuronal parameters and related proteins (PSP; AMPAR: (amino-methylisoxazole-propionic acid) ampa receptor; NMDAR: (*N*-methyl-D-aspartate acid) NMDA receptor; GABRA: (gamma-aminobutyric acid) $GABA_A$ receptor; GABRB: $GABA_B$ receptor; SCN: sodium voltage-gated channel; KCN: kalium (potassium) voltage-gated channel; CLC: chloride channel; PV: parvalbumin)

Neuronal parameter amplitude and time constants of	Protein
Fast excitation PSP	AMPAR
Slow excitation PSP	NMDAR
Fast inhibition PSP	GABRA
Slow inhibition PSP	GABRB
Firing threshold	SCN, KCN, CLC
Late excitatory PSP through GABRA	PV

characteristics, the following evolutionary computation procedure is used:

1. Generate a population of CNGMs, each with randomly, but constrained, generated values of coefficients for the GRN matrix **W**, initial gene expression values $g(0)$, initial values of SNN parameters $P(0)$, and different connectivity.
2. Run each SNN model over a period of time T and record the LFP.
3. Calculate the spectral characteristics of the LFP using FFT.
4. Compare the spectral characteristics of SNN LFP to the characteristics of the target EEG signal. Evaluate the closeness of the LFP signal for each SNN to the target EEG signal characteristics. Proceed further according to the standard GA algorithm to find a SNN model that matches the EEG spectral characteristics better than previous solutions.
5. Repeat steps 1 to 4 until the desired GRN and SNN model behavior is obtained.
6. Analyze the GRN and the SNN parameters for significant gene patterns that cause the SNN model to manifest similar spectral characteristics as the real data.

The proposed CNGM modeling framework can be used to find patterns of gene regulation related to brain functions. In [60.13] some preliminary results of analysis performed on real human interictal EEG data are presented. The model performance and the real EEG data are compared for the following relevant to the problem subbands: delta (0.5–3.5 Hz), theta (3.5–7.5 Hz), alpha (7.5–12.5 Hz), beta 1 (12.5–18 Hz), beta 2 (18–30 Hz), and gamma (above 30 Hz). This particular SNN had an evolved GRN with only 5 genes out of 16 (s100beta, GABRB, GABRA, mGLuR3, c-jun), all other genes having constant expression values. A GRN is obtained that has a meaningful interpretation and can be used to model what will happen if a gene/protein is suppressed by administering a drug, for example.

In *evolving CNGM* new genes can be added to the GRN model at a certain time, in addition to the new spiking neurons and connections created incrementally, as is the case in *evolving SNN*. Developing new evolving CNGM to model brain functions and brain diseases such as epilepsy, Alzheimer's, Parkinson's disease, schizophrenia, mental retardation, and others is a challenging problem for future research [60.13, 43].

60.4.3 Open Questions

Some questions emerged from the first CNGM experiments:

- How many different GRNs would lead to similar LFPs and what do they have in common?
- What neuronal parameters should be included in an ANN model and how can they be linked to activities of genes/proteins?
- What genes/proteins should be included in the model and can the gene interaction be represented over time within each neuron?
- How can the output activity of the ANN and the genes be integrated in time, as it is known that neurons spike in millisecond intervals and the process of gene transcription and translation into proteins takes minutes?
- How can a CNGM be created and evaluated in a situation of insufficient data?
- How can brain activity and the CNGM activity be measured in order to validate the model?
- What useful information (knowledge) can be derived from a CNG model?
- How can a CNGM model be adapted incrementally in a situation of new incoming data about brain functions and genes related to them?

Integrating principles from gene and neuronal information processing in a single ANN model raises many other, more general, questions that need to be addressed in the future, for example:

- Is it possible to create a truly adequate CNGM of the whole brain? Would gene-brain maps help in this respect [60.3]?

