Neurocomputing 214 (2016) 242–268

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Neural networks: An overview of early research, current frameworks and new challenges

Alberto Prieto*, Beatriz Prieto, Eva Martinez Ortigosa, Eduardo Ros, Francisco Pelayo, Julio Ortega, Ignacio Rojas

Department of Computer Architecture and Technology, CITIC-UGR, University of Granada, Spain

ARTICLE INFO

Article history:
Received 2 October 2015
Received in revised form
15 May 2016
Accepted 5 June 2016
Available online 8 June 2016

Keywords:
Neural modelling
Neural networks
Artificial neural networks
Learning algorithms
Neural hardware
Neural simulators
Applications of neural networks
Human Brain Project
Brain Initiative

ABSTRACT

This paper presents a comprehensive overview of modelling, simulation and implementation of neural networks, taking into account that two aims have emerged in this area: the improvement of our understanding of the behaviour of the nervous system and the need to find inspiration from it to build systems with the advantages provided by nature to perform certain relevant tasks. The development and evolution of different topics related to neural networks is described (simulators, implementations, and real-world applications) showing that the field has acquired maturity and consolidation, proven by its competitiveness in solving real-world problems. The paper also shows how, over time, artificial neural networks have contributed to fundamental concepts at the birth and development of other disciplines such as Computational Neuroscience, Neuro-engineering, Computational Intelligence and Machine Learning. A better understanding of the human brain is considered one of the challenges of this century, and to achieve it, as this paper goes on to describe, several important national and multinational projects and initiatives are marking the way to follow in neural-network research.

1. Introduction and goals of neural-network research

Generally speaking, the development of artificial neural networks or models of neural networks arose from a double objective: firstly, to better understand the nervous system and secondly, to try to construct information processing systems inspired by natural, biological functions and thus gain the advantages of these systems. Although currently computers are capable of carrying out some tasks more efficiently than the human brain, computers are not capable of equalling the brain's cognitive capacity, its flexibility, robustness and energy efficiency.

From the system engineering point of view a neural network is considered as a "**black-box**" as it imitates a behaviour rather than a structure and can reproduce any function; however studying the structure of the network does not provide any useful information about the system being modelled [1]. The physical organisation of the original system is not considered; instead a very flexible neural structure with a proven problem solving quality is used where problems of a similar nature are concerned. An advantage of the neural network is that it behaves as a non-linear black box, modelling and describing virtually any non-linear dynamics. As far as conventional statistics are concerned, the neural network may be considered as a non-identifiable model in the sense that various networks with varying topologies and parameters may be obtained which produce the same results.

Many of the topics thought up in the field of artificial neural networks, after a long and effective youth, have now acquired maturity and consolidation. They have proven to be very competitive in the resolution of real-world problems compared to more traditional data-analysis methods, usually based on explicit statistical modelling.

The concept of neural networks germinated independently but over time new contexts and disciplines have arisen, covering wider objectives which naturally include neural networks. In fact, artificial neural-network techniques combine naturally with others forming a set of computational procedures with a solid theoretical base, and with an unquestionable efficiency in the resolution of real problems in various fields of information processing. As a result, nowadays artificial neural networks are no longer considered as a self-contained field of research, rather they have become an integral part of new contexts and disciplines, among which we can find Computational Neuroscience, Neromorphic Computing, Neuroengineering, Natural Computation, Computational Intelligence, Soft Computing, Machine Learning and Neuroinformatics that will also be briefly considered in this paper.

* Corresponding author.
E-mail addresses: aprieto@ugr.es (A. Prieto), beap@ugr.es (B. Prieto), ortigosa@ugr.es (E.M. Ortigosa), eros@ugr.es (E. Ros), fpelayo@ugr.es (F. Pelayo), jortega@ugr.es (J. Ortega), irojas@ugr.es (I. Rojas).

http://dx.doi.org/10.1016/j.neucom.2016.06.014
0925-2312/

In recent years, government authorities in Europe and the USA have approved long term initiatives for the study of the human brain and have dedicated considerable economic resources to this end. Artificial neural networks, in various forms and at different levels, have been included in these research projects and the announcements of the projects have clearly set out the main challenges to be overcome in this field in the coming years.

Artificial neural systems for information processing constitute an inter-disciplinary subject, given that both neuroscientists and psychologists will benefit from the incorporation of methods and quantitative techniques allowing, via simulation, a greater in-depth knowledge of their field, whilst computer scientists and engineers will discover ideas inspired by biology, (such as learning models) allowing them to construct systems to satisfy the needs and challenges of the real world, and finally physicists and applied mathematicians will encounter new domains and challenges leading to advances in their fields. Currently, the computational models for artificial neural networks closest to biology ("bio-inspired" or "bio-mimetic") have a double objective:

1. *To carry out reverse engineering on the human brain;* that is to use computational models in the fields of neuroscience, cognitive science, and psychology to frame hypotheses that can be directly tested by biological or psychological experiments. The computer simulation of these models allows *in-virtual (in-silico)* experimentation, capable of predicting the behaviour of certain structures and functions and obtaining empirical results very close to those obtained from *in-vitro* or *in-vivo* experiments with biological samples. A greater knowledge of the structures and functions of the human brain is thus acquired without resorting to invasive methods to collect data or to carry out reflex-reaction tests. These techniques may even predict how different patterns of gene expression produce neurons with dissimilar morphologies identifying different molecules and diverse synaptic connections.
2. *To carry out artificial systems which attempt to imitate natural networks.* As we have already commented, modern computers are currently unable to equal the cognitive capacities, the flexibility, the robustness and the energy efficiency of the human brain. In fact, compared to computers, the brain works slowly (with spiking frequency signals of the order of hundreds of Hz) and with apparently low precision (stochastic individual neural processes). However, the whole brain carries out well organized computations in parallel (around 10^{16} synaptic operations per second), works in real time (in continuous interaction with the environment) with closed perception-action loops, and a very low energy consumption (approximately 30 W) beating the most powerful computers in certain "biologically relevant tasks", such as "manipulating objects", recognizing a scene after having viewed it once, etc. It also provides an elegant degradation of capabilities, self-repair, and modification through learning. These properties inspire scientists and engineers to search for new and disruptive computing models.

The origins of artificial neural networks were based on trying to mimic how the human brain performs a particular task via the use of simplified mathematical models. The basic concept consists of considering the brain as an information processing, highly complex, non-linear, parallel computer system. The most significant features of the domain are:

- The use of massive interconnection networks of simple processing units (neurons).
- Asynchronous parallel and distributed processing.
- Non-linear dynamics.
- Global interconnection of network elements.
- Self-organization.
- High-speed computational capability.
- Modification of the parameters of the network to carry out a specific task or adaptation to its environment via a **learning process**.

As will become evident in Section 2.3, artificial neural networks are being successfully applied in a wide range of areas and fields, contributing to the resolution of scientific and industrial real-world problems by performing specific tasks related to the capacity of inferring the underlying knowledge in observations and, in general, in conjunction with other information processing techniques or procedures.

These characteristics, as well as the general current state of the art and challenges for future research in the field of neural networks will be analysed in this paper. The text is organized in the following way: Section 2 analyses the concepts and the seminal research related to artificial neural networks that have arisen when models have been developed which aim to clarify human cognition and to build computer systems capable of resolving real-world problems. Section 3 focuses on describing the various frameworks and disciplines which artificial neural networks are currently integrated into and the role they play in each of these areas. Section 4 describes the main objectives and challenges of large governments with projects such as the Human Brain Project and the Brain Initiative, approved in recent years by government authorities of the European Commission and the USA, both of which are dedicating huge economic resources to this research, within which artificial neural networks appear in various forms. Section 5 presents our conclusions. Finally, we would like to point out that we have not tried to include exhaustive bibliographic references, citing every published contribution related to a specific topic, rather we have tried to provide support for our comments via some examples.

2. Topics related with ANNs

Various aspects of artificial neural networks may be considered from diverse points of view, such as: data problems, learning, models, structures and algorithms, simulators and hardware implementations, fields of use and real applications derived from biological inspiration.

Interest in artificial neural networks has evolved from their capacity to process information, which comes in data format. It is frequently necessary to carry out a pre-processing of the data before presenting it to the neural network. The main **data problems** which may occur are the following:

1. **Limited data for learning**. When only a limited amount of data is available cross-validation techniques are commonly used based on dividing the available data into two groups, one for learning and the other to validate the behaviour of the network. In order to gain a better knowledge of the network, the size and number of elements may be modified for training and evaluating the network in different situations [2,3].
2. **Imbalanced data**. A problem which occurs in learning, usually when in a classification problem there are many more elements of some classes than others [4]. There are several techniques to solve this problem, mainly focused either at the data level (sampling methods) or at the classifier level (modifying it internally). The sampling methods in imbalanced learning applications try to modify the imbalanced data set by some mechanisms in order to provide a balanced distribution by considering the representative proportions of class examples in the distribution. The cost-sensitive learning methods target the

imbalanced learning problem by using different cost matrices that describe the costs of misclassifying any particular data example [5]. Specific kernel-based learning methods and active learning methods for imbalanced learning have also been developed.

3. **Incomplete data.** Sometimes a collection of data to resolve a specific task is available but it has become incomplete due to being lost or because some of its variables or features are unknown. The solution to this problem is centred on approximating missing values, discovering a relationship between the known and the unknown data. Techniques based on neural networks [6] and from other perspectives, such as Multiple Kernel Learning [7], exist to solve this problem.
4. **Deluge of data**. We are now in the era of big data. The economist, K.N. Cukier [8], stated at the beginning of the present decade (2010s), there were about 1 trillion web pages and one hour of video downloaded to YouTube every second, amounting to 10 years of content every day; the genomes of 1000s of people, each of which has a length of $3.\ 8 \times 10^9$ base pairs, have been sequenced by various laboratories; Walmart handles more than 1M transactions per hour and has databases containing more than 2. 5 Petabytes of information, and so on. The generation of data is growing exponentially, and it is forecasted to reach more than 16 zettabytes in 2017 (1 zettabyte $= 2^{70} \approx 10^{21}$) [9]. Neural networks, based on real-time learning, provide fast and efficient learning, using massively parallel computations for big data analysis of the internet as well as in other contexts.
5. **High-dimensionality**. Data in real-world applications are frequently over abundant from the *resolution of a specific problem* point of view. As is shown in Section 2.1, there are models for neural networks that allow the discovery of latent factors with high dimensional data, reducing their dimensionality by projecting the data in a subspace of smaller dimension, thereby extracting the "substance" of the data.

Learning consists of estimating the parameters of a model of given data, and this concept is one of the most notable contributions of neural networks to the field of information processing systems. Learning makes it possible:

1. To not have to know the mechanisms (internal models) which are underlying a specific process in order to be able to implement it.
2. For the same neural network model to be used for various tasks. A neural network can be considered as a class of universal approximators which can implement a non-linear input-output mapping of a general nature [10].
3. To adapt the system to changes in the surrounding environment.

Three basic types of learning can be considered:

1. **Predictive** or **supervised learning**. The goal is to learn a mapping from inputs to outputs, given instances of a labelled training set of input-output pairs. It needs a knowledge of the desired answer for a certain input (input-output mapping), and the parameters are set, minimizing a cost function [11].
2. **Descriptive** or **unsupervised learning**. The learning is carried out on the basis of input patterns for which there is no specified output, i.e., “self-organized manner”. The aim is to produce new **knowledge** (“latent structures”) in the data, and to achieve better joint probability density estimators [12].
3. **Reinforcement** learning. In the same way that unsupervised learning is done without a teacher to provide the desired response at each step of the learning process, and based on the use of reward or punishment signals to estimate the parameters. The input-output mapping is performed through the continued interaction of a learning system with its environment so as to minimize a scalar index of performance [13].

There are variations of the basic types mentioned above such as **semi-supervised learning**, where traditional learning is combined with both unlabelled data and labelled data.

In the following sections we refer to the following topics in relation with neural networks: models, structures and algorithms (Section 2.1), simulators and hardware implementations (Section 2.2), and fields of use and real applications based on neural models (Section 2.3).

2.1. Models, structures and algorithms

As mentioned in Section 1, one of the objectives of the development of neural network models is to improve our understanding of human neural systems and to be able to carry out experiments and predictions with models without having to resort to the use of biological tissues, which frequently requires the use of invasive techniques. When the objective of a model is to facilitate research in the fields of neural biology or neuroscience, the most important and difficult task is to have the most detailed knowledge possible of the neural circuitry and mechanisms responsible for specific cognitive functions [14]. Obviously, we cannot simulate or emulate something, for example the human brain, with insufficient knowledge.

From the point of view of the engineering and construction of systems, which is the other objective of research into neural networks, it has not so far been possible to develop a model which could be considered as “universal” in the sense that it could applied efficiently to any information processing domain. The efficiency of a model can be established in terms of its precision in the resolution of a particular problem, and its complexity that determines the necessary resources for its implementation, with the aim of obtaining certain performances in issues such as processing speed, miniaturization and energy consumption. Generally speaking, a model functioning well in a specific domain or application may not be effective in other situations. As a result, a huge quantity of models has been developed with the aim of covering every type of problem in the real world. Besides various types of data, different algorithms and types of training have to be considered for each model. The engineer always has to try to find the best combination of models, algorithms and data in order to obtain the greatest possible efficiency as far as complexity and precision are concerned.

It can be considered that the development and maturation of artificial neural networks and model techniques have been produced in four periods, each of approximately two decades, the 1940s and 50s, the 1960s and 70s, the 1980s and 90s and the period of 2000 to the present day (see Table 1).

In the **first period** (the 1940s and 50s), models of individual neurons and their learning rules were proposed, as in the case of perceptron. The contributions of greatest influence in this period are described below.

The scientific community considers that the pioneers in the field of neural networks were McCulloch and Pitts [15], who were the first (1943) to introduce a **formal neuron model** which was both simple and unifying.

In 1949 the psychologist Hebb [16,17] launched the idea of biological neural networks which store information in the weights of their interconnections (synapses), and suggested a physiological learning procedure based on the modification of the synapses. This rule of adaptation and learning (unsupervised) has been widely used and continues to be used in numerous models.

Table 1
Periods of development of artificial neural networks.

Period	Facts	Concepts applied to artificial neural networks domain
1st period: 1940s and 1950s.	Models and learning rules of individual neurons.	Formal neuron model, perceptrons, associative memories.
2nd period: 1960s and 1970s.	Development of learning rules for single-layer networks, and the widespread application of techniques of statistical mechanics for recurrent networks.	Least mean-square algorithm (delta rule), Adaline, associative memories implementations, correlation matrix memory, Self-Organizing Maps (SOM), Adaptive Resonance Theory (ART), etc.
3rd period 1980s and 1990s.	Renewal of interest in the field of neural networks and a deepening study of self-organizing maps. Application and development of learning rules for multi-layer networks. Application of Bayesian methods and Gaussian processes.	Vector quantization (VQ), Discrete-Time Hopfield Neural Network, Principal Components Analysis (PCA), Boltzmann Machine (BM), Independent Component Analysis (ICA), Back-propagation learning (BP) (generalized delta rule), Radial Basis Functions (RBF), Cellular Neural Network (CNN), Natural gradient descent learning, Support Vector Machines (SVM), etc.
4th: 2000 until the present day.	Exhaustive theoretical studies to optimize and improve previous models: convergence analysis, statistical equilibrium, stability, estimation of states and control of synchronization.	Incremental Extreme Learning Machine (I-ELM). Deep Neural Networks (DNN).

Three years later (1952) Hodgkin and Huxley [18] established some dynamic equations which satisfactorily model the firing process and **spike propagation** in biological neurons. In this same line, Uttley [19] (1956) introduced the concept of the leaky integrate and fire neuron.

In 1956 Taylor [20] published a paper on **associative memories**, which are distributed units that learn or memorize by association, and try, to a certain degree, to imitate the process of association used in many different cognition models.

Later on, in 1958, Rosenblatt [21] presented a rule for learning (supervised) to establish the value of the weights and threshold for the McCulloch and Pitts neuron, with the aim of carrying out a specific information processing task. This showed that learning converges and these systems were denominated as **perceptrons**.

The **second period** (the 1960s and 1970s) is characterized by the development of learning rules for single-layer networks and the widespread application of techniques of statistical mechanics for recurrent networks.

In 1960 Widrow and Hoff [22] introduced the **least mean-square algorithm** (also known as the **delta rule**) and used it to propose the **Adaline** (Adaptive Linear Element), which is an adaptive pattern-classification neuron.

The **FitzHugh-Nagumo neuron model** [23,24,25] has been widely used to emulate the firing behaviour of the oscillating biological neurons of the sensory system. It uses phase space methods in conjunction with Hodgkin-Huxley equations and divides the phase plane into regions matching the physiological states of nerve fibres (active, resting, refractory, enhanced, depressed, etc.) forming a physiological state diagram, with the help of which many physiological phenomena can be condensed.

In 1967 Minsky [26] published a book on the study of the McCulloch and Pitts' model from the perspective of the automata and computation theories. Two years later (1969) Minsky himself, together with Paper [27], published a new book which showed the limitations of the isolated perceptrons (of one layer) to carry out certain information processing tasks, as these require the interconnection of perceptrons in various layers, when at that time the learning rules for these groupings were unknown. This book was very discouraging and considerably halted the study of neural networks for the next 15 years.

In 1972, Anderson [28], Kohonen [29] and Nakano [30], working individually, introduced the concept of **correlation matrix memory**. In this same year, also working independently, Kohonen and Amari, in a research line initiated by Taylor in 1956, proposed the idea of implementing **associated memories** with recurrent neural networks.

Nagumo and Sato [31] proposed a nonlinear model of a neuron that improved a previous model by Caianiello [32] where there is a critical value for the excitation level received by a neural unit below which an excitatory pulse cannot fire that driven unit. In the improvement suggested by Naguno and Sato, whenever a sequence of pulses with constant frequencies and progressively decreasing amplitudes is applied to the neural unit, the firing frequency of that unit decreases, as occurs in biological neurons.

In 1974, Little [33] introduced the concept of statistical mechanics related to the study of recurrent neural networks.

The first study on the formation of **self-organizing maps** using competitive learning and biologically inspired by the formation of topologically organized maps in the brain was done by Willshaw and von der Malsburg [34] (1976).

In the **third period** (the 1980s and 90s) there was a renewal of interest in the field of neural networks, the application and development of learning rules for multi-layer neural networks, and a deepening of the study of self-organizing maps was initiated. This period was characterized by the application of Bayesian methods and Gaussian processes, and the Support Vector Machines (SVMs) emerged. These exist alongside "biologically plausible" models which try to imitate, more or less faithfully, the behaviour of the brain, explain cognitive brain functions, such as learning and memory in terms of the behaviour of neurons and neuron populations, together with other models and algorithms orientated to the analysis of data whose origins are found in other domains of statistical mechanics, Shannon's information theory or multivariate statistics. The most relevant contributions of this third period are described below.

The techniques of obtaining **self-organizing maps** (**SOM**) (by unsupervised learning) proposed in 1976 by Willshaw and van der Malsburg, previously cited, were completed in 1980 by Amari [35] and in 1982 by Kohonen [36]. The self-organizing maps represent a model that searches for any low dimension structure which may be underlying the data, and sets a neighbourhood function with the aim of preserving the topological distances of the input space. The SOM are inspired by biological neural models, and it has already been proved that networks with a small number of nodes develop a process similar to K-means, whilst those formed by large quantities of nodes (thousands) may create emergent properties.

Another technique which uses unsupervised learning is **vector quantization** which obtains probability distribution from data, and represents an integration of unsupervised learning with Bayesian techniques [37].

In 1982 Hopfield applied the concept of the state of statistical physics in recurrent neural- network models, proposing networks with symmetric synaptic connections (**Discrete-Time Hopfield Neural Network** [38]).

Later, the **Generalized Hopfield networks** (GHN) including multilevel instead of two-level neurons and continuous-time functioning were proposed and analysed in some papers. As Zurada, Cloete, and van der Poel [39] proved in 1996, GHN show multiple stable states and preserve the stability properties of the original Hopfield networks. Other models based on the Hopfield networks have been proposed for different purposes, for example, the paper by Frasca5 applied Hopfield networks for learning from unbalanced data.

Although **Principal Components Analysis** (PCA) was already introduced in 1901 by K. Pearson with the aim of carrying out a linear regression analysis in the biological domain, but it was not widely used in the field of neural networks until the 1980s [40,41]. PCA is a technique for dimensionality reduction, finding the subspace in which data have the maximum variance.

A phenomenological model that reduces the number of differential equations of more complex previous models to only three was proposed by Hindmarsh and Rose in 1984 [42,43]. This relatively simple model accurately describes the dynamics of the action potentials and explains various phenomena of the oscillatory activity of the neuronal units.

An example of the application of the information theory for the design of algorithms is the **Bolzmann Machine** [44], which was proposed in 1985 by Ackley, Hinton and Sejnowski, based on the concept of simulated annealing, which produces a recurrent stochastic network, conveniently modelling the input patterns according to a Boltzmann distribution. The **simulated annealing**, mentioned previously, is a method to resolve optimization problems based on the principles of statistical mechanics thought up at that time by Kirpatrick, Gelatt and Vecchi [45] (1983) and Cerny [46] (1985).

Also in 1985 Herault, Jutten and Anns tackled the **blind source separation** problem [47,48], which consisted of extracting the original signals (the "*sources*") from an unknown mixture of them. The underlying concept of this problem became known as **independent component analysis** (ICA) and was later dealt with in detail by Comon [49], Hyvarinen and Oja [50] in 1994.

In 1986 Rumelhart, Hinton and Williams [51] applied and popularized the **Back Propagation** (BP) algorithm in the context of the neural networks, and answered the difficulties identified by Minsky and Paper in 1969 with respect to perceptron learning. They simply used the on-line gradient descent on the error function algorithm in feed-forward networks of graded-response neurons (**multilayer perceptrons**), and empirically proved the emergence of valuable internal representations in the hidden layers. It is interesting to note that Minsky and Papert referred to their doubts about the possibility of developing a method for multilayer perceptron learning starting from the Widrow-Hoff Delta Rule. In reality the BP algorithm was discovered and improved over almost three decades (1960s-1970s-1980s) by many researchers, among whom may be mentioned Bryson, Denham, and Dreyfus [52] (as pioneers), Ho, Linnainmaa, Parker, Werbos, LeCun, and Speelpenning [53].

As a result of the introduction by Grossberg [54] (1976) of the **Adaptive Resonance Theory (ART)** as a model of human cognitive information processing, from 1987 Carpenter and the same Grossberg proposed a real-time neural-network model that performed unsupervised and supervised learning, pattern recognition and prediction. Amongst these models we can find ART1 [55] (1987) for binary input patterns and fuzzy ART [56] (1991) for analog input patterns.

