LNCS 7824

Marco Tomassini
Alberto Antonioni
Fabio Daolio
Pierre Buesser (Eds.)

Adaptive and Natural Computing Algorithms

11th International Conference, ICANNGA 2013
Lausanne, Switzerland, April 2013
Proceedings

Springer

Marco Tomassini   Alberto Antonioni
Fabio Daolio   Pierre Buesser (Eds.)

Adaptive and Natural Computing Algorithms

11th International Conference, ICANNGA 2013
Lausanne, Switzerland, April 4-6, 2013
Proceedings

Springer

Volume Editors

Marco Tomassini
Alberto Antonioni
Fabio Daolio
Pierre Buesser

Université de Lausanne, Faculté des Hautes Etudes Commerciales
Département des Systèmes d'Information
UNIL-Dorigny, Bâtiment Internef, 1015 Lausanne, Switzerland
E-mail: {marco.tomassini, alberto.antonioni, fabio.daolio, pierre.buesser}@unil.ch

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-37212-4 e-ISBN 978-3-642-37213-1
DOI 10.1007/978-3-642-37213-1
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013933232

CR Subject Classification (1998): F.1-2, I.5, I.2

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

We are pleased to present in this LNCS volume the proceedings of the 11^{th} International Conference on Adaptive and Neural Computing Algorithms, ICANNGA 2013, that was held in Lausanne, Switzerland. The biennial ICANNGA series of conferences was started in 1993 in Innsbruck, Austria, and was followed by Ales, France (1995), Norwick, UK (1997), Portorož, Slovenia (1999), Prague, Czech Republic (2001), Rouen, France (2003), Coimbra, Portugal (2005), Warsaw, Poland (2007), Kuopio, Finland (2009), and Ljubljana, Slovenia (2011). The present edition thus marks the 20^{th} year of existence of this successful series of conferences.

We received 91 paper submissions for this edition coming from many different countries. Following an extensive review process, the Program Committee selected 51 manuscripts for inclusion in this volume. Of the 51 papers, 39 were presented in oral sessions and the rest as posters. The selected papers cover many aspects of soft computing techniques and adaptive algorithms, from artificial neural networks to evolutionary algorithms, system dynamics and identification, pattern recognition, machine learning techniques, and swarm computing among others. Both theoretical and fundamental contributions as well as applications were present, although theoretical and numerical models were the majority.

The conference featured three distinguished keynote speakers: Tom Heskes, Moshe Sipper, and Alessandro Villa. Their presentations were at the leading edge of today's research and of great inspirational value. Tom Heskes' talk was about Bayesian machine learning approaches to analyze complex data sets coming from the brain's functional data and their application to brain–computer interfaces, a really exciting perspective. Moshe Sipper focused on artificial intelligence techniques based on evolutionary computation within the domain of games, an activity in which Sipper's group has produced human-competitive and award-winning game strategies in games such as chess, checkers, and several others. Alessandro Villa's talk was on theoretical models of spatio-temporal patterns of recurrent neural networks based on dynamical systems theory. He showed that only selected meaningful patterns may contribute to extend the computational power of neural networks.

The success of a conference depends on the quality of the scientific contributions, as well as on the work of the reviewers and of the organizers. We are grateful to all the contributors for their hard and high-quality work that made the conference possible. And of course, we thank the reviewers for their time and careful work. We would also like to express our gratitude to the Advisory Committee, which guarantees the continuity of the conference series and provided advice, feedback, and discussion throughout. Finally, we thank the Economics Faculty of the University of Lausanne for the logistic support provided, which

proved very important in creating a nice and productive environment for all the conference activities.