SPIN: 12742608 (Springer Handbook of Bio-/Neuroinformatics)
MS ID: hb23-060 Proof 1
Created on: 17 April 2013 10:50 CET

- How can dynamic CNGM be used to trace over time and predict the progression of a brain diseases, such as epilepsy and Parkinson's?
- How can CNGM be used to model gene mutation effects?
- How can CNGM be used to predict drug effects?
- How can CNGM help us to understand brain functions better, such as memory and learning?
- What CI problems can be efficiently solved with the use of a brain-gene inspired ANN?

60.5 Quantum Inspired CI

60.5.1 Quantum Level of Information Processing

At the quantum level, particles (e.g., atoms, electrons, ions, photons, etc.) are in a complex evolving state all the time. The atoms are the material that everything is made of. They can change their characteristics due to the frequency of external signals. Quantum computation is based upon physical principles from the theory of quantum mechanics [60.48].

One of the basic principles is the *linear superposition* of states. At a macroscopic or classical level a system exists only in a single basis state as energy, momentum, position, spin, and so on. However, at a microscopic or quantum level a quantum particle (e.g., atom, electron, positron, ion) or a quantum system is in a superposition of all possible basis states. At the microscopic level any particle can assume different positions at the same time moment, can have different values of energy, can have different spins, and so on. This *superposition* principle is counterintuitive because in classical physics one particle has only one position, energy, spin, etc.

If a quantum system interacts in any way with its environment, the superposition is assumed to be destroyed and the system *collapses* into one single real state as in the classical physics (Heisenberg). This process is governed by a probability amplitude. The square of the intensity for the probability amplitude is the quantum probability to observe the state.

Another quantum mechanics principle is *entanglement* – two or more particles, regardless of their location, are in the same state with the same probability function. The two particles can be viewed as *correlated*, undistinguishable, *synchronized*, coherent. An example is a laser beam consisting of millions of photons having the same characteristics and states.

Quantum systems are described by a probability density ψ that exists in a Hilbert space. The Hilbert space has a set of states $|\varphi_i\rangle$ forming a basis. A system can exist in a certain quantum state $|\psi\rangle$, which is defined as

$$|\psi\rangle = \sum c_i |\varphi_i\rangle, \quad \sum |c_i|^2 = 1;, \tag{60.14}$$

where the coefficients c_i may be complex. $|\psi\rangle$ is said to be in a superposition of the basis states $|\varphi_i\rangle$. For example, the quantum inspired analog of a single *bit* in classical computers can be represented as a *qu-bit* in a quantum computer

$$|x\rangle = a|0\rangle + b|1\rangle;, \tag{60.15}$$

where $|0\rangle$ and $|1\rangle$ represent the states 0 and 1, and a and b their probability amplitudes, respectively. The *qu-bit* is not a single value entity, but is a function of parameters whose values are complex numbers. After the loss of coherence the *qu-bit* will *collapse* into one of the states $|0\rangle$ or $|1\rangle$ with the probability a^2 for the state $|0\rangle$ and probability b^2 for the state $|1\rangle$.

The state of a quantum particle (represented, for example, as a *qu-bit)* can be changed by an operator called a *quantum gate*. A quantum gate is a reversible gate and can be represented as a unitary operator U acting on the *qu-bit* basis states. The defining property of a unitary matrix is that its conjugate transpose is equal to its inverse. Several quantum gates have been introduced, such as the NOT gate, controlled NOT gate, rotation gate, Hadamard gate, etc. [60.49–52].

60.5.2 Why Quantum Inspired CI?

Quantum mechanical computers and quantum algorithms try to exploit the massive quantum parallelism which is expressed in the principle of *superposition*. The principle of superposition can be applied to many existing methods of CI, where instead of a single state (e.g., a parameter value, or a finite automaton state, or a connection weight, etc.) a superposition of states will be used, described by a wave probability function, so that all these states will be computed in parallel, resulting in an increased speed of computation by many orders of magnitude [60.5, 8, 9, 49–57].

