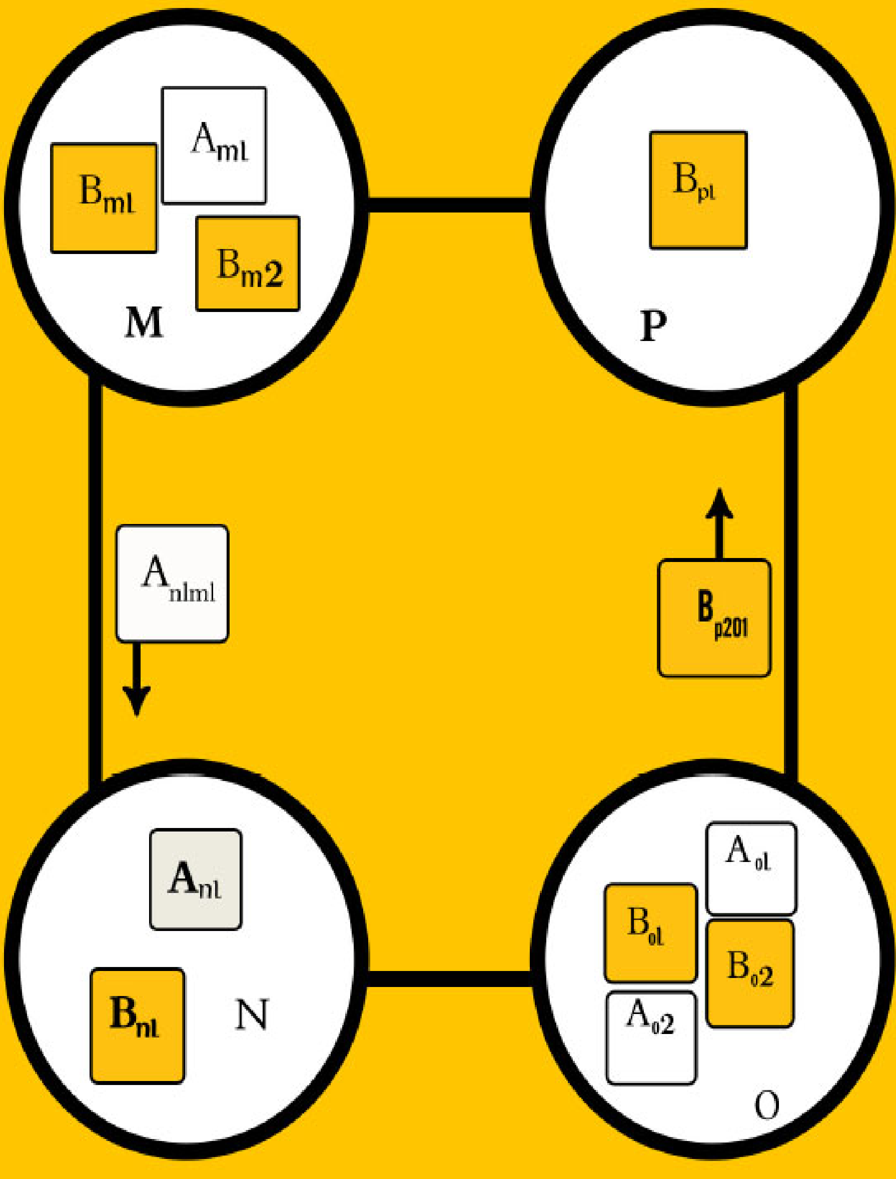
eISBN: 978-1-68108-090-1
ISBN: 978-1-68108-091-8

ARTIFICIAL NEURAL SYSTEMS

Principle and Practice



Pierre Lorrentz

Bentham e Books

ARTIFICIAL NEURAL SYSTEMS

Principle and Practice

Authored By

Pierre Lorrentz

University of Kent

United Kingdom

Bentham Science Publishers Ltd.
Executive Suite Y - 2
PO Box 7917, Saif Zone
Sharjah, U.A.E.
Email: subscriptions@benthamscience.org

BENTHAM SCIENCE

CONTENTS

FOREWORD

Neural Networks, Fuzzy Logic and Evolutionary Computing are members of Soft Computing class of techniques. The techniques are capable of identifying and handling inexact solutions for complex tasks and can deal with the real life uncertainties within the computational framework. Soft Computing has significantly matured over the years and we can find significant applications of soft computing in industry and research environment. Neural Networks is the most mature of other techniques in the Soft Computing. The networks have also benefitted from integrated Fuzzy Logic based systems to model complex engineering systems with the human expert knowledge and through robust system modelling.

Evolutionary Computing helps to optimise the design a Neural Network. Each members of the Soft Computing has several algorithms and concepts that need better understanding for application development.

This book on ‘artificial neural networks – principle and practice’ provides necessary foundation to understand the basics of Neural Networks and how to develop real life applications. From basic definitions to relevant theorems the book presents an algorithm approach to describe the foundations.

The book also emphasises systematic approach to Intelligent System analysis and design. In order to build a Neural Network or Artificial Neural Network application, one needs to apply knowledge of probability based methods as well as fuzzy sets for more uncertain aspects of the problem. The book then explains the motivation from our neural system to develop the Neural Networks.

Description of other network-based approaches using nodes and edges also strengthens the understanding about the Neural Networks. A major strength of the book is the fundamentals of quantum logic for emerging Neural Network development. This is major area for future development. A discussion on Neural Network hardware would have strengthened the book.

The second part of the book presents detailed discussion on learning algorithms, current and emerging Neural Network structures and application development. The chapters also present metrics to evaluate effectiveness of the network. Selection and integration of multiple Neural Networks to solve a real life and complex problem is a major aspect of the book. As mentioned before there are several algorithms and approaches to solve a problem, the systematic approach presented in the book is of major interest. Application of the network to solve a complex modelling task usually requires significant volume of data. A further discussion on the modelling approaches with less data would be very helpful.

The emphasis on probability based neural network and its application is significant because of its popularity. But the real strength of this part of the book is in describing the Quantum Neural Networks and the Deep Belief Network (DBN).

Finally the book also outlines the research and development in Neural Networks. Future Neural Networks are learning from specialised parts of our neural system and trying to scale up to solve even more complex engineering applications.

Rajkumar Roy
Cranfield University, UK

PREFACE

An intelligent system is that which exhibit characteristics of learning, adaptation, and problem-solving, among others. The group of intelligent system, conceived and designed by human, is loosely termed Artificial Neural Network (ANN) System. Such ANN system is the theme of the book. The book also describe nets (also called network or graphs), evolutionary methods, clustering algorithm, and others nets, most of which are complementary to ANN system.

The term "practice" in the title refers to design, analysis, performances assessment, and testing. The design and analysis may be facilitated by the explanations, equations, diagrams, and algorithms given. Performance assessments occur in any section that bear the name and apply to any ANN system because they are standard independent methods and most ANN system has an associated error feedback. Testing is exemplified by case studies and is given toward the end of most chapters.

An interest in artificial neural sciences is a sufficient requirement to understand the content of the book, though knowledge of signal processing, mathematics, and electrical/electronic communication is an advantage. The book specifically takes a developmental perspective, making it more beneficial for professionals. The book adopts a spiral method of description whereby various topics are revisited several times; each visit introduces fresh material at increasing level of sophistication. Each visit to a specific ANN type may also introduce new ANN system(s) and/or new algorithm(s) as the case may be.

The book is divided into two parts (I and II). Part I contain five chapters. Chapter 1 introduce the biological neurons and basic artificial neurons. From these, chapter 2 derive better neurons and introduce statistical methods. Chapter 3 describe a framework of dynamic fuzzy-neuron, and explain the fundamental principle governing the design and analysis of ANN system. To distinguish other algorithms (*e.g.* clustering algorithm) from learning algorithms, chapter 4 describe fundamentals of genetic algorithm, clustering algorithms, and those other algorithms complementary to ANN systems. Neural network is in chapter 3 introduced by graph. Chapter 5 concludes part 1 by introducing quantum neural network, quantum maths and logic. The chapter also describe Hodgkin-Huxley neuron, and memristance.

Similarly, part II consists of six chapters. In Chapter 6, artificial neuromorphic network, and Widrow-Hoff learning are visited; so is fuzzy ANN system. While chapter 7 describes the usual weighted, weightless ANN systems. It also introduces Bayesian ANNs, and discusses general performance assessment methods. On the other hand, chapter 8 considers various selection and combination strategy for ANN systems. Chapter 9 is dedicated to Bayesian networks. There are some promising ANN systems being considered in the research arena, and also now in chapter 10, these ANN may revolutionize ANN throughput in future. In chapter 11 implementation issues regarding Monte Carlo algorithm is visited, and also implementation issues regarding neuromorphic networks is revisited.