April 2013

Marco Tomassini
Alberto Antonioni
Fabio Daolio
Pierre Buesser

Organization

Advisory Committee

Rudolf Albrecht
Bartlomiej Beliczynski
Andrej Dobnikar
Mikko Kolehmainen
Vera Kurkova
David Pearson
Bernardete Ribeiro
Nigel Steele

Program Committee

Hernan Aguirre
Jarmo Alander
Rudolf Albrecht
Alberto Antonioni
Rubén Armañanzas
Wolfgang Banzhaf
Miguel Arturo Barreto-Sanz
Fülöp Bazsó
Bartlomiej Beliczynski
Lubica Benuskova
Pascal Bouvry
Hans Albert Braun
Stefano Cagnoni
Paolo Cazzaniga
Francesco Cerutti
Francisco Chicano
Carlos Coello Coello
Ernesto Costa
Carlos Cotta
Fabio Daolio
Christian Darabos
Ivanoe De Falco
Matteo De Felice
Antonio Della Cioppa
Federico Divina
Andrej Dobnikar
Bernabe Dorronsoro
António Dourado
Francisco Fernández de Vega
Stefan Figedy
Alexandru Floares
Mario Giacobini
Michele Giugliano
Juan A. Gomez-Pulido
Barbara Hammer
Jin-Kao Hao
Ignacio Hidalgo
Osamu Hoshino
Lazaros Iliadis
Marcin Iwanowski
Juan Luis Jimenez Laredo
Martti Juhola
Paul Kainen
Mario Koeppen
Mikko Kolehmainen
Stefanos Kollias
Petia Koprinkova-Hristova
Jozef Korbicz
Vera Kurkova
Giancarlo La Camera
Kauko Leiviskä
Tom Lenaerts
Uros Lotric
Francesco Masulli
Julian Miller
Francesco C. Morabito
Alberto Moraglio
Juan Manuel Moreno
Ferrante Neri
Roman Neruda

Ernst Niebur
Stanislaw Osowski
Jorge Peña
Carlos Andrés Peña-Reyes
Andrés Pérez-Uribe
Clara Pizzuti
Riccardo Poli
Mike Preuss
Bernardete Ribeiro
Conor Ryan
Jorge Santos
Henrik Saxen
Marc Schoenauer
Roberto Serra
Catarina Silva
Moshe Sipper
Branko Ster
Thomas Stuetzle
Miroslaw Swiercz
Ryszard Tadeusiewicz
El-Ghazali Talbi
Tatiana Tambouratzis
Marco Tomassini
Leonardo Vanneschi
Miguel A. Vega-Rodríguez
Sébastien Verel
Alessandro Villa
Stefan Wermter

Organizing Committee

Alberto Antonioni
Pierre Buesser
Fabio Daolio
Elisabeth Fournier Pulfer
Marco Tomassini

Table of Contents

On Appropriate Refractoriness and Weight Increment in Incremental Learning

Toshinori Deguchi[1], Junya Fukuta[1], and Naohiro Ishii[2]

[1] Gifu National College of Technology
[2] Aichi Institute of Technology

Abstract. Neural networks are able to learn more patterns with the incremental learning than with the correlative learning. The incremental learning is a method to compose an associate memory using a chaotic neural network. The capacity of the network is found to increase along with its size which is the number of the neurons in the network and to be larger than the one with correlative learning. In former work, the capacity was over the direct proportion to the network size with suitable pairs of the refractory parameter and the learning parameter. In this paper, the refractory parameter and the learning parameter are investigated through the computer simulations changing these parameters. Through the computer simulations, it turns out that the appropriate parameters lie near the origin with some relation between them.

1 Introduction

The incremental learning proposed by the authors is highly superior to the auto-correlative learning in the ability of pattern memorization[1,2]. The idea of the incremental learning is from the automatic learning[3]. In the incremental learning, the network keeps receiving the external inputs. If the network has already known an input pattern, it recalls the pattern. Otherwise, each neuron in it learns the pattern gradually. The neurons used in this learning are the chaotic neurons, and their network is the chaotic neural network, which was developed by Aihara[4].

In former work, we investigated the capacity of the networks[5] and the error correction capability[6]. Through the simulations, we found that the capacity is in proportion to the network size with the appropriate parameter which is inverse proportion to the size and that the capability decreases gradually as the number of the learned patterns increases.