Several remarkable concepts were proposed in 1988:

1. For example, Linsker [57] applied the concept of entropy, which comes from information theory, and establishes a **maximum mutual information (Infomax)** principle, discovering a new concept of self-organizing in a perceptual network which can be applied to the resolution of problems related to statistical inference.
2. Broomhead and Lowe presented a new model for the construction of neural networks, different from multilayer perceptrons, which was based on the use of **Radial Basis Functions** (RBF) [58]. A RBF is based on a kernel method which frequently uses Gaussian functions.
3. Finally, in the same year, (1988), Chua and Yang proposed a model known as the **Cellular Neural Network** [59]. These networks are formed by a two-dimensional structure of connected cells, each of which is completely characterized by a non-linear differential equation and connected with its close neighbours. They have applications in real-time signal processing [60], for example, being able to build artificial retinas [61].

One year later, in 1989, Mead [62] published the book **Analog VLSI and Neural Systems** in which he links concepts of neurobiology and VLSI "neuromorphic" chips such as artificial retinas and cochleas. In the same year Y. H. Pao describes a system architecture and a network computational approach compatible with a general-purpose artificial neural-net computer [63].

The nonlinear dynamics shown in biological neurons is not usually described with enough accuracy by artificial neuron models. In 1990, Aihara, Takabe and Toyoda [64] proposed a model for **single neurons with chaotic behaviour** also including the properties of graded response, relative refractoriness and spatio-temporal summation of inputs shown in biological neurons. Despite its simplicity, this model provides a qualitative description of the chaotic responses experimentally observed.

Schutter and Brower [65,66] in 1994 proposed a detailed compartmental model of a cerebellar Purkinje-cell dendritic membrane. This model is based on an anatomic reconstruction of single Purkinje cells and includes ten different types of voltage-dependent channels modelled by Hodgkin-Huxley equations, and the use of voltage-clamp empirical data of this type of neuron.

In 1995, Bell and Sejnowski [67] dealt with the problem of **blind deconvolution**, more exactly described as blind source separation, previously considered by Herault and Jutten, who introduced models based on the theory of information.

Also worth highlighting are the contributions of McKay [68] (1992), Bishop [69] (1995) and Ripley [70] (1996), who introduced Bayesian techniques for inference, regression and classification [71], and those of Willians and Rasmusen [72] (1996) and Seger [73] (2004) for the inclusion of using the methodology of Gaussian processes within the domain of neural networks. All in all, these years produced closer links between the fields of neural networks and the probability theory, thus bringing about a new domain with the name of **Statistical Machine Learning** [74,75].

In 1998 Amari [76] introduced the rule of **natural gradient descent learning,** which allows the application of geometry concepts to information processing in neural networks.

In 1998 Vapnik and collaborators, searching for a learning analysis for the worst case scenarios, came up with the concept of **Support Vector Machines** [77,78]. The underlying idea of the new model was to use kernels to permit non-linearity. This concept has become integrated into other domains, producing various *kernel methods* [79,80]. The SVM are used in problems of pattern recognition, regression and density estimation, and although it may be argued that they are within neural networks, they are considered to be a different field as they do not try to reproduce biological behaviour.

At the end of this stage starts the development of hybrid systems that try to obtain the best of different methodologies. For example, in 1999 J.H. Chiang proposed a fuzzy integral-based approach for hierarchical network implementation that involves replacing the max (or min) operator in information aggregation with a fuzzy integral-based neuron, resulting in increased flexibility for decision analysis [81].

Several excellent books dedicated to neural networks and machine learning were published in this period, such as those by Haykin [82] and Luo and Unbehauen [83].

In the fourth and last period, which began in approximately 2000 and continues until now, no models have become so popular and aroused such interest as those produced in previous phases, nevertheless the theoretical study of previous models has notably deepened, with exhaustive studies into topics such as convergence analysis, statistical equilibrium, stability [84–88], estimation of states and control of synchronization, aiming to optimize and improve the models [89–95].

The quantitative analysis of neural networks with discontinuous activation functions was also a hot topic in this period [96–99].

An increased interest in **Complex-Valued Networks** (CVNN) [100–105] also occurred in this period. They represent parameters and input values as the corresponding amplitude and phase of a complex number, also multiplied by a complex-valued weight [106], as this complex representation is well suited to express real-world phenomena involving waves such as communications and frequency-domain processing in general. Moreover, it has been observed that CVNN show smaller generalization errors in the coherent signal processing by feed-forward networks. The algorithms corresponding to the required *complex* back-propagation (CBP) had been proposed some years previously [107].

One of the new models which has aroused interest is the **Incremental Extreme Learning Machine**, proposed in 2006 by G.B. Huang, L. Chen, and C.K. Siew [108,109]. Unlike the conventional neural-network theories and implementations, these authors show that single-hidden-layer, feed-forward networks with randomly generated additive or RBF hidden nodes (according to any continuous sampling distribution) can work as universal approximators and the resulting incremental extreme learning machine (I-ELM) outperforms many common learning algorithms.

The **Deep Neural Networks** (**DNN**) make up another concept which has aroused great interest in recent years. They are built from a cascade of hidden layers of units between the input and output layers [53,110]. The characteristics represented by each processing layer define a hierarchy of abstraction levels, from the lowest abstract to the highest, which infers useful representations via a learning process from large-scale unlabelled data.

Different network models are suitable for forming a deep neural network, either feed-forward networks or recurrent neural networks, the latter being very useful in applications for speech processing [111,112], computer vision [113,114], and in natural language processing [115]. The weights and thresholds of the neurons in each layer can be determined by either supervised or unsupervised learning [116].

One of the most relevant issues in DNN systems is the high computational cost required for training/testing. An important development in this field has been the elaboration of layer-wise unsupervised pre-training methods, using a greedy algorithm that can learn DNN based on restricted Boltzmann machines (RBMS) [117]. Thanks to powerful computing platforms, such as a cluster of computers or GPU, DNN has been successfully used [118]. The existence of novel learning algorithms that try to reduce the computational time for deep learning is also very relevant; see, for example the proposal presented in [119].

To some extent, the idea behind deep hierarchical representations can be considered as inspired by recent results in the field of neuroscience regarding the interpretation of processing patterns and the communication of information within the nervous system, such as neural codification, which tries to establish a relationship between stimulation and neural responses with the electric activity of neurons in the brain [120].

In spite of the dominance of models based on the theory of probability, the bio-inspired models have not been forgotten and are frequently taken up again and improved, mainly due to the advances in measurement techniques in the field of neuroscience, allowing the discovery of new functions and the fine-tuning of behaviour which should mimic bio-inspired models. Amongst these, it is worth highlighting the great interest that models based on *spiking neurons* (integrate-and-fire neurons, in general) has produced.

The fundamental difference between the models of integrate-and-fire neurons compared to the traditional ones is based on the way of representing information. Whilst in the traditional models information is represented in the amplitude of the signals or is digitally codified, the hardware implementation models of spiking neurons are based on coded analog variables by time differences between pulses, which has practical advantages over other encoding methods, and are significantly more realistic as there is experimental evidence that many biological neural systems use the timing of single action potentials (or *spikes*) to encode information. In 1997W. Maass [121] showed that the networks of spiking neurons are computationally more powerful than conventional models based on threshold or sigmoidal functions. There are biologically relevant tasks that can be implemented by a simple spiking neuron that would require hundreds of units in a sigmoidal neural network. Moreover, any function that can be computed by a small sigmoidal neural net can also be computed by a small network of spiking neurons. There are a large number of studies dedicated to these types of networks [121,122].

As in previous periods, new biological models have been proposed in this fourth period (Table 2). In general, these models show that the ones proposed in earlier stages, such as the Hodgkin-Huxley model, could be simplified without drastic consequences to the description of the spiking behaviour. The goal is not only to simplify the differential equations governing the dynamic behaviour of neurons, but also to reduce the large number of parameters (a hundred or more) that must be extracted from experimental data to provide a detailed characterization of the electrophysiological neuron behaviour.

In 2002 Maass showed the inefficiency of conventional computing (Von Neumann machines) to process biological information in realistic scenarios where real-time processing of time-varying input streams is required, and introduced the concept of the **Liquid state machine** (LSM) [123–125]; where a readout unit (neuron) that receives inputs from hundreds or thousands of neurons in a neural circuit learns to extract relevant information from the high-dimensional transient states of the circuit and transforms transient circuit states into stable readout units.

In particular, each readout unit learns its own conception of equivalence among dynamical states within the neural circuit, and performs its task on different inputs. Precisely, this unexpected finding about the equivalent states assigned to readout units in a dynamical system explains how invariant readout units are possible despite the fact that the neural circuit may never reach the same state again.

The idea of LSM has been quite successful because it has led to obtaining reasonably realistic models for biological neurons ("spiking neurons") and biological synapses ("dynamical synapses") being able to reproduce some parts of the brain functionality. Moreover, multiple computations, in the same or in different time scales, can be done using the same LSM.

In 2003, Izhikevich introduced a simple model that mimics the spiking and bursting behaviour of known types of cortical neurons [126]. This simple model of spiking neurons combines the biological plausibility of Hodgkin-Huxley-type dynamics with the computational efficiency of integrate-and-fire neurons, thus making it possible to simulate large-scale brain models (about tens

Table 2
Some prominent mathematical biological models.

1943	Formal neuron model proposed by McCulloch & Pitts	[15]
1949	Synapsis behaviour proposed by Hebb	[16,17]
1952	Firing process and spike propagation model proposed by Hodgkin & Huxley	[18]
1961	Neuron model proposed by FitzHugh & Nagumo (FHN)	[23,26,27]
1972	Mathematical neural model proposed by Nagumo and Sato	[31]
1984	Neural model proposed by Hindmarsh-Rose.	[42,43]
1990	Chaotic neural networks proposed by Aihara, Takabe & Toyoda.	[62]
1994	Active membrane model of the cerebellar Purkinje cell proposed by Schutter & Bower	[65,66]
2002	Liquid state machine (LSM) proposed by Maass	[123–125]
2003	Simple model of spiking neurons proposed by Izhikevich	[126]
2005	Adaptive exponential integrate-and-fire model (AdEx or aEIF), proposed by Brette & Gerstnetl	[127,128]

of thousands of cortical spiking neurons in real time with a resolution of 1 ms) on a desktop PC. Depending on four parameters, the model reproduces spiking and bursting behaviour of known types of cortical neurons. This model is used today by various general neural network simulators (see Section. 2.2.1).

In 2005, Brette and Gerstnet [127,128] introduced a two-dimensional adaptive exponential integrate-and-fire model, known as AdEx or aEIF. It combines an exponential spike mechanism with an adaptation equation, whose parameters can be systematically extracted from a series of standard stimulations. The AdEx model (1) combines an extension to quadratic or exponential integrate-and-fire neurons with allows the replacement of the strict voltage threshold by a more realistic smooth spike initiation zone; (2) makes possible the inclusion of sub threshold resonances or adaptation as described in the Izhikevich model; and (3) changes the stimulation method from current injection to conductance injection, thus allowing the integrate-and-fire models to move closer to a situation that cortical neurons would experience *in vivo*. The model correctly predicts the timing of the spikes in response to injection of noisy synaptic conductances.

An important issue in artificial neural networks, which has been the subject of many research studies, is the choice of size and the optimization of the structure.

The Kolmogorov theorem [129] shows that any continuous function of n variables can be mapped to a single function of one variable. From this theorem, the existence of three layer feedforward networks can be proved for every pattern recognition problem that was classifiable. Nevertheless, a way to get the mapping that solves a practical problem is not obtained from this proof as the Kolmogorov theorem is strictly an existence theorem. Although many methods for designing neural networks have been developed, no method to obtain an optimal or suitable architecture to solve any problem is yet known, and therefore, given a problem and a data set, several sets of neurons and different layer alternatives must be tested in order to discover a suitable architecture to solve the proposed problem. This search is often carried out by using genetic algorithms [130,131].

As is well-known, a neural network with a small number of hidden units is not capable of learning training data and therefore the accuracy of the system is limited. On the contrary, an overfitting problem may be obtained if the network has a large number of hidden neurons, and again, the performance of the system is limited. Three techniques are extensively used in the bibliography to determine the neural network structure automatically, namely constructing, pruning and hybrid approaches.

Constructive methodologies begin with a reduced number of hidden units and incrementally add nodes during training with the goal of increasing the accuracy of the obtained network systems, until a satisfactory solution is found. Examples of these constructive methodologies are: growing cell structure [132], constructive back-propagation [133], or an extreme learning machine with adaptive growth of hidden nodes [134].

Pruning algorithms begin with a large network and during the training procedure; hidden units or weights are removed in order to reduce the complexity of the system, thus obtaining a minimum network size with good generalization performance. This methodology is generally based on some measure of the parameter's relevance or significance, including sensitivity or significance methods [135], regularization and penalty term methods [136], or using other techniques such as optimal brain damage [137] or optimally pruned extreme learning machine [138]. The hybrid approach is a complementary method where the advantages of growing and pruning are combined simultaneously during the learning process [139–141]. In recent years several meritorious books on the topic of neural networks and machine learning have continued to appear, such as those by Haykin [142], Coolen [37] and Murphy [74].

Currently, due to the increase both in the accessibility and in the processing power of computers, research on the theoretical and practical aspects of the models mentioned has increased considerably with the aim of searching for and advancing our knowledge of the brain and its capacity to solve problems, and also into the response to new engineering challenges within the field of information processing.

2.2. Simulators and specialized hardware

From the neurobiological point of view, the fundamental aim of obtaining models of the various neural functions is to build processing systems which imitate the behaviour of natural processing systems, thus producing virtual laboratories where experimental approaches and clinical studies can be emulated in the most realistic way possible, without the need to use frequently invasive techniques on humans or animals.

From the engineering point of view, the aim is to produce systems imitating the properties of natural systems, such as cognitive capabilities, flexibility, robustness, the ability to learn, develop and evolve over a wide range of temporal scales, energy-efficiency, integration density, and fault tolerance, which are not achievable when using traditional procedures.

To sum up, a model by itself is not very useful; it is just an intermediate stage to facilitate the imitation or replication of the behaviour of the neural function or system via a computer programme (simulator) or via its implementation with specific hardware (neural hardware).

The following section goes on to describe the bases of the two basic types of implementation of artificial neural networks: firstly via software simulators (Section 2.2.1) and secondly by means of neural hardware (Section 2.2.2).

2.2.1. Simulators

To understand how the brain works, we need to combine experimental studies of animal and human nervous systems with numerical simulation of large-scale brain models.

This section is about general neural-network simulators, as well as software applications that are used to mimic the behaviour of artificial and biological neural networks. With the term "general" we mean that these programmes: a) run on general purpose computers and b) are used to simulate different kinds of neurons and networks, being able to modify in each simulation different parameters of these networks and also their topology thus allowing the customization of specific functionalities that the user may require. Within this field, simulators that are accessible in terms of documentation and also the source itself (free open source) are the main focus.

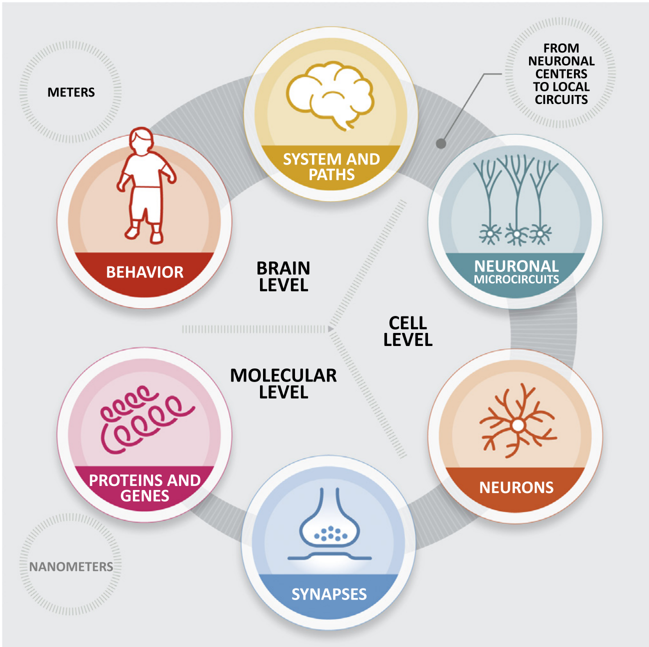

Fig. 1. Main levels of detail of a biological neural system.

Although there are a lot of simulators available (some of them developed only for specific structures and functions), in recent years different neural simulators have been developed for general purpose at different scales of detail (single detailed neurons, neural networks of different size and complexity, etc.) with libraries of models. They allow the user to set the neuron model and adaptation parameters, as well as to configure the network connectivity. With these simulators *in-virtual* (*in-silico*) experimentation can be done, predicting the behaviour of certain structures and functions, and obtaining empiric results towards matching measurements taken from biological structures (with *in-vitro* or *in-vivo* experimentation).

The simulations can be done at different scales and levels of detail. The structure and behaviour of the brain needs to be studied at different hierarchical levels of organization, covering proteins, genes, synapses, cells, microcircuits, brain regions, and cognition-behaviour (the whole brain) functions (Fig. 1). Different software applications have been developed to specifically perform simulations at one or various scales.

There are simulators which aim to mimic intra-cellular mechanisms in order to reproduce electrophysiological measurements with a strong focus on biological emulations leading towards a better understanding of the biological neural substrate. In this case, detailed biophysical representations of neurons are used, such as the Hodgkin-Huxley conductance model and the models described in Section 2.1 and listed in Table 2 (models proposed by FitzHugh-Nagumo, Hindmarsh-Rose, Schutter-Bower, Izhikevich, etc.).

In any case, the computational cost of neural-network simulation is proportional to its complexity; i. e. the level of detail of the neural models as well as the density and topology of the network. Whilst there are studies that require the simulation of the spiking generation processes or even a spike waveform, many other studies can adopt much simpler models such as the Integrate-and-Fire model. This model implements neural dynamics that are much simpler than the Hodgkin-Huxley model, simplifying the spike generation process by simple threshold firing mechanisms (when the membrane potential reaches a certain threshold the neuron fires a spike). Once the neuron generates a spike the cell undergoes a refractory period in which no further spikes are produced. There are other models, such as Izikevich's [126] and AdEx [143] that allow simulations using simple models (of just a few coupled differential equations) but with rich dynamics that facilitate the emulation of different neural biophysical behaviours (such as short term firing adaptation, burst response, etc.).

Simulations of large-scale networks typically use simpler models [144]. Furthermore, in the field of artificial neural networks, the focus is usually on computational primitives and they usually adopt rate-based models. In these models the neural response can be represented by an analog level (v_i) which is called the firing rate or activation of the neuron i. These models do not aim to be biologically plausible, because they leave out aspects of the biological substrate, such as how spike timing is related with processing and adaptation mechanisms (STDP, Spike Time Dependent Plasticity). This section focuses mainly on simulators that use spiking neural models, although rate-based models can be easily simulated with simpler software applications which are more suitable for artificial neural networks (allowing easier and faster simulations).

Generally speaking, neural biophysics is represented through hybrid mathematical models. On the one hand it is defined as neural state dynamics through differential equations (ordinary or partial and deterministic or stochastic), and on the other hand, a non-linear equation defines the synaptic trigger depending on the neural state and the received spikes. From a computational point of view, neural simulators can be seen as programmes that solve large scale coupled differential equations with numerical methods. The input and output variables are related (interconnected) according to the network topology and external stimuli.

There are three strategies to simulate spiking neural networks: time-driven, event-driven and hybrid (that integrates both the previous approaches). In the first, time-driven scheme, all the neural states are iteratively updated in each time slot, dt. These simulators are very easy to develop and can be applied to any model as the spike times are confined to temporal time slots. In terms of accuracy, the main problem comes from the discretization of time that leads to some error in the results, which is higher as dt increases. Thus, reducing dt (or even adopting methods to use dynamic dt that change throughout the simulation according to the neural state variable dynamics) leads to more accurate simulations, but the computational workload also increases dramatically as dt becomes shorter. Furthermore, in terms of scalability, the time driven computational load depends on the network size (in terms of number of neurons) because all the neural states of each neuron need to be updated periodically independently of their input/output variables.

With the event-driven strategy, the neural states are only updated when they receive or generate a spike. In this case, the neural model needs to allow discontinuous updates (for instance with linear or more complex dynamics that can be defined through analytic expressions). This will allow the calculation of a new neural state from a previous one (estimated in a previous simulation time). This can also be done using pre-calculated tables defining the neural dynamics along time intervals, which allows discontinuous simulations, making it unnecessary to iteratively calculate the neural state in intermediate instants intensively [145–148]. In this approach, the workload of a simulation depends on the global activity of the network and not on its size. For both strategies the computational complexity is optimal as it scales linearly with the number of synapses, although each scheme has its own assets and disadvantages [147,149].

A ranking of the different simulators cannot be done, due to changes and improvements being continuously carried out in most of them (and new ones that are being developed whilst others are discarded). Furthermore, the best choice depends very much on the research being addressed (either closer to biological plausibility or more focused on artificial neural system design). However, when choosing a specific simulator there are different factors that need to be taken into account [150]:

1. **Scale or level of the simulation**. For example, focus on the individual behaviour of a single neuron or global behaviour of a network (microcircuits, brain regions, etc.).
2. **Flexibility** related to the model neuron that can be used. There is a whole range of cell models from complex ones with internal dynamics defined at ion channel level and Hodgkin-Huxley models, to simpler ones, such as the adaptive exponential integrate-and-fire (AdEx) [143], Izhikevich [126], linear and non-linear IF neurons, etc.
3. **Precision** or biological plausibility: aiming to reproduce the neural response of biological neurons when stimulated with the same input patterns.
4. **Possibility of simulating short-term and long-term synaptic plasticity**.
5. **Simulation strategy**: event-driven (exact or approximated) or time-driven (with interpolation or not for spike times).
6. **Possibility** of simulating conductance-based (**COBA**) synaptic interactions.
7. **Support parallel processing** and distributed simulation for very large networks (for instance, distributed and multi-threaded simulations.) [151–154].
8. **Type of user interface**.
   a. Graphical or not.
   b. Possibility of interacting through the graphic interface for simple analyses (spike count, correlations, etc.) or complex (parameter fitting, FFT, matrix manipulations, etc.).
9. **Help and support**: e-mail, telephone, e-mail consulting, mailing lists, forum of users and community of developers.
10. **Documentation**. Tutorials, an on-line reference manual, published books concerning the simulator, list of publications of papers that use the simulator, etc.
11. **Price** of licence and maintenance.
12. Whether or not it is an **open source**.
13. Existence of a **web-site** for:
    a. Downloading the programme
    b. Obtaining all the documentation and
    c. Accessing the cell models that are used.
14. Possibility of importing and exporting (for instance in XML) model specifications, or other similar approaches such as PyNN (Python package for simulator-independent specification of neuronal network models [155]). Note that XLM is a standard for exchanging structured information between platforms.

Table 3
Some examples of simulators.