The book attempts to impart considerable knowledge of know-how of ANN to the reader in order to facilitate a novel development and research. Albeit also improve an ad-hoc ANN. This may encourage and help a developer to meet any industrial increasing demand for novel ANNs' implementation and application.

Pierre Lorrentz
University of Kent,
United Kingdom

Part 1

Principles

CHAPTER 1

Neurons

Abstract: The aim of this chapter is to explain what a natural biological neuron is, and what an artificial neuron is. To this end, the first section introduces the biological neuron, explains its structure and its information transmission methods. The second section explains how an artificial neuron may be obtained from a corresponding biological neuron. The resources for the artificial neuron may be purely electrical in nature and the behaviour of the resulting electric circuit is expected to be similar to that of information transmission of a biological neuron.

Keywords: Active transport, Axon, Calcium, Conduction, Conductance, Central nervous system, Dendrites , Diffusion, Depolarization, Electrogenesis, Ganglia, Hyperpolarization, Motor, Myelin sheath, Neurotransmitter, Neuron, Potassium, PRVP, Sodium, Soma, Sheath.

A BIOLOGICAL NEURON

Neurons form the fundamental components of the central nervous system (CNS) and the ganglia of the Peripheral nervous system (PNS). Neurons are also found in other locations which may accord them a corresponding name *e.g.* sensory neurons, motor neurons, and interneurons.

As shown in Fig. (**1**), a normal neuron has a soma (cell body), dendrites, and an axon. The term neurite refers to an axon, any dendrite, or other protrusions from the soma of the neuron without paying attention to their differences. Axon emerges from the soma at a base called the axon hillock and usually extends a longer distance than any dendrite of the neuron. Neurons do not undergo cell division but are generated by stem cells. Biology and Bio-scientific researchers have confirmed that the main features that distinguish a neuron are: (1) electrical excitability, and (2) the presence of synapses which are complicated junctions that permit signals to travel to other cells.

Pierre Lorrentz

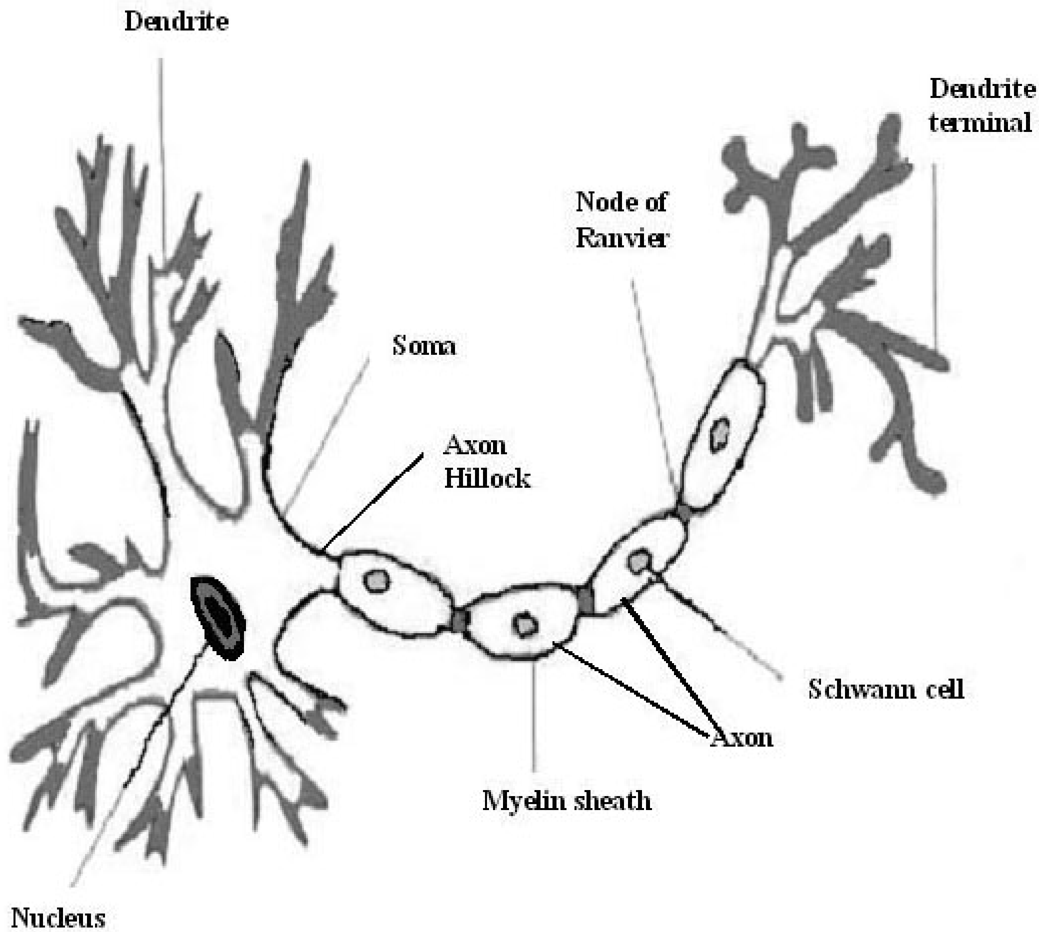


Fig. (1). A natural biological neuron.

Dendrites normally branched profusely from both the soma and the axon. Every neuron has only one axon which maintains the same approximate diameter throughout its length. The myelin sheath provides a protective coating around the axon. The myelin sheath allows the action potential to propagate faster than it would have been if compared with another axon of equal diameter. Neurons performs various specialized functions depending on their location and event received, Events are received by communication which is effected in two ways. One is by the release/absorption of neurotransmitter from the surrounding; this is a partly chemical process called neurotransmission. The second is the synaptic transmission. These modes of communication and the associated energy required for the communication is common to all natural biological neurons.

Synaptic Transmission

Synaptic signal is either excitatory or inhibitory. If the net signal excitation exceeds certain threshold and is sufficiently large, it generates a brief electrical pulse called action potential which originates at the soma. The action potential propagates down the axon as follows.

There are pores not covered by myelin sheath (see Fig. **2**) through which ion exchanges occur between the axon and the extrinsic fluid; these pores are known as nodes of Ranvier. The ion exchanges are responsible for the production of action potential. The action potential at one node is most often sufficient to initiate another action potential at a nearby node. A signal thus travels discretely rather than continuously along an axon. This mode of transmission along the axon is termed saltatory conduction.

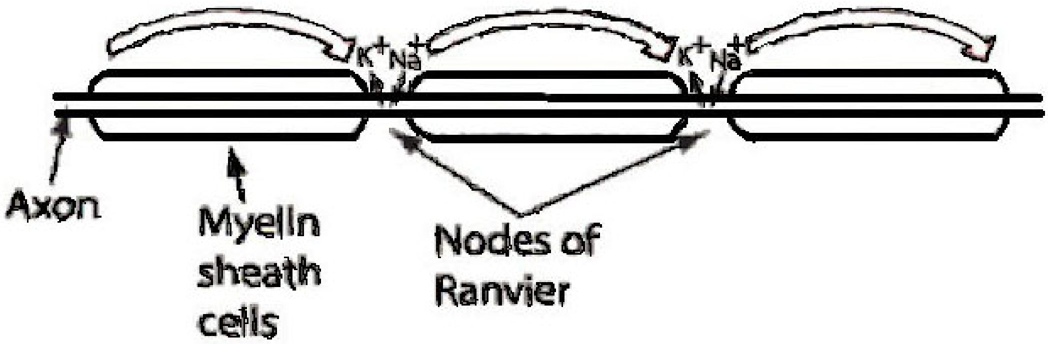


Fig. (2). A section of axon showing saltatory conduction.

TRANSMISSION ACROSS SYNAPSES

A presynaptic action potential propels the calcium Ca^{2+} ions through the voltage-gated calcium channel.

As depicted in Fig. (**3**), a Presynaptic Releasable Vesicle Pool (PRVP) constitutes the active synaptic region of the dendritic terminal ends. The concentration of the Ca^{2+} causes the PRVP vesicles to fuse with the membrane and release the neurotransmitters into the synaptic region. The neurotransmitters move by

CHAPTER 2

Basic Neurons

Abstract: The aim and objectives of this chapter is to present other types of artificial neuromorphic neurons with capability of reset and recovery. For this reason, the first section starts with the integrate-and-fire neuron, which has the propensity for reset. The second section introduces probability theory owing to the fact that many processes in the brain and central nervous system obey probability laws. The third section introduces another artificial neuromorphic neuron which employs a Poisson process and is closer in behaviour to a biological neuron.

