SpringerBriefs in Computer Science

Series editors

Stan Zdonik, Brown University, Providence, RI, USA
Shashi Shekhar, University of Minnesota, Minneapolis, MN, USA
Xindong Wu, University of Vermont, Burlington, VT, USA
Lakhmi C. Jain, University of South Australia, Adelaide, SA, Australia
David Padua, University of Illinois Urbana-Champaign, Urbana, IL, USA
Xuemin Sherman Shen, University of Waterloo, Waterloo, ON, Canada
Borko Furht, Florida Atlantic University, Boca Raton, FL, USA
V. S. Subrahmanian, Department of Computer Science, University of Maryland, College Park, MD, USA
Martial Hebert, Carnegie Mellon University, Pittsburgh, PA, USA
Katsushi Ikeuchi, Meguro-ku, University of Tokyo, Tokyo, Japan
Bruno Siciliano, Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione, Università di Napoli Federico II, Napoli, Italy
Sushil Jajodia, George Mason University, Fairfax, VA, USA
Newton Lee, Institute for Education, Research and Scholarships, Los Angeles, CA, USA

SpringerBriefs present concise summaries of cutting-edge research and practical applications across a wide spectrum of fields. Featuring compact volumes of 50 to 125 pages, the series covers a range of content from professional to academic.

Typical topics might include:

- A timely report of state-of-the art analytical techniques
- A bridge between new research results, as published in journal articles, and a contextual literature review
- A snapshot of a hot or emerging topic
- An in-depth case study or clinical example
- A presentation of core concepts that students must understand in order to make independent contributions

Briefs allow authors to present their ideas and readers to absorb them with minimal time investment. Briefs will be published as part of Springer's eBook collection, with millions of users worldwide. In addition, Briefs will be available for individual print and electronic purchase. Briefs are characterized by fast, global electronic dissemination, standard publishing contracts, easy-to-use manuscript preparation and formatting guidelines, and expedited production schedules. We aim for publication 8–12 weeks after acceptance. Both solicited and unsolicited manuscripts are considered for publication in this series.

More information about this series at http://www.springer.com/series/10028

Akka Zemmari • Jenny Benois-Pineau

Deep Learning in Mining of Visual Content

Springer

Akka Zemmari
Laboratoire Bordelais de Recherche
en Informatique (LaBRI)
University of Bordeaux
Talence Cedex, France

Jenny Benois-Pineau
Laboratoire Bordelais de Recherche
en Informatique (LaBRI)
University of Bordeaux
Talence Cedex, France

ISSN 2191-5768 ISSN 2191-5776 (electronic)
SpringerBriefs in Computer Science
ISBN 978-3-030-34375-0 ISBN 978-3-030-34376-7 (eBook)
https://doi.org/10.1007/978-3-030-34376-7

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2020
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Having written this book for young researchers and professionals, we dedicate it to those who encouraged us, as young researchers, to

Dominique BARBA and Yves METIVIER

Preface

Deep learning is a new trend in artificial intelligence pushing the performances of machine learning algorithms to their real-life applicability for solving a variety of problems the humanity faces nowadays.

Computer vision and multimedia indexing research have realized a general move from all previous approaches to those on the basis of deep learning.

This approach is built on the principles of supervised machine learning and in particular, artificial neural networks. Applied to visual information mining, it also incorporates our fundamental knowledge in image processing and analysis. Today, we translate all our know-how in visual information mining into this language.

The proliferation of software frameworks allows for easy design and implementation of deep architectures, for the choice and adequate parameterization of different optimization algorithms for training parameters of deep neural networks. The availability of graphical processing units (GPU) and of distributed computing made the computational times for learning quite reasonable. For young researchers and those who move to this kind of methods it is important, we think, to get very quickly into comprehension of underlying mathematical models and formalism, but also to make a bridge between the methods previously used for understanding of images and videos and these winning tools.

It is difficult today to write a book about deep learning, so numerous are different tutorials easily accessible on the Internet. What is the particularity of our book compared to them? We tried to keep a sufficient balance between the usage of mathematical formalism, graphical illustrations, and real-world examples. The book should be easy to understand for young researchers and professionals with engineering and computer science background. Deep learning without pain, this is our goal.

Acknowledgements We thank our PhD students, Abraham Montoya Obeso and Karim Aderghal, for their kind help in preparation of illustrative material for this book.