Acronym	Description	Original proposers	Web site
BRIAN [156,157]	Brian spiking neural network simulator	Romain Brette Dan Goodman Marcel Stimberg	http://briansimulator.org/
DigiCortex	Biological Neural Network Simulator	Ivan Dimkovic Ana Balevic	http://www.dimkovic.com/node/1; http://www.artificialbrains.com/digicortex
ECANSE [158] Siemens	Environment for Computer Aided Neural Software Engineering	Roman Blaško (Siemens)	http://www.trademarkia.com/ecanse-75026210.html
EDLUT [147,148]	Event Driven Look-Up-Table simulator	Eduardo Ros & col.	https://code.google.com/p/edlut/
emergent [159]	Emergent Neural Network Simulation System	Randall C. O'Reilly	https://grey.colorado.edu/emergent/
GENESIS [150,160,161]	GEneral NEural SImulation System	James Bower Dave Beeman	http://genesis-sim.org/
Mvaspike [150,162]	Modelling and simulating large, complex networks of biological neural networks, event-based.	Inria Sophia Antipolis (France)	http://mvaspike.gforge.inria.fr/
NCS [150,163,164]	NeoCortical simulator	Wilson C. E., Goodman P. H., Harris F.C.	http://www.cse.unr.edu/brain/ncs
NENGO [165–167]	Graphical and scripting based software for simulating large-scale neural systems.	Chris Eliasmith Terry Stewart Bryan Tripp	http://www.nengo.ca/
NEST [150,168]	Neural Simulation Tool	NEST Initiative	www.nest-initiative.org/
Neuron [150,169–172]	Neuron for empirically-based simulations of neurons and networks of neurons	Michael Hines	www.neuron.yale.edu/neuron/
Neuroph	Java neural network framework	Zoran Sevarac Ivan Goloskokovic Jon Tait	http://neuroph.sourceforge.net/
NN Toolbox	MATLAB Neural Network Toolbox	Mathworks	http://es.mathworks.com/products/neural-network/index.html
OpenNN	Open Neural Networks Library	Roberto López	http://www.intelnics.com/opennn/
PCSIM and CSIM [150]	Parallel neural Circuit SIMulator	Thomas Natschlager Pecevski Dejan	http://www.lsm.tugraz.at/pcsim/
SimBrain	Computer simulations of brain circuitry	Jeff Yoshimi	http://www.simbrain.net/
SNNAP	Simulator for Neural Networks and Action Potentials	John Byrne Douglas Baxter	http://nba.uth.tmc.edu/snnap/
SNNS [173]	Stuttgart Neural Network Simulator	University of Stuttgart, Maintained at University of Tübingen	http://www.ra.cs.uni-tuebingen.de/SNNS/
SpikeNET [174,175]	Neural simulator for modelling large networks of integrate and fire neurons	Arnaud Delorme Simon Thorpe	http://sccn.ucsd.edu/~arno/spikenet/
PSICS	Parallel Stochastic Ion Channel Simulator	Matthew Nolan	http://www.psics.org/
XNBC [176,177]	X- NeuroBioClusters	Jean-François VIBERT	http://ticemed-sa.upmc.fr/xnbc/,
XPP /XPPAUT [150,178]	General numerical tool for simulating, animating, and analyzing dynamical systems.	G. Bard Ermentrout John Rinzel	http://www.math.pitt.edu/~bard/xpp/xpp.html
VERTEX [179]	Virtual Electrode Recording Tool for EXtracellular potentials	Richard John Tomsett, and Marcus Kaiser	http://vertexsimulator.org/

15. **Requirements of the platform** (for instance, specific hardware requirements) that are needed for the simulator.
16. **Operating Systems** (LINUX, Windows, Mac-OS X, etc.) or framework (MATLAB, etc.), in which the simulations need to be run.
17. Possibility of **interfacing the simulator** to outside signals, such as a camera, or signals of real neurons.
18. Possibility of **interrupting the simulation**, storing the partial results and continuing the simulation from this point in the future.
19. Possibility of **coupling the simulation with front-ends**, such as Python or MATLAB, for interacting with the models, performing analysis, etc. without the necessity of modifying the original code.
20. Possibility of incorporating the simulator to a **user-extendable library** with new features, for example biophysical mechanisms without requiring modification of the original code.

Table 3 includes a list of widely known neural simulators. Most simulators in this table cover the main factors indicated, but at different levels. It is very complex to summarize to what extent they meet these factors. Furthermore, almost all of them are constantly being further developed with improvements and changes. Thus, it is preferable to include the website of each simulator, where it is possible to find the specific details and always in an updated form.

There are many other programs and frameworks, either of general purpose or that simulate functions or neural structures described in the literature, for instance: IQR [180], NeuroSpaces [181], NNET [182], NeuralSyns [183], NEUVISION [184], NeuroWeb [185], RSNNS [186], and See [187], or of particular models or levels as referenced in [188–192].

There is no specific simulator that is currently being used by the whole community (since some different approaches are more suitable than others, depending on the research task being addressed). There is no simulator that covers all simulation levels, scales, etc. On the other hand, there is the challenge of making the different simulation approaches compatible, being able to transfer models and structures from one simulation platform to the other, or being able to reproduce the same results easily with different simulators. For this purpose, platform independent cell-model descriptions are being proposed; for instance, PyNN, and also various simulators including interfaces with NEURON, which can be considered as the most widely used simulator for biologically plausible cell models.

Many of these simulations have been extended over the years to allow different kinds of simulations and increase their efficiency or provide support for different communities. NEURON and GENESIS are probably the most widely used simulators for detailed neuron models. They are usually used at a molecular level of abstraction (defining ionic channel densities and characteristics), multi-compartmental models, etc. In this framework, detailed model dynamics can be defined by hundreds or even thousands of differential equations, and a single detailed neuron simulation may require intensive numerical calculation. They have also been extended towards their use in parallel computers, but they are usually used for small scale neural systems (in terms of number of cells). Other simulators are more focused on "point neurons" (in which the neural model dynamics are defined by just a few coupled differential equations), as is the case of NEST, EDLUT, BRIAN, SpikeNET, etc. Even at a specific abstraction level, different simulators have been extended towards different goals, for instance, NEST has been extended with different libraries and interfaces towards facilitating network definition and simulation monitoring, and recently also optimized for massive parallel computers for large scale simulations. EDLUT has been extended for use on embedded real-time simulations, for instance in the framework of embodiment set-ups related with neuro-robotics or real-time simulations on conventional computers, without requiring specific neural hardware. Thus, depending on the ultimate goal of study, the level of abstraction and experimental set up, some tools may be more appropriate than others.

Another challenge is optimizing the networks [193] and adapting the simulators to new hardware platforms and new "neural friendly processors" [194], such as the ones described in the next section, in such a way that they can take full advantage of their computational capabilities to efficiently simulate large-scale neural networks.

2.2.2. Simulation platforms and neural hardware

As commented at the beginning of Section 2.2, a model (of neuron or network) by itself is not very useful and is just an intermediate stage to facilitate the emulation or replication of the behaviour of a neural function or system. The simulation or physical implementation may be carried out using conventional computers (where simulators described in the previous Section, 2.2.1 can be executed) or by using neural hardware when a greater velocity is required, or when a model is used for a specific application that, for example, requires being inserted in more complex systems with a reduced size and working in real time. There are two dimensions to consider when buying different simulation platforms or neural network emulators: firstly, ***flexibility***, which refers to the faculties the system offers to parameterize or scale the network model, the topology and the learning algorithms; and secondly, ***efficiency***, which is considered as the degree of adaptation (tuning) of the network with the application, as far as autonomy is concerned, the computing speed, miniaturization and energy consumption. Generally speaking, implementations in conventional computers are more versatile but less efficient. On the other hand, implementations in hardware systems and circuits specifically designed to emulate a function or neural network, are less flexible, but much more efficient (Fig. 2). In the next section the different implementation possibilities, with the help of examples from contributions of various research groups for each one are briefly described.

2.2.2.1. General purpose platforms. The most immediate, cheapest and flexible implementation (although the most inefficient), consists of using conventional computers, where the majority of the simulators described in Section 2.2.1 can be executed, but the advantage of parallelism, inherent to the functioning of the neural network due to the sequential execution of the programme (one instruction after another, data and data), is lost. Moreover, if the

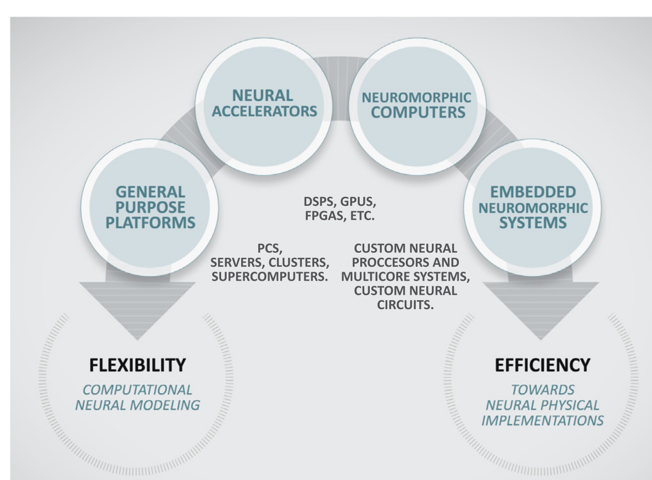

Fig. 2. Alternatives for platform simulation and neural hardware.

neural model is complex, as is the case of Deep Neural Networks, and/or the network to simulate is large this may be insufficient for obtaining results in a reasonable time on a conventional computer although it has a multi-core processor. In these cases it is necessary to opt for distributing the neural-network simulations in clusters or in supercomputers [153,195,196]; in other words, multi-computer platforms with parallel processing have to be used. Indeed, parallel processing in present and future high performance computers, implementing different levels of heterogeneous parallelism, constitutes an opportunity to simulate complex neural structures.

The most advanced simulators described in Section 2.2.1 include facilities or versions to be executed in multi-computers, using, for example, MPI interfaces. To carry out simulations in multi-computers it is necessary to work widely in the parallelization of the algorithms of various neural models, (back propagation [197,198], ART, neocognitron [199], recurrent NN, self-organizing NN, Bolzmann Machines [200], spiking models [201], etc.). Thus, in the EPFL, within the Blue Brain Project [202], a parallel version of the NEURON code has been developed, running on an IBM Blue Gene/P supercomputer with 16,384 computing cores and a peak performance of 56 TFlops, with this configuration facilitating the simulation of models at a cellular-level with up to 1 million detailed, multi-compartment neurons.

There are also platforms with resources to allow researchers to interactively visualize, analyse and "*steer*" within the simulation processes, and the platforms may even contain sensors for the acquisition of *in-vivo* signals [203] thus making bidirectional biology-silicon interfaces available.

Obviously, the computational load of the simulation of a neural network depends not only on the number of N neurons, but also on the topology of the network and the complexity of the neuron model, making the level of detail in which the simulation is carried out notably influence the computational resources required. To carry out single neuron to cellular level simulations of neural microcircuits, supercomputers with a processing power of the scale of Teraflops are essential, as the development of the Blue Brain Project, mentioned previously, has made clear [204]. Nevertheless, to carry out cellular-level simulations, for example of the whole mouse brain, or for molecular level simulations of single neurons, supercomputers with a processing power in the range of Petaflops are necessary. Once we achieve supercomputers with the potential of Exaflops, we will be able to carry out cellular level simulations of the complete human brain including dynamic switching to molecular-level simulation of some of its parts [205].

2.2.2.2. Neural hardware. There are certain applications or simulations for which general purpose computers and off-the-shelf devices and circuits are insufficient. These applications require specialized architectures, processors or systems specifically designed for the emulation of neural networks, in which the most common operations of the neural network have been boosted, and that really function with the inherent parallelism of these networks in such a way as to permit the processing of large quantities of data in a reasonable time (real time).

Generally speaking, the ad-hoc systems and hardware designed to emulate neural networks or abbreviated, **neural hardware**, try to offer a vehicle in which neural networks can be emulated directly in hardware rather than simulated on a general purpose computer which is structurally different from the neural network to be simulated, and thus obtain advantages such as: large scale network simulation, truly parallel processing capabilities, self-adaptation, fault tolerance [206], cost reduction, miniaturization and reduction of energy consumption. There are commercial applications available, such as video streaming, which requires the processing of large quantities of data, including learning and recognition, in real time. Other examples of real-world applications which use neural hardware, cited in [207], are optical character recognition, voice recognition, traffic monitoring, experiments in high energy physics, adaptive control, embedded microcontrollers for autonomous robots, and autonomous flight control.

To measure the performances of neural hardware the following parameters have traditionally been used:

1. *Processing speed:* Connections-per-second (CPS), measured as the rate of multiply/accumulate operations and transfer function computation during the recognition phase.
2. *Learning speed:* Connection-updates-per-second (CUPS) measured as the rate of weight changes during the learning phase involving calculation and update of weights.
3. *Synaptic energy:* average energy required to compute and update each synapse; measured as WCPS (watt per connection-per-second) or J per connection [208].
4. *Accuracy:* Bit connection primitives per second (CPPS) measured as the product of the number of connections per second multiplied by the precision (number of bits) of the input signals and by the precision (number of bits) of the weights.

2.2.2.3. Neuromorphic circuits. A **custom neural circuit** is an application-specific integrated circuit (ASIC) which replicates a neural function, structure or behaviour [209–213] more or less similar to the biological, counterpart and sometimes known as a **neurochip**. The term **silicon neuron** (SiN) is used to designate a hybrid analog/digital very large scale integration (VLSI) circuits that emulate the electrophysiological behaviour of real neurons [214]. Within the neural hardware, the qualification "**neuromorphic"**, thought up by Carver Mead [62], makes reference to the systems and circuits whose architecture and design principles are based on those of biological nervous systems. Among examples of neuromorphic circuits are silicon retinas [62,215–218] a silicon model of the cerebral cortex [219,220], auditory organ [221] and vestibular systems [222]. Neuromorphic systems deal with sharing important properties of the human brain, such as fault tolerance, the ability to learn and very low energy consumption.

In general, a custom neural circuit is made up of a multitude of processing elements (neurons) and synaptic connections which carry out more or less simple operations compared with the biological model. The operations are, for example, the product of every input for a weight, sums, mapping via the activation of a non-linear function, storage of weights, detection of the threshold, generation of spikes, etc. Some parameters of interest are the clock and data transfer rates, weight storage (analog/digital), hardwired or programmable interconnections, data representation in fixed point or floating-point, bits of precision, and degree of cascadability. It is common that some functions are carried out on a host computer, which is outside the neural network, particularly when a high level of programming is needed. Therefore, there are circuits with an on-chip/off-chip activation function, learning on-chip/off-chip or chip-in-the-loop training, etc.

Neural circuits may be digital, analog or hybrid. The benefit of using one or other technology depends on the application to be used. The analog implementations [223,224] have the advantage of being able to directly emulate the cerebral dynamics, obtain better response times and facilitate the integration of the networks with a greater number of neurons and synapses. On the other hand, the digital techniques are more flexible, they can be reprogrammed and reconfigured on-line, they allow the long-term storage of weights with greater precision, they have greater immunity to noise, the communication between chips is done without any loss of information and there is a good support of CAD tools in their design. To sum up, some operations are carried out more efficiently with digital technology, and others are better with analog technology. The hybrid custom neural chips try to take advantage of both designs, for example, the weights can be stored digitally, achieving a controlled precision, whilst analog processing elements increase the processing velocity.

As we mentioned in Section 2 there are various mathematical and topological network models that can be accurately simulated by computer programmes. However, many of these models are not able to exactly map on neural chips or on neural hardware, especially if they are based on analog circuits. This leads to a reduction in the efficacy of the learning and the precision of results. To resolve these problems researchers have suggested, see for example [193,194,225–228], changes in the models and learning processes with the aim of adapting them to the possibilities and limitations of the hardware, thus obtaining **hardware friendly models** or **hardware friendly learning algorithms.**

Neural chips may form part of various systems, such as neural accelerators, neuromorphic computers and embedded systems, as we will go on to describe.

2.2.2.4. Neural accelerators. A **neural accelerator** is a non-autonomous system which functions by being connected via a wide-band bus to a general purpose computer which acts as a host, in order to increase performance in the execution of some tasks inherent to neural computation. These accelerators are embedded in the host as co-processor boards which contain specialized processors, such as GPU [229–232], DSP [233,234], FPGA [235] or silicon neurons, as well as an additional solid-state memory. The system interface with the user is carried out via the peripherals of the host. Generally speaking, these heterogeneous platforms (with different types of processors) allow the simulation of different types of neurons and they can configure or programme various topologies efficiently on a single computer.

2.2.2.5. Neuromorphic computers and systems. A ***neuromorphic computer*** (sometimes known as a ***neurocomputer*** [236]) is an autonomous, customized, high-performance platform, built with the aim of emulating biological nervous tissue at different levels, mainly made up of individual synapses and neurons, and with similar programmability to a general purpose computer. These systems are based on very different computer architectures to those of von Neumann, with a structure and function inspired by the structure and function of the brain and usually containing neuromorphic circuits.

Among the actions or programmes proposed to build neurocomputers it is worth mentioning:

- ***SyNAPSE*** (Systems of Neuromorphic Adaptive Plastic Scalable Electronics) [237,238] which is a DARPA-funded program to develop an electronic neuromorphic machine that scales to biological levels and attempts to build a new kind of computer with similar form and function to the mammalian brain [239]. This project started in 2008 and it is scheduled to run until about 2016. The project is primarily contracted to IBM and HRL, who in turn sub-contract parts of the research to various US universities. The ultimate aim is to build an electronic microprocessor system that matches a mammalian brain in function, size, and power consumption. It should recreate 10 billion neurons, 100 trillion synapses, consume one kilowatt, and occupy less than two litres of space. The brain-inspired architecture consists of a network of neuro-synaptic cores, distributed and operating in parallel, communicating with one another via an on-chip event-driven net, without a common clock. Each core integrates memory, computation, and communication. Individual cores can fail and yet, like the brain, the architecture can still function. Chips communicate via an inter-chip interface leading to seamless scalability like the cortex, enabling creation of scalable neuromorphic systems. Within this field IBM has developed TrueNorth [240], a self-contained chip to achieve:

  1. One million individually programmable neurons.
  2. 256 million individually programmable synapses on one chip, which is a new paradigm.
  3. 5.4G transistors.
  4. 4,096 parallel and distributed cores, interconnected in an on-chip mesh network.
  5. Over 400 million bits of local on-chip memory ($\sim$100 Kb per core) to store synapses and neuron parameters.

- ***SpiNNaker*** (Spiking Neural Network Architecture) [241,242] is a massively-parallel multi-core computing system, based on a six-layer thalamocortical model, designed at the University of Manchester to improve the performance of human brain simulations. In contrast to traditional architectures for computers based on repeatable determinists and reliable computation, the communications between the Spinnaker nodes are carried out via simple messages, (40 or 72 bits), which are inherently unreliable and whose efficacy is based on the principles of massive parallel computation, having incorporated mechanisms for fault detection and recovery. The basic building block of the machine is the SpiNNaker multicore System-on-Chip, a Globally Asynchronous Locally Synchronous (GALS) system with 18 ARM968 processor nodes residing in synchronous islands, surrounded by a light-weight, packet-switched asynchronous communications infrastructure. Each computer node contains, besides 128 Mbytes off-die SDRAM, two silicon dice: the SpiNNaker chip and a Mobile DDR (Double Data Rate) SDRAM SpiNNaker. The project anticipates a total of 57 Knodes (in total up to 1,036,800 ARM9 cores and 7 Tbytes of distributed RAM) [243]. The machine has the ability to simulate different neural models (simultaneously, if necessary) in contrast to other neuromorphic hardware [244]. This platform is being further developed in the framework of the Human Brain Project (see Section 4).
- ***The neuromorphic computing system of Heidelberg University***. Various neuromorphic systems and circuits have been developed in this university, one of which is currently operational which features 200,000 neurons and 50,000,000 synapses on a complete silicon wafer manufactured in a 180 nm CMOS technology. The Brain Scales project (HICANN chip) aims to reduce the number of transistors of the electronic circuits required to emulate the neurons by using an analog approach [245]. Within the Human Brain Project (see Section 4), this research platform is being further developed using the concept of universal and configurable physical models of neural circuits which serve as prototypes for a completely new kind of computer architecture [246].
- ***Neurogrid*** is a neuromorphic system for simulating large-scale neural models in real time which is enough to include multiple cortical areas, yet detailed enough to account for distinct cellular properties [247]. The fundamental component is not a logic gate, as in a sequential, step-by-step Von Neumann architecture, but rather a silicon neuron, whose behaviour and connectivity are programmable. The design criteria were the following: 1) emulation of the synapse and dendritic tree with shared electronic circuits [248], in a way in which the number of synaptic connections are maximized; 2) all the electronic circuits are made with analog technology, except those that correspond to axonal arbours, thus maximising energy efficiency; and 3) the neural arrays interconnect with a tree network topology. These options facilitate the simulation of a million neurons, with billions of synaptic connections in real

time, using 16 neurocores integrated on a board, consuming 3 W, approximately 100,000 times less energy than if the simulation was done with a supercomputer. Creating and simulating cortical models on Neurogrid is straightforward: it is enough to describe the neural model by writing a Python script: assigning each of Neurogrid's sixteen neurocores to model a different cortical layer (or cell-type), tailoring its silicon neurons to emulate the cell-population's ion-channel repertoire and routing its softwires to match their synaptic connectivity. The Neurogrid software has a GUI to visualize the simulation results and change the model parameters interactively.

2.2.2.6. Embedded neuromorphic systems. Another category within neural hardware is that of the **embedded neuromorphic systems** for use in high-performance computing, advanced services, and products of consumer electronics [249], smart sensors, robotics, etc. But in contrast to the neuromorphic computers they are oriented towards very specific applications in real time, and do not function autonomously, rather they carry out their functions within a larger electronic or mechanic system where they are embedded. In these cases it is typical to use co-designed hardware/software techniques [250,251,235]. These systems may contain custom neural circuits, FPGAs or off-the-shelf hardware

2.2.2.7. Other trends and technologies. Custom neural chips are expensive due to their high development and production costs, which, like any integrated circuit, are only commercially profitable if they are produced in very large quantities. Therefore, in recent years the majority of hardware implementations have been carried out with Field Programmable Gate Arrays (FPGA) [252–254]. The same FPGA can be configured or reconfigured to implement various models of ANN. The FPGAs are integrated circuits which are extremely attractive due to the following peculiarities: a) flexibility and programmability (via hardware description language they can be tailored to the function or functions to be carried out), b) a reduced development time thanks to the well-proven CAD/CAE tools available, c) the possibility of including certain analog functions, such as comparators and A/D and D/A converters at some E/S pins, d) availability of many "hard" computing cores integrated in the FPGA itself (with performances close to those of ASIC), and e) very reasonable costs. The relatively low number of neurons (thousands) that can be implemented in an FPGA is one of the limitations presented in comparison with custom (ASIC) implementations.

Researchers in recent years have also notably increased their interest in the implementation of models based on the use of *spikes* for communication between neurons, which, as commented in Section 2, are closer to natural behaviour, and the hardware implementations of spiking neurons are extremely useful for a large variety of applications [255–259]. Therefore, various models with spiking dynamics have been proposed, which go from the simplest, such as the integrate-and-fire (I&F) basic model, to other more complex models which are closer to biology, such as the Hodgkin–Huxley, Morris–Lecar, FitzHugh-Nagumo [260] and Mihalas–Niebur. More details can be seen in [214].