In this paper, first, we explain the chaotic neural networks and the incremental learning and refer to the former work on the capacities[7], then the refractory parameter and the learning parameter are investigated, with simulations changing the refractory parameter and the learning parameter counting the capacity of the network in the 100 neuron network.

M. Tomassini et al. (Eds.): ICANNGA 2013, LNCS 7824, pp. 1–9, 2013.

2 Chaotic Neural Networks and Incremental Learning

The incremental learning was developed by using the chaotic neurons. The chaotic neurons and the chaotic neural networks were proposed by Aihara[4]. We presented the incremental learning which provided an associative memory[1]. The network type is an interconnected network, in which each neuron receives one external input, and is defined as follows[4]:

$$x_i(t+1) = f\big(\xi_i(t+1) + \eta_i(t+1) + \zeta_i(t+1)\big) \ , \tag{1}$$

$$\xi_i(t+1) = k_s \xi_i(t) + \upsilon A_i(t) \ , \tag{2}$$

$$\eta_i(t+1) = k_m \eta_i(t) + \sum_{j=1}^{n} w_{ij} x_j(t) \ , \tag{3}$$

$$\zeta_i(t+1) = k_r \zeta_i(t) - \alpha x_i(t) - \theta_i(1 - k_r) \ , \tag{4}$$

where $x_i(t+1)$ is the output of the i-th neuron at time $t+1$, f is the output sigmoid function described below in (5), k_s, k_m, k_r are the time decay constants, $A_i(t)$ is the input to the i-th neuron at time t, υ is the weight for external inputs, n is the size—the number of the neurons in the network, w_{ij} is the connection weight from the j-th neuron to the i-th neuron, and α is the parameter that specifies the relation between the neuron output and the refractoriness.

$$f(x) = \frac{2}{1 + \exp(\frac{-x}{\varepsilon})} - 1 \ . \tag{5}$$

The parameters in the chaotic neurons are assigned in Table 1.

Table 1. Parameters

$\upsilon = 2.0$,
$k_s = 0.95$,
$k_m = 0.1$,
$k_r = 0.95$,
$\theta_i = 0$,
$\varepsilon = 0.015$

In the incremental learning, each pattern is inputted to the network for some fixed steps before moving to the next. In this paper, this term is fixed to 50 steps, and 1 set is defined as a period for which all the patterns are inputted. The patterns are inputted repeatedly for some fixed sets.

During the learning, a neuron which satisfies the condition (6) changes the connection weights as in (7)[1].

$$\xi_i(t) \times (\eta_i(t) + \zeta_i(t)) < 0 \ . \tag{6}$$

$$w_{ij} = \begin{cases} w_{ij} + \Delta w, & \xi_i(t) \times x_j(t) > 0 \\ w_{ij} - \Delta w, & \xi_i(t) \times x_j(t) \le 0 \end{cases} \quad (i \neq j) \ , \tag{7}$$

where Δw is the learning parameter.

If the network has learned a currently inputted pattern, the mutual interaction $\eta_i(t)$ and the external input $\xi_i(t)$ are both positive or both negative at all the neurons. This means that if the external input and the mutual interaction have different signs at some neurons, a currently inputted pattern has not been learned completely. Therefore, a neuron in this condition changes its connection weights. To make the network memorize the patterns firmly, if the mutual interaction is less than the refractoriness $\zeta_i(t)$ in the absolute value, the neuron also changes its connection weights.

In this learning, the initial values of the connection weights can be 0, because some of the neurons' outputs are changed by their external inputs and this makes the condition establish in some neurons. Therefore, all initial values of the connection weights are set to be 0 in this paper. $\xi_i(0)$, $\eta_i(0)$, and $\zeta_i(0)$ are also set to be 0.

To confirm that the network has learned a pattern after the learning, the pattern is tested on the normal Hopfield's type network which has the same connection weights as the chaotic neural network. That the Hopfield's type network with the connection weights has the pattern in its memory has the same meaning as that the chaotic neural network recalls the pattern quickly when the pattern inputted. Therefore, it is the convenient way to use the Hopfield's type network to check the success of the learning.