SPIN: 12742608 (Springer Handbook of Bio-/Neuroinformatics)
MS ID: hb23-060 Proof 1
Created on: 17 April 2013 10:50 CET

Quantum mechanical computers were proposed in the early 1980s and a description was formalized in the late 1980s. These computers, when implemented, are expected to be superior to classical computers in various specialized problems. Much effort has been made to extend the principal ideas of quantum mechanics to other fields of interest. There are well-known quantum algorithms such as Shor's quantum factoring algorithm [60.58] and Grover's database search algorithm [60.50, 54].

The advantage of quantum computing is that while a system is *uncollapsed* it can carry out more computing than a collapsed system, because, in a sense, it is computing in *many universes* at once. The above quantum principles have inspired research in both computational methods and brain study.

New theories (some of them speculative at this stage) have already been formulated. For example, *Penrose* [60.8, 9] argues that solving the quantum measurement problem is prerequisite for understanding the mind and that consciousness emerges as a macroscopic quantum state due to a coherence of quantum-level events within neurons.

60.5.3 Quantum Inspired Evolutionary Computation and Connectionist Models

Quantum inspired methods of evolutionary computation (QIEC) and other techniques were proposed and discussed in [60.51, 55]. They include genetic programming [60.59], particle swarm optimizers [60.60], finite automata and Turing machines, etc.

In QIEC, a population of n *qu-bit* individuals at time t can be represented as

$$Q(t) = \{q_1^t, q_2^t, \ldots q_n^t\} \ , \qquad (60.16)$$

where n is the size of the population.

Evolutionary computing with *qu-bit* representation has a better characteristic of population diversity than other representations, since it can represent linear superposition of states probabilistically. The *qu-bit* representation leads to a quantum parallelism of the system as it is possible to evaluate the fitness function on a superposition of possible inputs. The output obtained is also in the form of superposition, which needs to be *collapsed* to obtain the actual solution.

Recent research activities have focussed on using quantum principles for ANN [60.4, 5, 61–63]. Considering quantum ANN seems to be important for at least two reasons. There is evidence for the role that quantum processes play in the living brain. *Penrose* argued that a new physics binding quantum phenomena with general relativity can explain such mental abilities as *understanding*, *awareness*, and *consciousness* [60.9]. The second motivation is the possibility that the field of classical ANN could be generalized to the promising new field of quantum computation [60.53]. Both considerations suggest a new understanding of mind and brain functions, as well as new unprecedented abilities in information processing. *Ezhov* and *Ventura* consider quantum neural networks as the next natural step in the evolution of neurocomputing systems [60.4].

Several quantum inspired ANN models have been proposed and illustrated on small examples. In [60.63] QIEA is used to train a MLP ANN. *Narayanan* and *Meneer* simulated classical and quantum inspired ANN and compared their performances [60.5]. Their work suggests that there are, indeed, certain types of problems for which quantum neural networks will prove superior to classical ones.

Other relevant work includes quantum decision making, quantum learning models [60.64], quantum networks for signal recognition [60.62], and quantum associative memory [60.61, 65]. There are also recent approaches to quantum competitive learning where the quantum system's potential for excellent performance is demonstrated on real-world data sets [60.66, 67].

60.6 Towards the Integration of Brain, Gene, and Quantum Information Processing Principles: A Conceptual Framework for Future Research

60.6.1 Quantum Inspired SNN

In Sect. 60.4 we presented a CNGM that integrated principles from neuronal information processing and gene information processing in the form of integrating SNN with GRN. Following some ideas from QI-ANN, we can expect that *QI-SNN* and *QI-CNGM* would open new possibilities for modeling gene–neuron interactions related to brain functions and to new efficient AI applications.

The CNGM from Sect. 60.4 linked principles of information processing in gene/protein molecules with

SPIN: 12742608 (Springer Handbook of Bio-/Neuroinformatics)
MS ID: hb23-060 Proof 1
Created on: 17 April 2013 10:50 CET

neuronal spiking activity, and then – to the information processing of a neuronal ensemble, that is measured as local field potentials (LFP). How the quantum information processes in the atoms and particles (ions, electrons, etc.), that make the large gene/protein molecules, relate to the spiking activity of a neuron and to the activity of a neuronal ensemble, is not known yet and it is a challenging question for the future.