Currently there is a growing and unusual interest in the development of neuromorphic circuits, using a new electric component known as a **memristor**. This component is a bipolar element, which complements the three other passive, basic elements which are used in circuit theory (resistor, capacitor, and inductor) just as L. Chua [261] predicted. An efficient physical implementation and a nanoscale of the memristor were not obtained until 2008 in the HP Laboratories [262]. The term memristor comes from the contraction of "memory resistor".

The resistance to the passing of the current within the memristor is a function of the electrical currents that previously flowed through it, that is to say, the more current that has flowed before, the more easily the current then flows. Therefore every memristor has the capacity to carry out simultaneous logical operations and storage, in a similar way to the synapses of the neural circuits of the brain [263]. It has been demonstrated that it is possible to design circuits with memristors to model different biological functions, models and systems [264,265] such as that of the Hodgkin-Huxley model [266], associative memories [267], cellular neural networks [268], and mixed time-varying delayed neural networks [269].

At the end of 2011, in the Hughes Research Laboratories (HRL Labs, LLC), researchers managed to construct a memristor array on a CMOS chip. This circuit has very low energy consumption and a density of memory components of 30 Gbits/cm [2], never previously reached by any other electronic technology. The use of memristors makes the construction of very large arrays of cores possible with sufficient numbers of neurons to match the human brain more closely [270].

Despite proposals regarding **optical technology** to implement weighted interconnections, associative memory and other neural functions that have been carried out for years [271–276], currently microelectronic technology continues to be the most adequate for physically implementing neural networks, as it provides stable processes, it is familiar, controllable and reproducible, besides being relatively cheap and available worldwide. **Quantum technology** [277,278] and **molecular technology** [279] have certain capacities, such as the possibility of functioning concurrently, but the research is still at a very early stage in the field of neural-network implementations. In fact, there are researchers who are exploring the possibilities of these technologies in this field [280–284] and have so far obtained sufficient practical results to be a substitute for electronic technology.

The most notable development of neurochips or silicon neurons took place in the 1980s and at the beginning of the 1990s, when it was realized that the computers at that time were not sufficiently powerful to satisfactorily emulate long, complicated neural algorithms. However the notable increase in the performance of general purpose hardware, faster than Moore's law, and the better scalability of the algorithms in these platforms combined to produce a lack of interest in developing specialised hardware [285]. It is necessary to remember that the characteristic that was required of the neurochips was parallelism and this was introduced little by little in conventional computer architectures, even at the level of processors.

Another inconvenience comes from the fact that the development of **neurochips** has a very limited commercial interest, due to their high specialisation. The more specialised a chip is, the less use it has in a number of problems, although its efficiency and yield is greater. This leads to there being no market for the production of microchips in large quantities as they would be commercially unviable. Nevertheless, the scientific and technological interest in these types of circuits is undeniable. As we mentioned previously, an alternative to custom-made chips, specialised in neural networks, is the use of FPGAs, as they can be produced in large quantities, making them commercially viable.

As a conclusion to this section, it can be stated that the development of neural hardware has significantly influenced progress in:

1. The development of specialised hardware to support neurobiological models of computation and apply it to obtain solutions for advanced services, industry and consumer electronics.
2. The exploration of new computer architectures inspired by the brain and based on new concepts of coding, learning, massive parallelism and processing with stochastic variables, far from the traditional concepts based on discrete logic as Robert Noyce

already stated in 1984 [285]: *"Until now we have been going the other way; that is, in order to understand the brain we have used the computer as a model for it. Perhaps it is time to reverse this reasoning: to understand where we should go with the computer, we should look to the brain for some clues".*

2.3. Areas of use and real world applications

This section shows some examples of the implementation of systems to solve real-world problems using computational models inspired by neural networks in nature. These implementations may be more or less close to reality and are frequently used in conjunction with other data processing techniques, such as the statistical learning theory and the information theory.

Neural networks are especially useful to infer underlying knowledge in observations or when the data or the tasks are so complex that they are unrealizable within reasonable times with traditional methods. The information processing systems based on neural networks have become standard tools, especially useful for solving real world problems, since the end of the 1980s [286].

In order to analyse the use of artificial neural networks, three dimensions can be considered: 1) areas of application, 2) fields of application, and 3) achievable tasks.

The neural network **areas of application** are varied, some of them are: astronomy, mathematics, physics, chemistry, earth and space sciences, life and medical sciences, social and behavioural sciences, economics, arts and humanities, and engineering. The neural network **fields of application** include disciplines related to the areas mentioned above, as shown in Table 4. The terminology used in both cases belongs to the fields and disciplines of the International Standard Nomenclature for Fields of Science and Technology of UNESCO.

This section includes references to examples that illustrate the variety of applications, without trying to be exhaustive as not all the cases reported in the literature are included.

It is difficult in each field of application and even for each task, to compare the use of neural networks versus other techniques (e.g., statistical methods or a support vector machine); therefore we prefer to make some general considerations.

There is a current trend in the neural network community towards more biologically plausible neural networks, and with the momentum of the Human Brain Project, models of plasticity, learning and memory capabilities or large-scale models of cognitive processes are mainly considered. It is important to highlight that in comparison with other statistical methods or the support vector machine, ANN has a relevant characteristic, since the time and temporal correlation of neuron activity is very relevant in neural signal processing.

Table 4
Some fields of application of neural networks.

- Administrative & Business management [287–289].
- Bioinformatics [290].
- Biometric identification [74,291–293].
- Control and industrial automation [294,295].
- Chemistry [296,297].
- Digital communications [298].
- Ecology [299,300].
- Electromechanics [301,302].
- Energy resources [303–306].
- Finances [307–313].
- Genetics [314].
- Geology [315,316].
- Internet (e-commerce, internet search, filtering) [317–319].
- Image processing [320–322].
- Manufacturing [323,324].
- Medical diagnosis [325–330].
- Medicine and health [331–334].
- Microbiology [335].
- Meteorology, climatology [336–341].
- Molecular biology [342,343].
- Natural resources [344–346].
- Organization and management of enterprises [347–350].
- Remote sensing.
- Robotics [351–353].
- Signal processing [49,50,354–360].
- Space [361,362].
- Speech and language processing [363–365].
- Scientific taxonomies [366].
- Etc.

An important characteristic of ANN is the self-adaptive behaviour, again essential in applications that need to adapt to changing environments (control and industrial automation, meteorology, climatology, etc). In fact, the morphology has great importance in the network's ability to learn, and it is necessary to implement dynamic framework structures that can be used to modify ANN configuration, such as "structural learning" as a mechanism to create specific network topologies that facilitate the processing roles of different neural layers. In this way, dendritic growth can be included, as well as increasing and decreasing neurons/layers, interconnection changes, morphology, etc.

Another important characteristic of ANN is the parallel computing architecture, with its essential behaviour in real-time applications, such as speech processing and image processing, using frameworks that can carry out these tasks as efficiently as human performance. This has a great impact in multiple disciplines and applications, from speech and natural language processing, to image processing, or problems in bioinformatics and biomedical engineering.

Nevertheless, SVM and statistical methods are very good tools that have been extensively used in many different classification problems. An advantage of these methods is that they provide a good generalization capability and the problem of over-fitting (very relevant in conventional ANN) is avoided. As disadvantages (in comparison with ANN) it is important to highlight that the model obtained is less transparent or easy to understand, consuming a great deal of time in training and frequently the determination of optimal parameters is difficult to obtain (mainly in nonlinearly separable data).

The most important basic tasks or processing information operations that neural networks can perform are: complex pattern recognition, function estimation, classification problems, and discovery of latent factors. The tasks, according the learning process, are [74,142]:

Supervised learning:

1. **Classification** [367–371] including pattern recognition, sequence recognition, identification of new or unknown data, and decision-making. The goal is to learn a mapping of each input into an output class.
2. **Pattern association** by means of associative memories.
3. **Regression analysis** [372] or **functional approximation**, including tasks such as system identification, modelling, fitness approximation, time series prediction [373–375] and forecasting [376,377]. This is similar to the classification task, but the output variable is continuous.

Unsupervised learning:

1. Discovering clusters (**clustering** data into groups [378–380]).
2. **Extraction of latent factors**. This task is usually done reducing their dimensionality by projecting the data in a subspace of smaller dimension; which extracts the "substance" or latent factors of the data. The most common approach to dimensionality reduction is the Principal Component Analysis (PCA) [41]. It is possible to adaptively extract the principal and minor components from the autocorrelation matrix of the input signals. The principal component subspace contains the relevant information and the significant characteristics of the input signal, whilst the minor component subspace represents an additive noise that corrupts the principal information.
3. **Discovering graph structures** to establish the degree of correlation between different variables.

Table 5
Some real-world problems resolvable with learning techniques and application fields.

Classification and clustering	
• Face detection and recognition [74,291–293]	Biometric identification
• Traffic sign recognition [320].	Image processing
• Texture classifier [321].	
• Handwriting recognition [322].	
• Document classification and e-mail spam filtering [74].	Internet
• Detecting intrusions and attacks through the Internet [317,318].	
• Biomedical images classification [325,326].	Medical diagnosis.
• Classification and diagnostic prediction of cancers [327,328].	
• Microarray gene expression cancer diagnosis [329].	
• Pattern recognition on medical images [330].	
• Supervised pattern recognition in food analysis [331].	Medicine & health
• Cloud classification [337] and detection via satellite remote sensing [338].	Meteorology
• Virtual screening of compounds [332].	Pharmacology
• Classifying flowers [74].	Scientific taxonomies
• Classification of EGG signals (in BCI, etc.) [355,356].	Signal processing
• Satellite selection for GPS navigation [361].	Space
Modelling, functional approximation and forecasting	
• Brand choice decisions [288].	Business management
• Modelling processes in Analytical Chemistry [297]	Chemistry
• Modelling the *Escherichia coli* fermentation process [296].	
• PID controllers design [295].	Control
• Prediction pollutant levels [300].	Ecology
• Forecasting financial and economic time series [309].	Economy and finances
• Corporate credit ratings [312]. Credit scoring and prediction [310,311].	
• Financial distress prediction [313].	
• Modelling in induction motors [301].	Electro-mechanics.
• Adaptive position tracking control of permanent magnet synchronous motor [302].	
• Modelling of energy systems [303–305].	Energy resources.
• Electrical load forecasting [306].	
• Model for analysis of the Drosophila Melanogaster genome [314].	Genetics
• Prediction of geological risks [315].	Geology
• Predicting the age of a viewer watching a given video on YouTube [74].	Internet
• Decision making [347,348]. Multiple criteria decision-making [349,350].	Management.
• Machinery diagnosis [323].	Mechanics
• Modelling for knee rehabilitation [333].	Medicine & health.
• Predicting the amount of prostate specific antigen (PSA) in the body [74].	
• Predicting climate variables (temperature, wind speed, etc.) [339–341]	Meteorology
• Protein function prediction [342].	Molecular biology
• Modelling, predicting and forecasting water resources [344–346].	Natural resources
• Tracking control of a biped robot [352].	Robotics
• Enhancing robot accuracy [353].	
Discovering clusters	
• Autoclass system [362], discovered a new type of star, based on clustering astrophysical measurements [74].	Astronomy & space
• Cluster users into groups, according to their web purchasing or browsing profile in order to customize the advertisements to be displayed to each group [74,289].	Business management.e-commerce
• Cluster flow-cytometry data into groups, to discover different sub-populations of cells [74,366].	Scientific taxonomies
Discovering latent factors	
• Motions capture data to a low dimensional space, and using it to create animations [74].	Computer graphics
• Using PCA to interpret gene microarray data [74].	Genetics
• Detection of changes on the Earth's surface [316].	Geology
• Filtering for network intrusion detection [319].	Internet
• Feature extraction in gearbox fault detection [324].	Mechanics
• Using latent semantic analysis (a PCA variant) for document retrieval [74].	Natural language.
• Speech processing and language modelling [363–365].	
• Signal processing in Brain Computer Interfaces (BCI) [334].	Medicine & health
• Signals separation into their different sources [49,50,357–360].	Signal processing
Discovering graph structure	
• Determination of the phosphorylation status of some proteins in a cell [343].	Molecular biology.
• Improving financial portfolio management by learning a sparse graph [74].	Economy and finances
• Predicting traffic jams on freeways [74].	Engineering
• Recovering, from time-series EEG data, the neural “wiring diagram” of a certain kind of bird [74].	Neuroscience
Matrix completion	
• Inpainting images to obtain realistic textures [74].	Computer graphics
• Market basket analysis and predicting in commercial mining data [74].	Economy and finances
• Collaborative filtering for, say, predicting the desired movies by a person on the basis of what they have previously seen [74].	Internet

4. **Matrix completion** and **beam forming** that, for example, makes it possible to estimate values for missing data or to reconstruct a signal, that is, to find a previously unknown signal from corrupted or imperfect observations.

Different examples of applications, including its application field and the kind of task achieved, are shown in Table 5.

Signal processing [83] is a particularly useful field in neural networks. This field includes different kinds of data such as audio, video, speech, image, communication, biomedical, geophysical, radar, sonar and seismic. Different techniques have been developed in order to perform:

- **Filtering**. It achieves some mapping from the input to the output signals, changing time-invariant or adapting the weight coefficients.
- **Signal detection** involves inferring from (imperfect) observational data whether or not a target signal is present.
- **System identification**, including blind source separation.
- **Signal reconstruction** [381].
- **Signal compression**, with the aim of reducing the bit rate of a signal digital representation with minimal losses in the signal quality.
- **Spectral estimation** for maximum entropy spectral discovering, harmonic retrieval, multichannel spectral estimation, two-dimensional spectral estimation, and high-order spectral estimation.
- **Array Signal Processing**, estimation of the direction of sources arrival from the signal array.

Most of the above applications can be developed with conventional techniques of information processing, therefore it is not enough to prove only that a particular task is achievable using an artificial neural network, but also to have to compare the solution provided with other alternatives, showing greater efficiency considering parameters, such as accuracy, robustness, speed, consumption energy, miniaturization and price. One of the notable advantages of neural networks is their function as a **black-box** which allows modelling systems in which no rules that underlie them and that determine their behaviour are explicitly known. In any case, the cited examples in this section clearly reveal that neural networks are being successfully applied in a wide range of areas and fields, contributing to the resolution of scientific and industrial real-world problems by performing specific tasks related to the capacity of inferring the underlying knowledge in the observations and, in general, in conjunction with other information processing techniques or procedures.

3. Current frameworks

In recent years, artificial neural networks are no longer considered as an isolated field, due to the development of new frameworks and disciplines which are naturally included in this domain. Within these contexts, artificial neural networks play an important role, being relevant to highlighting Computational Neuroscience, Neuromorphic Computing, Neuro-engineering, Natural Computing, Machine Learning, Computational Intelligence and Neuro-informatics. This section briefly describes the role of neural networks within each of these subjects.

Computational Neuroscience [382] is a discipline that seeks to understand how brains generate behaviours using computational approaches. The work of computational neuroscientists is mainly based on analysing how populations of highly interconnected neurons are formed during development and how they represent, process, store, act upon, and become altered by information present in the body and the environment [383]. The main objective is to create models as realistic as possible, from single-neuron modelling, to complex models of biological neurons, where there may be complex interactions between inhibitory and excitatory neurons, to handle challenges in the description of sensory processing, and to better understand the behaviour of our memory, its configuration, structure and evolution of the synapses, etc. In this discipline, the contributions presented in the development of new models and neural-network algorithms are able to summarize the essential biological characteristics at multiple spatial and temporal scales, from membrane currents, proteins, chemical coupling, etc.

As already mentioned above (Section 2.2.2), the notion "neuromorphic" was introduced by Carver Mead [62] referring to artificial neural networks whose architecture and design principles are based on the natural nervous system. The main goal of **Neuromorphic Computing** is the establishment of a computing field derived from knowledge of the structure and functioning of the brain in order to be able to apply basic research, applied research and industrial applications, and the development of systems and neuromorphic computing devices which are able to learn, act and evolve over a wide range of time scales and levels of energy consumption, integration density and fault tolerance better than those obtained with conventional high performance computers. Speed scales can range from biological real-time to ten thousand times faster. **Neuromorphic Engineering** is a discipline in which the synergy of different fields (such as biology, mathematics, computer science and electronic engineering) can be used simultaneously to design novel artificial neural systems for real application, with a clear inspiration from biological systems (for example for vision/perception/auditory/mobility system that emulate the biological nervous structure of real systems).

Neuroengineering or **Neural Engineering** [384,385] is a discipline from the field of biomedical engineering, the main objective being the application of a knowledge of relationships between neurons, neural networks and the functions of the nervous system in the development of engineering techniques to better understand, repair, replace, interact, or exploit the properties of neural systems. Relevant emphasis is focused on the design and the resolution of problems related to the interfaces between neural tissue and artificial devices or systems, such as neuroprosthetics and brain-machine interfaces.

Some research fields and applications of neuroingeneering are [386]: a) *Neural imaging*, with techniques such as Computed Axial Tomography (CAT), Magnetic Resonance Imaging (MRI), functional Magnetic Resonance Imaging (fMRI) and Positron Emission Tomography (PET) scans. This research field analyses the behaviour and organization of the brain; b) *Artificial Neural Networks* which provide their ability to obtain biologically plausible models of neural systems that can be used for the analysis and design of devices that can help patients with diseases or injuries (for example cochlear implants for hearing, vision, etc.; c) *Neural interfaces* and *Brain Computer Interfaces* (*BCI*); the first has the aim of designing engineered devices that can replace neuronal function, and BCI that enables a direct communication between the brain and the object to be controlled, mainly through the analysis of the signals of our brain, as well as diagnosing and treating intrinsic neurological dysfunction [355,387,388]; d) *Neural microsystems* and *microelectrode arrays*, focused on obtaining knowledge about the electrical, chemical, optical behaviour of neural systems; f) *Neural prostheses* are electronic and/or mechanical devices capable of supplementing or replacing functions lost by disease or failures of the nervous system, stimulating and recording their activity, often using microelectrode arrays; g) *Neurorobotics* or the development of autonomous systems, are robotic devices that have control systems based on principles and models of the brain, and therefore use brain-inspired algorithms [389–391]; and finally h) *Neuroregeneration*, or neural tissue regeneration, restoring the functions of damaged neurons.

Natural Computing [392] is a research field that explores the computational process presented in nature and human-designed computing inspired by nature. Natural computing can be categorized into three classes of methods: 1) development of nature-inspired models/techniques of computation (including neural computation inspired by the functioning of the brain, evolutionary algorithms, swarm intelligence inspired by the behaviour of groups of organisms); 2) methods that synthesize nature by means of computing (artificial life systems inspired by the properties of natural life in general, etc.); and 3) methods that use natural materials (e. g. , molecules) to compute (for example molecular computing, or DNA computing, and quantum computing, amongst others).

Computational Intelligence [393] includes a set of methodologies that are inspired by biological functions exhibited in natural systems to solve problems of information processing that are ineffective or unfeasible when solved with traditional approaches (explicit statistical modelling). Main approaches included within computational intelligence are *Fuzzy Logic* [394], which is a paradigm working with concepts and techniques for dealing with inaccuracy, granularity of information, approximate reasoning, and especially, to compute with words; *Evolutionary Computation*, which gives the possibility of a systematized random search with the achievement of optimal and efficient performance by algorithms that emulate certain aspects of biological evolution and genetics (crossover, mutation, and survival and reproduction of best adapted organisms); *Reconfigurable Computing*, which aims to build systems with plasticity in which, for example, the hardware (interconnected circuits) can dynamically and in real time adapt to the environment, or reconfigured to a failure of any of its elements in order to keep running, even with limited benefits; and *Neurocomputation*, which offers the possibility of designing artificial neural networks with the capacity to learn, generalize, self-organize, adapt and identify.

In fact, many hybrid computational intelligence methods have been developed that efficiently combine procedures in the areas of artificial neural networks, fuzzy logic and evolutionary computation to be applied in complex domains [395]. One example is the use of genetic algorithms for the choice of the size and the optimization of the structure of neural networks, as mentioned in Section 2.1.

Computational intelligence techniques have been successfully applied in the field of **machine learning** [74,75], dealing with complex problems in order to automatically identify patterns in the data and then predict future data from the discovered patterns, relevant examples being shared with Neurocomputation techniques, such as Support Vector Machines, Bayesian networks and Extreme Learning Machines.

Finally, **Neuroinformatics** [396] deals with all aspects related to the development of computer models, databases and software tools necessary to integrate, analyse and exploit the “deluge” data produced in the field of neuroscience. These data are related, for example, with the description of multiple levels of brain organization, ranging from genes and molecules to microcircuits, systems and behaviour. A relevant goal of data organization is to facilitate access to data for analysis, modelling and simulations.

4. Challenges towards understanding the biological substrate of the brain

Biological research on brain and neural systems has advanced significantly in recent decades, but the complexity of the mechanisms involved in different scales has also been discovered. Moreover, the different technologies that have been developed (genetically modified organisms, patch clamping, multi-cell recording, fMRI, etc.) have enabled us to more clearly identify the barriers that must be overcome for an in-depth investigation into the brain. We have also become aware of the important challenge of a "better understanding of the brain" and its high potential impact on our society.

Especially related to artificial neural networks, a deeper understanding of the brain would allow the reverse engineering of some computational networks and principles of biological neural systems and integrate them into new generations of artificial neural systems and machine learning approaches, particularly well-suited for our every-day biologically relevant tasks. Thus, the field of brain research can be seen as a continuous source of information, models, computational principles, and approaches that have a potential impact into the next generation of artificial neural networks.

The human brain is an extremely complex system and the existing experimental methodologies used for extracting data are very limited and are mainly rather indirect. Thus, the access to internal properties of the neural system, network topology, adaptation mechanisms, etc. is constrained. Although animal experimentation has advanced significantly in recent years, providing a great deal of data, there are many specifically human capabilities, such as high level cognition, reasoning, social capabilities, etc. that are unique to human beings and now need to be researched with non-invasive and indirect techniques. Furthermore, though the advances in recent years have been impressive, the lack of a general integration and standardization of the data, models, etc. makes the reproduction of the results from other studies difficult, and thus leads to very inefficient incremental and continuous advances. Many models, experimental setups, and so on need to be re-developed from scratch by different laboratories.

The better understanding of the human brain requires the massive international collaboration of many laboratories in different interdisciplinary fields, and it is one of the challenges of this century. Therefore, two flagship projects (HBP and BRAIN) have been defined in Europe and USA which are described below in this section, and in parallel with them other countries have been formulating their own national plans. Some of them are briefly mentioned below.

For example, Japan has been formulating its own project based on the following three objectives: to focus on studies of non-human primate brains that will directly link to a better understanding of the human brain, to elucidate the neural networks involved in brain disorders such as dementia and depression, and to promote close cooperation between basic and clinical research related to the brain. This project, dubbed Brain Mapping by Integrated Neurotechnologies for Disease Studies, (**Brain/MINDS**) [397], was launched in 2014 and will integrate new technologies and clinical research. In this programme, challenging goals will be achieved through long-term research carried out by linking core research institutions nationwide.

In Australia a specific programme has also been set up with preliminary funds of around $250 million over 10 years with the goal of developing the world's first bionic brain. The creation of the Australian Brain initiative (**AusBrain**) [398], is the main focus of a recently released report to improve and better coordinate Australia's efforts in brain research.