Keywords: Bayes theorem, Binomial, Bernoulli, Charging, Depolarization, Density function, Excitatory, Expected-value, Inhibitory, Mean, Moment, ODE, Pseudo-random-number-generator, Poisson, Steady-state, Synaptic strength, Spike, Threshold potential, Uniform distribution, Variance.

The first chapter has introduced one biological neuron and one artificial neuron. One advantage of developing ANN from principle is that reproduction is assured with minimal loss of resources and a target performance may often be achieved. Since the book is more about artificial neural network systems, chapter 1 contains the last item on biological neuron. Most development throughout the book however depends, directly or indirectly, on the biological neuron so that it may be regarded as an introduction to the rest of the book.

INTEGRATE-AND-FIRE NEURON

There is another version of artificial neuron model known as integrate-and-fire model; this is a version of figure 5 chapter 1 neuron with an inclusion of spike generation and reset. It states that when the membrane potential [1, 2] reaches or exceeds a threshold potential θ, firing an action potential [3] and discharging occurs. After that, it reset and (re-)build its potential again. The charging proceeds as follows.

Pierre Lorrentz

$$C_m \frac{\partial V}{\partial t} = -\left(\frac{V - E_m}{R_m}\right) - I_v \tag{1}$$

Multiplying (1) by R_m;

$$\tau_m \frac{\partial V}{\partial t} = -V + E + R_m I \tag{2}$$

where $\tau_m = R_m C_m$

Equation (2) is a first-order ODE, whose solution is given by:

$$V(t) = E_m + R_m I\left(1 - \exp\left(-\frac{T_s}{\tau_m}\right)\right) \tag{3}$$

One may be interested at what frequency $f(I)$ does the neuron fire. The neuron fire whenever the voltage V equals θ the threshold voltage or exceed it. Setting $Em = 0$ and $V = \theta$ in equation (3);

$$\theta = R_m I\left(1 - \exp\left(-\frac{T_s}{\tau_m}\right)\right) \tag{4}$$

$$\frac{\theta}{R_m I} = 1 - \exp\left(-\frac{T_s}{\tau_m}\right)$$

$$\left(\frac{\theta}{R_m I} - 1\right) = -\exp\left(-\frac{T_s}{\tau_m}\right)$$

$$\left(1 - \frac{\theta}{R_m I}\right) = \exp\left(-\frac{T_s}{\tau_m}\right)$$

$$Ln\left(1 - \frac{\theta}{R_m I}\right) = -\frac{T_s}{\tau_m}$$

$$-\tau_m Ln\left(1 - \frac{\theta}{R_m I}\right) = T_s$$

$$f(I) = \frac{1}{T_s + \tau_s} = \frac{1}{\tau_s - \tau_m Ln\left(1 - \frac{\theta}{R_m I}\right)} \quad \textbf{(5)}$$

where I is the injected current.

In order to apply this artificial neuron to model a stereo-typical situation found in CNS of some animals, a *distribution* known as *Poisson distribution* shall be introduced. A relevant introductory probability theory is presented now.

PROBABILITY

Definition 1.1: Probability is a set function *p* that assigns to each datum x_i in the sample space X a number $p(x_i)$ called the probability of the datum x_i, such that the following properties hold:

1) $p(x_i) \geq 0$

2) $p(X) = 1$

3) If $x_1, x_2, x_3, \ldots$ are data and $x_i \cap x_j = \emptyset$, $i \neq j$, then

$p(x_1 \cup x_2 \cup \ldots \cup x_k) = p(x_1) + p(x_2) + \ldots + p(x_k)$, for each positive integer *k*, and $p(x_1 \cup x_2 \cup x_3 \cup \ldots) = p(x_1) + p(x_2) + p(x_3 + \ldots)$, for an infinite, but countable number of data. For any datum x_i,;

$$p(x_i) = 1 - p(x_i^{'}) \text{: where } x_i^{'} = \text{complement of } x_i \quad \textbf{(6)}$$

If x_i and y_i are any two independent databases with no data in common, then:

$$p(x_i \cup y_i) = p(x_i) + p(y_i) \quad \textbf{(7)}$$

Otherwise;

$$p(x_i \cup y_i) = p(x_i) + p(y_i) - p(x_i \cap y_i) \quad \textbf{(8)}$$

CHAPTER 3

Basic Fuzzy Neuron and Fundamentals of ANN

Abstract: The chapter's aim and objectives are to provide an artificial neural-based fuzzy-logic foundation, and a general framework for design and analysis of ANN systems. The first section therefore introduces membership functions, define and give a relatively full operational description of fuzzy-logic neuron. Subsequent section two introduces ANN design principles and analysis from which a general wave neural network is derived. A full understanding of this chapter may be sufficient to design and analyse any artificial neural network system.

Keywords: Aggregation operator, Bell-shaped, Composite, Delta-function, Experience, Fuzzy-set, Fuzziness, Gaussian, Hessian, Lower bound, Lagrange, Membership grade, Operator, Peak, Relation, Stimuli, Support set, Upper bound.

A FUZZY NEURON

Just as a probability value may be assigned to a random variable, so also is a grade of membership μ assigned to a fuzzy variable. Assigning a grade of membership μ to a fuzzy variable [1] x is often based on prior knowledge regarding measured values of x. The prior knowledge may have been formulated into an equation known as *membership function*. Evaluation of a membership function for a variable x gives a *membership grade* μ for that variable. Membership function is to fuzzy logic what probability distribution is to probability theory. Some notable types of membership function are now presented.

Delta-function: The delta function may be used as a membership function, given a threshold y_0, as follows.

Let F denote a fuzzy variable Y, then

$$\mu_F(y) = \delta(y - y_0) \tag{1}$$

Pierre Lorrentz

Where y = general datum $y \in F$. This is subject to:

$$\delta(y - y_0) = \begin{cases} 1; & \text{when } y = y_0 \\ 0; & \text{otherwise} \end{cases} \tag{2}$$

This is a point-evaluation that has no spread. Assuming the fuzzy set F consist of

Y_i; $i = 1,2,...,n$ fuzzy variables $y = y_i(y_i \in Y_i)$; $i = 1,2,...n$ then

$$\mu_F(y) = \mu_{Y_i}(y)\delta(y - y_0) \tag{3}$$

$$Y_i \in (\bigcup_{i=1}^{n} Y_i)$$

where each $y = y_i$ is subject to the constraint equation (2).

<u>Triangular Membership Function:</u> Here we would employ the equation of a triangle as a membership function as follows.

$$\mu_F = 1 - \frac{|y - y_0|}{s}; \text{ when } |y - y_0| \le s \tag{4}$$

Where y_0 is the peak of the triangle, and s is the base of the triangle. The s is commonly referred to as the *support set* of the membership function. The y is a general point on the triangle. Assuming the fuzzy set F consist of Y_i; i =1,2,...,n fuzzy variable $y_i \in Y$ then

$$\mu_F(y) = \mu_{Y_i}(y_0)\mu_{A_i}(y) \tag{5}$$

$$\mu_{A_i}(y) = \begin{cases} 1 - \frac{|y - y_0|}{s_i}; & \text{when } |y - y_0| \le s_i \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

For any y_i, the parameter s may be chosen to give different level of fuzziness to the fuzzy value.

Gaussian function: we have come across Gaussian (normal) distribution function in previous sections. To derive a membership function from a Gaussian distribution, express the exponential part of the normal distribution as a membership function.

$$\mu_{A_i}(y) = \exp\left(\frac{y - y_0}{s_i}\right)^2 \tag{7}$$

The Gaussian membership function (7) is a bell-shaped membership function. The equation (5) also hold for composite Gaussian membership function as long as (7) is employed in equation (5).

There are other possible special-purpose membership functions. But it suffices to describe the common ones because they are the more frequently encountered.