What is known at present, is that the spiking activity of a neuron relates to the transmission of ions and neurotransmitter molecules across the synaptic clefts and to the emission of spikes. Spikes, as carriers of information, are electrical signals made of particles that are emitted in one neuron and transmitted along the nerves to many other neurons. These particles are characterized by their quantum properties. So, quantum properties may influence, under certain conditions, the spiking activity of neurons and of the whole brain, as brains obey the laws of quantum mechanics (as everything else in the material world does).

Similarly to a chemical effect of a drug to the protein and gene expression levels in the brain, which may affect the spiking activity and the functioning of the whole brain (modeling of these effects is subject of the computational neurogenetic modeling), external factors like radiation, light, high frequency signals, etc., can influence the quantum properties of the particles in the brain through gate operators. According to *Penrose* [60.9] microtubules in the neurons are associated with quantum gates, even though what constitutes a quantum gate in the brain is still a highly speculative topic.

So, the question is: *Is it possible to create an SNN model and a CNGM that incorporate some quantum principles?*

A *QI-SNN* can be developed as an extension of the concept of evolving SNN [60.39] using the superposition principle, where instead of many SNN maps, each representing one object (e.g., a face), there will be a single SNN, where both connections and neurons are represented as particles, being in many states at the same time defined as probability wave function. When an input vector is presented to the QI-SNN, the network collapses in a single SNN defining the class of the recognized input vector.

60.6.2 A Conceptual Framework of a QI-CNGM

Here we extend the concept of CNGM (60.9)–(60.11) by introducing the level of quantum information processing. This results in a conceptual and hypothetical QI-CNGM, ~~which we intend to investigate and develop as future research~~ CE8.

The following is a list of equations that include quantum particle states and functions (hypothetical at this stage) into (60.9)–(60.11) and (60.18)–(60.20), starting with a new (60.17) that is concerned only with the level of quantum particle states.

A future state Q' of a particle or a group of particles (e.g. ions, electrons, etc.) depends on the current state Q and on the frequency spectrum Eq of an external signal, according to the Max Planck constant

$$Q' = Fq\,(Q, Eq)\,. \tag{60.17}$$

A future state of a molecule M' or a group of molecules (e.g., genes, proteins) depends on its current state M, on the quantum state Q of the particles, and on an external signal Em:

$$M' = Fm\,(Q, M, Em)\,. \tag{60.18}$$

A future state N' of a spiking neuron or an ensemble of neurons will depend on its current state N, on the state of the molecules M, on the state of the particles Q, and on external signals En

$$N' = Fn\,(N, M, Q, En)\,. \tag{60.19}$$

Finally, a future neuronal state C' of the brain will depend on its current state C and also on the neuronal N, on the molecular M, and on the quantum Q states of the brain:

$$C' = Fc\,(C, N, M, Q, Ec)\,. \tag{60.20}$$

The above hypothetical model of integrated function representations is based on the following assumptions:

- A large number of atoms are characterized by the same quantum properties, possibly related to the same gene/protein expression profile of a large number of neurons characterized by spiking activity that can be represented as a function.
- A large neuronal ensemble can be represented by a single LFP function.
- A cognitive process can be represented, at an abstract level, as a function Fc that depends on all lower levels of neuronal, genetic, and quantum activities.

60.6.3 Open Questions

Several reasons can be given in support of the research on integrating principles from quantum, molecular, and brain information processing into future CI models:

CE8 Please check that this is the intended meaning.

Editor's or typesetter's annotations (will be removed before the final TeX run)

SPIN: 12742608 (Springer Handbook of Bio-/Neuroinformatics)
MS ID: hb23-060 Proof 1
Created on: 17 April 2013 10:50 CET

- This may lead to a better understanding of neuronal, molecular, and quantum information processes.
- This may lead to new computer devices – a million times faster and more accurate than the current ones.
- At the nanolevel of microelectronic devices, quantum processes would have a significant impact and new methods of computation would be needed anyway.

