**Evaluation Only. Created with Aspose.Words. Copyright 2003-2022 Aspose Pty Ltd.**

DEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNINGDEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING UNTRAINED CLOUD MULTIPROCESS MODEL DATA LOADING SAMPLE BATCH TRAINED TENSORS TENSOR PRODUCTION MODEL DATA (ONNX,JIT SOURCE TRAINING TORCHSCRIPT) LOOP DISTRIBUTED TRAINING PRODUCTION ON MULTIPLE SERVERS/GPUS SERVERDEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING UNTRAINED CLOUD MULTIPROCESS MODEL DATA LOADING SAMPLE BATCH TRAINED TENSORS TENSOR PRODUCTION MODEL DATA (ONNX,JIT SOURCE TRAINING TORCHSCRIPT) LOOP DISTRIBUTED TRAINING PRODUCTION ON MULTIPLE SERVERS/GPUS SERVER DEEP LEARNING WITH PYTORCH ELI STEVENS,LUCA ANTIGA AND THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING SHELTER ISLANDDEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING UNTRAINED CLOUD MULTIPROCESS MODEL DATA LOADING SAMPLE BATCH TRAINED TENSORS TENSOR PRODUCTION MODEL DATA (ONNX,JIT SOURCE TRAINING TORCHSCRIPT) LOOP DISTRIBUTED TRAINING PRODUCTION ON MULTIPLE SERVERS/GPUS SERVER DEEP LEARNING WITH PYTORCH ELI STEVENS,LUCA ANTIGA AND THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING SHELTER ISLAND FOR ONLINE INFORMATION AND ORDERING OF THIS AND OTHER MANNING BOOKS, PLEASE VISIT WWW.MANNING.COM. THE PUBLISHER OFFERS DISCOUNTS ON THIS BOOK WHEN ORDERED IN QUANTITY FOR MORE INFORMATION, PLEASE CONTACT SPECIAL SALES DEPARTMENT MANNING PUBLICATIONS CO. 20 BALDWIN ROAD PO BOX 761 SHELTER ISLAND, NY 11964 EMAIL: ORDERS@MANNING.COM @2020 BY MANNING PUBLICATIONS CO. ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED, STORED IN A RETRIEVAL SYSTEM, OR TRANSMITTED, IN ANY FORM OR BY MEANS ELECTRONIC, MECHANICAL, PHOTOCOPYING, OR OTHERWISE, WITHOUT PRIOR WRITTEN PERMISSION OF THE PUBLISHER. MANY OF THE DESIGNATIONS USED BY MANUFACTURERS AND SELLERS TO DISTINGUISH THEIR PRODUCTS ARE CLAIMED AS TRADEMARKS. WHERE THOSE DESIGNATIONS APPEAR IN THE BOOK, AND MANNING PUBLICATIONS WAS AWARE OF A TRADEMARK CLAIM, THE DESIGNATIONS HAVE BEEN PRINTED IN INITIAL CAPS OR ALL CAPS. @ RECOGNIZING THE IMPORTANCE OF PRESERVING WHAT HAS BEEN WRITTEN, IT IS MANNING'S POLICY TO HAVE THE BOOKS WE PUBLISH PRINTED ON ACID-FREE PAPER, AND WE EXERT OUR BEST EFFORTS TO THAT END. RECOGNIZING ALSO OUR RESPONSIBILITY TO CONSERVE THE RESOURCES OF OUR PLANET, MANNING BOOKS ARE PRINTED ON PAPER THAT IS AT LEAST 15 PERCENT RECYCLED AND PROCESSED WITHOUT THE USE OF ELEMENTAL CHLORINE. MANNING PUBLICATIONS CO. DEVELOPMENT EDITOR: FRANCES LEFKOWITZ 20 BALDWIN ROAD TECHNICAL DEVELOPMENT EDITOR: ARTHUR ZUBAREV PO BOX 761 REVIEW EDITOR: IVAN MARTINOVIC SHELTER ISLAND, NY 11964 PRODUCTION EDITOR: DEIRDRE HIAM COPYEDITOR: TIFFANY TAYLOR PROOFREADER: KATIE TENNANT TECHNICAL PROOFREADER: KOSTAS PASSADIS TYPESETTER: GORDAN SALINOVIC COVER DESIGNER: MARIJA TUDOR ISBN 9781617295263 PRINTED IN THE UNITED STATES OF AMERICADEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING UNTRAINED CLOUD MULTIPROCESS MODEL DATA LOADING SAMPLE BATCH TRAINED TENSORS TENSOR PRODUCTION MODEL DATA (ONNX,JIT SOURCE TRAINING TORCHSCRIPT) LOOP DISTRIBUTED TRAINING PRODUCTION ON MULTIPLE SERVERS/GPUS SERVER DEEP LEARNING WITH PYTORCH ELI STEVENS,LUCA ANTIGA AND THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING SHELTER ISLAND FOR ONLINE INFORMATION AND ORDERING OF THIS AND OTHER MANNING BOOKS, PLEASE VISIT WWW.MANNING.COM. THE PUBLISHER OFFERS DISCOUNTS ON THIS BOOK WHEN ORDERED IN QUANTITY FOR MORE INFORMATION, PLEASE CONTACT SPECIAL SALES DEPARTMENT MANNING PUBLICATIONS CO. 20 BALDWIN ROAD PO BOX 761 SHELTER ISLAND, NY 11964 EMAIL: ORDERS@MANNING.COM @2020 BY MANNING PUBLICATIONS CO. ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED, STORED IN A RETRIEVAL SYSTEM, OR TRANSMITTED, IN ANY FORM OR BY MEANS ELECTRONIC, MECHANICAL, PHOTOCOPYING, OR OTHERWISE, WITHOUT PRIOR WRITTEN PERMISSION OF THE PUBLISHER. MANY OF THE DESIGNATIONS USED BY MANUFACTURERS AND SELLERS TO DISTINGUISH THEIR PRODUCTS ARE CLAIMED AS TRADEMARKS. WHERE THOSE DESIGNATIONS APPEAR IN THE BOOK, AND MANNING PUBLICATIONS WAS AWARE OF A TRADEMARK CLAIM, THE DESIGNATIONS HAVE BEEN PRINTED IN INITIAL CAPS OR ALL CAPS. @ RECOGNIZING THE IMPORTANCE OF PRESERVING WHAT HAS BEEN WRITTEN, IT IS MANNING'S POLICY TO HAVE THE BOOKS WE PUBLISH PRINTED ON ACID-FREE PAPER, AND WE EXERT OUR BEST EFFORTS TO THAT END. RECOGNIZING ALSO OUR RESPONSIBILITY TO CONSERVE THE RESOURCES OF OUR PLANET, MANNING BOOKS ARE PRINTED ON PAPER THAT IS AT LEAST 15 PERCENT RECYCLED AND PROCESSED WITHOUT THE USE OF ELEMENTAL CHLORINE. MANNING PUBLICATIONS CO. DEVELOPMENT EDITOR: FRANCES LEFKOWITZ 20 BALDWIN ROAD TECHNICAL DEVELOPMENT EDITOR: ARTHUR ZUBAREV PO BOX 761 REVIEW EDITOR: IVAN MARTINOVIC SHELTER ISLAND, NY 11964 PRODUCTION EDITOR: DEIRDRE HIAM COPYEDITOR: TIFFANY TAYLOR PROOFREADER: KATIE TENNANT TECHNICAL PROOFREADER: KOSTAS PASSADIS TYPESETTER: GORDAN SALINOVIC COVER DESIGNER: MARIJA TUDOR ISBN 9781617295263 PRINTED IN THE UNITED STATES OF AMERICA TO MY WIFE (THIS BOOK WOULD NOT HAVE HAPPENED WITHOUT HER INVALUABLE SUPPORT AND PARTNERSHIP), MY PARENTS (I WOULD NOT HAVE HAPPENED WITHOUT THEM), AND MY CHILDREN (THIS BOOK WOULD HAVE HAPPENED A LOT SOONER BUT FOR THEM) THANK YOU FOR BEING MY HOME, MY FOUNDATION, AND MY JOY ELI STEVENS SAME :-) BUT, REALLY, THIS IS FOR YOU, ALICE AND LUIGI LUCA ANTIGA TO EVA, REBEKKA, JONATHAN, AND DAVID. THOMAS VIEHMANNDEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING UNTRAINED CLOUD MULTIPROCESS MODEL DATA LOADING SAMPLE BATCH TRAINED TENSORS TENSOR PRODUCTION MODEL DATA (ONNX,JIT SOURCE TRAINING TORCHSCRIPT) LOOP DISTRIBUTED TRAINING PRODUCTION ON MULTIPLE SERVERS/GPUS SERVER DEEP LEARNING WITH PYTORCH ELI STEVENS,LUCA ANTIGA AND THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING SHELTER ISLAND FOR ONLINE INFORMATION AND ORDERING OF THIS AND OTHER MANNING BOOKS, PLEASE VISIT WWW.MANNING.COM. THE PUBLISHER OFFERS DISCOUNTS ON THIS BOOK WHEN ORDERED IN QUANTITY FOR MORE INFORMATION, PLEASE CONTACT SPECIAL SALES DEPARTMENT MANNING PUBLICATIONS CO. 20 BALDWIN ROAD PO BOX 761 SHELTER ISLAND, NY 11964 EMAIL: ORDERS@MANNING.COM @2020 BY MANNING PUBLICATIONS CO. ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED, STORED IN A RETRIEVAL SYSTEM, OR TRANSMITTED, IN ANY FORM OR BY MEANS ELECTRONIC, MECHANICAL, PHOTOCOPYING, OR OTHERWISE, WITHOUT PRIOR WRITTEN PERMISSION OF THE PUBLISHER. MANY OF THE DESIGNATIONS USED BY MANUFACTURERS AND SELLERS TO DISTINGUISH THEIR PRODUCTS ARE CLAIMED AS TRADEMARKS. WHERE THOSE DESIGNATIONS APPEAR IN THE BOOK, AND MANNING PUBLICATIONS WAS AWARE OF A TRADEMARK CLAIM, THE DESIGNATIONS HAVE BEEN PRINTED IN INITIAL CAPS OR ALL CAPS. @ RECOGNIZING THE IMPORTANCE OF PRESERVING WHAT HAS BEEN WRITTEN, IT IS MANNING'S POLICY TO HAVE THE BOOKS WE PUBLISH PRINTED ON ACID-FREE PAPER, AND WE EXERT OUR BEST EFFORTS TO THAT END. RECOGNIZING ALSO OUR RESPONSIBILITY TO CONSERVE THE RESOURCES OF OUR PLANET, MANNING BOOKS ARE PRINTED ON PAPER THAT IS AT LEAST 15 PERCENT RECYCLED AND PROCESSED WITHOUT THE USE OF ELEMENTAL CHLORINE. MANNING PUBLICATIONS CO. DEVELOPMENT EDITOR: FRANCES LEFKOWITZ 20 BALDWIN ROAD TECHNICAL DEVELOPMENT EDITOR: ARTHUR ZUBAREV PO BOX 761 REVIEW EDITOR: IVAN MARTINOVIC SHELTER ISLAND, NY 11964 PRODUCTION EDITOR: DEIRDRE HIAM COPYEDITOR: TIFFANY TAYLOR PROOFREADER: KATIE TENNANT TECHNICAL PROOFREADER: KOSTAS PASSADIS TYPESETTER: GORDAN SALINOVIC COVER DESIGNER: MARIJA TUDOR ISBN 9781617295263 PRINTED IN THE UNITED STATES OF AMERICA TO MY WIFE (THIS BOOK WOULD NOT HAVE HAPPENED WITHOUT HER INVALUABLE SUPPORT AND PARTNERSHIP), MY PARENTS (I WOULD NOT HAVE HAPPENED WITHOUT THEM), AND MY CHILDREN (THIS BOOK WOULD HAVE HAPPENED A LOT SOONER BUT FOR THEM) THANK YOU FOR BEING MY HOME, MY FOUNDATION, AND MY JOY ELI STEVENS SAME :-) BUT, REALLY, THIS IS FOR YOU, ALICE AND LUIGI LUCA ANTIGA TO EVA, REBEKKA, JONATHAN, AND DAVID. THOMAS VIEHMANNDEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING UNTRAINED CLOUD MULTIPROCESS MODEL DATA LOADING SAMPLE BATCH TRAINED TENSORS TENSOR PRODUCTION MODEL DATA (ONNX,JIT SOURCE TRAINING TORCHSCRIPT) LOOP DISTRIBUTED TRAINING PRODUCTION ON MULTIPLE SERVERS/GPUS SERVER DEEP LEARNING WITH PYTORCH ELI STEVENS,LUCA ANTIGA AND THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING SHELTER ISLAND FOR ONLINE INFORMATION AND ORDERING OF THIS AND OTHER MANNING BOOKS, PLEASE VISIT WWW.MANNING.COM. THE PUBLISHER OFFERS DISCOUNTS ON THIS BOOK WHEN ORDERED IN QUANTITY FOR MORE INFORMATION, PLEASE CONTACT SPECIAL SALES DEPARTMENT MANNING PUBLICATIONS CO. 20 BALDWIN ROAD PO BOX 761 SHELTER ISLAND, NY 11964 EMAIL: ORDERS@MANNING.COM @2020 BY MANNING PUBLICATIONS CO. ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED, STORED IN A RETRIEVAL SYSTEM, OR TRANSMITTED, IN ANY FORM OR BY MEANS ELECTRONIC, MECHANICAL, PHOTOCOPYING, OR OTHERWISE, WITHOUT PRIOR WRITTEN PERMISSION OF THE PUBLISHER. MANY OF THE DESIGNATIONS USED BY MANUFACTURERS AND SELLERS TO DISTINGUISH THEIR PRODUCTS ARE CLAIMED AS TRADEMARKS. WHERE THOSE DESIGNATIONS APPEAR IN THE BOOK, AND MANNING PUBLICATIONS WAS AWARE OF A TRADEMARK CLAIM, THE DESIGNATIONS HAVE BEEN PRINTED IN INITIAL CAPS OR ALL CAPS. @ RECOGNIZING THE IMPORTANCE OF PRESERVING WHAT HAS BEEN WRITTEN, IT IS MANNING'S POLICY TO HAVE THE BOOKS WE PUBLISH PRINTED ON ACID-FREE PAPER, AND WE EXERT OUR BEST EFFORTS TO THAT END. RECOGNIZING ALSO OUR RESPONSIBILITY TO CONSERVE THE RESOURCES OF OUR PLANET, MANNING BOOKS ARE PRINTED ON PAPER THAT IS AT LEAST 15 PERCENT RECYCLED AND PROCESSED WITHOUT THE USE OF ELEMENTAL CHLORINE. MANNING PUBLICATIONS CO. DEVELOPMENT EDITOR: FRANCES LEFKOWITZ 20 BALDWIN ROAD TECHNICAL DEVELOPMENT EDITOR: ARTHUR ZUBAREV PO BOX 761 REVIEW EDITOR: IVAN MARTINOVIC SHELTER ISLAND, NY 11964 PRODUCTION EDITOR: DEIRDRE HIAM COPYEDITOR: TIFFANY TAYLOR PROOFREADER: KATIE TENNANT TECHNICAL PROOFREADER: KOSTAS PASSADIS TYPESETTER: GORDAN SALINOVIC COVER DESIGNER: MARIJA TUDOR ISBN 9781617295263 PRINTED IN THE UNITED STATES OF AMERICA TO MY WIFE (THIS BOOK WOULD NOT HAVE HAPPENED WITHOUT HER INVALUABLE SUPPORT AND PARTNERSHIP), MY PARENTS (I WOULD NOT HAVE HAPPENED WITHOUT THEM), AND MY CHILDREN (THIS BOOK WOULD HAVE HAPPENED A LOT SOONER BUT FOR THEM) THANK YOU FOR BEING MY HOME, MY FOUNDATION, AND MY JOY ELI STEVENS SAME :-) BUT, REALLY, THIS IS FOR YOU, ALICE AND LUIGI LUCA ANTIGA TO EVA, REBEKKA, JONATHAN, AND DAVID. THOMAS VIEHMANN CONTENTS FOREWORD XV PREFACE XVII ACKNOWLEDGMENTS XIX ABOUT THIS BOOK XXI ABOUT THE AUTHORS XXVII ABOUT THE COVER ILLUSTRATION XXVIII PART 1 CORE PYTORCH. INTRODUCING DEEP LEARNING AND THE PYTORCH LIBRARY3 1.1 THE DEEP LEARNING REVOLUTION 4 1.2 PYTORCH FOR DEEP LEARNING 6 1.3 WHY PYTORCH?7 THE DEEP LEARNING COMPETITIVE LANDSCAPE8 1.4 AN OVERVIEW OF HOW PYTORCH SUPPORTS DEEP LEARNING PROJECTS 10 1.5 HARDWARE AND SOFTWARE REQUIREMENTS 13 USING JUPYTER NOTEBOOKS 14 1.6 EXERCISES 15 1.7 SUMMARY 15DEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING UNTRAINED CLOUD MULTIPROCESS MODEL DATA LOADING SAMPLE BATCH TRAINED TENSORS TENSOR PRODUCTION MODEL DATA (ONNX,JIT SOURCE TRAINING TORCHSCRIPT) LOOP DISTRIBUTED TRAINING PRODUCTION ON MULTIPLE SERVERS/GPUS SERVER DEEP LEARNING WITH PYTORCH ELI STEVENS,LUCA ANTIGA AND THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING SHELTER ISLAND FOR ONLINE INFORMATION AND ORDERING OF THIS AND OTHER MANNING BOOKS, PLEASE VISIT WWW.MANNING.COM. THE PUBLISHER OFFERS DISCOUNTS ON THIS BOOK WHEN ORDERED IN QUANTITY FOR MORE INFORMATION, PLEASE CONTACT SPECIAL SALES DEPARTMENT MANNING PUBLICATIONS CO. 20 BALDWIN ROAD PO BOX 761 SHELTER ISLAND, NY 11964 EMAIL: ORDERS@MANNING.COM @2020 BY MANNING PUBLICATIONS CO. ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED, STORED IN A RETRIEVAL SYSTEM, OR TRANSMITTED, IN ANY FORM OR BY MEANS ELECTRONIC, MECHANICAL, PHOTOCOPYING, OR OTHERWISE, WITHOUT PRIOR WRITTEN PERMISSION OF THE PUBLISHER. MANY OF THE DESIGNATIONS USED BY MANUFACTURERS AND SELLERS TO DISTINGUISH THEIR PRODUCTS ARE CLAIMED AS TRADEMARKS. WHERE THOSE DESIGNATIONS APPEAR IN THE BOOK, AND MANNING PUBLICATIONS WAS AWARE OF A TRADEMARK CLAIM, THE DESIGNATIONS HAVE BEEN PRINTED IN INITIAL CAPS OR ALL CAPS. @ RECOGNIZING THE IMPORTANCE OF PRESERVING WHAT HAS BEEN WRITTEN, IT IS MANNING'S POLICY TO HAVE THE BOOKS WE PUBLISH PRINTED ON ACID-FREE PAPER, AND WE EXERT OUR BEST EFFORTS TO THAT END. RECOGNIZING ALSO OUR RESPONSIBILITY TO CONSERVE THE RESOURCES OF OUR PLANET, MANNING BOOKS ARE PRINTED ON PAPER THAT IS AT LEAST 15 PERCENT RECYCLED AND PROCESSED WITHOUT THE USE OF ELEMENTAL CHLORINE. MANNING PUBLICATIONS CO. DEVELOPMENT EDITOR: FRANCES LEFKOWITZ 20 BALDWIN ROAD TECHNICAL DEVELOPMENT EDITOR: ARTHUR ZUBAREV PO BOX 761 REVIEW EDITOR: IVAN MARTINOVIC SHELTER ISLAND, NY 11964 PRODUCTION EDITOR: DEIRDRE HIAM COPYEDITOR: TIFFANY TAYLOR PROOFREADER: KATIE TENNANT TECHNICAL PROOFREADER: KOSTAS PASSADIS TYPESETTER: GORDAN SALINOVIC COVER DESIGNER: MARIJA TUDOR ISBN 9781617295263 PRINTED IN THE UNITED STATES OF AMERICA TO MY WIFE (THIS BOOK WOULD NOT HAVE HAPPENED WITHOUT HER INVALUABLE SUPPORT AND PARTNERSHIP), MY PARENTS (I WOULD NOT HAVE HAPPENED WITHOUT THEM), AND MY CHILDREN (THIS BOOK WOULD HAVE HAPPENED A LOT SOONER BUT FOR THEM) THANK YOU FOR BEING MY HOME, MY FOUNDATION, AND MY JOY ELI STEVENS SAME :-) BUT, REALLY, THIS IS FOR YOU, ALICE AND LUIGI LUCA ANTIGA TO EVA, REBEKKA, JONATHAN, AND DAVID. THOMAS VIEHMANN CONTENTS FOREWORD XV PREFACE XVII ACKNOWLEDGMENTS XIX ABOUT THIS BOOK XXI ABOUT THE AUTHORS XXVII ABOUT THE COVER ILLUSTRATION XXVIII PART 1 CORE PYTORCH. INTRODUCING DEEP LEARNING AND THE PYTORCH LIBRARY3 1.1 THE DEEP LEARNING REVOLUTION 4 1.2 PYTORCH FOR DEEP LEARNING 6 1.3 WHY PYTORCH?7 THE DEEP LEARNING COMPETITIVE LANDSCAPE8 1.4 AN OVERVIEW OF HOW PYTORCH SUPPORTS DEEP LEARNING PROJECTS 10 1.5 HARDWARE AND SOFTWARE REQUIREMENTS 13 USING JUPYTER NOTEBOOKS 14 1.6 EXERCISES 15 1.7 SUMMARY 15 VI CONTENTS PRETRAINED NETWORKS 16 2.1 A PRETRAINED NETWORK THAT RECOGNIZES THE SUBJECT OF AN IMAGE 17 OBTAINING A PRETRAINED NETWORK FOR IMAGE RECOGNITION 19 ALEXNET 20 - RESNET 22 - READY, SET, ALMOST RUN 22 RUN! 25 2.2 A PRETRAINED MODEL THAT FAKES IT UNTIL IT MAKES IT 27 THE GAN GAME 28 : CYCLEGAN 29 - A NETWORK THAT TURNS HORSES INTO ZEBRAS 30 2.3 A PRETRAINED NETWORK THAT DESCRIBES SCENES 33 NEURALTALK2 34 2.4 TORCH HUB 35 2.5 CONCLUSION 37 2.6 EXERCISES 38 2.7 SUMMARY 38 IT STARTS WITH A TENSOR 39 3.1 THE WORLD AS FLOATING-POINT NUMBERS 40 3.2 TENSORS: MULTIDIMENSIONAL ARRAYS 42 FROM PYTHON LISTS TO PYTORCH TENSORS 42 - CONSTRUCTING OUR FIRST TENSORS 43 - THE ESSENCE OF TENSORS 43 3.3 INDEXING TENSORS 46 3.4 NAMED TENSORS 46 3.5 TENSOR ELEMENT TYPES 50 SPECIFYING THE NUMERIC TYPE WITH DTYPE 50 - A DTYPE FOR EVERY OCCASION 51 - MANAGING A TENSOR'S DTYPE ATTRIBUTE 51 3.6 THE TENSOR API 52 3.7 TENSORS: SCENIC VIEWS OF STORAGE 53 INDEXING INTO STORAGE 54 - MODIFYING STORED VALUES: IN-PLACE OPERATIONS 55 3.8 TENSOR METADATA: SIZE, OFFSET, AND STRIDE 55 VIEWS OF ANOTHER TENSOR'S STORAGE 56 - TRANSPOSING WITHOUT COPYING 58 : TRANSPOSING IN HIGHER DIMENSIONS 60 CONTIGUOUS TENSORS 60 3.9 MOVING TENSORS TO THE GPU 62 MANAGING A TENSOR'S DEVICE ATTRIBUTE 63DEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING UNTRAINED CLOUD MULTIPROCESS MODEL DATA LOADING SAMPLE BATCH TRAINED TENSORS TENSOR PRODUCTION MODEL DATA (ONNX,JIT SOURCE TRAINING TORCHSCRIPT) LOOP DISTRIBUTED TRAINING PRODUCTION ON MULTIPLE SERVERS/GPUS SERVER DEEP LEARNING WITH PYTORCH ELI STEVENS,LUCA ANTIGA AND THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING SHELTER ISLAND FOR ONLINE INFORMATION AND ORDERING OF THIS AND OTHER MANNING BOOKS, PLEASE VISIT WWW.MANNING.COM. THE PUBLISHER OFFERS DISCOUNTS ON THIS BOOK WHEN ORDERED IN QUANTITY FOR MORE INFORMATION, PLEASE CONTACT SPECIAL SALES DEPARTMENT MANNING PUBLICATIONS CO. 20 BALDWIN ROAD PO BOX 761 SHELTER ISLAND, NY 11964 EMAIL: ORDERS@MANNING.COM @2020 BY MANNING PUBLICATIONS CO. ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED, STORED IN A RETRIEVAL SYSTEM, OR TRANSMITTED, IN ANY FORM OR BY MEANS ELECTRONIC, MECHANICAL, PHOTOCOPYING, OR OTHERWISE, WITHOUT PRIOR WRITTEN PERMISSION OF THE PUBLISHER. MANY OF THE DESIGNATIONS USED BY MANUFACTURERS AND SELLERS TO DISTINGUISH THEIR PRODUCTS ARE CLAIMED AS TRADEMARKS. WHERE THOSE DESIGNATIONS APPEAR IN THE BOOK, AND MANNING PUBLICATIONS WAS AWARE OF A TRADEMARK CLAIM, THE DESIGNATIONS HAVE BEEN PRINTED IN INITIAL CAPS OR ALL CAPS. @ RECOGNIZING THE IMPORTANCE OF PRESERVING WHAT HAS BEEN WRITTEN, IT IS MANNING'S POLICY TO HAVE THE BOOKS WE PUBLISH PRINTED ON ACID-FREE PAPER, AND WE EXERT OUR BEST EFFORTS TO THAT END. RECOGNIZING ALSO OUR RESPONSIBILITY TO CONSERVE THE RESOURCES OF OUR PLANET, MANNING BOOKS ARE PRINTED ON PAPER THAT IS AT LEAST 15 PERCENT RECYCLED AND PROCESSED WITHOUT THE USE OF ELEMENTAL CHLORINE. MANNING PUBLICATIONS CO. DEVELOPMENT EDITOR: FRANCES LEFKOWITZ 20 BALDWIN ROAD TECHNICAL DEVELOPMENT EDITOR: ARTHUR ZUBAREV PO BOX 761 REVIEW EDITOR: IVAN MARTINOVIC SHELTER ISLAND, NY 11964 PRODUCTION EDITOR: DEIRDRE HIAM COPYEDITOR: TIFFANY TAYLOR PROOFREADER: KATIE TENNANT TECHNICAL PROOFREADER: KOSTAS PASSADIS TYPESETTER: GORDAN SALINOVIC COVER DESIGNER: MARIJA TUDOR ISBN 9781617295263 PRINTED IN THE UNITED STATES OF AMERICA TO MY WIFE (THIS BOOK WOULD NOT HAVE HAPPENED WITHOUT HER INVALUABLE SUPPORT AND PARTNERSHIP), MY PARENTS (I WOULD NOT HAVE HAPPENED WITHOUT THEM), AND MY CHILDREN (THIS BOOK WOULD HAVE HAPPENED A LOT SOONER BUT FOR THEM) THANK YOU FOR BEING MY HOME, MY FOUNDATION, AND MY JOY ELI STEVENS SAME :-) BUT, REALLY, THIS IS FOR YOU, ALICE AND LUIGI LUCA ANTIGA TO EVA, REBEKKA, JONATHAN, AND DAVID. THOMAS VIEHMANN CONTENTS FOREWORD XV PREFACE XVII ACKNOWLEDGMENTS XIX ABOUT THIS BOOK XXI ABOUT THE AUTHORS XXVII ABOUT THE COVER ILLUSTRATION XXVIII PART 1 CORE PYTORCH. INTRODUCING DEEP LEARNING AND THE PYTORCH LIBRARY3 1.1 THE DEEP LEARNING REVOLUTION 4 1.2 PYTORCH FOR DEEP LEARNING 6 1.3 WHY PYTORCH?7 THE DEEP LEARNING COMPETITIVE LANDSCAPE8 1.4 AN OVERVIEW OF HOW PYTORCH SUPPORTS DEEP LEARNING PROJECTS 10 1.5 HARDWARE AND SOFTWARE REQUIREMENTS 13 USING JUPYTER NOTEBOOKS 14 1.6 EXERCISES 15 1.7 SUMMARY 15 VI CONTENTS PRETRAINED NETWORKS 16 2.1 A PRETRAINED NETWORK THAT RECOGNIZES THE SUBJECT OF AN IMAGE 17 OBTAINING A PRETRAINED NETWORK FOR IMAGE RECOGNITION 19 ALEXNET 20 - RESNET 22 - READY, SET, ALMOST RUN 22 RUN! 25 2.2 A PRETRAINED MODEL THAT FAKES IT UNTIL IT MAKES IT 27 THE GAN GAME 28 : CYCLEGAN 29 - A NETWORK THAT TURNS HORSES INTO ZEBRAS 30 2.3 A PRETRAINED NETWORK THAT DESCRIBES SCENES 33 NEURALTALK2 34 2.4 TORCH HUB 35 2.5 CONCLUSION 37 2.6 EXERCISES 38 2.7 SUMMARY 38 IT STARTS WITH A TENSOR 39 3.1 THE WORLD AS FLOATING-POINT NUMBERS 40 3.2 TENSORS: MULTIDIMENSIONAL ARRAYS 42 FROM PYTHON LISTS TO PYTORCH TENSORS 42 - CONSTRUCTING OUR FIRST TENSORS 43 - THE ESSENCE OF TENSORS 43 3.3 INDEXING TENSORS 46 3.4 NAMED TENSORS 46 3.5 TENSOR ELEMENT TYPES 50 SPECIFYING THE NUMERIC TYPE WITH DTYPE 50 - A DTYPE FOR EVERY OCCASION 51 - MANAGING A TENSOR'S DTYPE ATTRIBUTE 51 3.6 THE TENSOR API 52 3.7 TENSORS: SCENIC VIEWS OF STORAGE 53 INDEXING INTO STORAGE 54 - MODIFYING STORED VALUES: IN-PLACE OPERATIONS 55 3.8 TENSOR METADATA: SIZE, OFFSET, AND STRIDE 55 VIEWS OF ANOTHER TENSOR'S STORAGE 56 - TRANSPOSING WITHOUT COPYING 58 : TRANSPOSING IN HIGHER DIMENSIONS 60 CONTIGUOUS TENSORS 60 3.9 MOVING TENSORS TO THE GPU 62 MANAGING A TENSOR'S DEVICE ATTRIBUTE 63 VII CONTENTS 3.10 NUMPY INTEROPERABILITY64 3.11 GENERALIZED TENSORS ARE TENSORS, TOO 65 3.12 SERIALIZING TENSORS 66 SERIALIZING TO HDF5 WITH H5PY 67 3.13 CONCLUSION 68 3.14 EXERCISES 68 3.15 SUMMARY 68 REAL-WORLD DATA REPRESENTATION USING TENSORS 70 4.1 WORKING WITH IMAGES 71 ADDING COLOR CHANNELS 72 - LOADING AN IMAGE FILE 72 CHANGING THE LAYOUT 73 : NORMALIZING THE DATA 74 4.2 3D IMAGES: VOLUMETRIC DATA 75 LOADING A SPECIALIZED FORMAT 76 4.3 REPRESENTING TABULAR DATA 77 USING A REAL-WORLD DATASET 77 - LOADING A WINE DATA TENSOR 78 REPRESENTING SCORES 81 : ONE-HOT ENCODING 81 : WHEN TO CATEGORIZE 83 - FINDING THRESHOLDS 84 4.4 WORKING WITH TIME SERIES 87 ADDING A TIME DIMENSION 88 - SHAPING THE DATA BY TIME PERIOD 89 - READY FOR TRAINING 90 4.5 REPRESENTING TEXT 93 CONVERTING TEXT TO NUMBERS94 - ONE-HOT-ENCODING CHARACTERS94 ONE-HOT ENCODING WHOLE WORDS 96 : TEXT EMBEDDINGS 98 TEXT EMBEDDINGS AS A BLUEPRINT 100 4.6 CONCLUSION 101 4.7 EXERCISES 101 4.8 SUMMARY 102 THE MECHANICS OF LEARNING 103 5.1 A TIMELESS LESSON IN MODELING 104 5.2 LEARNING IS JUST PARAMETER ESTIMATION 106 A HOT PROBLEM 107 : GATHERING SOME DATA 107 : VISUALIZING THE DATA 108 : CHOOSING A LINEAR MODEL AS A FIRST TRY 108 5.3 LESS LOSS IS WHAT WE WANT 109 FROM PROBLEM BACK TO PYTORCH 110DEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING UNTRAINED CLOUD MULTIPROCESS MODEL DATA LOADING SAMPLE BATCH TRAINED TENSORS TENSOR PRODUCTION MODEL DATA (ONNX,JIT SOURCE TRAINING TORCHSCRIPT) LOOP DISTRIBUTED TRAINING PRODUCTION ON MULTIPLE SERVERS/GPUS SERVER DEEP LEARNING WITH PYTORCH ELI STEVENS,LUCA ANTIGA AND THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING SHELTER ISLAND FOR ONLINE INFORMATION AND ORDERING OF THIS AND OTHER MANNING BOOKS, PLEASE VISIT WWW.MANNING.COM. THE PUBLISHER OFFERS DISCOUNTS ON THIS BOOK WHEN ORDERED IN QUANTITY FOR MORE INFORMATION, PLEASE CONTACT SPECIAL SALES DEPARTMENT MANNING PUBLICATIONS CO. 20 BALDWIN ROAD PO BOX 761 SHELTER ISLAND, NY 11964 EMAIL: ORDERS@MANNING.COM @2020 BY MANNING PUBLICATIONS CO. ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED, STORED IN A RETRIEVAL SYSTEM, OR TRANSMITTED, IN ANY FORM OR BY MEANS ELECTRONIC, MECHANICAL, PHOTOCOPYING, OR OTHERWISE, WITHOUT PRIOR WRITTEN PERMISSION OF THE PUBLISHER. MANY OF THE DESIGNATIONS USED BY MANUFACTURERS AND SELLERS TO DISTINGUISH THEIR PRODUCTS ARE CLAIMED AS TRADEMARKS. WHERE THOSE DESIGNATIONS APPEAR IN THE BOOK, AND MANNING PUBLICATIONS WAS AWARE OF A TRADEMARK CLAIM, THE DESIGNATIONS HAVE BEEN PRINTED IN INITIAL CAPS OR ALL CAPS. @ RECOGNIZING THE IMPORTANCE OF PRESERVING WHAT HAS BEEN WRITTEN, IT IS MANNING'S POLICY TO HAVE THE BOOKS WE PUBLISH PRINTED ON ACID-FREE PAPER, AND WE EXERT OUR BEST EFFORTS TO THAT END. RECOGNIZING ALSO OUR RESPONSIBILITY TO CONSERVE THE RESOURCES OF OUR PLANET, MANNING BOOKS ARE PRINTED ON PAPER THAT IS AT LEAST 15 PERCENT RECYCLED AND PROCESSED WITHOUT THE USE OF ELEMENTAL CHLORINE. MANNING PUBLICATIONS CO. DEVELOPMENT EDITOR: FRANCES LEFKOWITZ 20 BALDWIN ROAD TECHNICAL DEVELOPMENT EDITOR: ARTHUR ZUBAREV PO BOX 761 REVIEW EDITOR: IVAN MARTINOVIC SHELTER ISLAND, NY 11964 PRODUCTION EDITOR: DEIRDRE HIAM COPYEDITOR: TIFFANY TAYLOR PROOFREADER: KATIE TENNANT TECHNICAL PROOFREADER: KOSTAS PASSADIS TYPESETTER: GORDAN SALINOVIC COVER DESIGNER: MARIJA TUDOR ISBN 9781617295263 PRINTED IN THE UNITED STATES OF AMERICA TO MY WIFE (THIS BOOK WOULD NOT HAVE HAPPENED WITHOUT HER INVALUABLE SUPPORT AND PARTNERSHIP), MY PARENTS (I WOULD NOT HAVE HAPPENED WITHOUT THEM), AND MY CHILDREN (THIS BOOK WOULD HAVE HAPPENED A LOT SOONER BUT FOR THEM) THANK YOU FOR BEING MY HOME, MY FOUNDATION, AND MY JOY ELI STEVENS SAME :-) BUT, REALLY, THIS IS FOR YOU, ALICE AND LUIGI LUCA ANTIGA TO EVA, REBEKKA, JONATHAN, AND DAVID. THOMAS VIEHMANN CONTENTS FOREWORD XV PREFACE XVII ACKNOWLEDGMENTS XIX ABOUT THIS BOOK XXI ABOUT THE AUTHORS XXVII ABOUT THE COVER ILLUSTRATION XXVIII PART 1 CORE PYTORCH. INTRODUCING DEEP LEARNING AND THE PYTORCH LIBRARY3 1.1 THE DEEP LEARNING REVOLUTION 4 1.2 PYTORCH FOR DEEP LEARNING 6 1.3 WHY PYTORCH?7 THE DEEP LEARNING COMPETITIVE LANDSCAPE8 1.4 AN OVERVIEW OF HOW PYTORCH SUPPORTS DEEP LEARNING PROJECTS 10 1.5 HARDWARE AND SOFTWARE REQUIREMENTS 13 USING JUPYTER NOTEBOOKS 14 1.6 EXERCISES 15 1.7 SUMMARY 15 VI CONTENTS PRETRAINED NETWORKS 16 2.1 A PRETRAINED NETWORK THAT RECOGNIZES THE SUBJECT OF AN IMAGE 17 OBTAINING A PRETRAINED NETWORK FOR IMAGE RECOGNITION 19 ALEXNET 20 - RESNET 22 - READY, SET, ALMOST RUN 22 RUN! 25 2.2 A PRETRAINED MODEL THAT FAKES IT UNTIL IT MAKES IT 27 THE GAN GAME 28 : CYCLEGAN 29 - A NETWORK THAT TURNS HORSES INTO ZEBRAS 30 2.3 A PRETRAINED NETWORK THAT DESCRIBES SCENES 33 NEURALTALK2 34 2.4 TORCH HUB 35 2.5 CONCLUSION 37 2.6 EXERCISES 38 2.7 SUMMARY 38 IT STARTS WITH A TENSOR 39 3.1 THE WORLD AS FLOATING-POINT NUMBERS 40 3.2 TENSORS: MULTIDIMENSIONAL ARRAYS 42 FROM PYTHON LISTS TO PYTORCH TENSORS 42 - CONSTRUCTING OUR FIRST TENSORS 43 - THE ESSENCE OF TENSORS 43 3.3 INDEXING TENSORS 46 3.4 NAMED TENSORS 46 3.5 TENSOR ELEMENT TYPES 50 SPECIFYING THE NUMERIC TYPE WITH DTYPE 50 - A DTYPE FOR EVERY OCCASION 51 - MANAGING A TENSOR'S DTYPE ATTRIBUTE 51 3.6 THE TENSOR API 52 3.7 TENSORS: SCENIC VIEWS OF STORAGE 53 INDEXING INTO STORAGE 54 - MODIFYING STORED VALUES: IN-PLACE OPERATIONS 55 3.8 TENSOR METADATA: SIZE, OFFSET, AND STRIDE 55 VIEWS OF ANOTHER TENSOR'S STORAGE 56 - TRANSPOSING WITHOUT COPYING 58 : TRANSPOSING IN HIGHER DIMENSIONS 60 CONTIGUOUS TENSORS 60 3.9 MOVING TENSORS TO THE GPU 62 MANAGING A TENSOR'S DEVICE ATTRIBUTE 63 VII CONTENTS 3.10 NUMPY INTEROPERABILITY64 3.11 GENERALIZED TENSORS ARE TENSORS, TOO 65 3.12 SERIALIZING TENSORS 66 SERIALIZING TO HDF5 WITH H5PY 67 3.13 CONCLUSION 68 3.14 EXERCISES 68 3.15 SUMMARY 68 REAL-WORLD DATA REPRESENTATION USING TENSORS 70 4.1 WORKING WITH IMAGES 71 ADDING COLOR CHANNELS 72 - LOADING AN IMAGE FILE 72 CHANGING THE LAYOUT 73 : NORMALIZING THE DATA 74 4.2 3D IMAGES: VOLUMETRIC DATA 75 LOADING A SPECIALIZED FORMAT 76 4.3 REPRESENTING TABULAR DATA 77 USING A REAL-WORLD DATASET 77 - LOADING A WINE DATA TENSOR 78 REPRESENTING SCORES 81 : ONE-HOT ENCODING 81 : WHEN TO CATEGORIZE 83 - FINDING THRESHOLDS 84 4.4 WORKING WITH TIME SERIES 87 ADDING A TIME DIMENSION 88 - SHAPING THE DATA BY TIME PERIOD 89 - READY FOR TRAINING 90 4.5 REPRESENTING TEXT 93 CONVERTING TEXT TO NUMBERS94 - ONE-HOT-ENCODING CHARACTERS94 ONE-HOT ENCODING WHOLE WORDS 96 : TEXT EMBEDDINGS 98 TEXT EMBEDDINGS AS A BLUEPRINT 100 4.6 CONCLUSION 101 4.7 EXERCISES 101 4.8 SUMMARY 102 THE MECHANICS OF LEARNING 103 5.1 A TIMELESS LESSON IN MODELING 104 5.2 LEARNING IS JUST PARAMETER ESTIMATION 106 A HOT PROBLEM 107 : GATHERING SOME DATA 107 : VISUALIZING THE DATA 108 : CHOOSING A LINEAR MODEL AS A FIRST TRY 108 5.3 LESS LOSS IS WHAT WE WANT 109 FROM PROBLEM BACK TO PYTORCH 110 VIII CONTENTS 5.4 DOWN ALONG THE GRADIENT 113 DECREASING LOSS 113 - GETTING ANALYTICAL 114 - ITERATING TO FIT THE MODEL 116 - NORMALIZING INPUTS 119 : VISUALIZING (AGAIN) 122 5.5 PYTORCH'S AUTOGRAD: BACKPROPAGATING ALL THINGS 123 COMPUTING THE GRADIENT AUTOMATICALLY 123 - OPTIMIZERS A LA CARTE 127 : TRAINING, VALIDATION, AND OVERFITTING 131 AUTOGRAD NITS AND SWITCHING IT OFF. 137 5.6 CONCLUSION 139 5.7 EXERCISE 139 5.8 SUMMARY 139 USING A NEURAL NETWORK TO FIT THE DATA 141 6.1 ARTIFICIAL NEURONS142 COMPOSING A MULTILAYER NETWORK 144 - UNDERSTANDING THE ERROR FUNCTION 144 : ALL WE NEED IS ACTIVATION 145 : MORE ACTIVATION FUNCTIONS 147 : CHOOSING THE BEST ACTIVATION FUNCTION 148 WHAT LEARNING MEANS FOR A NEURAL NETWORK 149 6.2 THE PYTORCH NN MODULE 151 USING--CALL- RATHER THAN FORWARD 152 - RETURNING TO THE LINEAR MODEL 153 6.3 FINALLY A NEURAL NETWORK 158 REPLACING THE LINEAR MODEL 158 - INSPECTING THE PARAMETERS 159 COMPARING TO THE LINEAR MODEL 161 6.4 CONCLUSION 162 6.5 EXERCISES 162 6.6 SUMMARY 163 TELLING BIRDS FROM AIRPLANES: LEARNING FROM IMAGES 164 7.1 A DATASET OF TINY IMAGES 165 DOWNLOADING CIFAR-10 166 : THE DATASET CLASS 166 DATASET TRANSFORMS 168 - NORMALIZING DATA 170 7.2 DISTINGUISHING BIRDS FROM AIRPLANES 172 BUILDING THE DATASET 173 - A FULLY CONNECTED MODEL 174 OUTPUT OF A CLASSIFIER 175 : REPRESENTING THE OUTPUT AS PROBABILITIES 176 - A LOSS FOR CLASSIFYING 180 : TRAINING THE CLASSIFIER 182 - THE LIMITS OF GOING FULLY CONNECTED 189 7.3 CONCLUSION 191DEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING UNTRAINED CLOUD MULTIPROCESS MODEL DATA LOADING SAMPLE BATCH TRAINED TENSORS TENSOR PRODUCTION MODEL DATA (ONNX,JIT SOURCE TRAINING TORCHSCRIPT) LOOP DISTRIBUTED TRAINING PRODUCTION ON MULTIPLE SERVERS/GPUS SERVER DEEP LEARNING WITH PYTORCH ELI STEVENS,LUCA ANTIGA AND THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING SHELTER ISLAND FOR ONLINE INFORMATION AND ORDERING OF THIS AND OTHER MANNING BOOKS, PLEASE VISIT WWW.MANNING.COM. THE PUBLISHER OFFERS DISCOUNTS ON THIS BOOK WHEN ORDERED IN QUANTITY FOR MORE INFORMATION, PLEASE CONTACT SPECIAL SALES DEPARTMENT MANNING PUBLICATIONS CO. 20 BALDWIN ROAD PO BOX 761 SHELTER ISLAND, NY 11964 EMAIL: ORDERS@MANNING.COM @2020 BY MANNING PUBLICATIONS CO. ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED, STORED IN A RETRIEVAL SYSTEM, OR TRANSMITTED, IN ANY FORM OR BY MEANS ELECTRONIC, MECHANICAL, PHOTOCOPYING, OR OTHERWISE, WITHOUT PRIOR WRITTEN PERMISSION OF THE PUBLISHER. MANY OF THE DESIGNATIONS USED BY MANUFACTURERS AND SELLERS TO DISTINGUISH THEIR PRODUCTS ARE CLAIMED AS TRADEMARKS. WHERE THOSE DESIGNATIONS APPEAR IN THE BOOK, AND MANNING PUBLICATIONS WAS AWARE OF A TRADEMARK CLAIM, THE DESIGNATIONS HAVE BEEN PRINTED IN INITIAL CAPS OR ALL CAPS. @ RECOGNIZING THE IMPORTANCE OF PRESERVING WHAT HAS BEEN WRITTEN, IT IS MANNING'S POLICY TO HAVE THE BOOKS WE PUBLISH PRINTED ON ACID-FREE PAPER, AND WE EXERT OUR BEST EFFORTS TO THAT END. RECOGNIZING ALSO OUR RESPONSIBILITY TO CONSERVE THE RESOURCES OF OUR PLANET, MANNING BOOKS ARE PRINTED ON PAPER THAT IS AT LEAST 15 PERCENT RECYCLED AND PROCESSED WITHOUT THE USE OF ELEMENTAL CHLORINE. MANNING PUBLICATIONS CO. DEVELOPMENT EDITOR: FRANCES LEFKOWITZ 20 BALDWIN ROAD TECHNICAL DEVELOPMENT EDITOR: ARTHUR ZUBAREV PO BOX 761 REVIEW EDITOR: IVAN MARTINOVIC SHELTER ISLAND, NY 11964 PRODUCTION EDITOR: DEIRDRE HIAM COPYEDITOR: TIFFANY TAYLOR PROOFREADER: KATIE TENNANT TECHNICAL PROOFREADER: KOSTAS PASSADIS TYPESETTER: GORDAN SALINOVIC COVER DESIGNER: MARIJA TUDOR ISBN 9781617295263 PRINTED IN THE UNITED STATES OF AMERICA TO MY WIFE (THIS BOOK WOULD NOT HAVE HAPPENED WITHOUT HER INVALUABLE SUPPORT AND PARTNERSHIP), MY PARENTS (I WOULD NOT HAVE HAPPENED WITHOUT THEM), AND MY CHILDREN (THIS BOOK WOULD HAVE HAPPENED A LOT SOONER BUT FOR THEM) THANK YOU FOR BEING MY HOME, MY FOUNDATION, AND MY JOY ELI STEVENS SAME :-) BUT, REALLY, THIS IS FOR YOU, ALICE AND LUIGI LUCA ANTIGA TO EVA, REBEKKA, JONATHAN, AND DAVID. THOMAS VIEHMANN CONTENTS FOREWORD XV PREFACE XVII ACKNOWLEDGMENTS XIX ABOUT THIS BOOK XXI ABOUT THE AUTHORS XXVII ABOUT THE COVER ILLUSTRATION XXVIII PART 1 CORE PYTORCH. INTRODUCING DEEP LEARNING AND THE PYTORCH LIBRARY3 1.1 THE DEEP LEARNING REVOLUTION 4 1.2 PYTORCH FOR DEEP LEARNING 6 1.3 WHY PYTORCH?7 THE DEEP LEARNING COMPETITIVE LANDSCAPE8 1.4 AN OVERVIEW OF HOW PYTORCH SUPPORTS DEEP LEARNING PROJECTS 10 1.5 HARDWARE AND SOFTWARE REQUIREMENTS 13 USING JUPYTER NOTEBOOKS 14 1.6 EXERCISES 15 1.7 SUMMARY 15 VI CONTENTS PRETRAINED NETWORKS 16 2.1 A PRETRAINED NETWORK THAT RECOGNIZES THE SUBJECT OF AN IMAGE 17 OBTAINING A PRETRAINED NETWORK FOR IMAGE RECOGNITION 19 ALEXNET 20 - RESNET 22 - READY, SET, ALMOST RUN 22 RUN! 25 2.2 A PRETRAINED MODEL THAT FAKES IT UNTIL IT MAKES IT 27 THE GAN GAME 28 : CYCLEGAN 29 - A NETWORK THAT TURNS HORSES INTO ZEBRAS 30 2.3 A PRETRAINED NETWORK THAT DESCRIBES SCENES 33 NEURALTALK2 34 2.4 TORCH HUB 35 2.5 CONCLUSION 37 2.6 EXERCISES 38 2.7 SUMMARY 38 IT STARTS WITH A TENSOR 39 3.1 THE WORLD AS FLOATING-POINT NUMBERS 40 3.2 TENSORS: MULTIDIMENSIONAL ARRAYS 42 FROM PYTHON LISTS TO PYTORCH TENSORS 42 - CONSTRUCTING OUR FIRST TENSORS 43 - THE ESSENCE OF TENSORS 43 3.3 INDEXING TENSORS 46 3.4 NAMED TENSORS 46 3.5 TENSOR ELEMENT TYPES 50 SPECIFYING THE NUMERIC TYPE WITH DTYPE 50 - A DTYPE FOR EVERY OCCASION 51 - MANAGING A TENSOR'S DTYPE ATTRIBUTE 51 3.6 THE TENSOR API 52 3.7 TENSORS: SCENIC VIEWS OF STORAGE 53 INDEXING INTO STORAGE 54 - MODIFYING STORED VALUES: IN-PLACE OPERATIONS 55 3.8 TENSOR METADATA: SIZE, OFFSET, AND STRIDE 55 VIEWS OF ANOTHER TENSOR'S STORAGE 56 - TRANSPOSING WITHOUT COPYING 58 : TRANSPOSING IN HIGHER DIMENSIONS 60 CONTIGUOUS TENSORS 60 3.9 MOVING TENSORS TO THE GPU 62 MANAGING A TENSOR'S DEVICE ATTRIBUTE 63 VII CONTENTS 3.10 NUMPY INTEROPERABILITY64 3.11 GENERALIZED TENSORS ARE TENSORS, TOO 65 3.12 SERIALIZING TENSORS 66 SERIALIZING TO HDF5 WITH H5PY 67 3.13 CONCLUSION 68 3.14 EXERCISES 68 3.15 SUMMARY 68 REAL-WORLD DATA REPRESENTATION USING TENSORS 70 4.1 WORKING WITH IMAGES 71 ADDING COLOR CHANNELS 72 - LOADING AN IMAGE FILE 72 CHANGING THE LAYOUT 73 : NORMALIZING THE DATA 74 4.2 3D IMAGES: VOLUMETRIC DATA 75 LOADING A SPECIALIZED FORMAT 76 4.3 REPRESENTING TABULAR DATA 77 USING A REAL-WORLD DATASET 77 - LOADING A WINE DATA TENSOR 78 REPRESENTING SCORES 81 : ONE-HOT ENCODING 81 : WHEN TO CATEGORIZE 83 - FINDING THRESHOLDS 84 4.4 WORKING WITH TIME SERIES 87 ADDING A TIME DIMENSION 88 - SHAPING THE DATA BY TIME PERIOD 89 - READY FOR TRAINING 90 4.5 REPRESENTING TEXT 93 CONVERTING TEXT TO NUMBERS94 - ONE-HOT-ENCODING CHARACTERS94 ONE-HOT ENCODING WHOLE WORDS 96 : TEXT EMBEDDINGS 98 TEXT EMBEDDINGS AS A BLUEPRINT 100 4.6 CONCLUSION 101 4.7 EXERCISES 101 4.8 SUMMARY 102 THE MECHANICS OF LEARNING 103 5.1 A TIMELESS LESSON IN MODELING 104 5.2 LEARNING IS JUST PARAMETER ESTIMATION 106 A HOT PROBLEM 107 : GATHERING SOME DATA 107 : VISUALIZING THE DATA 108 : CHOOSING A LINEAR MODEL AS A FIRST TRY 108 5.3 LESS LOSS IS WHAT WE WANT 109 FROM PROBLEM BACK TO PYTORCH 110 VIII CONTENTS 5.4 DOWN ALONG THE GRADIENT 113 DECREASING LOSS 113 - GETTING ANALYTICAL 114 - ITERATING TO FIT THE MODEL 116 - NORMALIZING INPUTS 119 : VISUALIZING (AGAIN) 122 5.5 PYTORCH'S AUTOGRAD: BACKPROPAGATING ALL THINGS 123 COMPUTING THE GRADIENT AUTOMATICALLY 123 - OPTIMIZERS A LA CARTE 127 : TRAINING, VALIDATION, AND OVERFITTING 131 AUTOGRAD NITS AND SWITCHING IT OFF. 137 5.6 CONCLUSION 139 5.7 EXERCISE 139 5.8 SUMMARY 139 USING A NEURAL NETWORK TO FIT THE DATA 141 6.1 ARTIFICIAL NEURONS142 COMPOSING A MULTILAYER NETWORK 144 - UNDERSTANDING THE ERROR FUNCTION 144 : ALL WE NEED IS ACTIVATION 145 : MORE ACTIVATION FUNCTIONS 147 : CHOOSING THE BEST ACTIVATION FUNCTION 148 WHAT LEARNING MEANS FOR A NEURAL NETWORK 149 6.2 THE PYTORCH NN MODULE 151 USING--CALL- RATHER THAN FORWARD 152 - RETURNING TO THE LINEAR MODEL 153 6.3 FINALLY A NEURAL NETWORK 158 REPLACING THE LINEAR MODEL 158 - INSPECTING THE PARAMETERS 159 COMPARING TO THE LINEAR MODEL 161 6.4 CONCLUSION 162 6.5 EXERCISES 162 6.6 SUMMARY 163 TELLING BIRDS FROM AIRPLANES: LEARNING FROM IMAGES 164 7.1 A DATASET OF TINY IMAGES 165 DOWNLOADING CIFAR-10 166 : THE DATASET CLASS 166 DATASET TRANSFORMS 168 - NORMALIZING DATA 170 7.2 DISTINGUISHING BIRDS FROM AIRPLANES 172 BUILDING THE DATASET 173 - A FULLY CONNECTED MODEL 174 OUTPUT OF A CLASSIFIER 175 : REPRESENTING THE OUTPUT AS PROBABILITIES 176 - A LOSS FOR CLASSIFYING 180 : TRAINING THE CLASSIFIER 182 - THE LIMITS OF GOING FULLY CONNECTED 189 7.3 CONCLUSION 191 IX CONTENTS 7.4 EXERCISES 191 7.5 SUMMARY 192 USING CONVOLUTIONS TO GENERALIZE 193 8.1 THE CASE FOR CONVOLUTIONS 194 WHAT CONVOLUTIONS DO 194 8.2 CONVOLUTIONS IN ACTION 196 PADDING THE BOUNDARY 198 - DETECTING FEATURES WITH CONVOLUTIONS 200 - LOOKING FURTHER WITH DEPTH AND POOLING 202 PUTTING IT ALL TOGETHER FOR OUR NETWORK 205 8.3 SUBCLASSING NN.MODULE 207 OUR NETWORK AS AN NN.MODULE 208 - HOW PYTORCH KEEPS TRACK OF PARAMETERS AND SUBMODULES 209 : THE FUNCTIONAL API 210 8.4 TRAINING OUR CONVNET 212 MEASURING ACCURACY 214 : SAVING AND LOADING OUR MODEL 214 TRAINING ON THE GPU 215 8.5 MODEL DESIGN 217 ADDING MEMORY CAPACITY: WIDTH 218 - HELPING OUR MODEL TO CONVERGE AND GENERALIZE: REGULARIZATION 219 - GOING DEEPER TO LEARN MORE COMPLEX STRUCTURES: DEPTH223 - COMPARING THE DESIGNS FROM THIS SECTION 228 - IT'S ALREADY OUTDATED 229 8.6 CONCLUSION 229 8.7 EXERCISES 230 8.8 SUMMARY 231 PART 2 LEARNING FROM IMAGES IN THE REAL WORLD: EARLY DETECTION OF LUNG CANCER. 