The Fuzzy-logic Neuron

We assume that fuzzy variables interact *via* the minimum ($\wedge$) and the maximum ($\vee$) operator [2]. *Fuzzy relations* are functions defined on fuzzy operators. For example, *R* ;*R*2;*R*3;... may be defined as:

$$\mathrm{R}_1 = \underset{i}{\vee}\left\{X_{1(i)} \wedge Y_{(i)}\right\}; R_2 = \underset{i}{\vee}\left\{X_{2(i)} \wedge Y_{(i)}\right\}; R_3 = \underset{i}{\vee}\left\{X_{3(i)} \wedge Y_{(i)}\right\} \tag{8}$$

where $i = 1, 2, 3, \ldots$ and $n = 0, 1, 2, \ldots$

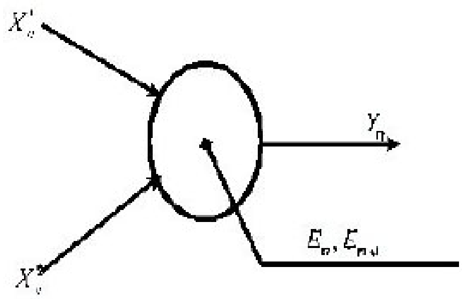


Fig. (1). A schematic of a dynamic fuzzy-logic neuron.

CHAPTER 4

Fundamental Algorithms and Methods

Abstract: An algorithm is a sequence of operations that is used to find a solution to a problem. The sequence of the operations are often well organised so that it may be representable as a flowchart or state-machine when possible. The first few sections of chapter 4 illustrate this by way of clustering algorithm and nature-inspired algorithms. Having laid the fundamental background of artificial neural networks (ANN) in previous chapters, in terms of definitions, theorems, and equations, it is now possible to organise one or more of these elements in such a way that it provide intelligent solution to some problems. The organisation of the elements has led to an attempt to give a formal definition of ANN in this chapter. Important and notable organisation of concepts and definitions may both help in reformulating problems as well as providing solution for them. The suitability of these solutions may be accessed by placing one or more performance metrics on the corresponding ANN as shown in the last section of this chapter. Such is the theme of this chapter.

Keywords: Allele, Chromosome, Covariance matrix, Cross-over, Deoxyribonucleic acid (DNA), Edges, Evolution, Fitness function, Flowchart, Genetic Algorithm (GA), Goodness-of-fit, Hypothesis testing, Mean-Square Error (MSE), Mutation, Nodes, Offspring, Principal Component Analysis (PCA), Ribonucleic acid (RNA), Selection, Walk.

INTRODUCTION

Chapter 4 begins with an introduction to density-based clustering algorithm.This is because density-based clustering algorithm is a clustering algorithm that may be developed from principle. The second section introduces evolution and nature-based algorithm like the genetic algorithm. The principle behind these algorithms lies in "survival-of-the-fittest" and "natural selection" from population genetics. A section on network method of analysis by using nodes and edges followed. Method of graphical analysis may be employed effectively to describe neural networks which are simple and completely tractable. Since it appears that sufficient introductions have been given in previous chapters, it is now adequate

Pierre Lorrentz

to formally define what an ANN system may be. So that the fifth section attempts to describe what intelligence may be and also formally define what an ANN system may be. Taking the definition of an ANN system for granted and designing an ANN system based on it, the performance of such an ANN system may be accessed as described in the last section of this chapter. Most sections of chapter three are completely introductory, and are meant to prepare the reader for subsequent chapters where they may be explained in detail and most definitely find use.

The methods of nodes and edges may find employment in explaining (away) most networks of other chapters *e.g.*; chapters 6, and 7. Most ANN systems are driven by their dynamics and so the definition of section four may apply to networks of chapter 6, 7, and 9 for example. Similarly, the performances of any ANN systems of subsequent chapters may be described by any of the concepts explained in the last section of chapter four.

Thus chapter 4 may have hopefully laid a fundamental background to ANN system design and analysis. And if not, it has nevertheless contributed, as one of the introductory chapters, to the principles on which lies the design and analysis of ANN systems in subsequent chapters.

DENSITY BASED ALGORITHMS: CLUSTERING ALGORITHMS

The concept of density modelling and mixtures of density modelling is hereby introduced. Density modelling is the usage of a probability density function (p.d.f.) in Bayes theorem [of chapter 2], this has very many implications and applications. If a p.d.f. is expressed as a linear combination of basis function in Bayes theorem, it is known as mixtures of density models or simply mixture models. In Bayes theorem, the unconditional probability $p(x)$, or the conditional probability $p(j|x)$, or both may be replaced by an equivalent expression which have been obtained from a given data. These equivalent expressions often represent a distribution over the given data. There are very many advantages derived from using a density model of given data, or a given problem in Bayes theorem as compared to any other alternatives. If in $p(x)p(j|x) = p(j)p(x|j)$, the $p(x)$ consist of M basis functions, then

$$p(x) = \sum_{j=1}^{M} p(j)p(x \mid j) \qquad \textbf{(1)}$$

have introduced a mixture into Bayes theorem. The p(j) is known as the mixing coefficient. The p(j|x) is the density function. Since p(j) is the prior probability (a probability term), it follows that:

$$\sum_{j=1}^{M} p(j) = 1; \text{ and } 0 < p(j) \leq 1 \qquad \textbf{(2)}$$

Recalling that by definition of a p.d.f.,

$$\int_{-\infty}^{\infty} p(x \mid j)dx = 1 \qquad \textbf{(3)}$$

From Bayes theorem therefore;

$$p(j \mid x) = \frac{p(x \mid j)p(j)}{p(x)} = \frac{p(x \mid j)p(j)}{\sum_{j=1}^{M} p(x \mid j)p(j)} \qquad \textbf{(4)}$$

One often seek p(x|j) from a given data, directly or indirectly. The next step is to ask what form of a density function does the given data satisfy, or what is the form of distribution over the data if any. This information may be obtained from the data by sampling and estimating both the mean μ and variance (or covariance) σ^2 of the data. The data mean μ and variance σ^2 are the main features of data, as the data can then be described by its distribution and its density function. A distribution or a density function is in turn described by its mean μ vector and its covariance σ^2 matrix. For a Gaussian distribution, there are three main density functions. Other possible density functions are mixture of these density functions in various degrees.

1. When the covariance matrix is a scalar multiple of identity matrix, then the density function is a “spherical” density function; *i.e.*,

CHAPTER 5

Quantum Logic and Classical Connectivity

Abstract: The aim and objectives of this chapter is to introduce the principles and fundamental concepts of elements which are used as building block of new and emerging neural systems. The first section introduces quantum logic and quantum algebra. Only those concepts relevant to ANNs' gates production are introduced, and as such the mathematics involved has been kept to a minimum. The second section introduce a new non-volatile memory elements; the memristance. Since these phenomena are well known prior to their formal discovery (formulation), the fundamental concepts and the phenomena forms the subject of discussion in this chapter. The chapter has striven to be self-contained and brief in order to describe and impart information relevant to understanding of the ANNs' elements.

Keywords: Adjoint, Bell state, Cauchy-Schwarz inequality, C-NOT, Conjugate, Eigenvalue, Eigenvector, EPR pair, Flux linkage, Hadamard gate, Hermittian, Memristance, NAND, Orthonormal, Pauli matrices, Quantum gate, Qubit, Qu-NOT, Universal gate, XOR.

INTRODUCTION

The first section of chapter five describes quantum gates and quantum algebra. Both are described together so as to facilitate q-gates' production, in a CAD tool for example, and to promote understanding of the principle behind their functions. Instantiating the q-gates in an ANN circuitry may automatically render such an ANN system a quantum ANN system. The second section introduce memristance as the latest non-volatile memory element which is much more robust as compared to other alternatives in maintaining weights of an ANN system at desired values, with minimum drift.

Quantum gates and quantum algebra are related directly to chapter nine where quantum expert systems are described. They may also be related implicitly to other neural systems of other chapters when their equivalent quantum neural system is formulated and implemented. Similarly, memristance of the second

Pierre Lorrentz

section of this chapter is related to chapters 10 and 11 where a non-volatile memory element is essential. Memristance is also an advantage to other ANN systems of other chapters when weight drift is considered.

Looking into the future of ANNs' development, this chapter has achieved its aim by introducing these gates and primitives.