3 Capacity

In this section, we retrace the simulations in the former work[7]. In the former works[5,6,7], it turned out to be important for the incremental learning that the connection weights were reinforced by the effect of the refractoriness $\zeta_i(t)$ in the learning condition (6). This suggests that α is significant to the incremental learning. Therefore, we regarded not only Δw but also α as learning parameters, although the other parameters are not meant to have no effect.

First, the refractory parameter α was fixed to be 2.0 and the capacity was inspected changing Δw. The capacity is proportional to its size with the proportional constant 0.92 as shown in Fig. 1. The capacity of the auto-correlative learning is also shown in Fig. 1 and the proportional constant is 0.07.

Under consideration of the former works[5,6], we took $\Delta w = 0.0001$ and $\alpha = 0.1$ and used a 200 neuron network with 250 input patterns. Fig. 2[7] shows the result. The horizontal axis is the learning sets which means learning period and the vertical axis is the number of learned patterns at the end of that sets. The result shows that the capacity of this network is equal to or more than 250 patterns, which means the capacity of the network exceeds the direct proportion to its size.

Next, to investigate the usable pair of Δw and α, the following simulations were carried out. In these simulation, 200 neuron network was used with 240 input patterns and the number of learned patterns was counted changing Δw and α. Because the appropriate value of Δw was 0.005 in former work[5,6], the parameter Δw was changed from 0.0001 to 0.01 in increments of 0.0001 to cover

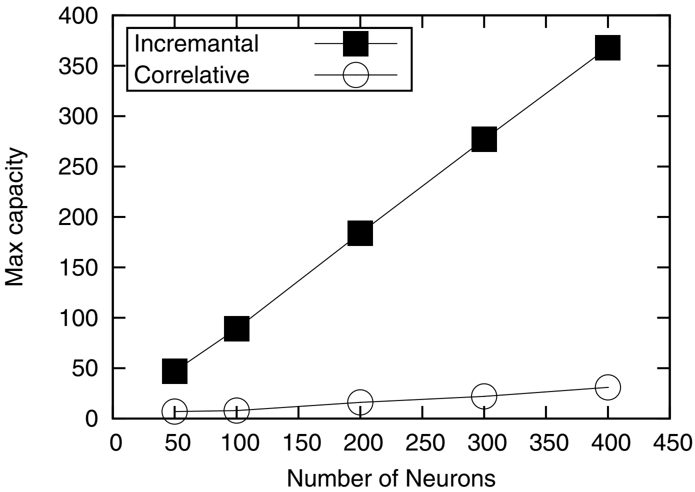

Fig. 1. Capacity of Network at $\alpha = 2.0$

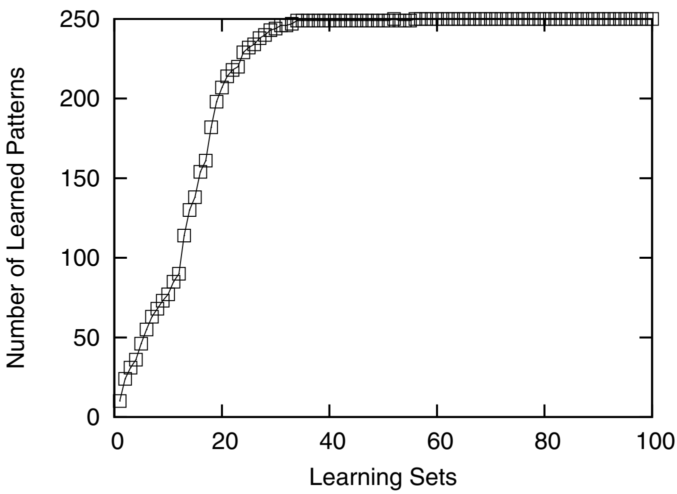

Fig. 2. Number of Learned Patterns in 200 Neuron Network

$\Delta w = 0.005$. The parameter α was changed from 0.01 to 2.00 in increments of 0.01 to cover $\alpha = 0.1$ which was used in Fig 2.