233 USING PYTORCH TO FIGHT CANCER 235 9.1 INTRODUCTION TO THE USE CASE 236 9.2 PREPARING FOR A LARGE-SCALE PROJECT237 9.3 WHAT IS A CT SCAN, EXACTLY?238 9.4 THE PROJECT: AN END-TO-END DETECTOR FOR LUNG CANCER 241 WHY CAN'T WE JUST THROW DATA AT A NEURAL NETWORK UNTIL IT WORKS? 245 - WHAT IS A NODULE? 249 - OUR DATA SOURCE: THE LUNA GRAND CHALLENGE 251 - DOWNLOADING THE LUNA DATA 251DEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING UNTRAINED CLOUD MULTIPROCESS MODEL DATA LOADING SAMPLE BATCH TRAINED TENSORS TENSOR PRODUCTION MODEL DATA (ONNX,JIT SOURCE TRAINING TORCHSCRIPT) LOOP DISTRIBUTED TRAINING PRODUCTION ON MULTIPLE SERVERS/GPUS SERVER DEEP LEARNING WITH PYTORCH ELI STEVENS,LUCA ANTIGA AND THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING SHELTER ISLAND FOR ONLINE INFORMATION AND ORDERING OF THIS AND OTHER MANNING BOOKS, PLEASE VISIT WWW.MANNING.COM. THE PUBLISHER OFFERS DISCOUNTS ON THIS BOOK WHEN ORDERED IN QUANTITY FOR MORE INFORMATION, PLEASE CONTACT SPECIAL SALES DEPARTMENT MANNING PUBLICATIONS CO. 20 BALDWIN ROAD PO BOX 761 SHELTER ISLAND, NY 11964 EMAIL: ORDERS@MANNING.COM @2020 BY MANNING PUBLICATIONS CO. ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED, STORED IN A RETRIEVAL SYSTEM, OR TRANSMITTED, IN ANY FORM OR BY MEANS ELECTRONIC, MECHANICAL, PHOTOCOPYING, OR OTHERWISE, WITHOUT PRIOR WRITTEN PERMISSION OF THE PUBLISHER. MANY OF THE DESIGNATIONS USED BY MANUFACTURERS AND SELLERS TO DISTINGUISH THEIR PRODUCTS ARE CLAIMED AS TRADEMARKS. WHERE THOSE DESIGNATIONS APPEAR IN THE BOOK, AND MANNING PUBLICATIONS WAS AWARE OF A TRADEMARK CLAIM, THE DESIGNATIONS HAVE BEEN PRINTED IN INITIAL CAPS OR ALL CAPS. @ RECOGNIZING THE IMPORTANCE OF PRESERVING WHAT HAS BEEN WRITTEN, IT IS MANNING'S POLICY TO HAVE THE BOOKS WE PUBLISH PRINTED ON ACID-FREE PAPER, AND WE EXERT OUR BEST EFFORTS TO THAT END. RECOGNIZING ALSO OUR RESPONSIBILITY TO CONSERVE THE RESOURCES OF OUR PLANET, MANNING BOOKS ARE PRINTED ON PAPER THAT IS AT LEAST 15 PERCENT RECYCLED AND PROCESSED WITHOUT THE USE OF ELEMENTAL CHLORINE. MANNING PUBLICATIONS CO. DEVELOPMENT EDITOR: FRANCES LEFKOWITZ 20 BALDWIN ROAD TECHNICAL DEVELOPMENT EDITOR: ARTHUR ZUBAREV PO BOX 761 REVIEW EDITOR: IVAN MARTINOVIC SHELTER ISLAND, NY 11964 PRODUCTION EDITOR: DEIRDRE HIAM COPYEDITOR: TIFFANY TAYLOR PROOFREADER: KATIE TENNANT TECHNICAL PROOFREADER: KOSTAS PASSADIS TYPESETTER: GORDAN SALINOVIC COVER DESIGNER: MARIJA TUDOR ISBN 9781617295263 PRINTED IN THE UNITED STATES OF AMERICA TO MY WIFE (THIS BOOK WOULD NOT HAVE HAPPENED WITHOUT HER INVALUABLE SUPPORT AND PARTNERSHIP), MY PARENTS (I WOULD NOT HAVE HAPPENED WITHOUT THEM), AND MY CHILDREN (THIS BOOK WOULD HAVE HAPPENED A LOT SOONER BUT FOR THEM) THANK YOU FOR BEING MY HOME, MY FOUNDATION, AND MY JOY ELI STEVENS SAME :-) BUT, REALLY, THIS IS FOR YOU, ALICE AND LUIGI LUCA ANTIGA TO EVA, REBEKKA, JONATHAN, AND DAVID. THOMAS VIEHMANN CONTENTS FOREWORD XV PREFACE XVII ACKNOWLEDGMENTS XIX ABOUT THIS BOOK XXI ABOUT THE AUTHORS XXVII ABOUT THE COVER ILLUSTRATION XXVIII PART 1 CORE PYTORCH. INTRODUCING DEEP LEARNING AND THE PYTORCH LIBRARY3 1.1 THE DEEP LEARNING REVOLUTION 4 1.2 PYTORCH FOR DEEP LEARNING 6 1.3 WHY PYTORCH?7 THE DEEP LEARNING COMPETITIVE LANDSCAPE8 1.4 AN OVERVIEW OF HOW PYTORCH SUPPORTS DEEP LEARNING PROJECTS 10 1.5 HARDWARE AND SOFTWARE REQUIREMENTS 13 USING JUPYTER NOTEBOOKS 14 1.6 EXERCISES 15 1.7 SUMMARY 15 VI CONTENTS PRETRAINED NETWORKS 16 2.1 A PRETRAINED NETWORK THAT RECOGNIZES THE SUBJECT OF AN IMAGE 17 OBTAINING A PRETRAINED NETWORK FOR IMAGE RECOGNITION 19 ALEXNET 20 - RESNET 22 - READY, SET, ALMOST RUN 22 RUN! 25 2.2 A PRETRAINED MODEL THAT FAKES IT UNTIL IT MAKES IT 27 THE GAN GAME 28 : CYCLEGAN 29 - A NETWORK THAT TURNS HORSES INTO ZEBRAS 30 2.3 A PRETRAINED NETWORK THAT DESCRIBES SCENES 33 NEURALTALK2 34 2.4 TORCH HUB 35 2.5 CONCLUSION 37 2.6 EXERCISES 38 2.7 SUMMARY 38 IT STARTS WITH A TENSOR 39 3.1 THE WORLD AS FLOATING-POINT NUMBERS 40 3.2 TENSORS: MULTIDIMENSIONAL ARRAYS 42 FROM PYTHON LISTS TO PYTORCH TENSORS 42 - CONSTRUCTING OUR FIRST TENSORS 43 - THE ESSENCE OF TENSORS 43 3.3 INDEXING TENSORS 46 3.4 NAMED TENSORS 46 3.5 TENSOR ELEMENT TYPES 50 SPECIFYING THE NUMERIC TYPE WITH DTYPE 50 - A DTYPE FOR EVERY OCCASION 51 - MANAGING A TENSOR'S DTYPE ATTRIBUTE 51 3.6 THE TENSOR API 52 3.7 TENSORS: SCENIC VIEWS OF STORAGE 53 INDEXING INTO STORAGE 54 - MODIFYING STORED VALUES: IN-PLACE OPERATIONS 55 3.8 TENSOR METADATA: SIZE, OFFSET, AND STRIDE 55 VIEWS OF ANOTHER TENSOR'S STORAGE 56 - TRANSPOSING WITHOUT COPYING 58 : TRANSPOSING IN HIGHER DIMENSIONS 60 CONTIGUOUS TENSORS 60 3.9 MOVING TENSORS TO THE GPU 62 MANAGING A TENSOR'S DEVICE ATTRIBUTE 63 VII CONTENTS 3.10 NUMPY INTEROPERABILITY64 3.11 GENERALIZED TENSORS ARE TENSORS, TOO 65 3.12 SERIALIZING TENSORS 66 SERIALIZING TO HDF5 WITH H5PY 67 3.13 CONCLUSION 68 3.14 EXERCISES 68 3.15 SUMMARY 68 REAL-WORLD DATA REPRESENTATION USING TENSORS 70 4.1 WORKING WITH IMAGES 71 ADDING COLOR CHANNELS 72 - LOADING AN IMAGE FILE 72 CHANGING THE LAYOUT 73 : NORMALIZING THE DATA 74 4.2 3D IMAGES: VOLUMETRIC DATA 75 LOADING A SPECIALIZED FORMAT 76 4.3 REPRESENTING TABULAR DATA 77 USING A REAL-WORLD DATASET 77 - LOADING A WINE DATA TENSOR 78 REPRESENTING SCORES 81 : ONE-HOT ENCODING 81 : WHEN TO CATEGORIZE 83 - FINDING THRESHOLDS 84 4.4 WORKING WITH TIME SERIES 87 ADDING A TIME DIMENSION 88 - SHAPING THE DATA BY TIME PERIOD 89 - READY FOR TRAINING 90 4.5 REPRESENTING TEXT 93 CONVERTING TEXT TO NUMBERS94 - ONE-HOT-ENCODING CHARACTERS94 ONE-HOT ENCODING WHOLE WORDS 96 : TEXT EMBEDDINGS 98 TEXT EMBEDDINGS AS A BLUEPRINT 100 4.6 CONCLUSION 101 4.7 EXERCISES 101 4.8 SUMMARY 102 THE MECHANICS OF LEARNING 103 5.1 A TIMELESS LESSON IN MODELING 104 5.2 LEARNING IS JUST PARAMETER ESTIMATION 106 A HOT PROBLEM 107 : GATHERING SOME DATA 107 : VISUALIZING THE DATA 108 : CHOOSING A LINEAR MODEL AS A FIRST TRY 108 5.3 LESS LOSS IS WHAT WE WANT 109 FROM PROBLEM BACK TO PYTORCH 110 VIII CONTENTS 5.4 DOWN ALONG THE GRADIENT 113 DECREASING LOSS 113 - GETTING ANALYTICAL 114 - ITERATING TO FIT THE MODEL 116 - NORMALIZING INPUTS 119 : VISUALIZING (AGAIN) 122 5.5 PYTORCH'S AUTOGRAD: BACKPROPAGATING ALL THINGS 123 COMPUTING THE GRADIENT AUTOMATICALLY 123 - OPTIMIZERS A LA CARTE 127 : TRAINING, VALIDATION, AND OVERFITTING 131 AUTOGRAD NITS AND SWITCHING IT OFF. 137 5.6 CONCLUSION 139 5.7 EXERCISE 139 5.8 SUMMARY 139 USING A NEURAL NETWORK TO FIT THE DATA 141 6.1 ARTIFICIAL NEURONS142 COMPOSING A MULTILAYER NETWORK 144 - UNDERSTANDING THE ERROR FUNCTION 144 : ALL WE NEED IS ACTIVATION 145 : MORE ACTIVATION FUNCTIONS 147 : CHOOSING THE BEST ACTIVATION FUNCTION 148 WHAT LEARNING MEANS FOR A NEURAL NETWORK 149 6.2 THE PYTORCH NN MODULE 151 USING--CALL- RATHER THAN FORWARD 152 - RETURNING TO THE LINEAR MODEL 153 6.3 FINALLY A NEURAL NETWORK 158 REPLACING THE LINEAR MODEL 158 - INSPECTING THE PARAMETERS 159 COMPARING TO THE LINEAR MODEL 161 6.4 CONCLUSION 162 6.5 EXERCISES 162 6.6 SUMMARY 163 TELLING BIRDS FROM AIRPLANES: LEARNING FROM IMAGES 164 7.1 A DATASET OF TINY IMAGES 165 DOWNLOADING CIFAR-10 166 : THE DATASET CLASS 166 DATASET TRANSFORMS 168 - NORMALIZING DATA 170 7.2 DISTINGUISHING BIRDS FROM AIRPLANES 172 BUILDING THE DATASET 173 - A FULLY CONNECTED MODEL 174 OUTPUT OF A CLASSIFIER 175 : REPRESENTING THE OUTPUT AS PROBABILITIES 176 - A LOSS FOR CLASSIFYING 180 : TRAINING THE CLASSIFIER 182 - THE LIMITS OF GOING FULLY CONNECTED 189 7.3 CONCLUSION 191 IX CONTENTS 7.4 EXERCISES 191 7.5 SUMMARY 192 USING CONVOLUTIONS TO GENERALIZE 193 8.1 THE CASE FOR CONVOLUTIONS 194 WHAT CONVOLUTIONS DO 194 8.2 CONVOLUTIONS IN ACTION 196 PADDING THE BOUNDARY 198 - DETECTING FEATURES WITH CONVOLUTIONS 200 - LOOKING FURTHER WITH DEPTH AND POOLING 202 PUTTING IT ALL TOGETHER FOR OUR NETWORK 205 8.3 SUBCLASSING NN.MODULE 207 OUR NETWORK AS AN NN.MODULE 208 - HOW PYTORCH KEEPS TRACK OF PARAMETERS AND SUBMODULES 209 : THE FUNCTIONAL API 210 8.4 TRAINING OUR CONVNET 212 MEASURING ACCURACY 214 : SAVING AND LOADING OUR MODEL 214 TRAINING ON THE GPU 215 8.5 MODEL DESIGN 217 ADDING MEMORY CAPACITY: WIDTH 218 - HELPING OUR MODEL TO CONVERGE AND GENERALIZE: REGULARIZATION 219 - GOING DEEPER TO LEARN MORE COMPLEX STRUCTURES: DEPTH223 - COMPARING THE DESIGNS FROM THIS SECTION 228 - IT'S ALREADY OUTDATED 229 8.6 CONCLUSION 229 8.7 EXERCISES 230 8.8 SUMMARY 231 PART 2 LEARNING FROM IMAGES IN THE REAL WORLD: EARLY DETECTION OF LUNG CANCER. 233 USING PYTORCH TO FIGHT CANCER 235 9.1 INTRODUCTION TO THE USE CASE 236 9.2 PREPARING FOR A LARGE-SCALE PROJECT237 9.3 WHAT IS A CT SCAN, EXACTLY?238 9.4 THE PROJECT: AN END-TO-END DETECTOR FOR LUNG CANCER 241 WHY CAN'T WE JUST THROW DATA AT A NEURAL NETWORK UNTIL IT WORKS? 245 - WHAT IS A NODULE? 249 - OUR DATA SOURCE: THE LUNA GRAND CHALLENGE 251 - DOWNLOADING THE LUNA DATA 251 CONTENTS X 9.5 CONCLUSION 252 9.6 SUMMARY 253 COMBINING DATA SOURCES INTO A UNIFIED DATASET 254 1 10.1 RAW CT DATA FILES 256 10.2 PARSING LUNA'S ANNOTATION DATA 256 TRAINING AND VALIDATION SETS 258 : UNIFYING OUR ANNOTATION AND CANDIDATE DATA 259 10.3 LOADING INDIVIDUAL CT SCANS 262 HOUNSFIELD UNITS 264 10.4 LOCATING A NODULE USING THE PATIENT COORDINATE SYSTEM 265 THE PATIENT COORDINATE SYSTEM 265 - CT SCAN SHAPE AND VOXEL SIZES 267 - CONVERTING BETWEEN MILLIMETERS AND VOXEL ADDRESSES 268 - EXTRACTING A NODULE FROM A CT SCAN 270 10.5 A STRAIGHTFORWARD DATASET IMPLEMENTATION 271 CACHING CANDIDATE ARRAYS WITH THE GETCTRAWCANDIDATE FUNCTION 274 : CONSTRUCTING OUR DATASET IN LUNADATASET \_INIT\_- 275 - A TRAINING/VALIDATION SPLIT 275 - RENDERING THE DATA 277 10.6 CONCLUSION 277 10.7 EXERCISES 278 10.8 SUMMARY 278 TRAINING A CLASSIFICATION MODEL TO DETECT SUSPECTED TUMORS 279 11.1 A FOUNDATIONAL MODEL AND TRAINING LOOP 280 11.2 THE MAIN ENTRY POINT FOR OUR APPLICATION 282 11.3 PRETRAINING SETUP AND INITIALIZATION 284 INITIALIZING THE MODEL AND OPTIMIZER 285 : CARE AND FEEDING OF DATA LOADERS 287 11.4 OUR FIRST-PASS NEURAL NETWORK DESIGN 289 THE CORE CONVOLUTIONS 290 : THE FULL MODEL 293 11.5 TRAINING AND VALIDATING THE MODEL 295 THE COMPUTEBATCHLOSS FUNCTION 297 - THE VALIDATION LOOP IS SIMILAR 299 11.6 OUTPUTTING PERFORMANCE METRICS 300 THE LOGMETRICS FUNCTION 301DEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING UNTRAINED CLOUD MULTIPROCESS MODEL DATA LOADING SAMPLE BATCH TRAINED TENSORS TENSOR PRODUCTION MODEL DATA (ONNX,JIT SOURCE TRAINING TORCHSCRIPT) LOOP DISTRIBUTED TRAINING PRODUCTION ON MULTIPLE SERVERS/GPUS SERVER DEEP LEARNING WITH PYTORCH ELI STEVENS,LUCA ANTIGA AND THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING SHELTER ISLAND FOR ONLINE INFORMATION AND ORDERING OF THIS AND OTHER MANNING BOOKS, PLEASE VISIT WWW.MANNING.COM. THE PUBLISHER OFFERS DISCOUNTS ON THIS BOOK WHEN ORDERED IN QUANTITY FOR MORE INFORMATION, PLEASE CONTACT SPECIAL SALES DEPARTMENT MANNING PUBLICATIONS CO. 20 BALDWIN ROAD PO BOX 761 SHELTER ISLAND, NY 11964 EMAIL: ORDERS@MANNING.COM @2020 BY MANNING PUBLICATIONS CO. ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED, STORED IN A RETRIEVAL SYSTEM, OR TRANSMITTED, IN ANY FORM OR BY MEANS ELECTRONIC, MECHANICAL, PHOTOCOPYING, OR OTHERWISE, WITHOUT PRIOR WRITTEN PERMISSION OF THE PUBLISHER. MANY OF THE DESIGNATIONS USED BY MANUFACTURERS AND SELLERS TO DISTINGUISH THEIR PRODUCTS ARE CLAIMED AS TRADEMARKS. WHERE THOSE DESIGNATIONS APPEAR IN THE BOOK, AND MANNING PUBLICATIONS WAS AWARE OF A TRADEMARK CLAIM, THE DESIGNATIONS HAVE BEEN PRINTED IN INITIAL CAPS OR ALL CAPS. @ RECOGNIZING THE IMPORTANCE OF PRESERVING WHAT HAS BEEN WRITTEN, IT IS MANNING'S POLICY TO HAVE THE BOOKS WE PUBLISH PRINTED ON ACID-FREE PAPER, AND WE EXERT OUR BEST EFFORTS TO THAT END. RECOGNIZING ALSO OUR RESPONSIBILITY TO CONSERVE THE RESOURCES OF OUR PLANET, MANNING BOOKS ARE PRINTED ON PAPER THAT IS AT LEAST 15 PERCENT RECYCLED AND PROCESSED WITHOUT THE USE OF ELEMENTAL CHLORINE. MANNING PUBLICATIONS CO. DEVELOPMENT EDITOR: FRANCES LEFKOWITZ 20 BALDWIN ROAD TECHNICAL DEVELOPMENT EDITOR: ARTHUR ZUBAREV PO BOX 761 REVIEW EDITOR: IVAN MARTINOVIC SHELTER ISLAND, NY 11964 PRODUCTION EDITOR: DEIRDRE HIAM COPYEDITOR: TIFFANY TAYLOR PROOFREADER: KATIE TENNANT TECHNICAL PROOFREADER: KOSTAS PASSADIS TYPESETTER: GORDAN SALINOVIC COVER DESIGNER: MARIJA TUDOR ISBN 9781617295263 PRINTED IN THE UNITED STATES OF AMERICA TO MY WIFE (THIS BOOK WOULD NOT HAVE HAPPENED WITHOUT HER INVALUABLE SUPPORT AND PARTNERSHIP), MY PARENTS (I WOULD NOT HAVE HAPPENED WITHOUT THEM), AND MY CHILDREN (THIS BOOK WOULD HAVE HAPPENED A LOT SOONER BUT FOR THEM) THANK YOU FOR BEING MY HOME, MY FOUNDATION, AND MY JOY ELI STEVENS SAME :-) BUT, REALLY, THIS IS FOR YOU, ALICE AND LUIGI LUCA ANTIGA TO EVA, REBEKKA, JONATHAN, AND DAVID. THOMAS VIEHMANN CONTENTS FOREWORD XV PREFACE XVII ACKNOWLEDGMENTS XIX ABOUT THIS BOOK XXI ABOUT THE AUTHORS XXVII ABOUT THE COVER ILLUSTRATION XXVIII PART 1 CORE PYTORCH. INTRODUCING DEEP LEARNING AND THE PYTORCH LIBRARY3 1.1 THE DEEP LEARNING REVOLUTION 4 1.2 PYTORCH FOR DEEP LEARNING 6 1.3 WHY PYTORCH?7 THE DEEP LEARNING COMPETITIVE LANDSCAPE8 1.4 AN OVERVIEW OF HOW PYTORCH SUPPORTS DEEP LEARNING PROJECTS 10 1.5 HARDWARE AND SOFTWARE REQUIREMENTS 13 USING JUPYTER NOTEBOOKS 14 1.6 EXERCISES 15 1.7 SUMMARY 15 VI CONTENTS PRETRAINED NETWORKS 16 2.1 A PRETRAINED NETWORK THAT RECOGNIZES THE SUBJECT OF AN IMAGE 17 OBTAINING A PRETRAINED NETWORK FOR IMAGE RECOGNITION 19 ALEXNET 20 - RESNET 22 - READY, SET, ALMOST RUN 22 RUN! 25 2.2 A PRETRAINED MODEL THAT FAKES IT UNTIL IT MAKES IT 27 THE GAN GAME 28 : CYCLEGAN 29 - A NETWORK THAT TURNS HORSES INTO ZEBRAS 30 2.3 A PRETRAINED NETWORK THAT DESCRIBES SCENES 33 NEURALTALK2 34 2.4 TORCH HUB 35 2.5 CONCLUSION 37 2.6 EXERCISES 38 2.7 SUMMARY 38 IT STARTS WITH A TENSOR 39 3.1 THE WORLD AS FLOATING-POINT NUMBERS 40 3.2 TENSORS: MULTIDIMENSIONAL ARRAYS 42 FROM PYTHON LISTS TO PYTORCH TENSORS 42 - CONSTRUCTING OUR FIRST TENSORS 43 - THE ESSENCE OF TENSORS 43 3.3 INDEXING TENSORS 46 3.4 NAMED TENSORS 46 3.5 TENSOR ELEMENT TYPES 50 SPECIFYING THE NUMERIC TYPE WITH DTYPE 50 - A DTYPE FOR EVERY OCCASION 51 - MANAGING A TENSOR'S DTYPE ATTRIBUTE 51 3.6 THE TENSOR API 52 3.7 TENSORS: SCENIC VIEWS OF STORAGE 53 INDEXING INTO STORAGE 54 - MODIFYING STORED VALUES: IN-PLACE OPERATIONS 55 3.8 TENSOR METADATA: SIZE, OFFSET, AND STRIDE 55 VIEWS OF ANOTHER TENSOR'S STORAGE 56 - TRANSPOSING WITHOUT COPYING 58 : TRANSPOSING IN HIGHER DIMENSIONS 60 CONTIGUOUS TENSORS 60 3.9 MOVING TENSORS TO THE GPU 62 MANAGING A TENSOR'S DEVICE ATTRIBUTE 63 VII CONTENTS 3.10 NUMPY INTEROPERABILITY64 3.11 GENERALIZED TENSORS ARE TENSORS, TOO 65 3.12 SERIALIZING TENSORS 66 SERIALIZING TO HDF5 WITH H5PY 67 3.13 CONCLUSION 68 3.14 EXERCISES 68 3.15 SUMMARY 68 REAL-WORLD DATA REPRESENTATION USING TENSORS 70 4.1 WORKING WITH IMAGES 71 ADDING COLOR CHANNELS 72 - LOADING AN IMAGE FILE 72 CHANGING THE LAYOUT 73 : NORMALIZING THE DATA 74 4.2 3D IMAGES: VOLUMETRIC DATA 75 LOADING A SPECIALIZED FORMAT 76 4.3 REPRESENTING TABULAR DATA 77 USING A REAL-WORLD DATASET 77 - LOADING A WINE DATA TENSOR 78 REPRESENTING SCORES 81 : ONE-HOT ENCODING 81 : WHEN TO CATEGORIZE 83 - FINDING THRESHOLDS 84 4.4 WORKING WITH TIME SERIES 87 ADDING A TIME DIMENSION 88 - SHAPING THE DATA BY TIME PERIOD 89 - READY FOR TRAINING 90 4.5 REPRESENTING TEXT 93 CONVERTING TEXT TO NUMBERS94 - ONE-HOT-ENCODING CHARACTERS94 ONE-HOT ENCODING WHOLE WORDS 96 : TEXT EMBEDDINGS 98 TEXT EMBEDDINGS AS A BLUEPRINT 100 4.6 CONCLUSION 101 4.7 EXERCISES 101 4.8 SUMMARY 102 THE MECHANICS OF LEARNING 103 5.1 A TIMELESS LESSON IN MODELING 104 5.2 LEARNING IS JUST PARAMETER ESTIMATION 106 A HOT PROBLEM 107 : GATHERING SOME DATA 107 : VISUALIZING THE DATA 108 : CHOOSING A LINEAR MODEL AS A FIRST TRY 108 5.3 LESS LOSS IS WHAT WE WANT 109 FROM PROBLEM BACK TO PYTORCH 110 VIII CONTENTS 5.4 DOWN ALONG THE GRADIENT 113 DECREASING LOSS 113 - GETTING ANALYTICAL 114 - ITERATING TO FIT THE MODEL 116 - NORMALIZING INPUTS 119 : VISUALIZING (AGAIN) 122 5.5 PYTORCH'S AUTOGRAD: BACKPROPAGATING ALL THINGS 123 COMPUTING THE GRADIENT AUTOMATICALLY 123 - OPTIMIZERS A LA CARTE 127 : TRAINING, VALIDATION, AND OVERFITTING 131 AUTOGRAD NITS AND SWITCHING IT OFF. 137 5.6 CONCLUSION 139 5.7 EXERCISE 139 5.8 SUMMARY 139 USING A NEURAL NETWORK TO FIT THE DATA 141 6.1 ARTIFICIAL NEURONS142 COMPOSING A MULTILAYER NETWORK 144 - UNDERSTANDING THE ERROR FUNCTION 144 : ALL WE NEED IS ACTIVATION 145 : MORE ACTIVATION FUNCTIONS 147 : CHOOSING THE BEST ACTIVATION FUNCTION 148 WHAT LEARNING MEANS FOR A NEURAL NETWORK 149 6.2 THE PYTORCH NN MODULE 151 USING--CALL- RATHER THAN FORWARD 152 - RETURNING TO THE LINEAR MODEL 153 6.3 FINALLY A NEURAL NETWORK 158 REPLACING THE LINEAR MODEL 158 - INSPECTING THE PARAMETERS 159 COMPARING TO THE LINEAR MODEL 161 6.4 CONCLUSION 162 6.5 EXERCISES 162 6.6 SUMMARY 163 TELLING BIRDS FROM AIRPLANES: LEARNING FROM IMAGES 164 7.1 A DATASET OF TINY IMAGES 165 DOWNLOADING CIFAR-10 166 : THE DATASET CLASS 166 DATASET TRANSFORMS 168 - NORMALIZING DATA 170 7.2 DISTINGUISHING BIRDS FROM AIRPLANES 172 BUILDING THE DATASET 173 - A FULLY CONNECTED MODEL 174 OUTPUT OF A CLASSIFIER 175 : REPRESENTING THE OUTPUT AS PROBABILITIES 176 - A LOSS FOR CLASSIFYING 180 : TRAINING THE CLASSIFIER 182 - THE LIMITS OF GOING FULLY CONNECTED 189 7.3 CONCLUSION 191 IX CONTENTS 7.4 EXERCISES 191 7.5 SUMMARY 192 USING CONVOLUTIONS TO GENERALIZE 193 8.1 THE CASE FOR CONVOLUTIONS 194 WHAT CONVOLUTIONS DO 194 8.2 CONVOLUTIONS IN ACTION 196 PADDING THE BOUNDARY 198 - DETECTING FEATURES WITH CONVOLUTIONS 200 - LOOKING FURTHER WITH DEPTH AND POOLING 202 PUTTING IT ALL TOGETHER FOR OUR NETWORK 205 8.3 SUBCLASSING NN.MODULE 207 OUR NETWORK AS AN NN.MODULE 208 - HOW PYTORCH KEEPS TRACK OF PARAMETERS AND SUBMODULES 209 : THE FUNCTIONAL API 210 8.4 TRAINING OUR CONVNET 212 MEASURING ACCURACY 214 : SAVING AND LOADING OUR MODEL 214 TRAINING ON THE GPU 215 8.5 MODEL DESIGN 217 ADDING MEMORY CAPACITY: WIDTH 218 - HELPING OUR MODEL TO CONVERGE AND GENERALIZE: REGULARIZATION 219 - GOING DEEPER TO LEARN MORE COMPLEX STRUCTURES: DEPTH223 - COMPARING THE DESIGNS FROM THIS SECTION 228 - IT'S ALREADY OUTDATED 229 8.6 CONCLUSION 229 8.7 EXERCISES 230 8.8 SUMMARY 231 PART 2 LEARNING FROM IMAGES IN THE REAL WORLD: EARLY DETECTION OF LUNG CANCER. 233 USING PYTORCH TO FIGHT CANCER 235 9.1 INTRODUCTION TO THE USE CASE 236 9.2 PREPARING FOR A LARGE-SCALE PROJECT237 9.3 WHAT IS A CT SCAN, EXACTLY?238 9.4 THE PROJECT: AN END-TO-END DETECTOR FOR LUNG CANCER 241 WHY CAN'T WE JUST THROW DATA AT A NEURAL NETWORK UNTIL IT WORKS? 245 - WHAT IS A NODULE? 249 - OUR DATA SOURCE: THE LUNA GRAND CHALLENGE 251 - DOWNLOADING THE LUNA DATA 251 CONTENTS X 9.5 CONCLUSION 252 9.6 SUMMARY 253 COMBINING DATA SOURCES INTO A UNIFIED DATASET 254 1 10.1 RAW CT DATA FILES 256 10.2 PARSING LUNA'S ANNOTATION DATA 256 TRAINING AND VALIDATION SETS 258 : UNIFYING OUR ANNOTATION AND CANDIDATE DATA 259 10.3 LOADING INDIVIDUAL CT SCANS 262 HOUNSFIELD UNITS 264 10.4 LOCATING A NODULE USING THE PATIENT COORDINATE SYSTEM 265 THE PATIENT COORDINATE SYSTEM 265 - CT SCAN SHAPE AND VOXEL SIZES 267 - CONVERTING BETWEEN MILLIMETERS AND VOXEL ADDRESSES 268 - EXTRACTING A NODULE FROM A CT SCAN 270 10.5 A STRAIGHTFORWARD DATASET IMPLEMENTATION 271 CACHING CANDIDATE ARRAYS WITH THE GETCTRAWCANDIDATE FUNCTION 274 : CONSTRUCTING OUR DATASET IN LUNADATASET \_INIT\_- 275 - A TRAINING/VALIDATION SPLIT 275 - RENDERING THE DATA 277 10.6 CONCLUSION 277 10.7 EXERCISES 278 10.8 SUMMARY 278 TRAINING A CLASSIFICATION MODEL TO DETECT SUSPECTED TUMORS 279 11.1 A FOUNDATIONAL MODEL AND TRAINING LOOP 280 11.2 THE MAIN ENTRY POINT FOR OUR APPLICATION 282 11.3 PRETRAINING SETUP AND INITIALIZATION 284 INITIALIZING THE MODEL AND OPTIMIZER 285 : CARE AND FEEDING OF DATA LOADERS 287 11.4 OUR FIRST-PASS NEURAL NETWORK DESIGN 289 THE CORE CONVOLUTIONS 290 : THE FULL MODEL 293 11.5 TRAINING AND VALIDATING THE MODEL 295 THE COMPUTEBATCHLOSS FUNCTION 297 - THE VALIDATION LOOP IS SIMILAR 299 11.6 OUTPUTTING PERFORMANCE METRICS 300 THE LOGMETRICS FUNCTION 301 XI CONTENTS 11.7 RUNNING THE TRAINING SCRIPT 304 NEEDED DATA FOR TRAINING 305 - INTERLUDE: THE ENUMERATEWITHESTIMATE FUNCTION 306 11.8 EVALUATING THE MODEL: GETTING 99.7% CORRECT MEANS WE'RE DONE, RIGHT? 308 11.9 GRAPHING TRAINING METRICS WITH TENSORBOARD 309 RUNNING TENSORBOARD 309 - ADDING TENSORBOARD SUPPORT TO THE METRICS LOGGING FUNCTION 313 11.10 WHY ISN'T THE MODEL LEARNING TO DETECT NODULES? 315 11.11 CONCLUSION 316 11.12 EXERCISES 316 11.13 SUMMARY 316 12 IMPROVING TRAINING WITH METRICS AND AUGMENTATION 318 12.1 HIGH-LEVEL PLAN FOR IMPROVEMENT 319 12.2 GOOD DOGS VS. BAD GUYS: FALSE POSITIVES AND FALSE NEGATIVES 320 12.3 GRAPHING THE POSITIVES AND NEGATIVES322 RECALL IS ROXIE'S STRENGTH 324 - PRECISION IS PRESTON'S FORTE 326 IMPLEMENTING PRECISION AND RECALL IN LOGMETRICS 327 : OUR ULTIMATE PERFORMANCE METRIC: THE F1 SCORE 328 - HOW DOES OUR MODEL PERFORM WITH OUR NEW METRICS? 332 12.4 WHAT DOES AN IDEAL DATASET LOOK LIKE? 334 MAKING THE DATA LOOK LESS LIKE THE ACTUAL AND MORE LIKE THE "IDEAL" 336 CONTRASTING TRAINING WITH A BALANCED LUNADATASET TO PREVIOUS RUNS 341 - RECOGNIZING THE SYMPTOMS OF OVERFITTING 343 12.5 REVISITING THE PROBLEM OF OVERFITTING 345 AN OVERFIT FACE-TO-AGE PREDICTION MODEL 345 12.6 PREVENTING OVERFITTING WITH DATA AUGMENTATION 346 SPECIFIC DATA AUGMENTATION TECHNIQUES 347 : SEEING THE IMPROVEMENT FROM DATA AUGMENTATION 352 12.7 CONCLUSION 354 12.8 EXERCISES 355 12.9 SUMMARY 356 USING SEGMENTATION TO FIND SUSPECTED NODULES 357 13.1 ADDING A SECOND MODEL TO OUR PROJECT 358 13.2 VARIOUS TYPES OF SEGMENTATION 360DEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING UNTRAINED CLOUD MULTIPROCESS MODEL DATA LOADING SAMPLE BATCH TRAINED TENSORS TENSOR PRODUCTION MODEL DATA (ONNX,JIT SOURCE TRAINING TORCHSCRIPT) LOOP DISTRIBUTED TRAINING PRODUCTION ON MULTIPLE SERVERS/GPUS SERVER DEEP LEARNING WITH PYTORCH ELI STEVENS,LUCA ANTIGA AND THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING SHELTER ISLAND FOR ONLINE INFORMATION AND ORDERING OF THIS AND OTHER MANNING BOOKS, PLEASE VISIT WWW.MANNING.COM. THE PUBLISHER OFFERS DISCOUNTS ON THIS BOOK WHEN ORDERED IN QUANTITY FOR MORE INFORMATION, PLEASE CONTACT SPECIAL SALES DEPARTMENT MANNING PUBLICATIONS CO. 20 BALDWIN ROAD PO BOX 761 SHELTER ISLAND, NY 11964 EMAIL: ORDERS@MANNING.COM @2020 BY MANNING PUBLICATIONS CO. ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED, STORED IN A RETRIEVAL SYSTEM, OR TRANSMITTED, IN ANY FORM OR BY MEANS ELECTRONIC, MECHANICAL, PHOTOCOPYING, OR OTHERWISE, WITHOUT PRIOR WRITTEN PERMISSION OF THE PUBLISHER. MANY OF THE DESIGNATIONS USED BY MANUFACTURERS AND SELLERS TO DISTINGUISH THEIR PRODUCTS ARE CLAIMED AS TRADEMARKS. WHERE THOSE DESIGNATIONS APPEAR IN THE BOOK, AND MANNING PUBLICATIONS WAS AWARE OF A TRADEMARK CLAIM, THE DESIGNATIONS HAVE BEEN PRINTED IN INITIAL CAPS OR ALL CAPS. @ RECOGNIZING THE IMPORTANCE OF PRESERVING WHAT HAS BEEN WRITTEN, IT IS MANNING'S POLICY TO HAVE THE BOOKS WE PUBLISH PRINTED ON ACID-FREE PAPER, AND WE EXERT OUR BEST EFFORTS TO THAT END. RECOGNIZING ALSO OUR RESPONSIBILITY TO CONSERVE THE RESOURCES OF OUR PLANET, MANNING BOOKS ARE PRINTED ON PAPER THAT IS AT LEAST 15 PERCENT RECYCLED AND PROCESSED WITHOUT THE USE OF ELEMENTAL CHLORINE. MANNING PUBLICATIONS CO. DEVELOPMENT EDITOR: FRANCES LEFKOWITZ 20 BALDWIN ROAD TECHNICAL DEVELOPMENT EDITOR: ARTHUR ZUBAREV PO BOX 761 REVIEW EDITOR: IVAN MARTINOVIC SHELTER ISLAND, NY 11964 PRODUCTION EDITOR: DEIRDRE HIAM COPYEDITOR: TIFFANY TAYLOR PROOFREADER: KATIE TENNANT TECHNICAL PROOFREADER: KOSTAS PASSADIS TYPESETTER: GORDAN SALINOVIC COVER DESIGNER: MARIJA TUDOR ISBN 9781617295263 PRINTED IN THE UNITED STATES OF AMERICA TO MY WIFE (THIS BOOK WOULD NOT HAVE HAPPENED WITHOUT HER INVALUABLE SUPPORT AND PARTNERSHIP), MY PARENTS (I WOULD NOT HAVE HAPPENED WITHOUT THEM), AND MY CHILDREN (THIS BOOK WOULD HAVE HAPPENED A LOT SOONER BUT FOR THEM) THANK YOU FOR BEING MY HOME, MY FOUNDATION, AND MY JOY ELI STEVENS SAME :-) BUT, REALLY, THIS IS FOR YOU, ALICE AND LUIGI LUCA ANTIGA TO EVA, REBEKKA, JONATHAN, AND DAVID. THOMAS VIEHMANN CONTENTS FOREWORD XV PREFACE XVII ACKNOWLEDGMENTS XIX ABOUT THIS BOOK XXI