QUANTUM LOGIC AND QUANTUM MATHEMATICS

A quantum bit (also called *qubit*) is a mathematical unit of quantum information and quantum computation. A *qubit* has two stable and measurable state $|1\rangle$ and $|0\rangle$. The sign "$|\ \rangle$" is used to indicate that the state "1" and "0" are not permanent and many other states are also likely. Some of other possible state may be a linear combination of $|1\rangle$ and $|0\rangle$ thus:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

Where α and β are real parts of a complex number of type: $z = xi + y$; (here $\alpha = x$ *and* $y = \beta$). By superposition of states $|1\rangle$ and $|0\rangle$ as in equation (1) above, an arbitrary $|\psi\rangle$ state may be derived by linear combination. The fundamental measurable states $|1\rangle$ and $|0\rangle$ forms an orthonormal basis of the vector space, and are known as computational *basic state*. However, in contrast to a classical bit, any particular state of a qubit cannot be determined accurately. That is, it is impossible to determine α and β accurately in equation (1). But a prolonged observation may reveal that only two states $|1\rangle$ and $|0\rangle$ can be determined with probabilities $|\alpha|^2$ and $|\beta|^2$ respectively such that

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2}$$

Thus a qubit is a unit two-dimensional vector space.

Prolonged observations made on qubit by several researches have revealed that a qubit has the characteristics of probability theories and not that of laws of physics. For example, equation (2) is always true, and its validity is irrespective of how $|\alpha|^2$ and $|\beta|^2$ is determined. Equation (1) may also be expressed in a more useful form as:

$$|\psi\rangle = e^{i\gamma}\left(\cos(\frac{\theta}{2})|0\rangle + e^{i\varphi}\sin(\frac{\theta}{2})|1\rangle\right) \qquad \textbf{(3)}$$

where θ,φ,γ are real numbers.

Equation (3) gives more explanation about the movement of qubit on a (unit) three-dimensional (3-D) sphere. The qubit $|\psi\rangle$ possesses certain probability of being in a continuum of states around the 3-D sphere. However, measurement attempts (of θ,φ,γ) is an interaction and cause changes in a state of qubit such that only two states are measurable, which are $|0\rangle$ with *probability amplitude* $|\alpha|^2$ and $|1\rangle$ with *probability amplitude* $|\beta|^2$. Thus it may be inferred that a single qubit is capable of expressing and representing connection weights between 0 and 1 in an ANN system irrespective of how complicated and large the ANN system might be. For multiple qubits, the number of possible state increase in respect of probability theory. As an example, two qubits have $2^2 = 4$ stable states 00,01,10,11. The linear combination of these four states by superposition gives

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \qquad \textbf{(4)}$$

Where $|\psi\rangle$= an arbitrary probable state. And

$$\sum_{i,j}\left|\alpha_{i,j}\right|^2 = 1;\ i,j \in [0,1]; \qquad \textbf{(5)}$$

$|\alpha_{i,j}|^2$ are corresponding probabilities of the states i, j. The $|\alpha_{i,j}|$ is sometimes simply called amplitude. For an n-qubit, the computational basic states are $|x_1,x_2,...,x_n\rangle$, and require 2^n amplitudes to express a quantum state.

Quantum Gates (Primitives)

Similar to classical binary logic, operations of a quantum logic gate (qugate) may be expressed by truth table. Since a single qubit has two measurable states $|0\rangle$ and $|1\rangle$, a NOT qugate may perform: $X|0\rangle \rightarrow |0\rangle$ and *vice versa*. Recalling the

CHAPTER 6

Learning Methods

Abstract: The first chapter of part II of this book introduces various common learning algorithms. The aim of chapter 6 is to acquaint the readers with the present-day knowledge in learning paradigms. Filters may be employed in implementation of learning algorithms, and *vice versa*. As such, the first few sections introduce Adaptive Linear Neuron (ADALINE) and recursive Least-Square (RLS) algorithms. Artificial intelligent systems may possess functional characteristics of living biological brain. The multi-agent network and neuromorphic network introduced in subsequent sections are examples of ANN systems with functional characteristics of living biological brain. The ability of the brain to process data is unparalleled; the human research efforts have however been able to discover a close match in Bayesian networks such that more than half of this chapter is devoted to presenting various types of probability-density-based learning algorithms. This is followed, in conclusion, by a hybrid neuro-fuzzy neural network section. By reading this chapter, one may fully understand the common ANN systems, and thus easily implement an ANN if required.

Keywords: Adaptive Linear Neuron (ADALINE), Adaptive Network-based Fuzzy Inference System (ANFIS), Agent, Capacitance, Efficacy, Expectation-Maximization (E-M) algorithm, Generative Topographic Mapping (GTM), Hodgkin-Huxley model, K-means, Knowledge base, Learning parameter, Membrane potential, Mixture models, Nodes, Radial Basis Function (RBF), Recursive Least-Square (RLS) algorithm, Sigmoid function, Sugeno-type Fuzzy system, Synaptic current, Weight matrix.

INTRODUCTION

The first section introduces one of the simplest types of neuron. These neuron exhibit main characteristics expected of a neuron. It is followed by the second section which illustrates also one of the simplest types of a learning algorithm. The Recursive Least-Square (RLS) algorithm possesses main characteristics of a learning algorithm. The third section describes the multi-agent network, an

Pierre Lorrentz

extended type of nature-inspired network. These three sections comprise a suitable start-off of ANN systems; for these reasons the three sections are on the first chapter of part II. The fourth section describes neuromorphic neural networks which closely mimic the functionality of a natural biological neuron. Because sufficient theory has been given in previous chapters, it is here suitable to introduce Bayesian neural networks in the fifth section. The last section presents the neuro-fuzzy system. This is because sufficient description of neuron and fuzzy logic has already been given in previous chapters.

Linear neuron may find use as a component of hierarchical neural networks of chapter 8. So also could RLS of the second section be employed in hierarchical architecture of chapter 8. Multi-agent networks of section three are better designed by using the graphical method (method of nodes and edges) of chapter 4. Genetic algorithm of chapter 4 could be an active agent at a node of a multi-agent network; so could a multi-layered perceptron of chapter 7. The best classical primitive (gate) suited to the implementation of the neuromorphic network of the fourth section of this chapter is the memristance of chapter 5. The neuromorphic network of chapter 6 is partially complete; it may be completed as discussed in chapter 11. The probability theory of chapters 2, form background studies to Bayesian networks of chapter 6. The Bayesian networks described here are standard networks which may be employed as components of a hierarchical mixture of experts that is described in chapters 8 and 9. Fuzzy logic introduced in chapters 3 are introductory texts to the neuro-fuzzy system described in the last section of this chapter.

As the first chapter of part II, chapter 6 has successfully describe various types of ANN systems and give standard neural network(s) for each type of ANN system mentioned.

THE ADAPTIVE LINEAR NEURON (ADALINE)

Filter is one of the standard means of weight initialisation and maintenance, but is not discussed in previous section. It is assumed that the reader is familiar with filters, but there are two exceptionally different filters that come in various disguises. These are Least-Mean-Square (LSM), and Recursive-Least-Square

(RLS) filters. The LSM and RLS filters will be discussed in this section in relation to learning algorithms.

A connectionist model of neuron called Adaptive Linear Neuron (ADALINE) was developed by Widrow [1] in 1962. The weights in this model are maintained according to LSM algorithm. The LSM algorithm is derived from a change in the gradient of a system's function. The change in the gradient is measured by a change in weight of ADALINE's function. The measurement is formal, methodical, and therefore given the name of its discoverer Widrow-Hoff learning rule [1]. The LSM algorithm adjusts the weight at every iteration step by an amount proportional to the gradient of the cumulative error E(x) of the network:

$$E(x) = \frac{1}{2}(t-o)^2 \tag{1}$$

where

t = target signal

O = output signal

E = expected error

$\Delta w = \eta(\Delta_w E(w))$

where η = learning parameter (or forgetting factor).

This weight adjustment constitutes the Widrow-Hoff leaning rule. The LSM that does the main weight adjustment is presented. Rather than an ad-hoc presentation, a slightly modified form of LSM which has been applied to human eye-iris recognition [2] is presented below.

The LSM Algorithm: An adaptive filter consists of two parts, namely; a digital filter with adjustable coefficient, and an adaptive algorithm which is employed in the adjustment and modification of the coefficient of the filter. In this report, an adaptive filter along with Widro-Hopf Least Mean Square (LMS) algorithm [1] provides a fan-in to a weightless classifier. The adaptive filter is produced from

CHAPTER 7

Neural Networks

Abstract: This chapter describe various types of ANN systems in relative detail. It is the aim of the chapter to give descriptions of advanced ANN system in such a detail as to facilitate easy implementation. The first few sections are dedicated to the recent weightless neural networks. This is followed by a weighted neural system section. Two advanced Bayesian network are introduced subsequently. The last section of the chapter explains the dynamics of ANN and how ANN nay be evaluated. The chapter has given a relatively extensive description of typical advanced neural networks from various categories of ANN systems.

