Natural Computing Series

For further volumes:
http://www.springer.com/series/4190

Daniel S. Yeung · Ian Cloete · Daming Shi ·
Wing W.Y. Ng

Sensitivity Analysis for Neural Networks

Springer

Authors

Prof. Daniel S. Yeung
School of Computer Science
and Engineering
South China University of Technology
Wushan Rd.
TianHe District
Guangzhou, China
danyeung@ieee.org

Prof. Daming Shi
School of Electrical Engineering and Computer Science
Kyungpook National University
Buk-gu, Daegu
South Korea
asdmshi@ntu.edu.sg

Prof. Ian Cloete
President
Campus 3
International University in Germany
76646 Bruchsal, Germany
ian.cloete@i-u.de; president@i-u.de

Dr. Wing W.Y. Ng
School of Computer Science
and Engineering
South China University of Technology
Wushan Rd.
TianHe District
Guangzhou, China
wingng@ieee.org

ISSN 1619-7127
ISBN 978-3-642-02531-0 e-ISBN 978-3-642-02532-7
DOI 10.1007/978-3-642-02532-7
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2009938187

ACM Computing Classification (1998): I.2.6, F.1.1

Cover design: KuenkelLopka GmbH

Printed on acid-free paper

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

Preface

Neural networks provide a way to realize one of our human dreams to make machines think like us. Artificial neural networks have been developed since Rosenblatt proposed the Perceptron in 1958. Today, many neural networks are not treated as black boxes any more. Issues such as robustness and generalization abilities have been brought to the fore. The advances in neural networks have led to more and more practical applications in pattern recognition, financial engineering, automatic control and medical diagnosis, to name just a few.

Sensitivity analysis dates back to the 1960s, when Widrow investigated the probability of misclassification due to weight perturbations, which are caused by machine imprecision and noisy input. For the purpose of analysis, these perturbations can be simulated by embedding disturbance into the original inputs or connection weights. The initial idea of sensitivity analysis was then extended to optimization and to applications of neural networks, such as sample reduction, feature selection, active learning and critical vector learning.

This text should primarily be of interest to graduate students, academics, and researchers in branches of neural networks, artificial intelligence, machine learning, applied mathematics and computer engineering where sensitivity analysis of neural networks and related concepts are used. We have made an effort to make the book accessible to such a cross-disciplinary audience.

The book is organized into eight chapters, of which Chap. 1 gives an introduction to the various neural network structures and learning schemes. A literature review on the methodologies of sensitivity analysis is presented in Chap. 2. Different from the traditional hypersphere model, the hyper-rectangle model described in Chap. 3 is especially suitable for the most popular and general feedforward network: the multilayer Perceptron. In Chap. 4, the activation function is also involved in the calculation of the sensitivity analysis by parameterizing. The sensitivity analysis of radial basis function networks is discussed in Chaps. 5 and 6, with the former giving a generalization error model whereas the latter concerns optimizing the hidden neurons. In Chap. 7, sensitivity is measured in order to encode prior knowledge into a neural network. In Chap. 8, sensitivity analysis is applied in many applications, such as dimensionality reduction, network optimization and selective learning.

We would like to express our thanks to many colleagues, friends and students who provided reviews of different chapters of this manuscript. They include Minh

Nhut Nguyen, Xiaoqin Zeng, Patrick Chan, Xizhao Wang, Fei Chen and Lu He. We often find ourselves struggling with many competing demands for our time and effort. As a result, our families, especially our beloved spouses, are the ones who suffer the most. We are delighted to dedicate this work to Foo-Lau Yeung, Wilma Cloete and Jian Liu.

It is with great humility that we would like to acknowledge our Good Lord as the true creator of all knowledge. This work is the result of our borrowing a small piece of knowledge from Him.

8 July 2009

Daniel S. Yeung
Ian Cloete
Daming Shi
Wing W.Y. Ng

Contents

Chapter 1
Introduction to Neural Networks

The human brain consists of ten billion densely interconnected nerve cells, called *neurons*; each connected to about 10,000 other neurons, with 60 trillion connections, *synapses*, between them. By using multiple neurons simultaneously, the brain can perform its functions much faster than the fastest computers in existence today. On the other hand, a neuron can be considered as a basic information-processing unit, whereas our brain can be considered as a highly complex, nonlinear and parallel biological information-processing network, in which information is stored and processed simultaneously. Learning is a fundamental and essential characteristic of biological neural networks. The ease with which they can learn led to attempts to emulate a biological neural network in a computer.

In the 1940s, McCulloch and Pitts proposed a model for biological neurons and biological neural networks. A stimulus is transmitted from dendrites to a soma via synapses, and axons transmit the response of one soma to another, as shown in Fig. 1.1. Inspired by the mechanism for learning in biological neurons, artificial neurons and artificial neural networks can perform arithmetic functions, with cells corresponding to neurons, activations corresponding to neuronal firing rates, connections corresponding to synapses, and connection weights corresponding to synaptic strengths, as shown in Fig. 1.1. The analogy between biological neurons and artificial neurons is made in Table 1.1. However, neural networks are far too simple to serve as realistic brain models on the cell level, but they might serve as very good models for the essential information processing tasks that organisms perform. This remains an open question because we have so little understanding of how the brain actually works (Gallant, 1993).

In a neural network, *neurons* are joined by directed arcs – *connections*. The neurons and arcs constitute the network *topology*. Each arc has a numerical *weight* that specifies the influence between two neurons. Positive weights indicate reinforcement; negative weights represent inhibition. The weights determine the behavior of the network, playing somewhat the same role as in a conventional computer program. Typically, there are many *inputs* for a single neuron, and a subsequent *output* of an *activation function* (or *transfer function*). Some frequently used activation functions include:

D.S. Yeung et al., *Sensitivity Analysis for Neural Networks*, Natural Computing Series,
DOI 10.1007/978-3-642-02532-7_1,