There is also another ambitious initiative in China (**Brainnetome**) [399], based on previous developments [400]. The goals of this programme are to dissect the functional atlas of brain networks for perception, memory, emotion, and their disorders as well as to develop advanced technologies to achieve these goals. It also aims to encourage collaboration among interdisciplinary researchers through continuous support for up to 25 top laboratories devoted to brain network studies in China (for more than 10 years).

The Norwegian Brain Initiative (**NORBRAIN**) [401,402] is a large-scale national infrastructure project. The objectives are to better understand the integrated functional systems of the brain and use this knowledge to develop new diagnostic tools and new treatment against neurological and neuropsychiatric disorders. In addition this national infrastructure allows neuroscientists, with different backgrounds, to use new generations of research tools and to provide insight into how complex mental functions and dysfunctions emerge from distributed neuronal activity in a local brain circuit.

Previously there have been other smaller initiatives with goals aligned with the challenges that are now being addressed by the

HBP, BRAIN, etc. Some examples of these initiatives in Europe are the following (according to their initial proposals and public web sites): **SpikeFORCE** [403] (2002–2005): its objective was to understand the neural principles that give an organism the ability to learn multiple tasks in real-time with minimal destructive interference between tasks, and to recreate this ability in real-time spiking neural networks; **SenseMaker** [404] (2002–2005): the aim of this project was to conceive and implement electronic architectures that are able to merge sensory information sampled through different modalities into a unified perceptual representation of our environment; **FACETS** [405] (2005–2009): the goal of this project was to create a theoretical and experimental foundation for the realisation of novel computing paradigms which exploit the concepts experimentally observed in biological nervous systems; **SENSOPAC** [406] (2006–2010): aimed at developing a cerebellum-based control system to solve complex haptic/touch problems using active sensing; **BrainScaleS** [407] (2011–2015): the goal was to understand multiple spatial and temporal scales in brain information processing based on *in-vivo* experimentation, computational analysis and computational synthesis; **BBP** [408] (started in 2005): the ultimate goal of the Blue Brain Project was to reverse engineer the mammalian brain; **REALNET** [409] (2011–2014) is a project aimed at investigating novel computational architectures inspired by the cerebellar network (the central role this circuit plays in generating sensory-motor integration and higher cognitive processing and by the unsurpassed level of detail that has been reached in understanding its neurobiological functions).

Among the largest initiatives that have caused greater expectation are the **Human Brain Project** (**HBP**) launched in Europe and **Brain Research through Innovative Neurotechnologies** (**BRAIN**) in the USA. Both initiatives share many aspects, such as addressing the long-term goal of better understanding the neurobiological substrate (anatomical, molecular and circuit bases) of computational primitives of the brain. Both initiatives address this long-term challenge with a plan that extends over 10 and 12 years, and will be supported with approximately 1000 million and 5000 million dollars, respectively, though these long term funding schemes require revisions and modifications according to different strategies and resource constraints that may arise in these long time periods.

The HBP initiative [410,411] is focused on an integrated effort and the development of the six general platforms that facilitate research in different fields related to the human brain:

1. *Neuroinformatics Platform* aimed at aggregating neuroscience data to build multi-level brain atlases and navigate in them.
2. *Brain Simulation Platform,* to develop software tools to build and simulate multi-scale brain models at different levels of detail.
3. *Medical Informatics Platform*, aimed at federating clinical data and records in order to identify and classify diseases.
4. *High Performance Computing*, aimed at developing and operating High Performance Computing (HPC) systems scaled and optimized for brain simulations.
5. *Neuromorphic Computing Platform* aimed at developing biologically inspired systems and devices approaching brain-like intelligence and other natural properties.
6. *Neurorobotics Platform* [412] aimed at developing robotic systems for closed loop cognitive experiments, allowing them to interface detailed brain models to a simulated body in a simulated environment.

   To sum up, the Human Brain Project's platforms will give scientists from all over the world a single point of access to neuroscience data, multi-omic clinical data and analysis tools. The platforms will allow them to reconstruct and simulate the brain on supercomputers coupled to virtual bodies acting in virtual environments (*in silico* behavioural experiments), and provide them with pipelines to develop simplified brain models for implementation in neuromorphic computing systems, with the ultimate goal of simulating the brain.

   The initiative BRAIN will be led by the US National Institutes of Health (NIH), the Defence Advanced Research Projects Agency (DARPA), and the National Science Foundation (NSF). BRAIN aims to accelerate the development and application of new technologies, so that neuroscientists will be able to produce an innovative dynamic image of the brain showing how individual cells and complex neural circuits interact in time and space. The following areas have been identified as high priorities [413].

   1. **Discovering diversity**: "to identify and provide experimental access to the different brain cell types to determine their roles in health and disease".
   2. **Maps at multiple scales**: "to generate circuit diagrams that vary in resolution from synapses to the whole brain. It is increasingly possible to map connected neurons in local circuits and distributed brain systems, enabling an understanding of the relationship between neuronal structure and function".
   3. **The brain in action**: "to produce a dynamic picture of the functioning brain by developing and applying improved methods for large-scale monitoring of neural activity".
   4. **Demonstrating causality**: to link brain activity to behaviour with precise interventional tools that change neural circuit dynamics.
   5. **Identifying fundamental principles**: "to produce conceptual foundations for understanding the biological basis of mental processes through the development of new theoretical and data analysis tools".
   6. **Advancing human neuroscience**: "to develop innovative technologies to understand the human brain and treat its disorders; create and support integrated human brain research networks".
   7. **From the BRAIN Initiative to the brain**: "to integrate new technological and conceptual approaches produced in goals 1–6 to discover how dynamic patterns of neural activity are transformed into cognition, emotion, perception, and action in health and disease. The most important outcome of the BRAIN Initiative will be a comprehensive, mechanistic understanding of mental function that emerges from the synergistic application of the new technologies and conceptual structures".

As indicated above, the American BRAIN initiative is more focused on the development of new technologies that will enable disruptive advances in the research fields related to the brain.

Paul M. Matthews [414] indicates that new tools for functional analysis of circuit activity are expected to be delivered to the BRAIN consortium in the first 4 years and to a wider community in the following years. New technologies, such as new sensor systems, will also be studied and potentially lead to new brain-machine interfaces. Therefore, he highlights that "new technologies and exploitation opportunities will emerge", and a plan towards their translation to actual products is envisaged in the BRAIN initiative. The HBP has also been structured with a starting ramp-up phase in which their different platforms will be released to the wider community.

On the other hand Erik Kandel [414] highlights more the impact of the HBP and BRAIN initiatives on the new capabilities to treat brain related diseases, such as schizophrenia, depression, bipolar disorders, post-traumatic stress disorder, addiction, Alzheimer's disease, amyotrophic lateral sclerosis, Parkinsonism, etc. He also highlights the opportunity of a better understanding of the unique characteristics of the human mind.

Christof Koch [414] highlights the goals addressing the construction of massive online databases of meso-scale connectivity, systematically classifying cell types, linking their electrophysiological properties with their dendritic and axonal projection patterns and the genes that they express in their cell bodies and finally arriving at the definition of functional properties of neurons towards creating in silico models of cortical regions. Some significant advances are being achieved by the Allen Institute for Brain Science [415]. He also indicates the importance of making all this data widely available through dedicated web tools. Interestingly, integration and availability are in fact, one of the main focuses of the HBP initiative that has been planned and is now finishing the first version of a general site where data, models, tools will be made available.

Seth Grant, co-leader of the Strategic Mouse Brain data subproject of HBP believes that [416] "A key goal of the Human Brain Project is to construct realistic simulations of the human brain–this will require molecular and cellular information and from that we will be able to model and understand biological and medical processes. In adittion, we will be able to use this information to design and implement new kinds of computer and robots".

5. Conclusions

As shown in this paper, over the years, neural models and simulations are making it possible to reveal fundamental principles of neural computation to neuroscientists. Moreover, we can affirm that the interest in artificial neural networks is growing so much that its models and algorithms are becoming standard tools in computer science and information engineering. The goal has been to obtain systems with similar cognitive capacities, flexibility, strength and energy efficiency to the human brain.

Section 2 describes the development and evolution of different topics concerning neural networks, such as data problems, learning techniques, models, simulators, hardware implementations and real-world applications. This highlights the fact that after a long and productive youth, neural networks have formed a robust set of computation procedures with a robust theoretical base and undeniable effectiveness in solving real problems in different fields of information processing.

One of the greatest contributions of neural networks to the information processing systems is the introduction to the concept and techniques of learning (Section 2) which have certain well-defined advantages over other forms based on statistical procedures. Among these advantages are [417]: (1) the possibility of doing the learning phase with all the data in an incremental way, online, instead of using samples; simple algorithms being highly proved, (2) adequate scalability, (3) algorithms easily parallelizable and executable in massive parallel platforms, and (4) neural hardware available for real time learning (Section 2.2).

Emphasis has been placed on the description of different models and algorithms developed over the years that have being grouped in four phases, each one lasting two decades (Section 2.3). Models of individual neurons and their learning rules, such as perceptron, were proposed in the first period (the 1940s and 50s). The second period (the 1960s and 70s) is characterized by the development of multilayer network learning rules and the application of statistical mechanics techniques to recurrent networks. The revival of neural networks took place in the third period (the 1980s and 90s) and started with a deeper study of self-organizing networks which was characterized by the application of Bayesian methods, Gaussian procedures and the development of support vector machines (SVM). In the last period (from 2000 to now) the theoretical study of previous models has deepened, dealing with issues such as convergence analysis and statistical balance, stability, state estimation and control synchronization. In this period of time new models and techniques have also been developed, such as Incremental Extreme Learning Machines and Deep Neural Networks (DNN).

Section 2.2 presents the analysis of the development of simulators and models of physical implementation of neural networks. Particularly Section 2.2.1 shows that, although there are a lot of programmes to simulate particular functions or structures, new generic neural simulators, at different scales (single neurons, neural networks, etc.) have been developed in recent decades. These new simulators have template libraries, are user expandable, allow the defining different parameters for each model and even network connectivity.

We can experiment with these *in-virtual* simulators, predicting the behaviour of certain structures and functions, and also obtain empiric results very close to those of biological material. Clearly, there is not one general purpose simulator which is better than the others, that is, there is no simulator that is able to simulate any neuron model or any network topology. It is a challenge to obtain compatibility with different simulators in order to easily transfer models or algorithms between them.

Another challenge is the optimization of networks and the adaptation of simulators to new hardware platforms and new processors, as we described in Section 2.2.2. This section is dedicated to neural hardware with the objective of improving all their possibilities and to achieve the simulation of high scale networks.

A challenge not yet achieved is the construction of neuromorphic circuits or computer chips that properly mimic the human brain. In fact, one of the top 10 emerging technologies in 2015, established by the World Economic Forum, is Neuromorphic Engineering [418]. As shown in Section 2.2.2, computers are based on data movement between memory circuits and processors through high-speed backbones. This causes bottle necks in the data and instruction transfer and to the consumption of a lot of energy wasted as heat. Nevertheless, the brain contains billions of elements (neurons) massively interconnected that make either distributed functions of memorization or processing, intimately related, being extremely efficient energetically. This is one of the main requirements for today's computer design [419]. Neuromorphic technology is estimated to produce a qualitative and quantitative leap in the progress of high performance computation, including intelligent chips that may be able to learn and adapt, with greater operational speed and less energy consumption.

Section 2.3 described several real-world applications for neural networks in information processing, showing their potential in tasks such as data classification, pattern recognition, functional estimation and discovery of latent factors, performed more efficiently than by using other techniques.

It is also remarkable that gradually the field of artificial neural networks has greatly contributed to the birth and development of other disciplines, where it has become integrated, naturally contributing with relevant concepts. Amongst these disciplines are Computational Intelligence, Machine Learning, Computational Neuroscience, Neuro-engineering, Natural Computing, and Neuroinformatics (Section 3).

In the scientific communities of several countries and in society in general, as described in Section 4, there is a great interest in deepening the knowledge of the human brain, this being one of the challenges of the twenty-first century. This has led the European Commission and the USA to establish two major projects, leading to studying the human brain thoroughly, and even trying to simulate it, either partially or totally, with the aid of high performance supercomputation. The aim is to get to know the "algorithms" that govern information processing within a neural circuit and the interactions between circuits in the brain as a whole, which, without doubt, will lead to new medical treatments and new computing technologies.

In the context of neural networks, the most daring question we could ask would be: are we currently capable of building a human brain? [420]. We cannot forget that the human brain has around 90 billion neurons shaping an extremely complex network, but we have ever more accurate models of the human brain and compiled data. The Human Brain Project literally states [421]: "Terascale computers have already allowed us to make the leap from simulations of single neurons to cellular level simulations of neuronal microcircuits. Petascale computers, now available, are potentially powerful enough for cellular-level simulations of the whole rodent brain, or for molecular level simulations of single neurons. Exascale computers, predicted for the end of this decade, could allow cellular level simulations of the complete human brain with dynamic switching to molecular-level simulation of parts of the brain when required".

Undoubtedly, the achievement of the challenges described in this paper will provide a better understanding of artificial neural networks and computational neuroscience in general. We will be able to get a glimpse of the way some unique features of the human mind are performed, such as high level cognition, reasoning, decision-making, consciousness, emotion, freewill and creativity [414].

The authors hope that this paper will be of interest to researchers who want to have a global vision of research and challenges in the world of neural networks, related to mechanisms of information processing and with the aim of a better understanding of the brain and the bio-inspiration to construct systems with amazing properties and functions inspired by nature.

Acknowledgements

This work has been partially supported by the Spanish National Grants TIN2013-47069-P and TIN2012-32039, and by the projects P11-TIC-7983 and P12-TIC-2082 of the Andalusian Regional Government (Spain), co-financed by the European Regional Development Fund (ERDF).

References

[1] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, A. Juditsky, Nonlinear black-box modeling in system identification: a unified overview, Automatica 31 (12) (1995) 1691–1724.
[2] S. Geisser, Predictive Inference, Chapman and Hall, 1993.
[3] R. Kohavi, R, A study of cross-validation and bootstrap for accuracy estimation and model selection, IJCAI 14 (2) (1995) 1137–1145.
[4] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[5] M. Frasca, A. Bertoni, M. Re, G. Valentini, A neural network algorithm for semi-supervised node label learning from unbalanced data, Neural Netw. 43 (2013) 84–98.
[6] Z. Ghahramani, M.I. Jordan, Dept. of Brain & Cognitive Sciences, MIT Center for Biological and Computational Learning. Technical Report 108, 16 pages. MIT, Cambridge, MA 02139, 1994. ⟨http://mlg.eng.cam.ac.uk/zoubin/papers/review.pdf⟩.
[7] R. Kumar, T. Chen, M. Hardt, D. Beymer, K. Brannon, T. Syeda-Mahmood, Multiple Kernel Completion and its application to cardiac disease discrimination. Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on, IEEE, 2013.
[8] V. Mayer-Schönberger, C. Kenneth, Big data: A revolution that will transform how we live, work, and think, Houghton Mifflin Harcourt, 2013.
[9] J. Bornholt, R. Lopez, D.M. Carmean, L. Ceze, G. Seelig, K. Strauss, A DNA-based archival storage system, in: Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems, ACM, 2016, pp. 637–649.
[10] K. Hornik, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (5) (1989) 359–366.
[11] M.I. Jordan, D.E. Rumelhart, Forward models: Supervised learning with a distal teacher, Cognit. Sci. 16 (3) (1992) 307–354.
[12] Z. Ghahramani, Unsupervised learning, in: O. Bousquet, G. Raetsch, U. von Luxburg (Eds.), Advanced Lectures on Machine Learning. Lecture Notes in Artificial Intelligence, 3176, Springer Verlag, Berlin, 2004.
[13] R.S. Sutton, A.G. Barto, Reinforcement Learning. An Introduction, MIT Press, 1998.
[14] M.I. Rabinovich, M.K. Muezzinoglu, Nonlinear dynamics of the brain: emotion and cognition, Phys.-Uspekhi 53 (4) (2010) 357–372.
[15] W.S. McCullough, W.H. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. 5 (1943) 115–133.
[16] D.O. Hebb, The organization of behavior: a neuropsychological theory, Psychology Press, 2005, Original edition: Wiley: New York; 1949.
[17] W. Gerstner, W.M. Kistler, Mathematical formulations of hebbian learning, Biol. Cybern. 87 (2002) 404–415.
[18] A.L. Hodgkin, A.F. Huxley, A quantitative description of membrane current and its applications to conduction and excitation in nerve, J. Physiol. 117 (1952) 500–544.
[19] A.M. Uttley, A Theory of the Mechanism of Learning Based on the Computation of Conditional Probabilities, Gauthier-Villars, Paris, 1956.
[20] W.K. Taylor, in: E.C. Cherry (Ed.), Electrical Simulation Of Some Nervous System Functional Activities Information Theory, 3, Butterworths, 1956, pp. 314–328.
[21] F. Rosenblatt, The Perceptron: a probabilistic model for information storage and organization in the brain, Psichol. Rev. 65 (1958) 386–408.
[22] B. Widrow and M.E. Hoff, Jr., Adaptive switching circuits, IRE WESCOM Convention Record, pp. 96–104.
[23] R. FitzHugh, Impulses and physiological states in theoretical models of nerve membrane, Biophys. J. 1 (1961).
[24] J. Nagumo, S. Arimoto, S. Yoshizawa, An active pulse transmission line simulating nerve axon, Proc. IRE 50 (10) (1962) 2061–2070.
[25] M. Ruth, Matthias, B. Hannon, Fitzhugh-Nagumo Neuron Model, Modeling Dynamic Biological Systems, Springer, New York 1997, pp. 82–86.
[26] M.L. Minsky, Computation: Finite and Infinite Machines, Prentice-Hall, Englewood Cliffs, N.J., 1967 1960.
[27] M.L. Minsky, S.A. Papert, Perceptrons, MIT Press, Cambridge, MA, 1969.
[28] J.A. Anderson, A simple neural network generating an interactive memory, Math. Biosci. 14 (1972) 197–220.
[29] T. Kohonen, Correlation matrix memories, IEEE Trans. Comput. C-21 (1972) 353–359.
[30] Nakano, Association: a model of associative memory, IEEE Trans. Syst., Man Cynbern. (1972) 380–388, SMC-2.
[31] J. Nagumo, S. Sato, On a response characteristic of a mathematical neuron model, Kybernetik 10 (3) (1972) 155–164.
[32] E.R. Caianiello, Outline of a theory of thought-processes and thinking machines, J. Theor. Biol. 1 (1961) 204–235.
[33] W.A. Little, The existence of persistent states in the brain, Math. Biosci. 19 (1974) 101–120.
[34] D.J. Willshaw, C. von der Malsburg, How patterned neural connections can be set up by self-organization, Proc. R. Soc. Lond. Ser. B 194 (1976) 431–445.
[35] S.I. Amari, Topographic organization of nerve fields, Bull. Math. Biol. 42 (1980) 339–364.
[36] T. Kohonen, Self-organized formation of topologically correct feature maps, Biol. Cybern. 43 (1982) 59–69.
[37] A.C.C. Coolen, R. Kühn, P. Sollich, Theory of Neural Information Processing Systems, Oxford University Press, 2005.
[38] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Natl. Acad. Sci. USA 79 (8) (1982) 2554–2558.
[39] J.M. Zurada, I. Cloete, E. van der Poel, Generalized Hopfield networks for associative memories with multi-valued stable states, Neurocomputing 13 (2) (1996) 135–149.
[40] E. Oja, A simplified neural model as a principal component analyzer, J. Math. Biol. 15 (1982) 267–273.
[41] E. Oja, Principal components, minor components and linear neural networks, Neural Netw. 5 (6) (1992) 927–936.
[42] J.L. Hindmarsh, R.M. Rose, A model of the nerve impulse using three coupled first-order differential equations, Proc. R. Soc. Lond. B221 (1984) 87–102.
[43] J.L. Hindmarsh, P. Cornelius, The development of the Hindmarsh-Rose model for bursting In Bursting, 3–18, „ World Science Publication, Hackensack, NJ, 2005.
[44] D.H. Ackley, G.E. Hinton, T.J. Sejnowski, A learning algorithm for Boltzmann Machines, Cognit. Sci. 9 (1985) 147–169.
[45] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Sci. New Ser. 220 (4598) (1983) 671–680.
[46] V. Černý, Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, Journal. Optim. theory Appl. 45 (1) (1985) 41–51.
[47] J. Herault, C. Jutten, B. Anns, Detection de grandeurs primitives dans un message composite par une architecture de calcul neuromimetique un aprentissage non service, Procedures of GRETSI, Nice, France, 1985.
[48] C. Jutten, J. Herault, Blind separation of sources, part I: an adaptive algorithm based on neuromimetic architecture, Signal Process. 24 (1) (1991) 1–10.
[49] P. Comon, Independent component analysis: a new concept, Signal Process. 36 (1994) 287–314.
[50] A. Hyvarinen, E. Oja, Independent component analysis: algorithms and applications, Neural Netw. 13 (4–5) (2000) 411–430.
[51] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations of back-propagation errors, Nature 323 (1986) 533–536.
[52] A.E. Bryson, W. Denham, S.E. Dreyfus, Optimal programming problems with inequality constraints, AIAA J. 1 (11) (1963) 2544–2550.
[53] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Netw. 61 (2015) 85–117.