Keywords: Adjustment, Back-propagation, Boltzmann distribution, Conditional probability, Division, Enhanced Probabilistic Convergent Network (EPCN), Generalized Likelihood Ration Test (GLRT), Helmholtz Machine, Kernel function, Kullback-Leibler divergence, Merging, Minimum Description Length, Mixture Density Network (MDN), Multi-classifier, Multi-expert System, Multi-Layered Perceptron (MLP), Probabilistic Convergent Network (PCN), Random Access Memory (RAM), Squared error, Wald test.

INTRODUCTION

In sections 1 and 2, some weightless neural networks are described in considerable detail. Weightless networks are presented here because they form a good alternative to weighted classical neural networks and also less prone to noise. A stereo-type weighted network, the Multi-Layered Perceptron (MLP), is described in the third section. The MLP is included because of its robustness and popularity; it represents a good example of weighted neural networks. Section four describes more advanced types of Bayesian classifier. They are suitably introduced here because the usual types of Bayesian classifiers have been described in chapter 6. The last section of chapter 7 presents the dynamics of an ANN system, and discussed the fusion mechanism of hierarchical network. This is followed by methods of independent evaluation of ANN systems.

Pierre Lorrentz

The first and second sections of chapter 7 show ANN systems whose learning and recognition algorithms are derived from Boolean logic.. The PCN and EPCN may be employed in selection mechanism described in chapter 8. Seeking minimal sets of weights by MLP may be synonymous to seeking a set of (minimal) basis functions. Whatever the structural architecture of MLP that has been determined may be implemented by using the classical primitives (gates) of chapter 5. The MLP can be used as a component neural network of neuro-fuzzy system of chapter 6. The probability theories of chapters 2 may have provided sufficient background principle to the Bayesian networks of the fourth section of chapter 7. Any of the Bayesian network may participate in the selection mechanism of chapter 8. The last section of this chapter may be regarded as a continuation of performance evaluation methods that has been introduced in chapter 4. The performance evaluation mechanism of the last section is algorithmic and in considerable detail, whereas that of chapter 4 is introductory. Most ANN systems of other chapters may be evaluated for performance by using the methods of the last section of chapter 7.

Chapter 7 has described large number of standard ANN systems in considerable detail.

WEIGHTLESS NETWORKS

This section introduces a neural network whose functionality depends essentially on Boolean logic.

Probabilistic Convergent Network (PCN)

Many prediction problems and pattern recognition problems can be solved by performing Boolean logic on them. In situations whereby prediction or recognition problems can be interpreted in terms of Boolean logic, a type of random access memory (RAM) based network called Probabilistic Convergent Network (PCN) becomes suitable. An added advantage of PCN over existing RAM-based network is the inclusion of confidence measure.

To carry out a logic representing, it is anticipated that all inputs be condensed to threshold image. Due to the complexity of architecture and function of PCN, some terminologies are worth introducing which will be used throughout this section. They are explained below

Binary inputs:- The Probabilistic Convergent Neuron (PCN) accepts as input, binary images only. Any input data is threshold-binarized and appropriately resized so that PCN may make sense of the data.

Compound Symbol:- Symbols are used to denote the neuron output of PCN. The architecture of PCN is shown in (Fig. **1**). Neuron outputs are inherently restricted to a small set of symbols often only "1" and "0". These are set of symbols often called base symbols. To extend the size of symbols which a neuron may employ, other symbols may be presented. For instance, if classes are labelled with numbers 0 – 9, the same set of characters 0 to 9 may be used also as additional symbol. This permit the storage and recall of those symbols which correspond with input class label. The RAM-based ANN concern is said to have used a compound symbol. As an illustration, a compound symbol that consist of class "5" and "6" may be "56" that is classes "5" and "6" might have been learned to the RAM-based ANN. The main difference in neuron output of PCN, as compared to other weightless nets, is the indication of frequency. The rate at which one class queries a certain memory is designated in PCN and Enhanced PCN (EPCN). For instance, if two classes querying a memory, if class "1" query the memory 75 times and class "2" query the same memory 25 times, the output of the RAM-neuron will be [75, 25]. If class "2" queries the memory 25 times and class "1" queries the memory 75 times, the result is: [25, 75]. Thus PCN and EPCN give the "probability" of incidence of every pattern class in a database.

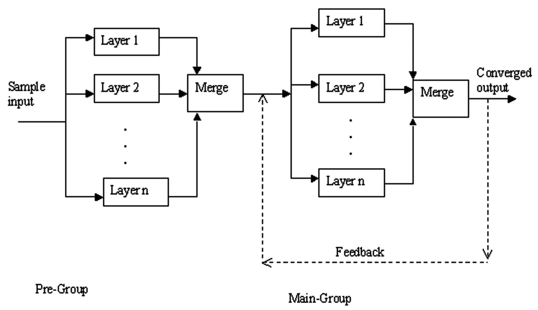


Fig. (1). A schematic representation of Probabilistic Convergent Network (PCN). This is an example of a RAM-based Neural Networks (NNs) [1].

CHAPTER 8

Selection and Combination Strategy of ANN Systems

Abstract: It is often required to select and combine two or more neural networks in order to process a given data. The aim and objectives of this chapter is to describe the selection and combination strategy of ANN systems. Two methods of ANNs' selection and combination that are derived from principle are described in detail. Manual selection and combination which is possible only if it involve few networks are not considered. Also not considered are heuristically determined set of networks, because of additional large experimentation that must be performed to select a suitable number and configuration of ANNs. These hindrances are relieved by the selection and combination strategy described in this chapter. The chapter has described two methods of selection and combination of ANNs which may be applied to minimize ANN's network errors. The selection and combination strategies descried in this chapter are principled, more robust, and of wider applicability than other alternatives.

Keywords: Classifier selection, Combiner configuration, Combiner engine, Combiner unit, Converter, Error – independent, Factorial selection, Fusion, Fuzzy – neuron, Group method, Interpreter, Kolmogorov-Gabor Polynomial, Main-group, Minimum complexity, Pool of networks, Pre-group, Statistical selection, Topology, Volterra series.

INTRODUCTION

The first section of chapter 8 describes the factorial selection of component classifier in an ensemble. Factorial selection solves class-dependent classification problems. The second section describes the group methods of selecting component classifier in an ensemble. Both methods may be developed from principle.

The probability theories of chapter 2 are fundamental to factorial selection strategy of the first section. The classifiers of chapter 6 and 7 may be component classifiers of the factorial selection strategy. Similar to the factorial selection, the

Pierre Lorrentz

group method accepts any classifier from chapter 5 and 6 provided the classifier achieves a performance beyond a set threshold. The threshold may be ascertained by one or more of the performances of chapter 4 and 7.

The selection and combination strategies of this chapter may be developed from principle and thus superior to those developed heuristically which are less tractable and may not be reproducible exactly.

The decision to include a Neural Networks (NN) in a Multi-Classifier System (MCS) is commonly referred to as *classifier selection*. Two selection strategies are well known. These are:

- The Direct method
- The "Pool of network" method.

The direct method is made up of neural network aggregate which are error-independent of each other. This is often an informed static selection made in advance. The "pool of network" method is a situation whereby an initial large number [1] of artificial neural systems are initialized, and in the course of experiments, smaller number (usually) of error-independent ANN have been selected by using measures such as error diversity measure.

Alternatively, the inclusion of a classifier in a hierarchy of classifiers may be divided into static and dynamic selection. Static selection mechanisms are those

methods employed in the selection of the based classifier which preclude alteration of composition during experiment. Dynamic selection mechanisms are meant to modify the composition of the ensemble during an experiment. There may be no fixed rule to this because, for example, a feedback of error correlation may convert a static method to a dynamic method.

FACTORIAL SELECTION

A factorial selection of base classifiers can either be dynamic or static depending on experimental setup. Customary selection method employed in choosing component classifiers can be grouped also as static or dynamic selection strategy which depends on whether there is a feedback system or not. Given $n = \sum_{i=1}^{r} n_i$ classes of r distinct types, where n_i are of type i and are otherwise indistinguishable, the number of permutation without repetition, of all n classes is:

$$M_m(n_1, \ldots, n_r) = \frac{n!}{\prod_{i=1}^{r} n_i!} \quad \textbf{(1)}$$

The number M_n is known as a multinormial coefficient. The multinormial coefficient M_m is a closed-form of the coefficient of equation (23) and/or equation (26) of chapter three. A special case is when r = 2. This is a case of binomial coefficient and is denoted by $M_n n_1, \ldots, n_n = {}_nC_r$ where;

$$\frac{n!}{r!(n-r)!} = {}_n C_r \quad \textbf{(2)}$$

Random variables X and Y are independent if, for all x and y;

$$f(x, y) = f_X(x) f_Y(y) \quad \textbf{(3)}$$

where f = the experimental outcome.

f_X = experimental outcome of X;

f_Y = experimental outcome of Y;

x = an element of X;

y = an element of Y.