Bordeaux, France
Talence, France
August 2019

Akka Zemmari
Jenny Benois-Pineau

Contents

1 Introduction 1

2 Supervised Learning Problem Formulation 5
2.1 Supervised Learning 5
2.2 Classification and Regression 6
2.3 Evaluation Metrics 7
2.3.1 Confusion Matrix 7
2.3.2 Metrics 9
2.3.3 AUC-ROC Curve 10
2.4 Conclusion 11

3 Neural Networks from Scratch 13
3.1 Formal Neuron 13
3.2 Artificial Neural Networks and Deep Neural Networks 16
3.3 Learning a Neural Network 17
3.3.1 Loss Function 18
3.3.2 One-Hot Encoding 18
3.3.3 Softmax or How to Transform Outputs into Probabilities 19
3.3.4 Cross-Entropy 19
3.4 Conclusion 20

4 Optimization Methods 21
4.1 Gradient Descent 21
4.2 Stochastic Gradient Descent 23
4.3 Momentum Based SGD 24
4.4 Nesterov Accelerated Gradient Descent 25
4.5 Adaptative Learning Rate 25
4.6 Extensions of Gradient Descent 27
4.7 Gradient Estimation in Neural Networks 28
4.7.1 A Simple Example 28
4.7.2 General Case: Backpropagation Algorithm 30
4.8 Conclusion 32

5 Deep in the Wild 35
5.1 Introduction 35
5.2 Convolution 36
5.3 Sub-sampling 42
5.3.1 Image Sampling 42
5.3.2 Sub-sampling of Images and Features 45
5.4 Conclusion 46

6 Convolutional Neural Networks as Image Analysis Tool 49
6.1 General Principles 49
6.2 Convolutional Layers 50
6.3 Max-Pooling Layers 52
6.4 Dropout 53
6.5 Some Well-Known CNNs Architectures 54
6.5.1 LeNet Architecture and MNIST Dataset 54
6.5.2 AlexNet Architecture 55
6.5.3 GoogLeNet 56
6.5.4 Other Important Architectures 56
6.6 Conclusion 58

7 Dynamic Content Mining 59
7.1 Hidden Markov Models 59
7.1.1 Likelihood Computation 61
7.1.2 Decoding: The Viterbi Algorithm 62
7.1.3 Learning an HMM 63
7.2 Recurrent Neural Networks 65
7.2.1 Definition 65
7.2.2 Training an RNN 67
7.3 Long-Short Term Memory Networks 68
7.3.1 The Model 68
7.4 Conclusion 69

8 Case Study for Digital Cultural Content Mining 71
8.1 Introduction 71
8.2 Visual Saliency 73
8.2.1 Top-Down Saliency Maps Built Upon Gaze Fixations 73
8.2.2 Co-saliency Propagation 75
8.3 Saliency in Deep CNNs 77
8.3.1 Attentional Mechanisms in Deep NNs 77
8.3.2 Squeeze-and-Excitation Networks 77
8.3.3 Double Attention Networks (A^2-Nets) 77
8.3.4 Visual Saliency Propagation 79
8.4 Conclusion 85

9 Introducing Domain Knowledge 87
9.1 Introduction 87
9.2 Domain Knowledge 88

9.2.1 Imaging Modalities 88
9.2.2 Selection of Brain ROI 90
9.2.3 Alignment of Different Imaging Modalities 91
9.3 Siamese Deep NNs for Fusion of Modalities 93
9.3.1 "2D+ϵ" Approach for Classification of Brain Scans for AD Detection 94
9.3.2 Siamese Architecture 95
9.4 Conclusion 96

Conclusion 99

Glossary 101

References 103

Acronyms

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
HMM	Hidden Markov Model
LSTM	Long Short-Term Memory Network
MLP	Multi-Layered Perceptron
RNN	Recurrent Neural Network

List of Figures

Fig. 2.1 Classification and regression examples given two features. In (**a**), a classification problem where red dots are class 0 and green dots are class 1, the black line represents the model to classify new data into two categories. In (**b**), a regression example where red dots are data and the black line represents the model to predict continuous values 7
Fig. 2.2 A 2×2 confusion matrix 8
Fig. 2.3 A 67×67 confusion matrix example in the task of architectural classification, see Chap. 8. Values are normalized in the range [0, 1] 8
Fig. 2.4 The AUC-ROC curve plot 11

Fig. 3.1 Two representations of a formal neuron: a detailed representation (**a**) and a compact one (**b**). Note that when it is a part of a neural network, the neuron is simply represented by a vertex 14
Fig. 3.2 Two particular activation functions: (**a**) the sigmoid function and (**b**) the Rectified Linear Unit (Relu) function 15
Fig. 3.3 Two neural networks: both have three inputs x_1, x_2 and x_3, and two outputs y_1 and y_2. Network (**a**) is a single-layer neural network and (**b**) a multilayers network with two hidden layers 16
Fig. 3.4 Two layers, the weights matrix between the two layers l and $l + 1$ and the bias vector of layer l 17
Fig. 3.5 Example of a one-hot encoding 18

Fig. 4.1 Two functions: (**a**) a convex function having a single minimum and (**b**) a non-convex one having two local minima 22
Fig. 4.2 A non-convex function with multiple local minima and plateaus 23