[54] S. Grossberg, Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors & II: Feedback, expectation, olfaction, and illusions, Biol. Cybern. 23 (121-134) (1976) 187–202.
[55] G.A. Carpenter, S. Grossberg, A massively parallel architecture for a self-organizing neural pattern recognition machine, Computer Vision, Graph., Image Process. 37 (1) (1987) 54–115.
[56] G.A. Carpenter, S. Grossberg, D.B. Rosen, Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system, Neural Netw. 4 (6) (1991) 759–771.
[57] R. Linsker, Self-organization in a perceptual network, Computer 21 (3) (1989) 105–117.
[58] D.S. Broomhead, D. Lowe, Multivariable functional interpolation and adaptive networks, Complex Syst. 2 (1988) 321–355.
[59] L. Chua, L. Yang, Cellular neural networks – theory, IEEE Trans. Circ. Syst. 35 (10) (1988) 257–1272.
[60] L. Chua, L. Yang, Cellular neural networks – applications, IEEE Trans. Circ. Syst. 35 (10) (1988) 1273–1290, Oct 1988.
[61] M. Anguita, F. Pelayo, F.J. Fernandez, A. Prieto, A low-power CMOS implementation of programmable CNN's with embedded photosensors, Circuits and Systems I: Fundamental Theory and Applications, IEEE Trans. 44.2 (1997) 149–153.
[62] C.A. Mead, Analog VLSI and Neural Systems, Addison-Wesley, Reading, MA, 1989.
[63] Y.H. Pao, Y. Takefji, Functional-link net computing, IEEE Comput. Journal. 25 (5) (1992) 76–79.
[64] K. Aihara, T. Takabe, M. Toyoda, M, Chaotic neural networks, Phys. Lett. A 144 (6) (1990) 333–340.
[65] E.R.I.K. De Schutter, J.M. Bower, An active membrane model of the cerebellar Purkinje cell. I. Simulation of current clamps in slice, J. Neurophysiol. 71 (1) (1994) 375–400.
[66] E.R.I.K. De Schutter, J.M. Bower, An active membrane model of the cerebellar Purkinje cell. II. Simulation of synaptic responses, J. Neurophysiol. 71 (1) (1994) 400–419.
[67] A.J. Bell, T.J. Sejnowski, An Information-maximizatium approach to blind separation and blind deconvolution, Neural Comput. 6 (1995) 1129–1159.
[68] D.J.D. MacKy, A practical Bayesian framework for backpropagation networks, Neural Comput. 4 (1992) 448–472.
[69] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford Universituy Press, Oxford, 1995.
[70] B.D. Ripley, Pattern Recognition and Neural Networks, Cambridge University Press, Cambridge, 1996.
[71] C. Bielza, P. Larrañaga, Bayesian networks in neuroscience: a survey, Front. Comput. Neurosci. 8 (2014) 131–153.
[72] C.K.I. Willian, C.E. Ramunsen, Gaussian processes for regression, in: D. S. Touretzky, M.C. Mozer, M.E. Hasselmo (Eds.), Advanced in Neural Information Processing Systems, 8, MIT Press, 1996, pp. 514–520.
[73] M. Seeger, Gaussian processes for machine learning, Int. J. Neural Syst. 14 (2004) 69–106.
[74] K. Murphy, Machine Learning: A Probabilistic Perspective, The MIT Press, 2012.
[75] F.M. l Schleif, M. Biehl, A. Vellido, Advances in machine learning and computational intelligence, Neurocomputing 72 (2009) 1377–1378.
[76] S.I. Amari, Natural gradient works efficiently in learning, Neural Comput. 10 (1998) 251–276.
[77] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[78] V. Vapnik, The Nature of Statistical Learning Theory, Springer Science & Business Media, 2013.
[79] B. Schölkopfand, A. Smola, Learning with Kernels, MIT Press, 2002.
[80] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, Cambridge, 2004.
[81] J.H. Chiang, Choquet fuzzy integral-based hierarchical networks for decision analysis, Fuzzy Syst. IEEE Trans. 7 (1) (1999) 63–71.
[82] S. Haykin, Neural Networks. A Comprensive foundation 1st and 2nd eds, Prentice Hall, New Jersey 1994, p. 1999.
[83] F.L. Luo, R. Unbehauen, Applied neural networks for signal processing, Cambridge University Press, 1998.
[84] C. Feng, R. Plamondon, On the stability analysis of delayed neural networks systems, Neural Netw. 14 (9) (2001) 1181–1188.
[85] C.H. Feng, R. Plamondon, Stability analysis of bidirectional associative memory networks with time delays, IEEE Trans. Neural Netw. 14 (6) (2003) 1560–1565.
[86] K. Gopalsamy, Stability of artificial neural networks with impulses, Appl. Math. Comput. 154 (3) (2004) 783–813.
[87] L. Wu, Z. Feng, W.X. Zheng, Exponential stability analysis for delayed neural networks with switching parameters: average dwell time approach, Neural Netw. IEEE Trans. 21 (9) (2010) 1396–1407.
[88] X. Zhang, L. Wu, S. Cui, An improved integral inequality to stability analysis of genetic regulatory networks with interval time-varying delays, IEEE/ACM Trans. Comput. Biol. Bioinforma. (TCBB) 12 (2) (2015) 398–409.
[89] M. Cottrell, J.C. Fort, G. Pages, Theoretical aspects of the SOM algorithm, Neurocomputing 21 (1-3) (1998) 119–138.
[90] S. Bermejo, J. Cabestany, The effect of finite sample size on on-line K-means, Neurocomputing 48 (1–4) (2002) 511–539.
[91] M.C. Fu, Optimization for simulation: Theory vs. practice, Informs J. Comput. 14 (3) (2002) 192–215.
[92] M. Gevrey, L. Dimopoulos, S. Lek, Review and comparison of methods to study the contribution of variables in artificial neural network models, Ecol. Model. 160 (3) (2003) 249–264.
[93] J. Ilonen, J.K. Kamarainen, J. Lampinen, Differential evolution training algorithm for feed-forward neural networks, Neural Process. Lett. 17 (1) (2003) 93–105.
[94] A. Abraham, Meta learning evolutionary artificial neural networks, Neurocomputing 56 (2004) 1–38.
[95] P.J. Zufiria, On the discrete-time dynamics of the basic Hebbian neural-network node, IEEE Trans. Neural Netw. 13 (6) (2002) 1342–1352.
[96] M. Forti, P. Nistri, Global convergence of neural networks with discontinuous neuron activations. Circuits and systems I: fundamental theory and applications, IEEE Trans. 50 (11) (2003) 1421–1435.
[97] M. Forti, P. Nistri, D. Papini, Global exponential stability and global convergence in finite time of delayed neural networks with infinite gain, Neural Netw. IEEE Trans. 16 (6) (2005) 1449–1463.
[98] W. Lu, T. Chen, Dynamical behaviors of Cohen-Grossberg neural networks with discontinuous activation functions, Neural Netw. 18 (3) (2005) 231–242.
[99] L. Duan, L. Huang, Z. Guo, Stability and almost periodicity for delayed high-order Hopfield neural networks with discontinuous activations, Nonlinear Dyn. 77 (4) (2014) 1469–1484.
[100] T. Kim, T. Adali, Fully complex multi-layer perceptron network for nonlinear signal processing, J. VLSI signal Process. Syst. Signal Image Video Technol. 32 (1-2) (2002) 29–43.
[101] T. Nitta, On the inherent property of the decision boundary in complex-valued neural networks, Neurocomputing 50 (2003) 291–303.
[102] I. Aizenberg, C. Moraga, Multilayer feedforward neural network based on multi-valued neurons (MLMVN) and a backpropagation learning algorithm, Soft Comput. 11 (2) (2007) 169–183.
[103] R. Savitha, S. Suresh, N. Sundararajan, A fully complex-valued radial basis function network and its learning algorithm, Int. J. Neural Syst. 19 (04) (2009) 253–267.
[104] M.F. Amin, K. Murase, Single-layered complex-valued neural network for real-valued classification problems, Neurocomputing 72 (4) (2009) 945–955.
[105] T. Xiong, Y. Bao, Z. Hu, R. Chiong, R, Forecasting interval time series using a fully complex-valued RBF neural network with DPSO and PSO algorithms, Inf. Sci. 305 (2015) 77–92.
[106] Hirose, Application fields and fundamental merits of complex-valued neural networks. Complex-Valued Neural Networks: Advances and Applications, IEEE Press, John Wiley, 2013.
[107] H. Leung, S. Haykin, The complex backpropagation algorithm, Signal Process. IEEE Trans. 39 (9) (1991) 2101–2104.
[108] G.-B. Huang, L. Chen, C.-K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Netw. 17 (4) (2006) 879–892.
[109] G.B. Huang, L. Chen, Neurocomputing 70 (16–18) (2007) 3056–3062.
[110] Y. Bengio, Learning Deep Architectures for AI, Found. Trends Mach. Learn. 2 (1) (2009) 1–27.
[111] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, B. Kingsbury, Deep neural networks for acoustic modelling in speech recognition, IEEE Signal Process. Mag. 29 (6) (2012) 82–97.
[112] L. Deng, G. Hinton, B. Kingsbury, New types of deep neural network learning for speech recognition and related applications: An overview, Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on (ICASSP), 2013, pp. 8599–8603.
[113] H. Lee, R. Grosse, R. Ranganath, A.Y. Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, in: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009, 609–616.
[114] D. Ciresan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 3642–3649.
[115] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, J. Mach. Learn. Res. 12 (2011) 2493–2537.
[116] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, Pattern Anal. Mach. Intell. IEEE Trans. 35 (8) (2013) 1798–1828.
[117] G. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.
[118] P. Zhou, C. Liu, Q. Liu, L. Dai, H. Jiang, A cluster-based multiple deep neural networks method for large vocabulary continuous speech recognition, in: Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on, IEEE, 2013, pp. 6650–6654.
[119] B. Chandra, R.K. Sharma, Fast learning in deep neural networks, Neurocomputing 171 (2016) 1205–1215.
[120] B.A. Olshausen, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Nature 381 (6583) (1996) 607–609.
[121] W. Maass, Networks of spiking neurons: The third generation of neural network models, Neural Netw. 10 (9) (1997) 1659–1671.
[122] S. Ghosh-Dastidar, Spiking Neural Networks, Int. J. Neural Syst. 19 (2009) 295–308.
[123] W. Maass, T. Natschläger, H. Markram, Real-time computing without stable states: a new framework for neural computation based on perturbations, Neural Comput. 14 (11) (2002) 2531–2560.

[124] W. Maass, H. Markram, On the computational power of recurrent circuits of spiking neurons, J. Comput. Syst. Sci. 69 (4) (2004) 593–616.
[125] W. Maass, Liquid computing. In Computation and Logic in the Real World, Springer, Berlin Heidelberg 2007, pp. 507–516.
[126] E.M. Izhikevich, Simple model of spiking neurons, IEEE Trans. Neural Netw. 14 (6) (2003) 1569–1572.
[127] R. Brette, W. Gerstner, Adaptive exponential integrate-and-fire model as an effective description of neuronal activity, J. Neurophysiol. 94 (5) (2005) 3637–3642.
[128] R. Naud, N. Marcille, C. Clopath, W. Gerstner, Firing patterns in the adaptive exponential integrate-and-fire model, Biol. Cybern. 99 (4–5) (2008) 335–347.
[129] A.N. Kolmogorov, On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition, Am. Math. Soc. Transl. 28 (1963) 55–59.
[130] J.D.Schaffer, D. Whitley, L.J. Eshelman, Combinations of genetic algorithms and neural networks: A survey of the state of the art, Combinations of Genetic Algorithms and Neural Networks, COGANN-92. International Workshop on, IEEE, 1992.
[131] D. Whitley, Genetic algorithms and neural networks, Genet. algorithms Eng. Comput. Sci. 3 (1995) 203–216.
[132] D. Heinke, F.H. Hamker, Comparing neural networks: a benchmark on growing neural gas, growing cell structures, and fuzzy ARTMAP, Neural Netw., IEEE Trans. 9 (6) (1998) 1279–1291.
[133] M. Lehtokangas, Modelling with constructive backpropagation, Neural Netw. 12 (4) (1999) 707–716.
[134] R. Zhang, Y. Lan, G.B. Huang, Z.,B. Xu, Z. B, Universal approximation of extreme learning machine with adaptive growth of hidden nodes, Neural Netw. Learn. Syst. IEEE Trans. 23 (2) (2012) 365–371.
[135] R. Reed, R, Pruning algorithms-a survey, Neural Netw. IEEE Trans. 4 (5) (1993) 740–747.
[136] B.E. Segee, M.J. Carter, IJCNN-91-Seattle International Joint Conference on Fault tolerance of pruned multilayer networks. In Neural Networks, IEEE, vol. 2 ,1991, pp. 447–452.
[137] Y. Le Cun, J.S. Denker, S.A. Solla, Optimal brain damage, NIPs 89 (1989).
[138] M. Yoan, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, A. Lendasse, OP-ELM: optimally pruned extreme learning machine, Neural Netw. IEEE Trans. 21 (1) (2010) 158–162.
[139] P.L. Narasimha, W.H. Delashmit, M.T. Manry, J. Li, F. Maldonado, An integrated growing-pruning method for feedforward network training, Neurocomputing 71 (13–15) (2008) 2831–2847.
[140] M.M. Islam, M.A. Sattar, M.F. Amin, X. Yao, K. Murase, A new adaptive merging and growing algorithm for designing artificial neural networks, systems, man, and cybernetics, Part B: cybernetics, IEEE Trans. 39 (3) (2009) 705–722.
[141] M. Bortman, M. Aladjem, A growing and pruning method for radial basis function networks, Neural Netw., IEEE Trans. 20 (6) (2009) 1039–1045.
[142] S. Haykin, Neural Networks and Learning Machines, 3rd ed, Pearson, 2009.
[143] W. Gerstner, R. Brette, Adaptive exponential integrate-and-fire model, Scholarpedia 4 (6) (2009) 8427.
[144] E. Claverol, A. Brown, J. Chad, Discrete simulation of large aggregates of neurons, Neurocomputing 47 (2002) 277–297.
[145] M. Mattia, P. del Giudice, Efficient event-driven simulation of large networks of spiking neurons and dynamical synapses. Neural Computation, vol. 12 (200), pp. 2305–2329.
[146] J. Reutimann, M. Giugliano, S. Fusi, Event-driven simulation of spiking neurons with stochastic dynamics, Neural Comput. 15 (4) (2003) 811–830.
[147] E. Ros, R. Carrillo, E.M. Ortigosa, B. Barbour, R. Agís, Event-Driven Simulation Scheme For Spiking Neural Networks Using Lookup Tables To Characterize Neuronal Dynamics, Neural Comput. 18 (2006) 2959–2993.
[148] F. Naveros, N.R. Luque, J.A. Garrido, R.R. Carrillo, M. Anguita, E. Ros, A spiking neural simulator integrating event-driven and time-driven computation schemes using parallel CPU-GPU co-processing, IEEE Trans. Neural Netw. 26 (7) (2014) 1567–1574.
[149] M. Rudolph, A. Destexhe, A. How much can we trust neural simulation strategies? Neurocomputing 70 (10–12) (2007) 1966–1969.
[150] R. Brette, M. Rudolph, T. Carnevale, M. Hines, D. Beeman, J.M. Bower, M. Diesmann, A. Morrison, P.H. Goodman, F.C. Harris, J.,M. Zirpe, T. Natschläger, D. Pecevski, B. Ermentrout, M. Djurfeldt, A. Lansner, O. Rochel, T. Vieville, E. Muller, A.P. Davison, S.E. Boustani, A. Destexhe, Simulation of networks of spiking neurons: a review of tools and strategies, J. Comput. Neurosci. 23 (2007) 349–398.
[151] P. Hammarlund, Ö. Ekeberg, Large neural network simulations on multiple hardware platforms, J. Comput. Neurosci. 5 (4) (1998) 443–459.
[152] M. Hereld, R.L. Stevens, J. Teller, W. van Drongelen, Large neural simulations on large parallel computers, Int. J. Bioelectromagn. 7 (1) (2005) 44–46.
[153] U. Seiffert, Artificial neural networks on massively parallel computer hardware, Neurocomputing 57 (2004) 135–150.
[154] H. de Garis, C. Shou, B. Goertzel, L. Ruiting, A world survey of artificial brain projects, Part I: Large-scale brain simulations, Neurocomputing 74 (1-3) (2010) 3–29.
[155] A.P. Davison, D. Brüderle, J. Eppler, J. Kremkow, E. Muller, D.A. Pecevski, L. Perrinet, P. Yge, PyNN: a common interface for neuronal network simulators, Front. Neuroinform. 2 (2008) 11.
[156] D.F. Goodman, R. Brette, The Brian simulator, Front. Neurosci. 3 (2) (2009) 192–197.
[157] M. Stimberg, D.F.M. Goodman, V. Benichoux, R. Brette, Equation-oriented specification of neural models for simulations, Front. Neuroinform. 8 (6) (2014) 1–14.
[158] R. Blaško, Soft Computing Applications, Developed by ECANSE, in: P. Sinák, et al., (Eds.), The State of the Art in Computational Intelligence Advances, 5, Springer-Verlag, Berlin, 2000, pp. 233–237.
[159] R.C. O'Reilly, Y. Munakata, Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain, MIT Press, 2000.
[160] J.M. Bower, D. Beeman, The Book of GENESIS: Exploring Realistic Neural Models with the General Neural Simulation System, second edition, Springer, New York, 1998.
[161] J.D. Johnsen, P. Nielsen, H.J. Luhmann, G. Northoff, R. Kötter, Multi-level network modelling of cortical dynamics built on the GENESIS environment, Neurocomputing 44-46 (2002) 863–868.
[162] O. Rochel, D. Martinez, An event-driven framework for the simulation of networks of spiking neurons, Proceedings of the 11th European Symposium on Artificial Neural Networks (ESANN) 2003, 295–300.
[163] C.E. Wilson, P.H. Goodman, F.C. Harris, Implementation of a biologically realistic parallel neocortical-neural network simulator, in: Proceedings of the Tenth SIAM on Conference on Parallel Process. Sci. Comp. (PPSC), 2001.
[164] J.B. Maciokas, P.H. Goodman, J. Kenyon, M. Toledo-Rodriguez, H. Markram, Accurate dynamical models of interneuronal GABaergic channel physiologies, Neurocomputing 65 (2005) 5–14.
[165] C. Eliasmith, How to Build a Brain: A Neural Architecture for Biological Cognition, Oxford University Press, 2013.
[166] C. Eliasmith, T.C. Stewart, X. Choo, T. Bekolay, T. DeWolf, Y. Tang, D. Rasmussen, A large-scale model of the functioning brain, Science 338 (6111) (2012) 1202–1205.
[167] T.C. Stewart, B. Tripp, C. Eliasmith, Python scripting in the Nengo simulator, Front. Neuroinform. 3 (7) (2009).
[168] M. Diesmann, M.O. Gewaltig, NEST: An environment for neural systems simulations, in Forschung und wisschenschaftliches Rechnen, Beitr. zum Heinz-Billing-Preis 58 (2001) 43–70.
[169] M.L. Hines, N.T. Carnevale, N. T. NEURON: a tool for neuroscientists, Neuroscientist 7 (2) (2001) 123–135.
[170] N.T. Carnevale, M.L. Hines, The NEURON Book, Cambridge University Press, Cambridge, UK, 2006.
[171] M.L. Hines, N.T. Carnevale, Discrete event simulation in the NEURON environment, Neurocomputing 58–60 (2004) 1117–1122.
[172] M. Migliore, C. Cannia, W.W. Lytton, H. Markram, M.L. Hines, Parallel network simulations with NEURON, J. Comput. Neurosci. 21 (2) (2006) 119–129.
[173] N. Zell, R. Mache, G. Hübner, M. Mamier, M. Vogt, K.U. Schmalzl, K. Herrmann, SNNS (Stuttgart Neural Network Simulator): In Neural Network Simulation Environments, Springer, US 1994, pp. 165–186.
[174] A. Delorme, J. Gautrais, R. van Rullen, S. Thorpe, SpikeNET: a simulator for modelling large networks of integrate and fire neurons, Neurocomputing 26–27 (1999) 989–996.
[175] S.J. Thorpe, R. Guyonneau, N. Guilbaud, J.M. Allegraud, R. VanRullen, SpikeNet: real-time visual processing with one spike per neuron, Neurocomputing 58 (2004) 857–864.
[176] J.F. Vibert, N. Azmy, Neuro-bio-clusters: a tool for interacting biological neural networks simulation, in: T. Kohonen, K. Makisara, O. Simula, J. Kangas (Eds.), Artificial Neural Networks, Elsevier S. P. North-Holland Pub, 1991, pp. 551–556.
[177] J.F. Vibert, F. Alvarez, E.K. Kosmidis, XNBC V9: A user friendly simulation and analysis tool for neurobiologists, Neurocomputing 38–40 (2001) 1715–1723.
[178] B. Ermentrout. Simulating, analyzing, and animating dynamical systems: A guide to XPPAUT for researchers and students, SIAM, vol. 14, 2002.
[179] K.H. Pettersen, H. Lindén, A.M. Dale, G.T. Einevoll, Extracellular spikes and CSD, in: R. Brette, A. Destexhe (Eds.), Handbook of Neural Activity Measurement, Cambridge University Press, 2012, pp. 92–135.
[180] U. Bernardet, M. Blanchard, P. FMJ Verschure, IQR: a distributed system for real-time real-world neuronal simulation, Neurocomputing 44–46 (2002) 1043–1048.
[181] H. Cornelis, E. de Schutter, NeuroSpaces: separating modelling and simulation, Neurocomputing 52 (54) (2003) 227–231.
[182] F.K. Skinner, J.B. Liu, NNET: linking small- and large-scale network models, Neurocomputing 52 (54) (2003) 381–387.
[183] M. Sousa, P. Aguiar, Building, simulating and visualizing large spiking neural networks with NeuralSyns, Neurocomputing 123 (2014) 372–380.
[184] M. Mulas, P. Massobrio, NEUVISION: a novel simulation environment to model spontaneous and stimulus-evoked activity of large-scale neuronal networks, Neurocomputing 122 (2013) 441–457.
[185] E. Schikuta, NeuroWeb: An Internet-based neural network simulator, in: Proc. of the 14th IEEE International Conference on Tools with Artificial Intelligence, Washington, IEEE Computer Society, 2002, pp. 407–412.
[186] C. Bergmeir, J.M. Benitez, Neural networks in R using the stuttgart neural network simulator: RSNNS, J. Stat. Softw. 46 (7) (2012) 1–26.
[187] M. Djurfeldt, A. Sandberg, O. Ekeberg, A. Lansner, SEE—a framework for simulation of biologically detailed and artificial neural networks and systems, Neurocomputing 26–27 (1999) 997–1003.
[188] K.M.L. Menne, A. Folkers, T. Malina, U.G. Hofmann, Test of spike-sorting algorithms on the basis of simulated network data, Neurocomputing 44 (2002) 1119–1126.
[189] D. Hansel, G. Mato, C. Meunier, L. Neltner, On numerical simulations of integrate-and-fire neural networks, Neural Comput. 10 (2) (1998) 467–483.
[190] M. Resta, An agent-based simulator driven by variants of self-organizing maps, Neurocomputing 147 (2015) 207–224.