60.7 Conclusions and Directions for Further Research

This chapter presents some CI models inspired by principles from different levels of information processing in the brain – including neuronal level, gene/protein level, and quantum level, and argues that CI models that integrate principles from different levels of information processing would be useful tools for a better understanding of brain functions and for the creation of more powerful methods and systems of computational intelligence.

Many open questions need to be answered in the future, some of these are:

- How do quantum processes affect the functioning of a living system in general?
- How do quantum processes affect cognitive and mental functions?
- Is it true that the brain is a quantum machine – working in a probabilistic space with many states (e.g., thoughts) being in a superposition all the time and it is only when we formulate our thought through speech or writing that the brain *collapses* in a single state?
- Is fast pattern recognition in the brain, involving far away segments, a result of both parallel spike transmissions and particle entanglement?
- Is communication between people and between living organisms in general a result of entanglement processes?
- How does the energy in the atoms relate to the energy of the proteins, the cells, and the whole brain?
- Would it be beneficial to develop different QI computational intelligence techniques, such as QI-SVM, QI-GA, QI-decision trees, QI-logistic regression, QI-cellular automata, and QI-ALife?
- How do we implement QI computational intelligence algorithms in order to benefit from their high speed and accuracy? Should we wait for the quantum computers to be realized many years from now, or we can implement them efficiently on specialized computing devices based on classical principles of physics?

Further directions in our research are:

- Building a brain–gene-quantum ontology system that integrates facts, information, knowledge, and CI models of different levels of information processing in the brain and their interaction.
- Building novel brain, gene, and quantum inspired CI models, studying their characteristics, and interpreting the results.
- Applying the new methods to solving complex CI problems in neuroinformatics and brain diseases, bioinformatics and cancer genetics, multimodal information processing, and biometrics.

References

60.1 C. Bishop: *Neural Networks for Pattern Recognition* (Oxford Univ. Press, Oxford, UK 1995)
60.2 N. Kasabov, L. Benuskova: Computational neurogenetics, Int. J. Theor. Comput. Nanosci. **1**(1), 47–61 (2004)
60.3 G. Marcus: *The Birth of the Mind: How a Tiny Number of Genes Creates the Complexity of the Human Mind* (Basic, New York 2004)
60.4 A. Ezhov, D. Ventura: Quantum neural networks. In: *Future Directions for Intelligent Systems and Information Sciences*, ed. by N. Kasabov (Springer, TS9 2000) pp. 213–234
60.5 A. Narayanan, T. Meneer: Quantum artificial neural network architectures and components, Inf. Sci. TS12 199–215 (2000)
60.6 D.B. Fogel: *Evolutionary Computation – Toward a New Philosophy of Machine Intelligence* (IEEE, New York 1995)
60.7 X. Yao: Evolutionary artificial neural networks, Int. J. Neural Syst. **4**(3), 203–222 (1993)
60.8 R. Penrose: *Shadows of the Mind. A Search for the Missing Science of Conscious* (Oxford Univ. Press, Oxford 1994)

TS9 *Please supply the publisher's location.*

Editor's or typesetter's annotations (will be removed before the final TeX run)

Part A | 60

SPIN: 12742608 (Springer Handbook of Bio-/Neuroinformatics)
MS ID: hb23-060 Proof 1
Created on: 17 April 2013 10:50 CET