Fig. 3 shows the result of these simulations. From this result, not only Δw but also α strongly affects the number of learned patterns. Around $\Delta w = 0.0011$ and $\alpha = 0.41$, all the 240 input patterns are learned.

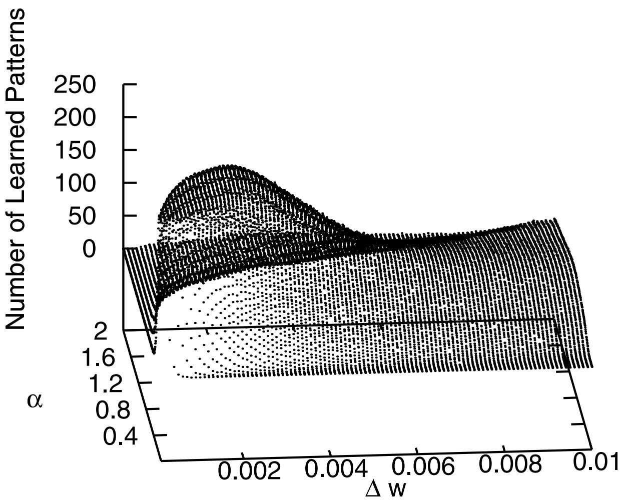

Fig. 3. Number of Success with α and Δw

4 Appropriate Pairs

In this section, we investigate the appropriate pair of Δw and α, with which the network was able to learn all the inputted patterns. To reduce computational complexity, we took the 100 neuron network in this section. The refractory parameter α is changed in increments of 0.01 and the learning parameter Δw is changed in increments of 0.0001.

Through the simulations, the capacity turns out to be 131 with α and Δw listed in Table 2. Fig. 4 shows the result of the simulation with $(\alpha, \Delta w) = (0.02, 0.0001)$. Both axes are the same as the ones in Fig. 2. Because the appropriate pairs listed in Table 2 are located in low area of Δw, the simulations have to be carried out with finer values of Δw in this area.

In the next simulations, the refractory parameter α is changed from 0.001 to 0.2 in increments of 0.001 and the learning parameter Δw is changed from 10^{-6} to 10^{-4} in increments of 10^{-6}. In these simulations, the capacity of the network grew to 145. Fig.5 shows the number of learned patterns when 145 patterns are inputted to the network. The appropriate pairs are plotted with large dots, so that the area of these pairs looks blacker than the other. In Fig.5, the appropriate pairs are lined in an area near $\alpha = 0.01$.

From the result, next simulations are carried out with smaller parameters. In these simulations, the refractory parameter α is changed from 10^{-5} to 10^{-3} in increments of 10^{-5} and the learning parameter Δw is changed from 10^{-7} to 10^{-5} in increments of 10^{-7}. Again, the capacity of the network grew to 149. Fig.6 shows the number of learned patterns when 149 patterns are inputted to the network. The appropriate pairs are lined in an area around $\alpha = 180\Delta w$.

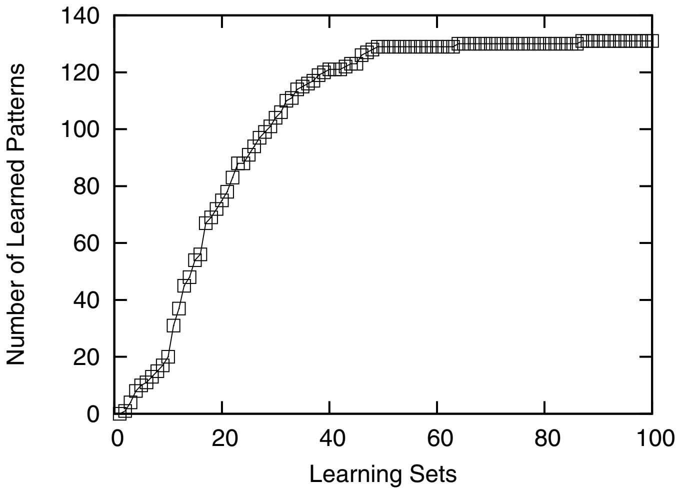

Fig. 4. Number of Learned Patterns in 100 Neuron Network with $\alpha = 0.02$ and $\Delta w = 0.0001$