[191] K.G. Spiliotis, C.I. Siettos, A timestepper-based approach for the coarse-grained analysis of microscopic neuronal simulators on networks: Bifurcation and rare-events micro-to macro-computations, Neurocomputing 74 (17) (2011) 3576–3589.
[192] I. Ziv, Da Baxter, Jh Byrne, Simulator for neural networks and action potentials: description and application, J. Neurophysiol. 71 (1) (1994) 294–308.
[193] M. Sanchez-Montanez, Strategies for the optimization of large scale networks of integrate and fire neurons, in: J. Mira, A. Prieto (Eds.), Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence, 2084, Lecture Notes in Computer Science, 2001, pp. 117–125.
[194] J.M. Nageswaran, N. Dutt, J.L. Krichmar, A. Nicolau, A.V. Veidenbaum, A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphics processors, Neural Netw. 22 (5–6) (2009) 791–800.
[195] H.E. Plesser, J. Eppler, A. Morrison, Abigail; et al. Efficient parallel simulation of large-scale neuronal networks on clusters of multiprocessor computers, Lect. Notes Comput. Sci. 4641 (2007) 672–681.
[196] P. Pacheco, M. Camperi, T. Uchino, PARALLEL NEUROSYS: a system for the simulation of very large networks of biologically accurate neurons on parallel computers, Neurocomputing 32 (2000) 1095–1102.
[197] A. d'Acierno, Back-propagation learning algorithm and parallel computers: the CLEPSYDRA mapping scheme, Neurocomputing 31 (2000) 67–85.
[198] V. Kumar, S. Shekhar, M.B. Amin, A scalable parallel formulation of the back-propagation algorithm for hypercubes and related architectures, IEEE Trans. Parallel Distrib. Syst. 5 (10) (1994) 1073–1090.
[199] L.M. Patnaik, R.N. Rao, Parallel implementation of neocognitron on star topology: theoretical and experimental evaluation, Neurocomputing 41 (2001) 109–124.
[200] J. Ortega, I. Rojas, A.F. Díaz, A. Prieto, Parallel coarse grain computing of boltzmann machines June, Neural Process. Lett. 7 (No. 3) (1998) 169–184.
[201] C. Chen, T.M. Taha, Spiking neural networks on high performance computer clusters Sep., Proc. SPIE, Opt. Photon-. Inf. Process. 8134 (2011) 813406.
[202] H. Markram, The Blue Brain project, Nat. Rev. Neurosci. 7 (2) (2006) 153–160.
[203] R. Fontaine, F. Belanger, N. Viscogliosi, H. Semmaoui, M.A. Tetrault, J. B. Michaud, C. Pepin, J. Cadorette, R. Lecomte, The hardware and signal processing architecture of LabPET (TM), a small animal APD-based digital PET scanner, IEEE Trans. Nucl. Sci. 56 (1) (2009) 3–9.
[204] The Blue Brain Project. 2011; Available from: 〈http://bluebrain.epfl.ch/〉. The Blue Brain Project. EPFL.
[205] H. Markrama, K. Meierb, T. Lippertc, S. Grillnerd, R. Frackowiake, S. Dehaenef, A. Knollg, H. Sompolinskyh, K. Verstrekeni, J. DeFelipe j, S. Grantk, J.-P. Changeuxl, A. Sariam, Introducing the Human Brain Project, Procedia Comput. Sci. 7 (2011) 39–42.
[206] J. Soto, J.M. Moreno, J. Cabestany, A self-adaptive hardware architecture with fault tolerance capabilities, Neurocomputing 121 (2013) 25–31.
[207] J. Misra, I. Saha, Artificial neural networks in hardware: a survey, Neurocomputing 74 (2010) 239–255.
[208] L. Reyneri, On the performance of pulsed and spiking neurons, Analog. Integr. Circ. Signal Process. 30 (2) (2002) 101–119.
[209] K. Goser, U. Ramacher, Mikroelektronische Realisierung von künstlichen neuronalen Netzen/Microelectronic Realizations of artificial neural networks, Informationstechnik 34 (4) (1992) 241–247.
[210] M. Glesner, W. Poechmueller, Neurocomputers: An Overview of Neural Networks in VLSI, Chapman and Hall, London, 1994.
[211] A. Prieto, A. Andreou, Microelectronics for bio-inspired systems, Analog. Integr. Circ. Signal Process. 30 (2002) 87–90.
[212] J. Lachmair, E. Merényi, M. Porrmann, U. Rückert, A reconfigurable neuroprocessor for self-organizing feature maps, Neurocomputing 112 (2013) 189–199.
[213] M.L. Rossmann, A. Bühlmeier, G. Manteuffel, K. Goser, Dynamic Hebbian learning strategies for VLSI-systems, Neurocomputing 28 (1-3) (1999) 157–164.
[214] G. Indiveri, B. Linares-Barranco, T.J. Hamilton, A. van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud, J. Schemmel, G. Cauwenberghs, J. Arthur, K. Hynna, F. Folowosele, S. Saighi, T. Serrano-Gotarredona, J. Wijekoon, Y. Wang, K. Boahen, Neuromorphic silicon neuron circuits, Front. Neurosci. 5 (2011) 73.
[215] K. Zaghloul, K. Boahen, A silicon retina that reproduces signals in the optic nerve, J. Neural Eng. 3 (4) (2006) 257–267.
[216] M. Mahowald, The silicon retina, An Analog VLSI System for Stereoscopic Vision, Springer 1994, pp. 4–65.
[217] M. Anguita, F.J. Pelayo, A. Prieto, J. Ortega, Analog CMOS implementation of a cellular neural networks with programmable cloning templates, IEEE Trans. Circuits Syst. 40 (3) (1993).
[218] T. Delbrück, B. Linares-Barranco, E. Culurciello, C. Posch, Activity-driven, event-based vision sensors. In Circuits and Systems (ISCAS), in: Proceedings of 2010 IEEE International Symposium on, 2010, pp. 2426–2429.
[219] K. Boahen, Neurogrid: emulating a million neurons in the cortex, in: Grand Challenges in Neural Computation, 2006, 6702.
[220] C. Johansson, A. Lansner, Towards cortex sized artificial neural systems, Neural Netw. 20 (1) (2007) 48–61.
[221] J. Lazzaro, J. Wawrzynek, M. Mahowald, M. Sivilotti, D. Gillespie, Silicon auditory processors as computer peripherals, Neural Netw. IEEE Trans. 4 (3) (1993) 523–528.
[222] F. Corradi, D. Zambrano, M. Raglianti, G. Passetti, C. Laschi, G. Indiveri, Towards a neuromorphic vestibular system, IEEE Trans. Biomed. Circ. Syst. 8 (5) (2014) 669–680.
[223] S.M. Fakhraie, H. Farshbaf, K.C. Smith, Scalable closed-boundary analog neural networks, IEEE Trans. Neural Netw. 15 (2004) 492–504.
[224] M. Verleysen, P. Thissen, J.L. Voz, J. Madrenas, An analog processor architecture for a neural network classifier, IEEE Micro 14 (1994) 16–28.
[225] U. Lotric, P. Bulic, Applicability of approximate multipliers in hardware neural networks, Neurocomputing 96 (2012) 57–65.
[226] J.L. Bernier, J. Ortega, I. Rojas, A. Prieto, Improving the tolerance of multilayer perceptrons by minimizing thestatistical sensitivity to weight deviations, Neurocomputing 31 (1-4) (2000) 87–103.
[227] J.L. Bernier, J. Ortega, E. Ros, I. Rojas, A. Prieto, A quantitative study of fault tolerance, noise immunity and generalization ability of MLPs, Neural Comput. 12 (2000) 2941–2964.
[228] C. Johansson, A. Lansner, Implementing plastic weights in neural networks using low precision arithmetic, Neurocomputing 72 (4–6) (2009) 968–972.
[229] A.K. Fidjeland, M.P. Shanahan, Accelerated simulation of spiking neural networks using GPUs in Proc., IJCNN, Barcelona, Spain, 2010.
[230] J.M. Nageswaran, N. Dutt, J.L. Krichmar, A. Nicolau, A. Veidenbaum, Efficient simulation of large-scale spiking neural networks using CUDA graphics processors, in: Proc. IJCNN, Atlanta, GA, USA, June 2009.
[231] R. Brette, D.F. Goodman, Simulating spiking neural networks on GPU, Network 23 (4) (2012) 167–182.
[232] A. Ahmadi, H. Soleimani, A GPU based simulation of multilayer spiking neural networks in Proc. 19th, ICEE, Tehran, Iran, 2011.
[233] Y. Lu, D.W. Li, Z.H. Xu, Y.G. Xi, Convergence analysis and digital implementation of a discrete-time neural network for model predictive control, IEEE Trans. Ind. Electron. 61 (12) (2014) 7035–7045.
[234] J. Moreno, M.E. Ortuzar, J.W. Dixon, Energy-management system for a hybrid electric vehicle, using ultracapacitors and neural networks, IEEE Trans. Ind. Electron. 53 (2) (2006) 614–623.
[235] E. Ros, E.M. Ortigosa, R. Agís, R. Carrillo, M. Arnold, Real-time computing platform for spiking neurons (RT-spike), IEEE Trans. Neural Netw. 17 (4) (2006) 1050–1063.
[236] A. Strey, N. Avellana, A new concept for parallel neurocomputer architectures, in: Proceedings of EuroPar'96, Lyon, France, 1996, pp. 470–477.
[237] 〈http://www.artificialbrains.com/darpa-synapse-program〉.
[238] P. Merolla, J. Arthur, F. Akopyan, N. Imam, R. Manohar, D. Modha, A digital neurosynaptic core using embedded crossbar memory with 45 pJ per spike in 45 nm, in Proc. Custom Integr. Circuits Conf., 2011.
[239] R. Preissl, T.M. Wong, P. Datta, M. Flickner, R. Singh, S.K. Esser, W.P. Risk, H.D. Simon, D.S. Modha, Compass: A scalable simulator for an architecture for Cognitive Computing; in: SC '12 Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, Article No. 54, IEEE Computer Society Press Los Alamitos, CA, USA, 2012.
[240] 〈http://www.research.ibm.com/articles/brain-chip.shtml〉.
[241] S.B. Furber, D.R. Lester, L.A. Plana, J.D. Garside, E. Painkras, S. Temple, A. D. Brown, Overview of the SpiNNaker system architecture December, IEEE Trans. Comput. 62 (12) (2013) 2454–2467.
[242] S.B. Furber, F. Gallupi, S. Temple, L.A. Plana, The SpiNNaker project: a massively-parallel computer architecture for neural simulations May, Proc. IEEE 102 (no. 5) (2014).
[243] 〈http://apt.cs.manchester.ac.uk/projects/SpiNNaker/architecture/〉.
[244] A. Rast, F. Galluppi, S. Davies, L. Plana, C. Patterson, T. Sharp, D. Lester, S. Furber, Concurrent heterogeneous neural model simulation on real-time neuromimetic hardware, Neural Netw. 24 (9) (2011) 961–978.
[245] J. Schemmel D. Brüderle A. Grübl M. Hock K. Meier S. Millner A wafer-scale neuromorphic hardware system for large-scale neural modelling, in: Proc. IEEE Int. Symp. Circuits Syst., 2010, pp. 1947–1950.
[246] 〈http://www.uni-heidelberg.de/presse/news2013/pm20130128_hbp_en.html〉.
[247] B.V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A.R. Chandrasekaran, J.-M. Bussat, R. Alvarez-Icaza, J.V. Arthur, P.A. Merolla, K. Boahen, Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations, Proc. IEEE 102 (5) (2014) 699–716.
[248] R. Silver, K. Boahen, S. Grillner, N. Kopell, K.L. Olsen, Neurotech for neuroscience: unifying concepts, organizing principles, emerging tools, J. Neurosci. 27 (44) (2007) 11807–11819.
[249] F. Yang, M. Paindavoine, Implementation of an RBF neural network on embedded systems: real-time face tracking and identity verification, IEEE Trans. Neural Netw. 14 (5) (2003) 1162–1175.
[250] L. Reyneri, Implementation issues of neuro-fuzzy hardware: going toward HW/SW codesign, IEEE Trans. Neural Netw. 14 (1) (2003) 176–194.
[251] B. Guo, D.H. Wang, Y. Shen, Z. Liu, Hardware-software partitioning of real-time operating systems using Hopfield neural networks, Neurocomputing 69 (16–18) (2006) 2379–2384.
[252] J. Zhu, P. Sutton, FPGA implementations of neural networks—a survey of a decade of progress, Field-Program. Log. Appl. 2778 (2003) 1062–1066.
[253] L.P. Maguire, T.M. McGinnity, B. Glackin, A. Ghani, A. Belatreche, J. Harkin, Challenges for large-scale implementations of spiking neural networks on FPGAs, Neurocomputing 71 (1-3) (2007) 13–29.
[254] M. Atencia, H. Boumeridja, G. Joya, F. Garcia-Lagos, F. Sandoval, FPGA implementation of a systems identification module based upon Hopfield networks, Neurocomputing 70 (16–18) (2007) 2828–2835.
[255] W. Gerstner, Spiking Neuron Models: Single Neurons, Populations, Plasticity, Cambridge University Press, Cambridge, UK, 2002.
[256] F.J. Pelayo, E. Ros, X. Arreguit, A. Prieto, VLSI neural model using spikes, Analog. Integr. Circ. Signal Process. 13 (1–2) (1997) 111–121.

[257] E. Chicca, F. Stefanini, C. Bartolozzi, G. Indiveri, Neuromorphic electronic circuits for building autonomous cognitive systems, Proc. IEEE 102 (9) (2014) 1367–1388.
[258] M.R. Azghadi, N. Iannella, S.F. Al-Sarawi, G. Indiveri, Spike-based synaptic plasticity in silicon: design, implementation, application, and challenges, Proc. IEEE 102 (5) (2014) 717–737.
[259] M. Schaefer, T. Schoenauer, C. Wolff, C, et al., Simulation of spiking neural networks - architectures and implementations, Neurocomputing 48 (2002) 647–679.
[260] B. Linares-Barranco, E. Sánchez-Sinencio, A. Rodríguez-Vázquez, J.L. Huertas, CMOS implementation of FitzHugh-Nagumo neuron model, IEEE J. Solid-State Circ. 26 (7) (1991) 956–965.
[261] L.O. Chua, Memristor—the missing circuit element, IEEE Trans. Circuit Theory CT-18 (5) (1971) 507–519.
[262] D.B. Strukov, G.S. Snider, D.R. Stewart, S.R. Williams, The missing memristor found, Nature 453 (7191) (2008) 80–83.
[263] A. Thomas, Memristor-based neural networks, J. Phys. D: Appl. Phys. 46 (9) (2013) 093001.
[264] S. Li, F. Zeng, C. Chen, H. Liu, G. Tang, S. Gao, C. Song, Y. Lin, F. Pan, D. Guob, Synaptic plasticity and learning behaviours mimicked through Ag interface movement in an Ag/conducting polymer/Ta memristive system, J. Mater. Chem. C 1.34 (2013) 5292–5298.
[265] G. Indiveri, B. Linares-Barranco, R. Legenstein, G. Deligeorgis, T. Prodromakis, Integration of nanoscale memristor synapses in neuromorphic computing architectures, Nanotechnology 24 (38) (2013) 384010.
[266] L. Chua, V. Sbitnev, H. Kim, Hodgkin-Huxley axon is made of memristors, Int. J. Bifurc. Chaos 22 (3) (2012).
[267] Y.V. Pershin, M. Di Ventra, Experimental demonstration of associative memory with memristive neural networks, Neural Netw. 23 (7) (2010) 881–886.
[268] M. Itoh, L.O. Chua, Memristor cellular automata and memristor discrete-time cellular neural networks, Int. J. Bifurc. Chaos 19 (11) (2009) 3605–3656.
[269] L. Duan, L. Huang, Periodicity and dissipativity for memristor-based mixed time-varying delayed neural networks via differential inclusions, Neural Netw. 57 (2014) 12–22.
[270] Artificial Brains. DARPA SyNAPSE Program. 〈http://www.artificialbrains.com/darpa-synapse-program#memristor-chip〉.
[271] Y.S. AbuMostafa, D. Psaltis, Optical neural computers, Sci. Am. 255 (1987) 88–95.
[272] E. Lange, Y. Nitta, K. Kyuma, Optical neural chips, IEEE Micro 14 (6) (1994) 29–41.
[273] A.K. Datta, S.K. Sen, S. Bandyopadhyay, et al., Optical computing techniques, IETE Tech. Rev. 12 (2) (1995) 93–105.
[274] F.T.S. Yu, C.M. Uang, Optical neural networks, in: R.G. Driggers (Ed.), Encyclopedia of Optical Engineering, 1:1, CRC Press, New York, NY, USA, 2003, pp. 1763–1777.
[275] P.E.X. Silveira, Optoelectronic neural networks, in: R.G. Driggers (Ed.), Encyclopedia of Optical Engineering, 1:1, CRC Press, New York, NY, USA, 2003, pp. 1887–1902.
[276] A. Serrano-Heredia, C.M. Hinojosa, R. Ponce, et al., Opto-digital implementation of a neural network using a Joint Transform Correlator based in a Hopfield inner product model for character recognition, Conference on Optical Information Systems, Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE), San Diego, CA, Aug 04–05, Vol. 5202, 2003, pp. 365–372.
[277] V.P. Shmerko, S.N. Yanushkevich, Computing paradigms for predictable nanoelectronics, J. Comput. Theor. Nanosci. 7 (2) (2010) 303–324.
[278] M. Schuld, I. Sinayskiy, F. Petruccione, The quest for a quantum neural network, Quantum Inf. Process. 13 (11) (2014) 2567–2586.
[279] W. Up, A. Renn, Molecular computing - A review. 1. Data and image storage, J. Mol. Electron. 7 (1) (1991) 1–20.
[280] M. Conrad, The lure of molecular computing, IEEE Spectr. 23 (1988) 55–60.
[281] J.C. Chen, R.D. Chen, Toward an evolvable neuromolecular hardware: a hardware design fbr a multilevel artificial brain with digital circuits, Neurocomputing 45 (2002) 9–34.
[282] F. Alibart, S. Pleutin, D. Gue´rin, C. Novembre, S. Lenfant, K. Lmimouni, C. Gamrat, D. Vuillaume, An organic nanoparticle transistor behaving as a biological spiking synapse, Adv. Funct. Mater. 20 (2) (2009) 330–337.
[283] K.L. Wang, Issues of nanoelectronics: A possible roadmap, Journal. Nanosci. Nanotechnol. 2 (3–4) (2002) 235–266.
[284] L. Nunes de Castro, Fundamentals of natural computing: an overview, Phys. Life Rev. 4 (1) (2007) 1–36.
[285] D. Hammerstrom, A survey of bio-inspired and other alternative architectures, Nanotechnology (2010).
[286] A.F. Murray (Ed.), Applications of Neural Networks, Springer, 1995, ISBN: 978-1-4419-5140-3.
[287] M. Tkáč, R. Verner, Artificial neural networks in business: two decades of research, Appl. Soft Comput. 38 (2016) 788–804.
[288] Y. Bentz, D. Merunka, Neural networks and the multinomial logit for brand choice modelling: a hybrid approach, J. Forecast. 19 (3) (2000) 177–200.
[289] P. Berkhin, A survey of dataminig techniques, in: J. Kogan, C. Nicholas, M. Tebouille (Eds) Grouping Multidimensional Data: Recent Advances in Clustering, Springer, pp. 25–71.
[290] L.J. Lancashire, C. Lemetre, G.R. Ball, An introduction to artificial neural networks in bioinformatics—application to complex microarray and mass spectrometry datasets in cancer studies, Briefings Bioinform. (2009), bbp012.
[291] W. Zhao, R. Chellappa, P.J. Phillips, A. Rosenfeld, Face recognition: a literature survey, ACM Comput. Surv. (CSUR) 35 (4) (2003) 399–458.
[292] E. Hjelmås, B.K. Low, Face detection: A survey, Comput. Vis. Image Underst. 83 (3) (2001) 236–274.
[293] J. Li, W. Hao, X. Zhang, X. Learning kernel subspace for face recognition, Neurocomputing 151 (2001) 1187–1197.
[294] M. Cannon, J.J.E. Slotine, Space-frequency localized basis function networks for nonlinear system estimation and control, Neurocomputing 9 (3) (1995) 293–342.
[295] Y.X. Zhao, X. Du, G.L. Xia, L.G. Wu, A novel algorithm for wavelet neural networks with application to enhanced PID controller design, Neurocomputing 158 (2015) 257–267.
[296] M.R. Warnes, J. Glassey, G.A. Montaguen, B. Kara, Application of radial basis function and feedforward artificial neural networks to the Escherichia coli fermentation process, Neurocomputing 2 (1–3) (1998) 67–82.
[297] M.A. Bezerra, R.E. Santelli, E.P. Oliveira, L.S. Villar, L.A. Escaleira, Response surface methodology (RSM) as a tool for optimization in analytical chemistry, Talanta 76 (5) (2008) 965–977.
[298] M. Ibnkahla, Applications of neural networks to digital communications – a survey, Signal Process. 80 (7) (2000) 1185–1215.
[299] A. Guisan, N.E. Zimmermann, Predictive habitat distribution models in ecology, Ecol. Model. 135 (2–3) (2000) 147–186.
[300] W.Z. Lu, H.Y. Fan, S.M. Lo, Application of evolutionary neural network method in predicting pollutant levels in downtown area of Hong Kong, Neurocomputing 51 (2003) 387–400.
[301] J.M. Gutierrez-Villalobos, J. Rodriguez-Resendiz, E.A. Rivas-Araiza, V. H. Mucino, A review of parameter estimators and controllers for induction motors based on artificial neural networks, Neurocomputing 118 (2013) 87–100.
[302] L. Qi, H.B. Shi, Adaptive position tracking control of permanent magnet synchronous motor based on RBF fast terminal sliding mode control, Neurocomputing 115 (2013) 23–30.
[303] S.A. Kalogirou, Applications of artificial neural-networks for energy systems, Appl. Energy 67 (1–2) (2000) 17–35.
[304] C. Booth, J.R. McDonald, The use of artificial neural networks for condition monitoring of electrical power transformers, Neurocomputing 23 (1–3) (1998) 97–109.
[305] P.K. Wong, Z.X. Yang, C.M. Vong, J.H. Zhong, Real-time fault diagnosis for gas turbine generator systems using extreme learning machine, Neurocomputing 128 (2014) 249–257.
[306] H.S. Hippert, C.E. Pedreira, R.C. Souza, Neural networks for short-term load forecasting: a review and evaluation, IEEE Trans. Power Syst. 16 (1) (2001) 44–55.
[307] R.R. Trippi, E. Turban, Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real World Performance, McGraw-Hill, Inc, 1992.
[308] J.R. Coakley, C.E. Brown, Artificial neural networks in accounting and finance: modeling issues, Int. J. Intell. Syst. Acc. Financ. Manag. 9 (2) (2000) 119–144.
[309] I. Kaastra, M. Boyd, Designing a neural network for forecasting financial and economic time series, Neurocomputing 10 (3) (1996) 215–236.
[310] G. Wang, J. Hao, J. Ma, H. Jiang, A comparative assessment of ensemble learning for credit scoring, Expert. Syst. Appl. 38 (1) (2011) 223–230.
[311] S. Khemakhem, Y. Boujelbènea, Credit risk prediction: a comparative study between discriminant analysis and the neural network approach, J. Acc. Manag. Inf. Syst. 14 (1) (2015) 60–78.
[312] H.M. Zhong, C.Y. Miao, Z.Q. Shen, Y.H. Feng, Comparing the learning effectiveness of BP, ELM, I-ELM, and SVM for corporate credit ratings, Neurocomputing 128 (2014) 285–295.
[313] E.I. Altman, G. Marco, F. Varetto, Corporate distress diagnosis: comparisons using linear discriminant analysis and neural networks (the Italian experience), J. Bank. Financ. 18 (3) (1994) 505–529.
[314] M.G. Reese, Application of a time-delay neural network to promoter annotation in the Drosophila melanogaster genome, Comput. Chem. 26 (1) (2001) 51–56.
[315] M. Liu, Y.D. He, J.X. Wang, H.P. Lee, Y.C. Liang, Hybrid intelligent algorithm and its application in geological hazard risk assessment, Neurocomputing 149 (2015) 847–853.
[316] D. Lu, P. Mausel, E. Brondizio, E. Moran, Change detection techniques, Int. J. Remote. Sens. 25 (12) (2004) 2365–2407.
[317] S. Mukkamala, G. Janoski, A. Sung, Intrusion detection using neural networks and support vector machines, in: Proceeding of the 2002 International Joint Conference on Neural Networks, vol. 13, IEEE Neural Network Soc., pp. 1702–1707, 2002.
[318] S.C. Lee, D.V. Heinbuch, Training a neural-network based intrusion detector to recognize novel attacks, Syst., Man. Cybern. 31 (4) (2001) 294–299.
[319] E. De la Hoz, E. De La Hoz, A. Ortiz, J. Ortega, B. Prieto, PCA filtering and probabilistic SOM for network intrusion detection, Neurocomputing 164 (2015) 71–81.
[320] Z.L. Sun, H. Wang, W.S. Lau, G. Seet, D.W. Wang, Application of BW-ELM model on traffic sign recognition, Neurocomputing 128 (2014) 153–159.
[321] R.W. Swiniarski, L. Hargis, Rough sets as a front end of neural-networks texture classifiers, Neurocomputing 36 (2001) 85–102.
[322] R. Plamondon, S.N. Srihari, Online and off-line handwriting recognition: a comprehensive survey, Patten Anal. Mach. Intell. IEEE Trans. 22 (1) (2000) 63–84.