60.9 R. Penrose: *The Emperor's New Mind* (Oxford Univ. Press, Oxford 1989)
60.10 S. Amari, N. Kasabov: *Brain-like Computing and Intelligent Information Systems* (Springer, New York 1998)
60.11 M. Arbib: *Brains, Machines and Mathematics* (Springer, Berlin 1987)
60.12 M. Arbib (Ed.): *The Handbook of Brain Theory and Neural Networks* (MIT, Cambridge 2003)
60.13 L. Benuskova, N. Kasabov: *Towards Computational Neurogenetic Modelling* (Springer, New York 2007), ~~In print~~ TS[10]
60.14 G. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, D.B. Rosen: Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analogue multi-dimensional maps, IEEE Trans. Neural Netw. **3**(5), 698–713 (1991)
60.15 G. Carpenter, S. Grossberg: *Pattern Recognition by Self-Organizing Neural Networks* (The MIT, Cambridge, USA 1991)
60.16 N. Kasabov: Foundations of neural networks. In: *Fuzzy Systems and Knowledge Engineering* (The MIT, MA 1996)
60.17 T. Kohonen: *Self-Organizing Maps* (Springer, Cambridge 1997)
60.18 E. Rolls, A. Treves: *Neural Networks and Brain Function* (Oxford Univ. Press, Oxford 1998)
60.19 F. Rosenblatt: *Principles of Neurodynamics* (Spartan Books, New York 1962)
60.20 D.E. Rumelhart, G.E. Hinton, R.J. Williams (Eds.): Learning Internal Representations by Error Propagation, Parallel Distrib, Processing: Explorations in the Microstructure of Cognition (MIT/Bradford Books, Cambridge 1986)
60.21 G.A. Rummery, M. Niranjan: *On-line Q-learning Using Connectionist System* (Cambridge Univ. Press, Cambridge 1994), 166 pp., CUED/F-INENG/TR
60.22 S. Schaal, C. Atkeson: Constructive incremental learning from only local information, Neural Comput. **10**, 2047–2084 (1998)
60.23 S. Grossberg: *Studies of Mind and Brain* (Reidel, Boston 1982)
60.24 J.G. Taylor: *The Race for Consciousness* (MIT, Cambridge 1999)
60.25 N. Kasabov: Evolving fuzzy neural networks – algorithms, applications and biological motivation. In: *Methodologies for the Conception, Design and Application of Soft Computing*, ed. by T. Yamakawa, G. Matsumoto (World Scientific, TS[9] 1998) pp. 271–274
60.26 B. Fritzke: A growing neural gas network learns topologies, Adv. Neural Inf. Process. Syst. **7**, 625–632 (1995)
60.27 J. Platt: A resource allocating network for function interpolation, Neural Comput. **3**, 213–225 (1991)
60.28 J. Freeman, D. Saad: On-line learning in radial basis function networks, Neural Comput. **9**(7), TS[13] (1997)
60.29 T. Poggio: Regularization theory, radial basis functions and networks. In: *From Statistics to Neural Networks: Theory and Pattern Recognition Applications*, NATO ASI Series, Vol. 136, ~~ed. by~~ TS[18] (NATO, TS[9] 1994) pp. 83–104
60.30 N. Kasabov: *Evolving Connectionist Systems: The Knowledge Engineering Approach* (Springer, London 2007)
60.31 W. Maass, C.M. Bishop (Eds.): *Pulsed Neural Networks* (The MIT, Cambridge 1999)
60.32 N. Kasabov, Q. Song: DENFIS: Dynamic, evolving neural-fuzzy inference systems and its application for time-series prediction, IEEE Trans. Fuzzy Syst. **10**, 144–154 (2002)
60.33 N. Kasabov: Evolving fuzzy neural networks for on-line supervised/unsupervised, knowledge – based learning, SMC B: Cybern. **31**(6), 902–918 (2001)
60.34 T. Yamakawa, H. Kusanagi, E. Uchino, T. Miki: A new effective algorithm for neo fuzzy neuron model, Proc. Fifth IFSA World Congr. (IFSA, Seoul, Korea 1993) pp. 1017–1020
60.35 L.A. Zadeh: Fuzzy Sets, Inf. Control. **8**, 338–353 (1965)
60.36 Q. Song, N. Kasabov: ~~TWNFI – transductive neural-fuzzy inference system with weighted data normalization and its application in medicine, IEEE Trans. Fuzzy Syst.~~ ~~TS[12]~~, ~~TS[13]~~ ~~(2005)~~
60.37 W. Gerstner, W.M. Kistler: *Spiking Neuron Models* (Cambridge Univ. Press, Cambridge 2002)
60.38 A. Destexhe: Spike-and-wave oscillations based on the properties of $GABA_B$ receptors, J. Neurosci. **18**, 9099–9111 (1998)
60.39 S. Wysoski, L. Benuskova, N. Kasabov: On-line learning with structural adaptation in a network of spiking neurons for visual pattern recognition, Artificial Neural Networks – ICANN 2006 **4131**, 61–70 (2006)
60.40 C. Brown, M. Shreiber, B. Chapman, G. Jacobs: Information science and bioinformatics. In: *Future Directions of Intelligent Systems and Information Sciences*, ed. by N. Kasabov (Physica, Heidelberg 2000) pp. 251–287
60.41 D.S. Dimitrov, I. Sidorov, N. Kasabov: Computational biology. In: *Handbook of Theoretical and Computational Nanotechnology*, Vol. 1, ed. by M. Rieth, W. Sommers (American Scientific, TS[9] 2004), Chap. 21
60.42 Z. Chan, N. Kasabov, L. Collins: A two-stage methodology for gene regulatory network extraction from time-course gene expression data, Expert Syst. Appl. **30**(1), 59–63 (2006)
60.43 H. Chin, S. Moldin (Eds.): *Methods in Genomic Neuroscience* (CRC, TS[9] 2001)
60.44 N. Kasabov, S.H. Chan, V. Jain, I. Sidirov, S.D. Dimitrov: Gene Regulatory Network Discovery from Time-Series Gene Expression Data – A Computational Intelligence Approach, LNCS **3316**, 1344–1353 (2004)