The statistical selection method [2] exemplified the factorial selection method. An outline of the method is as follows.

CHAPTER 9

Probability-based Neural Network Systems

Abstract: Since Gaussian distribution may be employed as a universal approximator, it is clear that most modelling and optimisation problems could be solved by probability-based ANN systems. For this reason, chapter 9 concentrate on probability-based ANN systems. The first section introduces the random number generator, which has application in Markov-Chain and its hybrid, in subsequent sections. The fifth section describes the Restricted Boltzmann Machine (RBM) in detail. The Boltzmann machine may be a component network of Deep Belief Networks (DBN), which is described in the last section. The chapter has explained many algorithms related to DBN with great intuition, as this may facilitate better understanding and therefore implementation.

Keywords: Annealed Importance Sampling (AIS), Boltzmann machine, Contrastive divergence, Detailed balance, Distribution, Dynamic architecture, Energy function, Ergodic, Gibbs, Hamiltonian, Markov chain, Metropolis-Hasting criteria, Molecular dynamics, Momentum heat-bath, Partition function, Pseudo-random number, Random number, Sampling, Stationary distribution, Time-reversible, Verlet integrator.

INTRODUCTION

Chapter 9 is essentially a description of Bayesian ANN systems. This is because a very large number of ANN systems could be described from probability density consideration and one chapter may not be sufficient to describe them all. The first section describes a random number generator which is required by subsequent sections. For example, Markov chain of the second section may require a random number generator to describe a stochastic sampled path. Subjecting the random walk of a Markov chain to a dynamics gives a hybrid Markov chain of the third section. The first four sections may be essential in order to understand other ANN systems introduced in other sections that follow. The first system that follows is the restricted Boltzmann machine of the fifth section. The (restricted) Boltzmann machine is in the sixth section suitably introduced because it has distinguished

Pierre Lorrentz

itself as probably the most flexible ANN system. It demonstrates a wide range of architectural possibility and more than two learning and/or recognition algorithms may be employed in combination. This possibility may not be commonly found in other ANN systems. An illustration of this is seen in the last section of chapter 9 where deep believe network is introduced.

The random number generator of the first section of chapter 9 may be employed in the generation of uniform distribution, Gaussian distribution, and other distributions. It is then possible to sample from these distribution by Markov process of the second section and/or hybrid Markov chain of the third section of this chapter. Results of hybrid Monte Carlo of chapter 11 may be employed in the Markov chain (or it hybrid) of chapter 8. The (restricted) Boltzmann machine may be developed from scratch by starting from the multinomial series or Generalized factorial of chapter three. Both the (restricted) Boltzmann machine and the hierarchical neural system derivable from it, called the deep believe network, may be employed in industrial framework as explained in chapter 10. Assuming a new integrator is discovered from chapter 11, it may be tested on the Boltzmann machine of chapter 9 and/or in an industrial perspective (the third section) of chapter 10. In any case, the deep believe network of the last section of chapter 9 may be employed in industrial perspective as described in chapter 10.

This chapter has introduced some advanced Bayesian networks and explained their algorithms.

RANDOM-NUMBER GENERATORS

If we sample from a random variable x_i, i = 1, 2, 3,… un-correlated values over certain interval, from a prescribed distribution *e.g.*, uniform distribution, we have generated “random” numbers. For these numbers to be truly random they need to come from a prescribed distribution; but for the numbers to be random only over a specified range, the distribution from which the numbers are sampled from should be close to an ideal true distribution. When the distribution from which the random sample originates is close to the true distribution, and x_i values are such that:

$$x_{i+1} = \alpha x_i + b \pmod{n} \tag{1}$$

then the samples generated are pseudo-random. The device (or algorithm) *i.e.*; implementation of equation (1) that generates the pseudo-random numbers is a pseudo-random-number generator. Values for α and b are constant and positive. The value n determines the ranges (1,n) of random numbers. This (pseudo-) random number generator is suitable for most practical purposes, more so because data samples are normally always processed. The essential constraint to be satisfied is that the distribution from which x_i is sampled from should be arbitrary close (if not exactly equal) to the distribution of interest, over the range (1, n). Dividing the range (1,n) by n gives $\left(\frac{1}{n},1\right) \approx (0,1)$ ≈ (0,1) where n is a sufficient positive large number. So that by a simple division by n, a uniform N(0,1) distribution has been sampled from. From henceforth, a pseudo-random number of the sort described here will be taken as a random number. Also, a corresponding pseudo- random number generator will be assumed throughout as a random-number generator. This is what is normally required in practice. The range of x_i is also sometimes called the period of the x_i random variable. It is essential, for reproduction of random values, to select a number called *seed* around which the random values are produced. To reproduce a specific set of random numbers, set the range (1,n), α, and the corresponding seed of the generator. By using a random-number generator to sample from a distribution, the distribution of choice is said to be sampled at random.

MARKOV CHAIN

If a distribution exist for which no direct access is possible quantitatively, an ideal approach is to sample from the distribution until a sufficient sample is obtained. Taking random samples from partly unknown distribution π(x) is known as Markov Chain (MC) sampling. Taking a random sample from the posterior distribution of partly unknown distribution is called Markov Chain Monte-Carlo (MCMC) sampling. A process is Markov if a next state-space is dependent only on the previous state-space as:

$$\pi(x_{i+1}) = k\pi(x_i) \quad \textbf{(2)}$$

where the k is called the transition kernel of the Markov process, and $\pi(x_{i+1})$ is a target distribution.

CHAPTER 10

Emerging Networks

Abstract: Considering the possible availability of non-volatile memory element, chapter 10 has presented a memristive neural network in its first section. Generally, the chapter aims to present those neural systems that, at present, exist as proof of concept, some of which are undergoing industrial tests. For this reason, quantum neural networks are described in relative detail, and without losing its generality. A category of Bayesian network commonly known as Deep Belief Networks (DBN) is revisited, this time to illustrate their industrial tests. This chapter has provided some informational resources on the present state of ANNs′ development. It has also striven to bridge the gap between researches, developments, and applications.

Keywords: Amplifier, Bridge Circuit, Coupling Strength, Doublet Generator, Entanglement, Green Function, Hierarchical Bayesian Network, K_{th} phonon, Memristance, Probability Density, OFF-state, ON-state, Period, Polarization, Pulse, Q-dot, Schroedinger Equation, Synapses, Time-slice, Transconductance, Transistor.

INTRODUCTION

After having explored the meaning of memristance in chapter 5, the memristance is used in the first section of this chapter to compute synapsis. A quantum neural system based on quantum principles is described in the second section. This follows because methods of producing quantum gates have already been explained. The Deep Believe Network (DBN) of the last section of chapter 10 may be regarded as the application of DBN that is described in the previous chapter.

Chapter 5 has given sufficient description of memristance which is required for the synaptic computation of the first section of this chapter. Similarly, the sections that describe quantum logic and quantum algebra of chapter 5 contain sufficient information for qu-gate primitives which may be used for quantum expert system of the second section of this chapter. The Bayesian networks of chapters 7 and 9

Pierre Lorrentz

may be components of a deep believe network which in the third section of this chapter is set in industrial perspective.

Chapter 10 has explained emerging networks in considerable detail, and also describe how deep believe networks may be employed in industries.

MEMRISTIC NEURAL NETWORKS

The making of memristive (resistance based on memristor) primitives have been introduced in chapter 5. The design of a memristive artificial synapse is introduced here by using the memristor primitives of chapter 5. The memristive artificial synapses consist of five memristor and one differential amplifier. A ciruit composed with five memristances and one differential amplifier in order to generate a synapse, is termed memristive synaptic circuit. The memristive synaptic circuit described is able to set both positive and negative weights.

We assume that the reader is familiar with high-school or undergraduate Wheatstone Bridge, and amplifier. The first step is simply to replace the resistances of the bridge-circuit with memristances, and connect the circuit as shown in Fig. (**1**). Fig. (**1**) essentially functions as an artificial synaptic circuit as follows.