Fig. 4.3 Three examples of learning rates: (**a**) a large rate can yield to a divergent behaviour, (**b**) a too small rate yield to a slow convergence and (**c**) an adaptive rate can ensure a quick convergence 26
Fig. 4.4 A very simple neural network: two layers with a single unit in each layer. The two neurons have the same activation function σ 29

Fig. 5.1 Low-pass filtering of an image: (**a**) source image and (**b**) Gaussian filtered image with a 3×3 kernel 38
Fig. 5.2 (**a**) Source image and (**b**) Sobel filtering on x-axis 39
Fig. 5.3 AlexNet architecture [KSH12] 40
Fig. 5.4 Difference of Gaussians filtering on two samples. Samples are shown in top and bottom rows with $\sigma = 1.0$ and $\sigma = 1.5$, respectively 40
Fig. 5.5 Feature maps from two samples by applying Difference of Gaussians (DoG) filtering and the output of the first convolutional layer (Conv) of AlexNet architecture [KSH12] 41
Fig. 5.6 Sampled image spectra: (**a**) original image spectrum, (**b**) sampled image spectrum and (**c**) reconstructed image spectrum 44
Fig. 5.7 Filtering in Fourier domain: (**a**) with Box filter, (**b**) with Gaussian filter 45
Fig. 5.8 Illustration of sub-sampling process: (**a**) by a Gaussian pyramid, (**b**) sub-sampling by convolution and pooling layers in AlexNet architecture [KSH12], (**c**) the output (6×6 pixels) of both pipelines 47

Fig. 6.1 In a CNN, the dimensions decrease layers after layers 50
Fig. 6.2 Image encoding: (**a**) a handwritten digit and (**b**) its encoding 51
Fig. 6.3 The convolution of a matrix (6×6) corresponding to the image in Fig. 6.2 and a filter (3×3) and the resulting matrix 51
Fig. 6.4 The first two layers of the CNN for the previous example: the two units (red and blue) are connected to a subset of the input layer units and not to all of them and the two units share the same weights for some of the previous units 52
Fig. 6.5 An illustration of a max-pooling operation: the size of the image is reduced 53
Fig. 6.6 Illustration of the dropout regularization: (**a**) the neural network and (**b**) the dropout application. At the beginning of any iteration in the training phase, each unit have probability p to be removed, together with its connections making the network to become thinner. During the test phase, if the dropout probability is p, each weight is multiplied by $1 - p$ 54
Fig. 6.7 GoogLeNet architecture: structure of an inception block 56

Fig. 6.8 Architecture of the ResNet network. It contains 34 layers and shortcuts each two convolutions 57

Fig. 7.1 A node in a hidden Markov model: possible transitions are from a hidden state $X_t = s_i$ to another one, and a node can emit an observation $o_t = o_j$. The conditional probabilities correspond to the transition and emission probabilities 61
Fig. 7.2 A very simple representation of a recurrent neural network 65
Fig. 7.3 A recurrent network and its unfolding 66
Fig. 7.4 A unit of a Short Long-Term Memory network: σ and tanh are respectively sigmoid and tanh functions as introduced in Chap. 3. Operators $\times$ and $+$ are simply the multiplication and addition operations 68

Fig. 8.1 Gaze fixations and subjective saliency map generated: (**a**) Gaze fixations over the image and (**b**) saliency map over the source image. Red: more attention, Blue: less attention 75
Fig. 8.2 Reference and projected saliency map of reference and target images by matching points. A red line depicts keypoints matching and green polygon over target image shows the transformation given by the estimated homography. Here, the parameters are Lowe's ratio and $\alpha = 20°$ as the angle threshold for the homography validation step. Red: more attention, Blue: less attention 76
Fig. 8.3 Squeeze-and-Excitation block, re-drawn from [FSM+17] 78
Fig. 8.4 A^2: double attention block, re-drawn from [CKL+18] 78
Fig. 8.5 Random pooling of features for different values of k-parameter; original image is given as a feature map with its respective saliency map 81
Fig. 8.6 Saliency maps introduced in pooling layers. Feature maps are not resized for a better visualization 81
Fig. 8.7 Adaptive spatial saliency dropping. Visual attention maps are displayed in order to verify where activations are dropped. We stated that $k = 4$ brings better results after training several models. For this sample $\mu_{sm} = 0.22$ and λ^* is computed by Eq. (8.13) for each value of k 84

Fig. 9.1 Example of sMRI and DTI modalities on AD brain. (**a**) sMRI modality, (**b**) MD-DTI modality 90
Fig. 9.2 Examples of AD, MCI and Healthy brain on SMRI and DTI Modalities. From left to right AD, MCI, NC 91
Fig. 9.3 Diagram of the Preprocessing for sMRI dataset 93
Fig. 9.4 Single Network for ROI classification on one projection of one modality 94
Fig. 9.5 Siamese Network built on all modalities and all projections 96