TS[10] Please update.
TS[18] *Please supply the editor(s).*
TS[12] *Please provide the volume number.*
TS[13] *Please supply the page range.*
TS[15] *Please supply more information.*

Editor's or typesetter's annotations (will be removed before the final TeX run)

SPIN: 12742608 (Springer Handbook of Bio-/Neuroinformatics)
MS ID: hb23-060 Proof 1
Created on: 17 April 2013 10:50 CET

60.45 http://alleninstitute.org, TS15
60.46 http://www.kedri.info, TS15
60.47 W. Freeman: *Neurodynamics* (Springer, London 2000)
60.48 R.P. Feynman, R.B. Leighton, M. Sands: *The Feynman Lectures on Physics* (Addison-Wesley Publishing Company, Massachusetts 1965)
60.49 T. Hey: Quantum computing: An introduction. In, Comput. Control Eng. J. **10**(3), 105–112 (1999)
60.50 T. Hogg, D. Portnov: Quantum optimization, Inf. Sci. **128**, 181–197 (2000)
60.51 J.-S. Jang, K.-H. Han, J.-H. Kim: Quantum-inspired evolutionary algorithm-based face verification, LNCS TS12, 2147–2156 (2003)
60.52 S.C. Kak: *Quantum Neural Computation, Research Report* (Louisiana State Univ., Baton Rouge, TS16)
60.53 M. Brooks: *Quantum Computing and Communications* (Springer, Berlin, Heidelberg 1999)
60.54 L.K. Grover: A fast quantum mechanical algorithm for database search, STOC '96: Proc. Twenty-Eighth Ann. ACM Symp. Theory Comput. (ACM, New York, USA 1996) pp. 212–219
60.55 K.-H. Han, J.-H. Kim: Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, IEEE Trans. Evol. Comput. TS12, 580–593 (2002)
60.56 G.E. Hinton: Connectionist learning procedures, Artif. Intell. **40**, 185–234 (1989)
60.57 M.A. Perkowski: Multiple-valued quantum circuits and research challenges for logic design and computational intelligence communities, IEEE Comput. Intell. Soc. Mag. TS12, TS13 (2005)
60.58 P.W. Shor: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput. **26**, 1484–1509 (1997)
60.59 L. Spector: *Automatic Quantum Computer Programming: A Genetic Programming Approach* (Kluwer Academic, TS9 2004)
60.60 J. Liu, W. Xu, J. Sun: Quantum-Behaved Particle Swarm Optimization with Mutation Operator, 17th IEEE Int. Conf. Tools Artif. Intell. (ICTAI'05) (2005)
60.61 C.A. Trugenberger: Quantum pattern recognition, Quantum Inf. Process. **1**, 471–493 (2002)
60.62 X.-Y. Tsai, H.-C. Huang, S.-J. Chuang: Quantum NN vs. NN in signal recognition, in: ICITA'05, Proc. Third Int. Conf. Inf. Technol. Appl. (ICITA'05), Vol. 2 (IEEE Computer Society, Washington, DC, USA 2005) pp. 308–312
60.63 G.K. Venayagamoorthy, S. Gaurav: Quantum-inspired evolutionary algorithms and binary particle swarm optimization for training MLP and SRN neural networks, J. Theor. Comput. Nanosci. TS12, TS13 (2006)