Table 2. Appropriate α and Δw in the 100 Neuron Network

α	Δw
0.02	0.0001
0.03	0.0002
0.04	0.0002
0.05	0.0003
0.06	0.0003
0.07	0.0004
0.08	0.0005

From above results, the capacity of the network is increasing as the parameters are decreasing. Then, for the third time, the simulations are carried out, with the refractory parameter α changed from 4×10^{-7} to 4×10^{-5} in increments of 4×10^{-7} and with the learning parameter Δw changed from 10^{-9} to 10^{-7} in increments of 10^{-9}.

These simulations revealed that the capacity of the network is still 149. Fig.7 shows the number of learned patterns when 149 patterns are inputted to the network. The appropriate pairs are lined in an area around $\alpha = 193\Delta w$. From the fact that these parameters are two digits smaller than the ones in Fig 6, and that the capacities are the same, 149 can be the maximum capacity of the network.

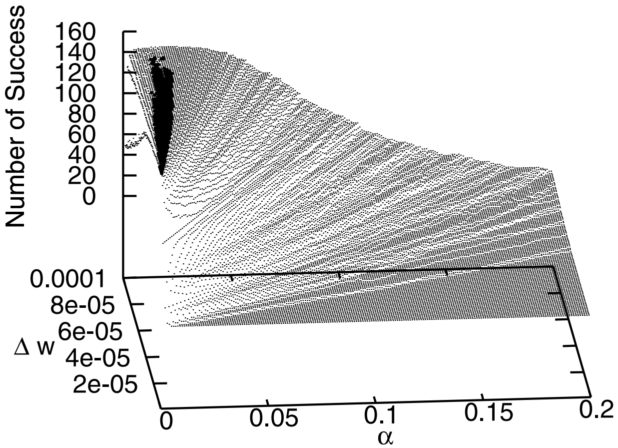

Fig. 5. Number of Learned Patterns in 100 Neuron Network with $\alpha = 0.001$ to 0.2 and $\Delta w = 10^{-6}$ to 10^{-4}

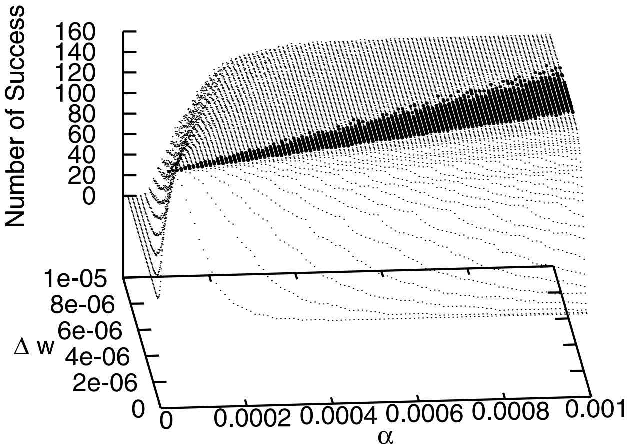

Fig. 6. Number of Learned Patterns in 100 Neuron Network with $\alpha = 10^{-5}$ to 10^{-3} and $\Delta w = 10^{-7}$ to 10^{-5}

For α and Δw, there are appropriate pairs of these parameters, which are in an area around $\alpha \simeq 190\Delta w$ near the origin. We consider that smaller Δw means finer tuning of the connection weights, that smaller α is needed for smaller Δw to keep the balance in the condition (6), and that thus the appropriate pairs lie near the origin. Because the capacity can change with the parameters, to use the maximum capacity of the network, it is important to verify the these parameters which tend to be near the origin with the relation of $\alpha \simeq 190\Delta w$.