[323] A.K.S. Jardine, D.M. Lin, D. Banjevic, A review on machinery diagnostics and prognostics implementing condition-based maintenance, Mech. Syst. Signal Process. 20 (7) (2006) 1483–1510.
[324] H.H. Bafroui, A. Ohadi, Application of wavelet energy and Shannon entropy for feature extraction in gearbox fault detection under varying speed conditions, Neurocomputing 133 (2014) 437–445.
[325] J.A. Noble, D. Boukerroui, Ultrasound image segmentation: a survey, IEEE Trans. Med. imaging 25 (8) (2006) 987–1010.
[326] D.J. Hemanth, C.K.S. Vijila, A.I. Selvakumar, J. Anitha, Performance improved iteration-free artificial neural networks for abnormal magnetic resonance brain image classification, Neurocomputing 130 (2014) 98–107.
[327] J. Khan, J.S. Wei, M. Ringner, L.H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu, C. Peterson, Meltzer. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, Nat. Med. 7 (6) (2001) 673–679.
[328] J.D. Wulfkuhle, L.A. Liotta, E.F. Petricoin, Proteomic applications for the early detection of cancer, Nat. Rev. Cancer 3 (4) (2003) 267–275.
[329] A. Statnikov, C.F. Aliferis, I. Tsamardinos, D. Hardin, A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis, Bioinformatics 21 (5) (2005) 631–643.
[330] J.C. Lindon, E. Holmes, J.K. Nicholson, Pattern recognition methods and applications in biomedical magnetic resonance, Prog. Nucl. Magn. Reson. Spectrosc. 39 (1) (2001) 40.
[331] L.A. Berrueta, R.M. Alonso-Salces, K. Héberger, Supervised pattern recognition in food analysis, J. Chromatogr. A 1158 (1) (2007) 196–214.
[332] A. Afantitis, G. Melagraki, P.A. Koutentis, H. Sarimveis, G. Kollias, Ligand-based virtual screening procedure for the prediction and the identification of novel β-amyloid aggregation inhibitors using Kohonen maps and Counterpropagation Artificial Neural Networks, Eur. J. Med. Chem. 46 (2) (2011) 497–508.
[333] M.M. Ardestan, M. Moazen, Z.X. Chen, J. Zhang, Z.M. Jin, A real-time topography of maximum contact pressure distribution at medial tibiofemoral knee implant during gait: application to knee rehabilitation, Neurocomputing 154 (2015) 174–188.
[334] M.A. Lopez-Gordo, F. Pelayo, A. Prieto, E. Fernandez, An auditory brain-computer interface with accuracy prediction, Int. J. Neural Syst. 22 (3) (2012) 1250009.
[335] I.A. Basheer, M. Hajmeer, Artificial neural networks: fundamentals, computing, design, and application, J. Microbiol. Methods 43 (1) (2000) 3–31.
[336] L. Carro-Calvo L, S. Salcedo-Sanz, J. Luterbache, Neural computation in paleoclimatology: General methodology and a case study, Neurocomputing 113 (2013) 262–268.
[337] C. Ambroise, G. Seze, F. Badran, S. Thiria, Hierarchical clustering of self-organizing maps for cloud classification, Neurocomputing 30 (1-4) (2000) 47–52.
[338] P. Li, L. Dong, H. Xiao, M. Xu, A cloud image detection method based on SVM vector machine, Neurocomputing 169 (2015) 34–42.
[339] P.M. Ferreira, E.A. Faria, A.E. Ruano, Neural network models in greenhouse air temperature prediction, Neurocomputing 43 (2002) 51–75.
[340] T.G. Barbounis, J.B. Theocharis, A locally recurrent fuzzy neural network with application to the wind speed prediction using spatial correlation, Neurocomputing 70 (7-9) (2007) 1525–1542.
[341] T.G. Barbounis, J.B. Theocharis, Locally recurrent neural networks for long-term wind speed and power prediction, Neurocomputing 69 (4-6) (2006) 466–496.
[342] M. Frasca, A. Bertoni, G. Valentini, UNIPred: Unbalance-aware Network Integration and Prediction of protein functions, J. Comput. Biol. 22 (12) (2015) 1057–1074.
[343] K. Sachs, O. Perez, D. Pe'e, D. Lauffenburger, G. Nolan, Causal protein-signaling network derived from multiparameter single-cell data, Science 308 (5721) (2005) 523–529.
[344] H.R. Maier, G.C. Dandy, Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications, Environ. Model. Softw. 15 (1) (2000) 101–124.
[345] J.W. Labadie, Optimal operation of multireservoir systems: State-of-the-art review, Journal. water Resour. Plan. Manag. 130 (2) (2004) 93–111.
[346] B. Bhattacharya, D.P. Solomatine, Neural networks and M5 model trees in modelling water level-discharge relationship, Neurocomputing 63 (2005) 381–396.
[347] T. Hill, L. Marquez, M. O'Connor, W. Remus, Artificial neural network models for forecasting and decision making, Int. J. Forecast. 10 (1) (1994) 5–15.
[348] D. West, S. Dellana, J. Qian, Neural network ensemble strategies for financial decision applications, Comput. Oper. Res. 32 (10) (2005) 2543–2559.
[349] B. Malakooti, Y.Q. Zhou, Feedforward artificial neural networks for solving discrete multiple criteria decision making problems, Manag. Sci. 40 (11) (1994) 1542–1561.
[350] L. Chen, S. Lin, An interactive neural network-based approach for solving multiple criteria decision-making problems, Decis. Support. Syst. 36 (2) (2003) 137–146.
[351] D. Floreano, F. Mondada, Evolutionary neurocontrollers for autonomous mobile robots, Neural Netw. 11 (7-8) (1998) 1461–1478.
[352] M.J. Mahmoodabadi, M. Taherkhorsand, A. Bagheri, Optimal robust sliding mode tracking control of a biped robot based on ingenious multi-objective PSO, Neurocomputing 124 (2014) 194–209.
[353] H.N. Nguyen, J. Zhou, H.J. Kang, A calibration method for enhancing robot accuracy through integration of an extended Kalman filter algorithm and an artificial neural network, Neurocomputing 151 (2015) 996–1005.
[354] F.H. Lu, R. Undehauen, Applied Neural Networks for Signal Processing, Cambridge Universitu Press, 1998.
[355] F. Lotte, M. Congedo, A. Lecuyer, F. Lamarche, B. Arnaldi, A review of classification algorithms for EEG-based brain-computer interfaces, J. Neural Eng. 4 (2) (2007).
[356] Y. Wu, Y.B. Ge, A novel method for motor imagery EEG adaptive classification based biomimetic pattern recognition, Neurocomputing 116 (2013) 280–290.
[357] E. Oja, The nonlinear PCA learning rule in independent component analysis, Neurocomputing 17 (1) (1997) 25–45.
[358] C.G. Puntonet, A. Prieto, Neural net approach for blind separation of sources based on geometric properties, Neurocomputing 18 (1-3) (1998) 141–164.
[359] A. Prieto, C.G. Puntonet, B. Prieto, A neural learning algorithm for blind separation of sources based on geometric properties, Signal Process. 64 (3) (1998) 315–331.
[360] A. Cichocki, J. Karhunen, W. Kasprzak, R. Vigario, Neural networks for blind separation with unknown number of sources, Neurocomputing 24 (1–3) (1999) 55–93.
[361] D. Simon, H. Elsherief, Navigation satellite selection using neural networks, Neurocomputing 7 (3) (1995) 247–258.
[362] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, D. Freeman, Autoclass: A Bayessian Classification Systems: In Readings in knowledge acquisition and learning, Morgan Kaufmann Publishers 1993, pp. 431–441.
[363] T. Sagara, M. Hagiwara, Natural language neural network and its application to question-answering system, Neurocomputing 142 (2014) 201–208.
[364] S.M. Siniscalchi, T. Svendsen, C.H. Lee, An artificial neural network approach to automatic speech processing, Neurocomputing 140 (2014) 326–338.
[365] L. Gajecki, Architectures of neural networks applied for LVCSR language modelling, Neurocomputing 133 (2014) 46–53.
[366] K. Lo, F. Hahne, R. Brinkman, R. Ryan, R. Gottardo, Flow-class: a bioconductor package for automated gating of flow cytometry data, BMC Inform. 10 (2009) 145.
[367] G.M. Foody, A. Mathur, A relative evaluation of multiclass image classification by support vector machines, IEEE Trans. Geosci. Remote. Sens. 42 (6) (2004) 1335–1343.
[368] A.K. Jain, R.P.W. Duin, J.C. Mao, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 4–37.
[369] E.R. Hruschka, N.F. Ebecken, Extracting rules from multilayer perceptrons in classification problems: a clustering-based approach, Neurocomputing 70 (1) (2006) 384–397.
[370] A. Nazemi, M. Dehghan, A neural network method for solving support vector classification problems, Neurocomputing 152 (2015) 369–376.
[371] G.B. Huang, X. Ding, H. Zhou, Optimization method based extreme learning machine for classification, Neurocomputing 74 (1) (2010) 155–163.
[372] Y. Lan, Y.C. Soh, G.B. Huang, Constructive hidden nodes selection of extreme learning machine for regression, Neurocomputing 73 (16-18) (2010) 3191–3199.
[373] T.Y. Kim, K.J. Oh, C.H. Kim, J.D. Don, Artificial neural networks for non-stationary time series, Neurocomputing 61 (2004) 439–447.
[374] T. Kuremloto, S. Kimura, K. Kobayashi, M. Obayashim, Time series forecasting using a deep belief network with restricted Boltzmann machines, Neurocomputing 137 (2014) 47–56.
[375] C. Alippi, V. Piuri, Experimental neural networks for prediction and identification, IEEE Trans. Instrum. Meas. 45 (2) (1996) 670–676.
[376] D.K. Wedding, K.J. Cios, Time series forecasting by combining RBF networks, certainty factors, and the Box-Jenkins model, Neurocomputing 10 (2) (1996) 149–168.
[377] L.J. Herrera, H. Pomares, I. Rojas, A. Guillén, A. Prieto, O. Valenzuela, Recursive prediction for long term time series forecasting using advanced models, Neurocomputing 70 (16–18) (2007) 2870–2880.
[378] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. (CSUR) 31 (3) (1999) 264–323.
[379] G.P. Zhang, Neural networks for classification: a survey. systems, man, and cybernetics, part C: applications and reviews, IEEE Trans. 30 (4) (2000) 451–462.
[380] R. Xu, D. Wunsch, Survey of clustering algorithms, Neural Netw. IEEE Trans. Neural Netw. 16 (3) (2005) 645–678.
[381] D.J. Hemanth, C.K.S. Vijila, A.I. Selvakumar, J. Anitha, An adaptive filtering approach for electrocardiogram (ECG) signal noise reduction using neural networks, Neurocomputing 117 (2013) 206–213.
[382] P.S. Churchland, C. Koch, T.J. Sejnowski, What is computational neuroscience?, in: Eric L. Schwartz (Ed.), Computational Neuroscience, MIT Press, 1993, pp. 46–55.
[383] T.J. Sejnowski, Computational Neuroscience International Encyclopedia of the Social & Behavioral Sciences, Second Edition, Elsevier 2015, pp. 480–484.
[384] M. Akay (Edt), Handbook of Neural Engineering, Wiley-IEEE Press, 2007.
[385] D.J. DiLorenzo, J.D. Bronzino (Edts), Neuroengineering, 6th Ed, CRC Press, 2008.
[386] C.E. Schmidt, J.B. Leach, Neural tissue engineering: strategies for repair and regeneration, Annu. Rev. Biomed. Eng. 5 (1) (2003) 293–347.
[387] J.R. Wolpaw, N. Birbaumer, D.J. McFarland, G. Pfurtschellere, T.M. Vaughana, Brain-computer interfaces for communication and control, Clin. Neurophysiol. 113 (6) (2002) 767–791.
[388] M.A. Lopez, A. Prieto, F. Pelayo, C. Morillas, Use of Phase in Brain-Computer Interfaces based on Steady-State Visual Evoked Potentials, Neural Process. Lett. 32 (1) (2010) 1–9.

[389] E. Cattin, S. Roccella, N. Vitiello, I. Sardellitti, P.K. Artemiadis, P. Vacalebri, F. Vecchi, M.C. Carrozza, K.J. Kyriakopoulos, P. Dario, Design and development of a novel robotic platform for neuro-robotics applications: the neurobotics arm (NEURARM), Adv. Robot. 22 (1) (2008) 3–37.
[390] N. Burgess, J.G. Donnett, J. O'Keefe, Using a Mobile Robot to Test a Model of the Rat Hippocampus, Connect. Sci. 10 (3-4) (1998) 291–300.
[391] N.R. Luque, J.A. Garrido, R.R. Carrillo, E. D'Angelo, E. Ros, Fast convergence of learning requires plasticity between inferior olive and deep cerebellar nuclei in a manipulation task: a closed-loop robotic simulation, Front. Comput. Neurosci. (2014) 1–16.
[392] G. Rozenberg, T. Back, J. Kok (Eds.), Handbook of Natural Computing, Springer Verlag, 2012.
[393] W. Pedrycz, Computational Intelligence: An Introduction, CRC Press, 1997.
[394] L.A. Zadeh, Fuzzy sets, Inf. Control. 8 (3) (1965) 338–353.
[395] A. Prieto, M. Atencia, F. Sandoval, Advances in artificial neural networks and machine learning, Neurocomputing 121 (2013) 1–4.
[396] G.M. Shepherd, J.S. Mirsky, M.D. Healy, M.S. Singer, E. Skoufos, M.S. Hines, P. M. Nadkarni, P.L. Miller, The Human Brain Project: neuroinformatics tools for integrating, searching and modelling multidisciplinary neuroscience data, Trends Neurosci. 21 (11) (1998) 460–468.
[397] Brain Mapping by Integrated Neurotechnologies for Disease Studies. Official website: 〈http://brainminds.jp/en/〉. Last modification: 15.07.15.
[398] Website: 〈https://www.science.org.au/publications/inspiring-smarter-brain-research-australia〉. Last modification: 24.02.14.
[399] Brainnetome Project. Official website: 〈http://www.brainnetome.org/en/brainnetomeproject.html〉. Last modification: 22.07.15.
[400] J. TianZi, Brainnetome and related projects, Sci. Sin. Vitae 57 (4) (2014) 462–466.
[401] Norwegian University of Science and Technology. 〈http://www.ntnu.edu/kavli/research/norbrain〉.
[402] University of Oslo. 〈http://www.med.uio.no/imb/english/research/about/infrastructure/norbrain/〉.
[403] SpikeFORCE Project in Information Society Technologies World. Website: 〈http://www.ist-world.org/ProjectDetails.aspx?ProjectId=5e284098967d4471961edde067abd27a〉.
[404] Sensemaker Project in Information Society Technologies World. Website: 〈http://www.ist-world.org/ProjectDetails.aspx?ProjectId=e9a2613ab2d64ef7b8ea8ab113f11976〉.
[405] The FACETS project. Website: 〈http://facets.kip.uni-heidelberg.de/〉.
[406] The SENSOPAC Project. Website: 〈http://www.sensopac.org/〉.
[407] The BrainScaleS Project. Website: 〈http://brainscales.kip.uni-heidelberg.de/〉.
[408] The Blue Brain Project. Website: 〈http://bluebrain.epfl.ch/〉.
[409] The REALNET Project. Website: 〈http://www.realnet-fp7.eu/〉.
[410] The Human Brain Project. A Report to the European Commission. The HBP-PS Consortium, Lausanne, April 2012. 〈https://goo.gl/3G6HMd〉.
[411] Human Brain Project. Official website: 〈https://www.humanbrainproject.eu/〉.
[412] The Neurorobotics platform (HBP). Website: 〈http://neurorobotics.net/the-human-brain-project/〉.
[413] BRAIN 2025, A Scientific Vision June, 5, National Institutes of Health, 2014 〈http://braininitiative.nih.gov/2025/BRAIN2025.pdf〉.
[414] E.R. Kandel, H. Markram, P.M. Matthews, R. Yuste, C. Koch, Neuroscience thinks big (and collaboratively), Neuroscience 14 (2013) 659–664.
[415] Allen Institute for Brain Science. Official website: 〈http://alleninstitute.org/〉.
[416] Human Brain Project. Press Officer. What People are saying. 〈https://www.humanbrainproject.eu/es/media〉.
[417] A. Roy; 〈http://www.neuroinf.org/pipermail/comp-neuro/2014-June/004822.html〉.
[418] B. Meyerson, Top 10 emerging technologies of 2015, The World Economic Forum (2015), 〈https://agenda.weforum.org/2015/03/top-10-emerging-technologies-of-2015-2/〉.
[419] L.A. Barroso, U. Hölzle, The case for energy-proportional computing, Computer 12 (2007) 33–37.
[420] M. Costandi, How to buil a brain July, Sciencefocus.com 2012, pp. 32–38.
[421] The Blue Brain Project. EPFL. 〈http://bluebrain.epfl.ch/〉.

Prof. Alberto Prieto earned his BSc in Physics (Electronics) in 1968 from the Complutense University in Madrid, Spain. In 1976, he completed a PhD at the University of Granada (UGR). This thesis received the PhD Award in the field of Physics of the UGR and the Citema Foundation National Award (Sain).
From 1971 to 1984 he was founder and Head of the Computing Centre, and he headed the Computer Science and Technology Studies at the UGR from 1985 to 1990. Also he was founder and Head of the Research Centre in Information Technology and Communications (CITIC-UGR) from 2011 to 2013. He is currently Full Professor at the Department of Computer Architecture and Technology at the University of Granada.
He is the co-author of four books (one of them with four editions) in the field of Computer Technology and Architecture for McGraw-Hill and Thomson publishing houses, and has co-edited five volumes of the series Lecture Notes in Computer Science (LNCS). More than 250 articles written by him have been published in specialized journals and conferences. He has co-directed 26 PhD theses. He is on the Editorial Board of "Neurocomputing" (Elsevier), "Neural Processing Letters" (Springer), and "International Journal of High Performance Systems Architecture" (Inderscience). Prof. Prieto is Life Senior Member of the IEEE, and founder and Chairman (until 2013) of the Spanish Chapter of the IEEE Computational Intelligence Society.
His area of research primarily focuses on advanced computing architectures and artificial neural networks.

Dr. Beatriz Prieto has a B.S. degree in Electronics Engineering and a PhD degree in Electronics from the University of Granada (Spain). She is currently associate professor at the Department of Computer Architecture and Technology of the same University. She has worked in the areas of blind separation of sources and artificial neural networks. His current research interest lies in the fields of neural networks and neuroengineering.

Dr. Eva M. Ortigosa received the Ph.D. degree in computer engineering from the University of Málaga, Spain, in 2002. She was with the Computer Architecture Department, University of Málaga, from 1996 to 2002. Since then, she has been with the Department of Computer Architecture and Technology, University of Granada, Granada, Spain, where currently she is an Assistant Professor. Her research interests include hardware implementation of digital circuits for real time processing in embedded systems, spiking neuron models and synchronization processes.

Prof. Eduardo Ros received the Ph.D. degree in 1997. He is currently Full Professor at the Department of Computer Architecture and Technology at the University of Granada. He received the Andalucia's award for best young researchers in 2002. He has published more than 70 papers on international journals.
He has three patents under exploitation and has participated in the creation of a technology based enterprise, spin-off from the University of Granada, which has received different awards such as the Entrepreneur prize XXI 2009 (Caixa), Young entrepreneur prize 2008 (Bancaja) and Young Enterprize 2008 (AJE).
He is a very active researcher at an international level, he has participated as IP in 6 European Grants (of the 5th, 6th and 7th EU frameworks). He leads an interdisciplinary lab, with interest in computational neuroscience, neuromorphic engineering real-time image processing, biomedical applications, etc. In particular, his main research interests include simulation of biologically plausible processing schemes with spiking neural networks, high performance computer vision, hardware implementation of digital circuits for real-time processing in embedded systems, etc.

Prof. Francisco Pelayo, received the B.Sc. degree in Physics in 1982, the M.Sc. degree in Electronics in 1983, and the Ph.D. degree in 1989, all from the University of Granada, Spain. He is currently a full professor at the Department of Computer Architecture and Technology (http://atc.ugr.es) of the same university. He has worked in the areas of VLSI design and test, reconfigurable hardware, artificial neural networks and fuzzy systems. His current research interest lies in the fields of bioinspired processing and control systems, robotics, neuroengineering and brain-computer interfaces.

Prof. Julio Ortega received his B.Sc.degree in Electronic Physics in1985, M.Sc.degree in Electronics in 1986, and Ph.D.degree in 1990, all from the University of Granada, Spain. His Ph.D. dissertation has received the Award of Ph.D. dissertations of the University of Granada. He was at the Open University, U.K., Department of Electronics (University of Dortmund, Germany) and Department of Computer Science and Electrical Engineering (University of Essex, UK), as invited researcher. Currently he is a Full Professor at the Department of Computer Technology and Architecture of the University of Granada. He has published more than 50 technical papers on international journals included in the JCR, more than 130 contributions to international conferences. His research interests include parallel processing and parallel computer architectures, multiobjective optimization, artificial neural networks, and evolutionary computation. He has led research projects in the area of parallel algorithms and architectures for classification and optimization problems. He is Senior Member of the IEEE Computer Society.

Prof. Ignacio Rojas received the B.Sc. degree in electronic physics in 1992 and the Ph.D. degree in Intelligent Systems with honors in 1996, both from the University of Granada, Spain. He has been visiting professor at the University of Dortmund (Germany) at the Department of Electrical Engineering, as a visiting researcher with the BISC Group of Prof. L.Zadeh, University of California, Berkeley and as visiting professor at the Muenster University of Applied Sciences, Germany. He is currently Full Professor with the Department of Computer Architecture and Computer Technology, University of Granada. He is member of the IEEE Computational Intelligence Society and researcher in the CITIC-UGR (Information and Communications Technology Centre of the University of Granada).
His research interests include hybrid system, hardware-software implementation, combination of intelligent system for adaptive control, self-organizing neuro-fuzzy systems, neural networks, time series forecasting, e-monitoring, e-health, bioinformatics, computational biology, data mining and architectures for complex optimization problems.