60.64 N. Kouda, N. Matsui, H. Nishimura, F. Peper: Qu-bit neural network and its learning efficiency, Neural Comput. Appl. **14**, 114–121 (2005)
60.65 D. Ventura, T. Martinez: Quantum associative memory, Inf. Sci. Inf. Comput. Sci. **124**, 273–296 (2000)
60.66 D. Ventura: Implementing competitive learning in a quantum system. In, Proc. Int. Jt. Conf. Neural Netw. (IEEE 1999)
60.67 G. Xie, Z. Zhuang: A quantum competitive learning algorithm, Liangzi Dianzi Xuebao/Chin. J. Quantum Electron. (China) **20**, 42–46 (2003)
60.68 TS17 C.P. Williams, S.H. Clearwater: *Explorations in Quantum Computing* (Springer, Berlin, Heidelberg 1998)
60.69 TS17 S. Grossberg: On learning and energy – entropy dependence in recurrent and nonrecurrent signed networks, J. Stat. Phys. **1**, 319–350 (1969)
60.70 TS17 S. Haykin: *Neural Networks – A Comprehensive Foundation* (Prentice Hall, Engelwood Cliffs, NJ 1994)
60.71 TS17 D. Hebb: *The Organization of Behavior* (John Wiley, New York 1949)
60.72 TS17 T.M. Heskes, B. Kappen: On-line learning processes in artificial neural networks. In: *Mathematic Foundations of Neural Networks*, ed. by TS18 (Elsevier, Amsterdam 1993) pp. 199–233
60.73 TS17 J. Moody, C. Darken: Fast learning in networks of locally-tuned processing units, Neural Comput. **1**, 281–294 (1989)
60.74 TS17 K. Pribram: Rethinking neural networks: Quantum fields and biological data, Proc. First Appalach. Conf. Behav. Neurodyn. (Lawrence Erlbaum Associates, Hillsdate New Yersy 1993)
60.75 TS17 G. Resconi, A.J. van Der Wal: A data model for the morphogenetic neuron, Int. J. Gen. Syst. **29**(1), 141–149 (2000)
60.76 TS17 G. Resconi, G.J. Klir, E. Pessa: Conceptual foundations of quantum mechanics the role of evidence theory, quantum sets and modal logic, Int. J. Mod. Phys. C **10**(1), 29–62 (1999)
60.77 TS17 V. Vapnik: *Statistical Learning Theory* (Wiley, TS9 1998)

Acknowledgements

The work presented in this chapter was partially supported by grant AUTX0201 funded by the Foundation of Research Science and Technology of New Zealand and also by the Knowledge Engineering and Discovery Research Institute KEDRI (http://www.kedri.info), Auckland University of Technology.

TS15 *Please supply more information.*
TS16 *Please supply the year of publication.*
TS17 This reference is not cited in the text.
TS18 *Please supply the editor(s).*

Editor's or typesetter's annotations (will be removed before the final TeX run)

SPIN: 12742608 (Springer Handbook of Bio-/Neuroinformatics)
MS ID: hb23-060 Proof 1
Created on: 17 April 2013 10:50 CET