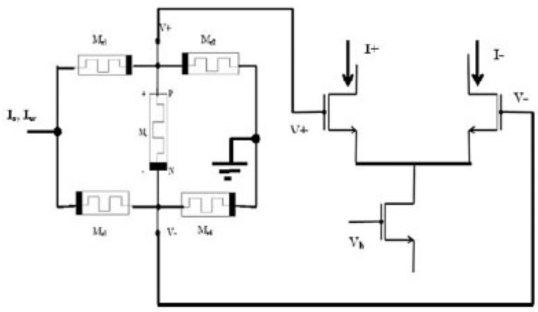


Fig. (1). A diagram to demonstrate the use of memristors in a synaptic circuit.

The memristance M_w at the centre is the weight memristance, and M_{r1}, M_{r2}, M_{r3}, and M_{r4} acts in pairs, similar to bridge circuit, to give the weight a sign. The sign is either positive (+) or negative (-). A positive signal (broken signal, Fig. (**2**)) increases memristances M_{r1} and M_{r4}, whereas the same pulse decreases the memristances M_{r2} and M_{r3}. This result in [M_{r1},M_{r4}] switching to ON-state and [M_{r2},M_{r3}] switching to OFF-state.So that current flows through M_{r1}, M_w, M_{r4} in that order, leading to increasing V+ value and decreasing V- value of voltages. A negative strong pulse (dotted signal, Fig. (**2**)) decreases the memristances M_{r1} and M_{r4}, whereas it increases the memristances M_{r2} and M_{r3}. This switches [M_{r1},M_{r4}] to OFF-state, and [M_{r2},M_{r3}] to ON-state, such that current flows in opposite direction through M_{r2}, M_w, M_{r3}, in that order.But when the conventional direction of left-t--right current flow is maintained, the voltage shown on Fig. (**1**) and Fig. (**2**) gets the opposite signs. The effect is a small positive voltage on the lower side of Fig. (**1**) and Fig. (**2**), and large negative voltage on the upper side of Fig. (**1**) and Fig. (**2**), giving a net negative voltage as shown on Table **1**.

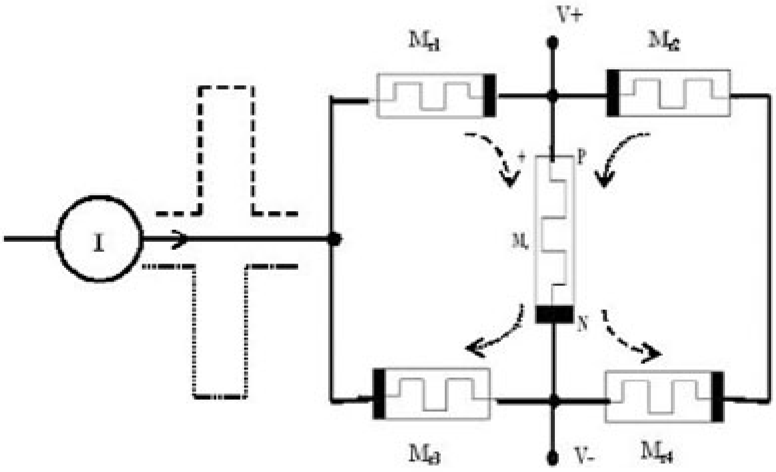


Fig. (2). The functionality of a memristive synaptic circuit.

Table 1. **A tabular explanation of the switching circuit.**

Applied wave-pulses	Memristance configuration	Current sign	Voltage sign [(V+)-(V-)]	Sign of weights M_w
Positive wave-pules (broken line; Fig. (**2**))	Mr1 = Mr4 = min.	+	+	+
	Mr2 = Mr3 = max.	-	-	
Negative wave-pulses (dotted line; Fig. (**2**))	Mr1 = Mr4 = max.	+	-	-
	Mr2 = Mr3 = min.	-	+	

CHAPTER 11

Research and Developments in Neural Networks

Abstract: Two categories of ANN systems are able to model any intelligent Expert in great detail. These are Bayesian network, and neuromorphic network. Both are hampered by lack of adequate resources and lack of human knowledge. Research and development on these two categories of ANN systems is the subject of this chapter. The first section of chapter 11, on Bayesian network, specifically describes Hybrid Monte Carlo (HMC) and associated algorithms, after which areas of possible researches are highlighted. The second section is on neuromorphic network. It presents the current state of industrial development. The chapter has taken care to omit those conceptual developments which may not be achievable in near future. Illustration of the recent neuromorphic design has been given in concluding the second section. The chapter has provided research and development information resources on two advanced ANN systems.

Keywords: Average error, CMOS, Comlex-conjugate Eigenvalue, Efficacy, Hamiltonian system, Harmonic oscillator, Hybrid Monte Carlo (HMC), Leapfrog algorithm, Long-Term Depression (LTD), Long-Term Potentiation (LTP), Markov-Chain Monte Carlo (MCMC), Multi-dimensional matrix, Omyleyan integrator, Pulse-Width Modulation (PWM), Random-number generator, Shadow Hamiltonian, Simplectic, Spike-Timing-Dependent-Plasticity (STDP) computation, Titanium dioxide, Verlet velocity.

INTRODUCTION

The first section of chapter 11 explore the possibility of improving hybrid Markov chain through search for more efficient integrator for hybrid Monte Carlo algorithm. The second section presents a complete neuromorphic network. Without this, the book may not have completed an industrial illustration of a neuromorphic network.

Section one of this chapter considers various types of integrators which may be employed in Bayesian networks of chapters 7 and 9. While the second section

Pierre Lorrentz

presents a general framework of neuromorphic network to which the euromorphic networks of chapter 6 and 5 applies.

This chapter has provided some ways of extending neuromorphic networks, and improving Bayesian ANN systems.

EXTENSION OF HYBRID MONTE CARLO

At each stage of the Markov chain HMC, a numerical integration of a Hamiltonian system of differential equation is required. The integration relies often heavily on Verlet's algorithm. Since much of the computation done by Markov chain HMC is the integration of the Hamiltonian system of differential equation, the efficiency and exact-ness of Verlet algorithm is of paramount importance. A single integration stage requires a step size h in Verlet integrator (see chapter 9 for details), which is expected to be sufficiently large and to promote stability of the HMC algorithm. We therefore would not seek an order of accuracy as h tends to zero. Rather h dictates a region of stability when inserted into a function that models the behaviour of the integrator of choice as the Markov chain progresses. This function may be denoted by f(h).

HMC may be defined as a Markov chain which samples from a probability distribution in $\mathbb{R}^d$ with a density function π *(v);*

$$\pi(v) \propto \exp(-V(v)) \tag{1}$$

where v = velocity, and *V (v);* = potential energy.

A kinetic energy term is added to equation (1). The kinetic energy term consist of momentum p and a mass matrix M in a Hamiltonian $H(v, p)$ energy of equation (2);

$$H(v, p) = \frac{1}{2} p^T M^{-1} p + V(v) \tag{2}$$

In view of the energy (2), another probability distribution $\prod(v,p)$ is also defined as:

$$\prod(v,p) = K.\exp(-H(v,p)) \tag{3}$$

$$\prod(v,p) = K.\exp\left(-\frac{1}{2}p^T M^{-1} p\right).\exp(-V(v))$$

The K of equation (3) is a scaling constant, whereas p and v are stochastic and independent. The momentum p has a Gaussian N(0, M) distribution. Markov chain sampling by HMC from $v^{(n)}$ and $p^{(n)}$ is expected to be symplectic (*e.g.*; volume preserving), and ergodic. This means that a map

$$\psi : (v^n, p^n) \rightarrow (v^*, p^*) \tag{4}$$

is such that

$$\det\left\|\psi'(v,p)\right\| = 1 \tag{5}$$

<u>*Hybrid Monte Carlo (HMC) Algorithm*</u>

1) Given: $v^{(0)} \in \mathbb{R}^d$; $N \geq 1$; $n = 0$

2) Sample $p^{(n)} \sim N(0,M)$; calculate $(v^*, p^*) = \psi(v^{(n)}, p^{(n)})$

3) Compute $a^{(n)} = \min\left(1, \exp\left(H(v^{(n)}, p^{(n)}) - H(v^*, p^*)\right)\right)$

4) Sample $u^{(n)} = U(0,1)$; if $a^{(n)} > u^{(n)}$; set

   $v^{(n+1)} = v^*$; otherswise set $v^{(n+1)} = v^{(n)}$

5) Set $u = u + 1$; if $n = N$ stop otherwise go to step 1.

Fig. (1). Normal HMC.

Hybrid Monte Carlo satisfies these criteria, and has been introduced in chapter 9.

SUBJECT INDEX

Pierre Lorrentz