ABOUT THE AUTHORS XXVII ABOUT THE COVER ILLUSTRATION XXVIII PART 1 CORE PYTORCH. INTRODUCING DEEP LEARNING AND THE PYTORCH LIBRARY3 1.1 THE DEEP LEARNING REVOLUTION 4 1.2 PYTORCH FOR DEEP LEARNING 6 1.3 WHY PYTORCH?7 THE DEEP LEARNING COMPETITIVE LANDSCAPE8 1.4 AN OVERVIEW OF HOW PYTORCH SUPPORTS DEEP LEARNING PROJECTS 10 1.5 HARDWARE AND SOFTWARE REQUIREMENTS 13 USING JUPYTER NOTEBOOKS 14 1.6 EXERCISES 15 1.7 SUMMARY 15 VI CONTENTS PRETRAINED NETWORKS 16 2.1 A PRETRAINED NETWORK THAT RECOGNIZES THE SUBJECT OF AN IMAGE 17 OBTAINING A PRETRAINED NETWORK FOR IMAGE RECOGNITION 19 ALEXNET 20 - RESNET 22 - READY, SET, ALMOST RUN 22 RUN! 25 2.2 A PRETRAINED MODEL THAT FAKES IT UNTIL IT MAKES IT 27 THE GAN GAME 28 : CYCLEGAN 29 - A NETWORK THAT TURNS HORSES INTO ZEBRAS 30 2.3 A PRETRAINED NETWORK THAT DESCRIBES SCENES 33 NEURALTALK2 34 2.4 TORCH HUB 35 2.5 CONCLUSION 37 2.6 EXERCISES 38 2.7 SUMMARY 38 IT STARTS WITH A TENSOR 39 3.1 THE WORLD AS FLOATING-POINT NUMBERS 40 3.2 TENSORS: MULTIDIMENSIONAL ARRAYS 42 FROM PYTHON LISTS TO PYTORCH TENSORS 42 - CONSTRUCTING OUR FIRST TENSORS 43 - THE ESSENCE OF TENSORS 43 3.3 INDEXING TENSORS 46 3.4 NAMED TENSORS 46 3.5 TENSOR ELEMENT TYPES 50 SPECIFYING THE NUMERIC TYPE WITH DTYPE 50 - A DTYPE FOR EVERY OCCASION 51 - MANAGING A TENSOR'S DTYPE ATTRIBUTE 51 3.6 THE TENSOR API 52 3.7 TENSORS: SCENIC VIEWS OF STORAGE 53 INDEXING INTO STORAGE 54 - MODIFYING STORED VALUES: IN-PLACE OPERATIONS 55 3.8 TENSOR METADATA: SIZE, OFFSET, AND STRIDE 55 VIEWS OF ANOTHER TENSOR'S STORAGE 56 - TRANSPOSING WITHOUT COPYING 58 : TRANSPOSING IN HIGHER DIMENSIONS 60 CONTIGUOUS TENSORS 60 3.9 MOVING TENSORS TO THE GPU 62 MANAGING A TENSOR'S DEVICE ATTRIBUTE 63 VII CONTENTS 3.10 NUMPY INTEROPERABILITY64 3.11 GENERALIZED TENSORS ARE TENSORS, TOO 65 3.12 SERIALIZING TENSORS 66 SERIALIZING TO HDF5 WITH H5PY 67 3.13 CONCLUSION 68 3.14 EXERCISES 68 3.15 SUMMARY 68 REAL-WORLD DATA REPRESENTATION USING TENSORS 70 4.1 WORKING WITH IMAGES 71 ADDING COLOR CHANNELS 72 - LOADING AN IMAGE FILE 72 CHANGING THE LAYOUT 73 : NORMALIZING THE DATA 74 4.2 3D IMAGES: VOLUMETRIC DATA 75 LOADING A SPECIALIZED FORMAT 76 4.3 REPRESENTING TABULAR DATA 77 USING A REAL-WORLD DATASET 77 - LOADING A WINE DATA TENSOR 78 REPRESENTING SCORES 81 : ONE-HOT ENCODING 81 : WHEN TO CATEGORIZE 83 - FINDING THRESHOLDS 84 4.4 WORKING WITH TIME SERIES 87 ADDING A TIME DIMENSION 88 - SHAPING THE DATA BY TIME PERIOD 89 - READY FOR TRAINING 90 4.5 REPRESENTING TEXT 93 CONVERTING TEXT TO NUMBERS94 - ONE-HOT-ENCODING CHARACTERS94 ONE-HOT ENCODING WHOLE WORDS 96 : TEXT EMBEDDINGS 98 TEXT EMBEDDINGS AS A BLUEPRINT 100 4.6 CONCLUSION 101 4.7 EXERCISES 101 4.8 SUMMARY 102 THE MECHANICS OF LEARNING 103 5.1 A TIMELESS LESSON IN MODELING 104 5.2 LEARNING IS JUST PARAMETER ESTIMATION 106 A HOT PROBLEM 107 : GATHERING SOME DATA 107 : VISUALIZING THE DATA 108 : CHOOSING A LINEAR MODEL AS A FIRST TRY 108 5.3 LESS LOSS IS WHAT WE WANT 109 FROM PROBLEM BACK TO PYTORCH 110 VIII CONTENTS 5.4 DOWN ALONG THE GRADIENT 113 DECREASING LOSS 113 - GETTING ANALYTICAL 114 - ITERATING TO FIT THE MODEL 116 - NORMALIZING INPUTS 119 : VISUALIZING (AGAIN) 122 5.5 PYTORCH'S AUTOGRAD: BACKPROPAGATING ALL THINGS 123 COMPUTING THE GRADIENT AUTOMATICALLY 123 - OPTIMIZERS A LA CARTE 127 : TRAINING, VALIDATION, AND OVERFITTING 131 AUTOGRAD NITS AND SWITCHING IT OFF. 137 5.6 CONCLUSION 139 5.7 EXERCISE 139 5.8 SUMMARY 139 USING A NEURAL NETWORK TO FIT THE DATA 141 6.1 ARTIFICIAL NEURONS142 COMPOSING A MULTILAYER NETWORK 144 - UNDERSTANDING THE ERROR FUNCTION 144 : ALL WE NEED IS ACTIVATION 145 : MORE ACTIVATION FUNCTIONS 147 : CHOOSING THE BEST ACTIVATION FUNCTION 148 WHAT LEARNING MEANS FOR A NEURAL NETWORK 149 6.2 THE PYTORCH NN MODULE 151 USING--CALL- RATHER THAN FORWARD 152 - RETURNING TO THE LINEAR MODEL 153 6.3 FINALLY A NEURAL NETWORK 158 REPLACING THE LINEAR MODEL 158 - INSPECTING THE PARAMETERS 159 COMPARING TO THE LINEAR MODEL 161 6.4 CONCLUSION 162 6.5 EXERCISES 162 6.6 SUMMARY 163 TELLING BIRDS FROM AIRPLANES: LEARNING FROM IMAGES 164 7.1 A DATASET OF TINY IMAGES 165 DOWNLOADING CIFAR-10 166 : THE DATASET CLASS 166 DATASET TRANSFORMS 168 - NORMALIZING DATA 170 7.2 DISTINGUISHING BIRDS FROM AIRPLANES 172 BUILDING THE DATASET 173 - A FULLY CONNECTED MODEL 174 OUTPUT OF A CLASSIFIER 175 : REPRESENTING THE OUTPUT AS PROBABILITIES 176 - A LOSS FOR CLASSIFYING 180 : TRAINING THE CLASSIFIER 182 - THE LIMITS OF GOING FULLY CONNECTED 189 7.3 CONCLUSION 191 IX CONTENTS 7.4 EXERCISES 191 7.5 SUMMARY 192 USING CONVOLUTIONS TO GENERALIZE 193 8.1 THE CASE FOR CONVOLUTIONS 194 WHAT CONVOLUTIONS DO 194 8.2 CONVOLUTIONS IN ACTION 196 PADDING THE BOUNDARY 198 - DETECTING FEATURES WITH CONVOLUTIONS 200 - LOOKING FURTHER WITH DEPTH AND POOLING 202 PUTTING IT ALL TOGETHER FOR OUR NETWORK 205 8.3 SUBCLASSING NN.MODULE 207 OUR NETWORK AS AN NN.MODULE 208 - HOW PYTORCH KEEPS TRACK OF PARAMETERS AND SUBMODULES 209 : THE FUNCTIONAL API 210 8.4 TRAINING OUR CONVNET 212 MEASURING ACCURACY 214 : SAVING AND LOADING OUR MODEL 214 TRAINING ON THE GPU 215 8.5 MODEL DESIGN 217 ADDING MEMORY CAPACITY: WIDTH 218 - HELPING OUR MODEL TO CONVERGE AND GENERALIZE: REGULARIZATION 219 - GOING DEEPER TO LEARN MORE COMPLEX STRUCTURES: DEPTH223 - COMPARING THE DESIGNS FROM THIS SECTION 228 - IT'S ALREADY OUTDATED 229 8.6 CONCLUSION 229 8.7 EXERCISES 230 8.8 SUMMARY 231 PART 2 LEARNING FROM IMAGES IN THE REAL WORLD: EARLY DETECTION OF LUNG CANCER. 233 USING PYTORCH TO FIGHT CANCER 235 9.1 INTRODUCTION TO THE USE CASE 236 9.2 PREPARING FOR A LARGE-SCALE PROJECT237 9.3 WHAT IS A CT SCAN, EXACTLY?238 9.4 THE PROJECT: AN END-TO-END DETECTOR FOR LUNG CANCER 241 WHY CAN'T WE JUST THROW DATA AT A NEURAL NETWORK UNTIL IT WORKS? 245 - WHAT IS A NODULE? 249 - OUR DATA SOURCE: THE LUNA GRAND CHALLENGE 251 - DOWNLOADING THE LUNA DATA 251 CONTENTS X 9.5 CONCLUSION 252 9.6 SUMMARY 253 COMBINING DATA SOURCES INTO A UNIFIED DATASET 254 1 10.1 RAW CT DATA FILES 256 10.2 PARSING LUNA'S ANNOTATION DATA 256 TRAINING AND VALIDATION SETS 258 : UNIFYING OUR ANNOTATION AND CANDIDATE DATA 259 10.3 LOADING INDIVIDUAL CT SCANS 262 HOUNSFIELD UNITS 264 10.4 LOCATING A NODULE USING THE PATIENT COORDINATE SYSTEM 265 THE PATIENT COORDINATE SYSTEM 265 - CT SCAN SHAPE AND VOXEL SIZES 267 - CONVERTING BETWEEN MILLIMETERS AND VOXEL ADDRESSES 268 - EXTRACTING A NODULE FROM A CT SCAN 270 10.5 A STRAIGHTFORWARD DATASET IMPLEMENTATION 271 CACHING CANDIDATE ARRAYS WITH THE GETCTRAWCANDIDATE FUNCTION 274 : CONSTRUCTING OUR DATASET IN LUNADATASET \_INIT\_- 275 - A TRAINING/VALIDATION SPLIT 275 - RENDERING THE DATA 277 10.6 CONCLUSION 277 10.7 EXERCISES 278 10.8 SUMMARY 278 TRAINING A CLASSIFICATION MODEL TO DETECT SUSPECTED TUMORS 279 11.1 A FOUNDATIONAL MODEL AND TRAINING LOOP 280 11.2 THE MAIN ENTRY POINT FOR OUR APPLICATION 282 11.3 PRETRAINING SETUP AND INITIALIZATION 284 INITIALIZING THE MODEL AND OPTIMIZER 285 : CARE AND FEEDING OF DATA LOADERS 287 11.4 OUR FIRST-PASS NEURAL NETWORK DESIGN 289 THE CORE CONVOLUTIONS 290 : THE FULL MODEL 293 11.5 TRAINING AND VALIDATING THE MODEL 295 THE COMPUTEBATCHLOSS FUNCTION 297 - THE VALIDATION LOOP IS SIMILAR 299 11.6 OUTPUTTING PERFORMANCE METRICS 300 THE LOGMETRICS FUNCTION 301 XI CONTENTS 11.7 RUNNING THE TRAINING SCRIPT 304 NEEDED DATA FOR TRAINING 305 - INTERLUDE: THE ENUMERATEWITHESTIMATE FUNCTION 306 11.8 EVALUATING THE MODEL: GETTING 99.7% CORRECT MEANS WE'RE DONE, RIGHT? 308 11.9 GRAPHING TRAINING METRICS WITH TENSORBOARD 309 RUNNING TENSORBOARD 309 - ADDING TENSORBOARD SUPPORT TO THE METRICS LOGGING FUNCTION 313 11.10 WHY ISN'T THE MODEL LEARNING TO DETECT NODULES? 315 11.11 CONCLUSION 316 11.12 EXERCISES 316 11.13 SUMMARY 316 12 IMPROVING TRAINING WITH METRICS AND AUGMENTATION 318 12.1 HIGH-LEVEL PLAN FOR IMPROVEMENT 319 12.2 GOOD DOGS VS. BAD GUYS: FALSE POSITIVES AND FALSE NEGATIVES 320 12.3 GRAPHING THE POSITIVES AND NEGATIVES322 RECALL IS ROXIE'S STRENGTH 324 - PRECISION IS PRESTON'S FORTE 326 IMPLEMENTING PRECISION AND RECALL IN LOGMETRICS 327 : OUR ULTIMATE PERFORMANCE METRIC: THE F1 SCORE 328 - HOW DOES OUR MODEL PERFORM WITH OUR NEW METRICS? 332 12.4 WHAT DOES AN IDEAL DATASET LOOK LIKE? 334 MAKING THE DATA LOOK LESS LIKE THE ACTUAL AND MORE LIKE THE "IDEAL" 336 CONTRASTING TRAINING WITH A BALANCED LUNADATASET TO PREVIOUS RUNS 341 - RECOGNIZING THE SYMPTOMS OF OVERFITTING 343 12.5 REVISITING THE PROBLEM OF OVERFITTING 345 AN OVERFIT FACE-TO-AGE PREDICTION MODEL 345 12.6 PREVENTING OVERFITTING WITH DATA AUGMENTATION 346 SPECIFIC DATA AUGMENTATION TECHNIQUES 347 : SEEING THE IMPROVEMENT FROM DATA AUGMENTATION 352 12.7 CONCLUSION 354 12.8 EXERCISES 355 12.9 SUMMARY 356 USING SEGMENTATION TO FIND SUSPECTED NODULES 357 13.1 ADDING A SECOND MODEL TO OUR PROJECT 358 13.2 VARIOUS TYPES OF SEGMENTATION 360 XII CONTENTS 13.3 SEMANTIC SEGMENTATION: PER-PIXEL CLASSIFICATION 361 THE U-NET ARCHITECTURE 364 13.4 UPDATING THE MODEL FOR SEGMENTATION 366 ADAPTING AN OFF-THE-SHELF MODEL TO OUR PROJECT 367 13.5 UPDATING THE DATASET FOR SEGMENTATION 369 U-NET HAS VERY SPECIFIC INPUT SIZE REQUIREMENTS 370 - U-NET TRADE YINU PUNOU8 OYP BUIPING - OLEDYPP AZ 'SA AE IOF SFFO DATA 371 : IMPLEMENTING LUNA2DSEGMENTATIONDATASET 378 DESIGNING OUR TRAINING AND VALIDATION DATA 382 - IMPLEMENTING TRAININGLUNA2DSEGMENTATIONDATASET 383 - AUGMENTING ON THE GPU 384 13.6 UPDATING THE TRAINING SCRIPT FOR SEGMENTATION 386 INITIALIZING OUR SEGMENTATION AND AUGMENTATION MODELS 387 USING THE ADAM OPTIMIZER 388 - DICE LOSS 389 - GETTING IMAGES INTO TENSORBOARD 392 : UPDATING OUR METRICS LOGGING 396 SAVING OUR MODEL 397 13.7 RESULTS 399 13.8 CONCLUSION 401 13.9 EXERCISES 402 13.10 SUMMARY 402 END-TO-END NODULE ANALYSIS, AND WHERE TO GO NEXT 404 14.1 TOWARDS THE FINISH LINE 405 14.2 INDEPENDENCE OF THE VALIDATION SET 407 14.3 BRIDGING CT SEGMENTATION AND NODULE CANDIDATE CLASSIFICATION 408 SEGMENTATION 410 - GROUPING VOXELS INTO NODULE CANDIDATES 411 DID WE FIND A NODULE? CLASSIFICATION TO REDUCE FALSE POSITIVES 412 14.4 QUANTITATIVE VALIDATION 416 14.5 PREDICTING MALIGNANCY 417 GETTING MALIGNANCY INFORMATION 417 - AN AREA UNDER THE CURVE BASELINE: CLASSIFYING BY DIAMETER 419 - REUSING PREEXISTING WEIGHTS: FINE-TUNING 422 - MORE OUTPUT IN TENSORBOARD 428 14.6 WHAT WE SEE WHEN WE DIAGNOSE 432 TRAINING, VALIDATION, AND TEST SETS 433 14.7 WHAT NEXT? ADDITIONAL SOURCES OF INSPIRATION (AND DATA) 434 PREVENTING OVERFTTING: BETTER REGULARIZATION 434 - REFINED TRAINING DATA 437 : COMPETITION RESULTS AND RESEARCH PAPERS 438DEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING UNTRAINED CLOUD MULTIPROCESS MODEL DATA LOADING SAMPLE BATCH TRAINED TENSORS TENSOR PRODUCTION MODEL DATA (ONNX,JIT SOURCE TRAINING TORCHSCRIPT) LOOP DISTRIBUTED TRAINING PRODUCTION ON MULTIPLE SERVERS/GPUS SERVER DEEP LEARNING WITH PYTORCH ELI STEVENS,LUCA ANTIGA AND THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING SHELTER ISLAND FOR ONLINE INFORMATION AND ORDERING OF THIS AND OTHER MANNING BOOKS, PLEASE VISIT WWW.MANNING.COM. THE PUBLISHER OFFERS DISCOUNTS ON THIS BOOK WHEN ORDERED IN QUANTITY FOR MORE INFORMATION, PLEASE CONTACT SPECIAL SALES DEPARTMENT MANNING PUBLICATIONS CO. 20 BALDWIN ROAD PO BOX 761 SHELTER ISLAND, NY 11964 EMAIL: ORDERS@MANNING.COM @2020 BY MANNING PUBLICATIONS CO. ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED, STORED IN A RETRIEVAL SYSTEM, OR TRANSMITTED, IN ANY FORM OR BY MEANS ELECTRONIC, MECHANICAL, PHOTOCOPYING, OR OTHERWISE, WITHOUT PRIOR WRITTEN PERMISSION OF THE PUBLISHER. MANY OF THE DESIGNATIONS USED BY MANUFACTURERS AND SELLERS TO DISTINGUISH THEIR PRODUCTS ARE CLAIMED AS TRADEMARKS. WHERE THOSE DESIGNATIONS APPEAR IN THE BOOK, AND MANNING PUBLICATIONS WAS AWARE OF A TRADEMARK CLAIM, THE DESIGNATIONS HAVE BEEN PRINTED IN INITIAL CAPS OR ALL CAPS. @ RECOGNIZING THE IMPORTANCE OF PRESERVING WHAT HAS BEEN WRITTEN, IT IS MANNING'S POLICY TO HAVE THE BOOKS WE PUBLISH PRINTED ON ACID-FREE PAPER, AND WE EXERT OUR BEST EFFORTS TO THAT END. RECOGNIZING ALSO OUR RESPONSIBILITY TO CONSERVE THE RESOURCES OF OUR PLANET, MANNING BOOKS ARE PRINTED ON PAPER THAT IS AT LEAST 15 PERCENT RECYCLED AND PROCESSED WITHOUT THE USE OF ELEMENTAL CHLORINE. MANNING PUBLICATIONS CO. DEVELOPMENT EDITOR: FRANCES LEFKOWITZ 20 BALDWIN ROAD TECHNICAL DEVELOPMENT EDITOR: ARTHUR ZUBAREV PO BOX 761 REVIEW EDITOR: IVAN MARTINOVIC SHELTER ISLAND, NY 11964 PRODUCTION EDITOR: DEIRDRE HIAM COPYEDITOR: TIFFANY TAYLOR PROOFREADER: KATIE TENNANT TECHNICAL PROOFREADER: KOSTAS PASSADIS TYPESETTER: GORDAN SALINOVIC COVER DESIGNER: MARIJA TUDOR ISBN 9781617295263 PRINTED IN THE UNITED STATES OF AMERICA TO MY WIFE (THIS BOOK WOULD NOT HAVE HAPPENED WITHOUT HER INVALUABLE SUPPORT AND PARTNERSHIP), MY PARENTS (I WOULD NOT HAVE HAPPENED WITHOUT THEM), AND MY CHILDREN (THIS BOOK WOULD HAVE HAPPENED A LOT SOONER BUT FOR THEM) THANK YOU FOR BEING MY HOME, MY FOUNDATION, AND MY JOY ELI STEVENS SAME :-) BUT, REALLY, THIS IS FOR YOU, ALICE AND LUIGI LUCA ANTIGA TO EVA, REBEKKA, JONATHAN, AND DAVID. THOMAS VIEHMANN CONTENTS FOREWORD XV PREFACE XVII ACKNOWLEDGMENTS XIX ABOUT THIS BOOK XXI ABOUT THE AUTHORS XXVII ABOUT THE COVER ILLUSTRATION XXVIII PART 1 CORE PYTORCH. INTRODUCING DEEP LEARNING AND THE PYTORCH LIBRARY3 1.1 THE DEEP LEARNING REVOLUTION 4 1.2 PYTORCH FOR DEEP LEARNING 6 1.3 WHY PYTORCH?7 THE DEEP LEARNING COMPETITIVE LANDSCAPE8 1.4 AN OVERVIEW OF HOW PYTORCH SUPPORTS DEEP LEARNING PROJECTS 10 1.5 HARDWARE AND SOFTWARE REQUIREMENTS 13 USING JUPYTER NOTEBOOKS 14 1.6 EXERCISES 15 1.7 SUMMARY 15 VI CONTENTS PRETRAINED NETWORKS 16 2.1 A PRETRAINED NETWORK THAT RECOGNIZES THE SUBJECT OF AN IMAGE 17 OBTAINING A PRETRAINED NETWORK FOR IMAGE RECOGNITION 19 ALEXNET 20 - RESNET 22 - READY, SET, ALMOST RUN 22 RUN! 25 2.2 A PRETRAINED MODEL THAT FAKES IT UNTIL IT MAKES IT 27 THE GAN GAME 28 : CYCLEGAN 29 - A NETWORK THAT TURNS HORSES INTO ZEBRAS 30 2.3 A PRETRAINED NETWORK THAT DESCRIBES SCENES 33 NEURALTALK2 34 2.4 TORCH HUB 35 2.5 CONCLUSION 37 2.6 EXERCISES 38 2.7 SUMMARY 38 IT STARTS WITH A TENSOR 39 3.1 THE WORLD AS FLOATING-POINT NUMBERS 40 3.2 TENSORS: MULTIDIMENSIONAL ARRAYS 42 FROM PYTHON LISTS TO PYTORCH TENSORS 42 - CONSTRUCTING OUR FIRST TENSORS 43 - THE ESSENCE OF TENSORS 43 3.3 INDEXING TENSORS 46 3.4 NAMED TENSORS 46 3.5 TENSOR ELEMENT TYPES 50 SPECIFYING THE NUMERIC TYPE WITH DTYPE 50 - A DTYPE FOR EVERY OCCASION 51 - MANAGING A TENSOR'S DTYPE ATTRIBUTE 51 3.6 THE TENSOR API 52 3.7 TENSORS: SCENIC VIEWS OF STORAGE 53 INDEXING INTO STORAGE 54 - MODIFYING STORED VALUES: IN-PLACE OPERATIONS 55 3.8 TENSOR METADATA: SIZE, OFFSET, AND STRIDE 55 VIEWS OF ANOTHER TENSOR'S STORAGE 56 - TRANSPOSING WITHOUT COPYING 58 : TRANSPOSING IN HIGHER DIMENSIONS 60 CONTIGUOUS TENSORS 60 3.9 MOVING TENSORS TO THE GPU 62 MANAGING A TENSOR'S DEVICE ATTRIBUTE 63 VII CONTENTS 3.10 NUMPY INTEROPERABILITY64 3.11 GENERALIZED TENSORS ARE TENSORS, TOO 65 3.12 SERIALIZING TENSORS 66 SERIALIZING TO HDF5 WITH H5PY 67 3.13 CONCLUSION 68 3.14 EXERCISES 68 3.15 SUMMARY 68 REAL-WORLD DATA REPRESENTATION USING TENSORS 70 4.1 WORKING WITH IMAGES 71 ADDING COLOR CHANNELS 72 - LOADING AN IMAGE FILE 72 CHANGING THE LAYOUT 73 : NORMALIZING THE DATA 74 4.2 3D IMAGES: VOLUMETRIC DATA 75 LOADING A SPECIALIZED FORMAT 76 4.3 REPRESENTING TABULAR DATA 77 USING A REAL-WORLD DATASET 77 - LOADING A WINE DATA TENSOR 78 REPRESENTING SCORES 81 : ONE-HOT ENCODING 81 : WHEN TO CATEGORIZE 83 - FINDING THRESHOLDS 84 4.4 WORKING WITH TIME SERIES 87 ADDING A TIME DIMENSION 88 - SHAPING THE DATA BY TIME PERIOD 89 - READY FOR TRAINING 90 4.5 REPRESENTING TEXT 93 CONVERTING TEXT TO NUMBERS94 - ONE-HOT-ENCODING CHARACTERS94 ONE-HOT ENCODING WHOLE WORDS 96 : TEXT EMBEDDINGS 98 TEXT EMBEDDINGS AS A BLUEPRINT 100 4.6 CONCLUSION 101 4.7 EXERCISES 101 4.8 SUMMARY 102 THE MECHANICS OF LEARNING 103 5.1 A TIMELESS LESSON IN MODELING 104 5.2 LEARNING IS JUST PARAMETER ESTIMATION 106 A HOT PROBLEM 107 : GATHERING SOME DATA 107 : VISUALIZING THE DATA 108 : CHOOSING A LINEAR MODEL AS A FIRST TRY 108 5.3 LESS LOSS IS WHAT WE WANT 109 FROM PROBLEM BACK TO PYTORCH 110 VIII CONTENTS 5.4 DOWN ALONG THE GRADIENT 113 DECREASING LOSS 113 - GETTING ANALYTICAL 114 - ITERATING TO FIT THE MODEL 116 - NORMALIZING INPUTS 119 : VISUALIZING (AGAIN) 122 5.5 PYTORCH'S AUTOGRAD: BACKPROPAGATING ALL THINGS 123 COMPUTING THE GRADIENT AUTOMATICALLY 123 - OPTIMIZERS A LA CARTE 127 : TRAINING, VALIDATION, AND OVERFITTING 131 AUTOGRAD NITS AND SWITCHING IT OFF. 137 5.6 CONCLUSION 139 5.7 EXERCISE 139 5.8 SUMMARY 139 USING A NEURAL NETWORK TO FIT THE DATA 141 6.1 ARTIFICIAL NEURONS142 COMPOSING A MULTILAYER NETWORK 144 - UNDERSTANDING THE ERROR FUNCTION 144 : ALL WE NEED IS ACTIVATION 145 : MORE ACTIVATION FUNCTIONS 147 : CHOOSING THE BEST ACTIVATION FUNCTION 148 WHAT LEARNING MEANS FOR A NEURAL NETWORK 149 6.2 THE PYTORCH NN MODULE 151 USING--CALL- RATHER THAN FORWARD 152 - RETURNING TO THE LINEAR MODEL 153 6.3 FINALLY A NEURAL NETWORK 158 REPLACING THE LINEAR MODEL 158 - INSPECTING THE PARAMETERS 159 COMPARING TO THE LINEAR MODEL 161 6.4 CONCLUSION 162 6.5 EXERCISES 162 6.6 SUMMARY 163 TELLING BIRDS FROM AIRPLANES: LEARNING FROM IMAGES 164 7.1 A DATASET OF TINY IMAGES 165 DOWNLOADING CIFAR-10 166 : THE DATASET CLASS 166 DATASET TRANSFORMS 168 - NORMALIZING DATA 170 7.2 DISTINGUISHING BIRDS FROM AIRPLANES 172 BUILDING THE DATASET 173 - A FULLY CONNECTED MODEL 174 OUTPUT OF A CLASSIFIER 175 : REPRESENTING THE OUTPUT AS PROBABILITIES 176 - A LOSS FOR CLASSIFYING 180 : TRAINING THE CLASSIFIER 182 - THE LIMITS OF GOING FULLY CONNECTED 189 7.3 CONCLUSION 191 IX CONTENTS 7.4 EXERCISES 191 7.5 SUMMARY 192 USING CONVOLUTIONS TO GENERALIZE 193 8.1 THE CASE FOR CONVOLUTIONS 194 WHAT CONVOLUTIONS DO 194 8.2 CONVOLUTIONS IN ACTION 196 PADDING THE BOUNDARY 198 - DETECTING FEATURES WITH CONVOLUTIONS 200 - LOOKING FURTHER WITH DEPTH AND POOLING 202 PUTTING IT ALL TOGETHER FOR OUR NETWORK 205 8.3 SUBCLASSING NN.MODULE 207 OUR NETWORK AS AN NN.MODULE 208 - HOW PYTORCH KEEPS TRACK OF PARAMETERS AND SUBMODULES 209 : THE FUNCTIONAL API 210 8.4 TRAINING OUR CONVNET 212 MEASURING ACCURACY 214 : SAVING AND LOADING OUR MODEL 214 TRAINING ON THE GPU 215 8.5 MODEL DESIGN 217 ADDING MEMORY CAPACITY: WIDTH 218 - HELPING OUR MODEL TO CONVERGE AND GENERALIZE: REGULARIZATION 219 - GOING DEEPER TO LEARN MORE COMPLEX STRUCTURES: DEPTH223 - COMPARING THE DESIGNS FROM THIS SECTION 228 - IT'S ALREADY OUTDATED 229 8.6 CONCLUSION 229 8.7 EXERCISES 230 8.8 SUMMARY 231 PART 2 LEARNING FROM IMAGES IN THE REAL WORLD: EARLY DETECTION OF LUNG CANCER. 233 USING PYTORCH TO FIGHT CANCER 235 9.1 INTRODUCTION TO THE USE CASE 236 9.2 PREPARING FOR A LARGE-SCALE PROJECT237 9.3 WHAT IS A CT SCAN, EXACTLY?238 9.4 THE PROJECT: AN END-TO-END DETECTOR FOR LUNG CANCER 241 WHY CAN'T WE JUST THROW DATA AT A NEURAL NETWORK UNTIL IT WORKS? 245 - WHAT IS A NODULE? 249 - OUR DATA SOURCE: THE LUNA GRAND CHALLENGE 251 - DOWNLOADING THE LUNA DATA 251 CONTENTS X 9.5 CONCLUSION 252 9.6 SUMMARY 253 COMBINING DATA SOURCES INTO A UNIFIED DATASET 254 1 10.1 RAW CT DATA FILES 256 10.2 PARSING LUNA'S ANNOTATION DATA 256 TRAINING AND VALIDATION SETS 258 : UNIFYING OUR ANNOTATION AND CANDIDATE DATA 259 10.3 LOADING INDIVIDUAL CT SCANS 262 HOUNSFIELD UNITS 264 10.4 LOCATING A NODULE USING THE PATIENT COORDINATE SYSTEM 265 THE PATIENT COORDINATE SYSTEM 265 - CT SCAN SHAPE AND VOXEL SIZES 267 - CONVERTING BETWEEN MILLIMETERS AND VOXEL ADDRESSES 268 - EXTRACTING A NODULE FROM A CT SCAN 270 10.5 A STRAIGHTFORWARD DATASET IMPLEMENTATION 271 CACHING CANDIDATE ARRAYS WITH THE GETCTRAWCANDIDATE FUNCTION 274 : CONSTRUCTING OUR DATASET IN LUNADATASET \_INIT\_- 275 - A TRAINING/VALIDATION SPLIT 275 - RENDERING THE DATA 277 10.6 CONCLUSION 277 10.7 EXERCISES 278 10.8 SUMMARY 278 TRAINING A CLASSIFICATION MODEL TO DETECT SUSPECTED TUMORS 279 11.1 A FOUNDATIONAL MODEL AND TRAINING LOOP 280 11.2 THE MAIN ENTRY POINT FOR OUR APPLICATION 282 11.3 PRETRAINING SETUP AND INITIALIZATION 284 INITIALIZING THE MODEL AND OPTIMIZER 285 : CARE AND FEEDING OF DATA LOADERS 287 11.4 OUR FIRST-PASS NEURAL NETWORK DESIGN 289 THE CORE CONVOLUTIONS 290 : THE FULL MODEL 293 11.5 TRAINING AND VALIDATING THE MODEL 295 THE COMPUTEBATCHLOSS FUNCTION 297 - THE VALIDATION LOOP IS SIMILAR 299 11.6 OUTPUTTING PERFORMANCE METRICS 300 THE LOGMETRICS FUNCTION 301 XI CONTENTS 11.7 RUNNING THE TRAINING SCRIPT 304 NEEDED DATA FOR TRAINING 305 - INTERLUDE: THE ENUMERATEWITHESTIMATE FUNCTION 306 11.8 EVALUATING THE MODEL: GETTING 99.7% CORRECT MEANS WE'RE DONE, RIGHT? 308 11.9 GRAPHING TRAINING METRICS WITH TENSORBOARD 309 RUNNING TENSORBOARD 309 - ADDING TENSORBOARD SUPPORT TO THE METRICS LOGGING FUNCTION 313 11.10 WHY ISN'T THE MODEL LEARNING TO DETECT NODULES? 315 11.11 CONCLUSION 316 11.12 EXERCISES 316 11.13 SUMMARY 316 12 IMPROVING TRAINING WITH METRICS AND AUGMENTATION 318 12.1 HIGH-LEVEL PLAN FOR IMPROVEMENT 319 12.2 GOOD DOGS VS. BAD GUYS: FALSE POSITIVES AND FALSE NEGATIVES 320 12.3 GRAPHING THE POSITIVES AND NEGATIVES322 RECALL IS ROXIE'S STRENGTH 324 - PRECISION IS PRESTON'S FORTE 326 IMPLEMENTING PRECISION AND RECALL IN LOGMETRICS 327 : OUR ULTIMATE PERFORMANCE METRIC: THE F1 SCORE 328 - HOW DOES OUR MODEL PERFORM WITH OUR NEW METRICS? 332 12.4 WHAT DOES AN IDEAL DATASET LOOK LIKE? 334 MAKING THE DATA LOOK LESS LIKE THE ACTUAL AND MORE LIKE THE "IDEAL" 336 CONTRASTING TRAINING WITH A BALANCED LUNADATASET TO PREVIOUS RUNS 341 - RECOGNIZING THE SYMPTOMS OF OVERFITTING 343 12.5 REVISITING THE PROBLEM OF OVERFITTING 345 AN OVERFIT FACE-TO-AGE PREDICTION MODEL 345 12.6 PREVENTING OVERFITTING WITH DATA AUGMENTATION 346 SPECIFIC DATA AUGMENTATION TECHNIQUES 347 : SEEING THE IMPROVEMENT FROM DATA AUGMENTATION 352 12.7 CONCLUSION 354 12.8 EXERCISES 355 12.9 SUMMARY 356 USING SEGMENTATION TO FIND SUSPECTED NODULES 357 13.1 ADDING A SECOND MODEL TO OUR PROJECT 358 13.2 VARIOUS TYPES OF SEGMENTATION 360 XII CONTENTS 13.3 SEMANTIC SEGMENTATION: PER-PIXEL CLASSIFICATION 361 THE U-NET ARCHITECTURE 364 13.4 UPDATING THE MODEL FOR SEGMENTATION 366 ADAPTING AN OFF-THE-SHELF MODEL TO OUR PROJECT 367 13.5 UPDATING THE DATASET FOR SEGMENTATION 369 U-NET HAS VERY SPECIFIC INPUT SIZE REQUIREMENTS 370 - U-NET TRADE YINU PUNOU8 OYP BUIPING - OLEDYPP AZ 'SA AE IOF SFFO DATA 371 : IMPLEMENTING LUNA2DSEGMENTATIONDATASET 378 DESIGNING OUR TRAINING AND VALIDATION DATA 382 - IMPLEMENTING TRAININGLUNA2DSEGMENTATIONDATASET 383 - AUGMENTING ON THE GPU 384 13.6 UPDATING THE TRAINING SCRIPT FOR SEGMENTATION 386 INITIALIZING OUR SEGMENTATION AND AUGMENTATION MODELS 387 USING THE ADAM OPTIMIZER 388 - DICE LOSS 389 - GETTING IMAGES INTO TENSORBOARD 392 : UPDATING OUR METRICS LOGGING 396 SAVING OUR MODEL 397 13.7 RESULTS 399 13.8 CONCLUSION 401 13.9 EXERCISES 402 13.10 SUMMARY 402 END-TO-END NODULE ANALYSIS, AND WHERE TO GO NEXT 404 14.1 TOWARDS THE FINISH LINE 405 14.2 INDEPENDENCE OF THE VALIDATION SET 407 14.3 BRIDGING CT SEGMENTATION AND NODULE CANDIDATE CLASSIFICATION 408 SEGMENTATION 410 - GROUPING VOXELS INTO NODULE CANDIDATES 411 DID WE FIND A NODULE? CLASSIFICATION TO REDUCE FALSE POSITIVES 412 14.4 QUANTITATIVE VALIDATION 416 14.5 PREDICTING MALIGNANCY 417 GETTING MALIGNANCY INFORMATION 417 - AN AREA UNDER THE CURVE BASELINE: CLASSIFYING BY DIAMETER 419 - REUSING PREEXISTING WEIGHTS: FINE-TUNING 422 - MORE OUTPUT IN TENSORBOARD 428 14.6 WHAT WE SEE WHEN WE DIAGNOSE 432 TRAINING, VALIDATION, AND TEST SETS 433 14.7 WHAT NEXT? ADDITIONAL SOURCES OF INSPIRATION (AND DATA) 434 PREVENTING OVERFTTING: BETTER REGULARIZATION 434 - REFINED TRAINING DATA 437 : COMPETITION RESULTS AND RESEARCH PAPERS 438 XIII CONTENTS 14.8 CONCLUSION 439 BEHIND THE CURTAIN 439 14.9 EXERCISES 441 14.10 SUMMARY 441 PART 3 DEPLOYMENT : 443 DEPLOYING TO PRODUCTION445 15.1 SERVING PYTORCH MODELS 446 OUR MODEL BEHIND A FLASK SERVER 446 - WHAT WE WANT FROM DEPLOYMENT 448 : REQUEST BATCHING 449 15.2 EXPORTING MODELS 455 INTEROPERABILITY BEYOND PYTORCH WITH ONNX 455 - PYTORCH'S OWN EXPORT: TRACING 456 - OUR SERVER WITH A TRACED MODEL 458 15.3 INTERACTING WITH THE PYTORCH JIT 458 WHAT TO EXPECT FROM MOVING BEYOND CLASSIC PYTHON/PYTORCH 458 THE DUAL NATURE OF PYTORCH AS INTERFACE AND BACKEND 460 TORCHSCRIPT 461 - SCRIPTING THE GAPS OF TRACEABILITY 464 15.4 LIBTORCH: PYTORCH IN C++ 465 RUNNING JITED MODELS FROM C++ 465 : C++ FROM THE START: THE C++ API 468 15.5 GOING MOBILE 472 IMPROVING EFFICIENCY: MODEL DESIGN AND QUANTIZATION 475 15.6 EMERGING TECHNOLOGY: ENTERPRISE SERVING OF PYTORCH MODELS 476 15.7 CONCLUSION 477 15.8 EXERCISES 477 15.9 SUMMARY 477 INDEX 479DEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING UNTRAINED CLOUD MULTIPROCESS MODEL DATA LOADING SAMPLE BATCH TRAINED TENSORS TENSOR PRODUCTION MODEL DATA (ONNX,JIT SOURCE TRAINING TORCHSCRIPT) LOOP DISTRIBUTED TRAINING PRODUCTION ON MULTIPLE SERVERS/GPUS SERVER DEEP LEARNING WITH PYTORCH ELI STEVENS,LUCA ANTIGA AND THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING SHELTER ISLAND FOR ONLINE INFORMATION AND ORDERING OF THIS AND OTHER MANNING BOOKS, PLEASE VISIT WWW.MANNING.COM. THE PUBLISHER OFFERS DISCOUNTS ON THIS BOOK WHEN ORDERED IN QUANTITY FOR MORE INFORMATION, PLEASE CONTACT SPECIAL SALES DEPARTMENT MANNING PUBLICATIONS CO. 20 BALDWIN ROAD PO BOX 761 SHELTER ISLAND, NY 11964 EMAIL: ORDERS@MANNING.COM @2020 BY MANNING PUBLICATIONS CO. ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED, STORED IN A RETRIEVAL SYSTEM, OR TRANSMITTED, IN ANY FORM OR BY MEANS ELECTRONIC, MECHANICAL, PHOTOCOPYING, OR OTHERWISE, WITHOUT PRIOR WRITTEN PERMISSION OF THE PUBLISHER. MANY OF THE DESIGNATIONS USED BY MANUFACTURERS AND SELLERS TO DISTINGUISH THEIR PRODUCTS ARE CLAIMED AS TRADEMARKS. WHERE THOSE DESIGNATIONS APPEAR IN THE BOOK, AND MANNING PUBLICATIONS WAS AWARE OF A TRADEMARK CLAIM, THE DESIGNATIONS HAVE BEEN PRINTED IN INITIAL CAPS OR ALL CAPS. @ RECOGNIZING THE IMPORTANCE OF PRESERVING WHAT HAS BEEN WRITTEN, IT IS MANNING'S POLICY TO HAVE THE BOOKS WE PUBLISH PRINTED ON ACID-FREE PAPER, AND WE EXERT OUR BEST EFFORTS TO THAT END. RECOGNIZING ALSO OUR RESPONSIBILITY TO CONSERVE THE RESOURCES OF OUR PLANET, MANNING BOOKS ARE PRINTED ON PAPER THAT IS AT LEAST 15 PERCENT RECYCLED AND PROCESSED WITHOUT THE USE OF ELEMENTAL CHLORINE. MANNING PUBLICATIONS CO. DEVELOPMENT EDITOR: FRANCES LEFKOWITZ 20 BALDWIN ROAD TECHNICAL DEVELOPMENT EDITOR: ARTHUR ZUBAREV PO BOX 761 REVIEW EDITOR: IVAN MARTINOVIC SHELTER ISLAND, NY 11964 PRODUCTION EDITOR: DEIRDRE HIAM COPYEDITOR: TIFFANY TAYLOR PROOFREADER: KATIE TENNANT TECHNICAL PROOFREADER: KOSTAS PASSADIS TYPESETTER: GORDAN SALINOVIC COVER DESIGNER: MARIJA TUDOR ISBN 9781617295263 PRINTED IN THE UNITED STATES OF AMERICA TO MY WIFE (THIS BOOK WOULD NOT HAVE HAPPENED WITHOUT HER INVALUABLE SUPPORT AND PARTNERSHIP), MY PARENTS (I WOULD NOT HAVE HAPPENED WITHOUT THEM), AND MY CHILDREN (THIS BOOK WOULD HAVE HAPPENED A LOT SOONER BUT FOR THEM) THANK YOU FOR BEING MY HOME, MY FOUNDATION, AND MY JOY ELI STEVENS SAME :-) BUT, REALLY, THIS IS FOR YOU, ALICE AND LUIGI LUCA ANTIGA TO EVA, REBEKKA, JONATHAN, AND DAVID. THOMAS VIEHMANN CONTENTS FOREWORD XV PREFACE XVII ACKNOWLEDGMENTS XIX ABOUT THIS BOOK XXI ABOUT THE AUTHORS XXVII ABOUT THE COVER ILLUSTRATION XXVIII PART 1 CORE PYTORCH. INTRODUCING DEEP LEARNING AND THE PYTORCH LIBRARY3 1.1 THE DEEP LEARNING REVOLUTION 4 1.2 PYTORCH FOR DEEP LEARNING 6 1.3 WHY PYTORCH?7 THE DEEP LEARNING COMPETITIVE LANDSCAPE8 1.4 AN OVERVIEW OF HOW PYTORCH SUPPORTS DEEP LEARNING PROJECTS 10 1.5 HARDWARE AND SOFTWARE REQUIREMENTS 13 USING JUPYTER NOTEBOOKS 14 1.6 EXERCISES 15 1.7 SUMMARY 15 VI CONTENTS PRETRAINED NETWORKS 16 2.1 A PRETRAINED NETWORK THAT RECOGNIZES THE SUBJECT OF AN IMAGE 17 OBTAINING A PRETRAINED NETWORK FOR IMAGE RECOGNITION 19 ALEXNET 20 - RESNET 22 - READY, SET, ALMOST RUN 22 RUN! 25 2.2 A PRETRAINED MODEL THAT FAKES IT UNTIL IT MAKES IT 27 THE GAN GAME 28 : CYCLEGAN 29 - A NETWORK THAT TURNS HORSES INTO ZEBRAS 30 2.3 A PRETRAINED NETWORK THAT DESCRIBES SCENES 33 NEURALTALK2 34 2.4 TORCH HUB 35 2.5 CONCLUSION 37 2.6 EXERCISES 38 2.7 SUMMARY 38 IT STARTS WITH A TENSOR 39 3.1 THE WORLD AS FLOATING-POINT NUMBERS 40 3.2 TENSORS: MULTIDIMENSIONAL ARRAYS 42 FROM PYTHON LISTS TO PYTORCH TENSORS 42 - CONSTRUCTING OUR FIRST TENSORS 43 - THE ESSENCE OF TENSORS 43 3.3 INDEXING TENSORS 46 3.4 NAMED TENSORS 46 3.5 TENSOR ELEMENT TYPES 50 SPECIFYING THE NUMERIC TYPE WITH DTYPE 50 - A DTYPE FOR EVERY OCCASION 51 - MANAGING A TENSOR'S DTYPE ATTRIBUTE 51 3.6 THE TENSOR API 52 3.7 TENSORS: SCENIC VIEWS OF STORAGE 53 INDEXING INTO STORAGE 54 - MODIFYING STORED VALUES: IN-PLACE OPERATIONS 55 3.8 TENSOR METADATA: SIZE, OFFSET, AND STRIDE 55 VIEWS OF ANOTHER TENSOR'S STORAGE 56 - TRANSPOSING WITHOUT COPYING 58 : TRANSPOSING IN HIGHER DIMENSIONS 60 CONTIGUOUS TENSORS 60 3.9 MOVING TENSORS TO THE GPU 62 MANAGING A TENSOR'S DEVICE ATTRIBUTE 63 VII CONTENTS 3.10 NUMPY INTEROPERABILITY64 3.11 GENERALIZED TENSORS ARE TENSORS, TOO 65 3.12 SERIALIZING TENSORS 66 SERIALIZING TO HDF5 WITH H5PY 67 3.13 CONCLUSION 68 3.14 EXERCISES 68 3.15 SUMMARY 68 REAL-WORLD DATA REPRESENTATION USING TENSORS 70 4.1 WORKING WITH IMAGES 71 ADDING COLOR CHANNELS 72 - LOADING AN IMAGE FILE 72 CHANGING THE LAYOUT 73 : NORMALIZING THE DATA 74 4.2 3D IMAGES: VOLUMETRIC DATA 75 LOADING A SPECIALIZED FORMAT 76 4.3 REPRESENTING TABULAR DATA 77 USING A REAL-WORLD DATASET 77 - LOADING A WINE DATA TENSOR 78 REPRESENTING SCORES 81 : ONE-HOT ENCODING 81 : WHEN TO CATEGORIZE 83 - FINDING THRESHOLDS 84 4.4 WORKING WITH TIME SERIES 87 ADDING A TIME DIMENSION 88 - SHAPING THE DATA BY TIME PERIOD 89 - READY FOR TRAINING 90 4.5 REPRESENTING TEXT 93 CONVERTING TEXT TO NUMBERS94 - ONE-HOT-ENCODING CHARACTERS94 ONE-HOT ENCODING WHOLE WORDS 96 : TEXT EMBEDDINGS 98 TEXT EMBEDDINGS AS A BLUEPRINT 100 4.6 CONCLUSION 101 4.7 EXERCISES 101 4.8 SUMMARY 102 THE MECHANICS OF LEARNING 103 5.1 A TIMELESS LESSON IN MODELING 104 5.2 LEARNING IS JUST PARAMETER ESTIMATION 106 A HOT PROBLEM 107 : GATHERING SOME DATA 107 : VISUALIZING THE DATA 108 : CHOOSING A LINEAR MODEL AS A FIRST TRY 108 5.3 LESS LOSS IS WHAT WE WANT 109 FROM PROBLEM BACK TO PYTORCH 110 VIII CONTENTS 5.4 DOWN ALONG THE GRADIENT 113 DECREASING LOSS 113 - GETTING ANALYTICAL 114 - ITERATING TO FIT THE MODEL 116 - NORMALIZING INPUTS 119 : VISUALIZING (AGAIN) 122 5.5 PYTORCH'S AUTOGRAD: BACKPROPAGATING ALL THINGS 123 COMPUTING THE GRADIENT AUTOMATICALLY 123 - OPTIMIZERS A LA CARTE 127 : TRAINING, VALIDATION, AND OVERFITTING 131 AUTOGRAD NITS AND SWITCHING IT OFF. 137 5.6 CONCLUSION 139 5.7 EXERCISE 139 5.8 SUMMARY 139 USING A NEURAL NETWORK TO FIT THE DATA 141 6.1 ARTIFICIAL NEURONS142 COMPOSING A MULTILAYER NETWORK 144 - UNDERSTANDING THE ERROR FUNCTION 144 : ALL WE NEED IS ACTIVATION 145 : MORE ACTIVATION FUNCTIONS 147 : CHOOSING THE BEST ACTIVATION FUNCTION 148 WHAT LEARNING MEANS FOR A NEURAL NETWORK 149 6.2 THE PYTORCH NN MODULE 151 USING--CALL- RATHER THAN FORWARD 152 - RETURNING TO THE LINEAR MODEL 153 6.3 FINALLY A NEURAL NETWORK 158 REPLACING THE LINEAR MODEL 158 - INSPECTING THE PARAMETERS 159 COMPARING TO THE LINEAR MODEL 161 6.4 CONCLUSION 162 6.5 EXERCISES 162 6.6 SUMMARY 163 TELLING BIRDS FROM AIRPLANES: LEARNING FROM IMAGES 164 7.1 A DATASET OF TINY IMAGES 165 DOWNLOADING CIFAR-10 166 : THE DATASET CLASS 166 DATASET TRANSFORMS 168 - NORMALIZING DATA 170 7.2 DISTINGUISHING BIRDS FROM AIRPLANES 172 BUILDING THE DATASET 173 - A FULLY CONNECTED MODEL 174 OUTPUT OF A CLASSIFIER 175 : REPRESENTING THE OUTPUT AS PROBABILITIES 176 - A LOSS FOR CLASSIFYING 180 : TRAINING THE CLASSIFIER 182 - THE LIMITS OF GOING FULLY CONNECTED 189 7.3 CONCLUSION 191 IX CONTENTS 7.4 EXERCISES 191 7.5 SUMMARY 192 USING CONVOLUTIONS TO GENERALIZE 193 8.1 THE CASE FOR CONVOLUTIONS 194 WHAT CONVOLUTIONS DO 194 8.2 CONVOLUTIONS IN ACTION 196 PADDING THE BOUNDARY 198 - DETECTING FEATURES WITH CONVOLUTIONS 200 - LOOKING FURTHER WITH DEPTH AND POOLING 202 PUTTING IT ALL TOGETHER FOR OUR NETWORK 205 8.3 SUBCLASSING NN.MODULE 207 OUR NETWORK AS AN NN.MODULE 208 - HOW PYTORCH KEEPS TRACK OF PARAMETERS AND SUBMODULES 209 : THE FUNCTIONAL API 210 8.4 TRAINING OUR CONVNET 212 MEASURING ACCURACY 214 : SAVING AND LOADING OUR MODEL 214 TRAINING ON THE GPU 215 8.5 MODEL DESIGN 217 ADDING MEMORY CAPACITY: WIDTH 218 - HELPING OUR MODEL TO CONVERGE AND GENERALIZE: REGULARIZATION 219 - GOING DEEPER TO LEARN MORE COMPLEX STRUCTURES: DEPTH223 - COMPARING THE DESIGNS FROM THIS SECTION 228 - IT'S ALREADY OUTDATED 229 8.6 CONCLUSION 229 8.7 EXERCISES 230 8.8 SUMMARY 231 PART 2 LEARNING FROM IMAGES IN THE REAL WORLD: EARLY DETECTION OF LUNG CANCER. 233 USING PYTORCH TO FIGHT CANCER 235 9.1 INTRODUCTION TO THE USE CASE 236 9.2 PREPARING FOR A LARGE-SCALE PROJECT237 9.3 WHAT IS A CT SCAN, EXACTLY?238 9.4 THE PROJECT: AN END-TO-END DETECTOR FOR LUNG CANCER 241 WHY CAN'T WE JUST THROW DATA AT A NEURAL NETWORK UNTIL IT WORKS? 245 - WHAT IS A NODULE? 249 - OUR DATA SOURCE: THE LUNA GRAND CHALLENGE 251 - DOWNLOADING THE LUNA DATA 251 CONTENTS X 9.5 CONCLUSION 252 9.6 SUMMARY 253 COMBINING DATA SOURCES INTO A UNIFIED DATASET 254 1 10.1 RAW CT DATA FILES 256 10.2 PARSING LUNA'S ANNOTATION DATA 256 TRAINING AND VALIDATION SETS 258 : UNIFYING OUR ANNOTATION AND CANDIDATE DATA 259 10.3 LOADING INDIVIDUAL CT SCANS 262 HOUNSFIELD UNITS 264 10.4 LOCATING A NODULE USING THE PATIENT COORDINATE SYSTEM 265 THE PATIENT COORDINATE SYSTEM 265 - CT SCAN SHAPE AND VOXEL SIZES 267 - CONVERTING BETWEEN MILLIMETERS AND VOXEL ADDRESSES 268 - EXTRACTING A NODULE FROM A CT SCAN 270 10.5 A STRAIGHTFORWARD DATASET IMPLEMENTATION 271 CACHING CANDIDATE ARRAYS WITH THE GETCTRAWCANDIDATE FUNCTION 274 : CONSTRUCTING OUR DATASET IN LUNADATASET \_INIT\_- 275 - A TRAINING/VALIDATION SPLIT 275 - RENDERING THE DATA 277 10.6 CONCLUSION 277 10.7 EXERCISES 278 10.8 SUMMARY 278 TRAINING A CLASSIFICATION MODEL TO DETECT SUSPECTED TUMORS 279 11.1 A FOUNDATIONAL MODEL AND TRAINING LOOP 280 11.2 THE MAIN ENTRY POINT FOR OUR APPLICATION 282 11.3 PRETRAINING SETUP AND INITIALIZATION 284 INITIALIZING THE MODEL AND OPTIMIZER 285 : CARE AND FEEDING OF DATA LOADERS 287 11.4 OUR FIRST-PASS NEURAL NETWORK DESIGN 289 THE CORE CONVOLUTIONS 290 : THE FULL MODEL 293 11.5 TRAINING AND VALIDATING THE MODEL 295 THE COMPUTEBATCHLOSS FUNCTION 297 - THE VALIDATION LOOP IS SIMILAR 299 11.6 OUTPUTTING PERFORMANCE METRICS 300 THE LOGMETRICS FUNCTION 301 XI CONTENTS 11.7 RUNNING THE TRAINING SCRIPT 304 NEEDED DATA FOR TRAINING 305 - INTERLUDE: THE ENUMERATEWITHESTIMATE FUNCTION 306 11.8 EVALUATING THE MODEL: GETTING 99.7% CORRECT MEANS WE'RE DONE, RIGHT? 308 11.9 GRAPHING TRAINING METRICS WITH TENSORBOARD 309 RUNNING TENSORBOARD 309 - ADDING TENSORBOARD SUPPORT TO THE METRICS LOGGING FUNCTION 313 11.10 WHY ISN'T THE MODEL LEARNING TO DETECT NODULES? 315 11.11 CONCLUSION 316 11.12 EXERCISES 316 11.13 SUMMARY 316 12 IMPROVING TRAINING WITH METRICS AND AUGMENTATION 318 12.1 HIGH-LEVEL PLAN FOR IMPROVEMENT 319 12.2 GOOD DOGS VS. BAD GUYS: FALSE POSITIVES AND FALSE NEGATIVES 320 12.3 GRAPHING THE POSITIVES AND NEGATIVES322 RECALL IS ROXIE'S STRENGTH 324 - PRECISION IS PRESTON'S FORTE 326 IMPLEMENTING PRECISION AND RECALL IN LOGMETRICS 327 : OUR ULTIMATE PERFORMANCE METRIC: THE F1 SCORE 328 - HOW DOES OUR MODEL PERFORM WITH OUR NEW METRICS? 332 12.4 WHAT DOES AN IDEAL DATASET LOOK LIKE? 334 MAKING THE DATA LOOK LESS LIKE THE ACTUAL AND MORE LIKE THE "IDEAL" 336 CONTRASTING TRAINING WITH A BALANCED LUNADATASET TO PREVIOUS RUNS 341 - RECOGNIZING THE SYMPTOMS OF OVERFITTING 343 12.5 REVISITING THE PROBLEM OF OVERFITTING 345 AN OVERFIT FACE-TO-AGE PREDICTION MODEL 345 12.6 PREVENTING OVERFITTING WITH DATA AUGMENTATION 346 SPECIFIC DATA AUGMENTATION TECHNIQUES 347 : SEEING THE IMPROVEMENT FROM DATA AUGMENTATION 352 12.7 CONCLUSION 354 12.8 EXERCISES 355 12.9 SUMMARY 356 USING SEGMENTATION TO FIND SUSPECTED NODULES 357 13.1 ADDING A SECOND MODEL TO OUR PROJECT 358 13.2 VARIOUS TYPES OF SEGMENTATION 360 XII CONTENTS 13.3 SEMANTIC SEGMENTATION: PER-PIXEL CLASSIFICATION 361 THE U-NET ARCHITECTURE 364 13.4 UPDATING THE MODEL FOR SEGMENTATION 366 ADAPTING AN OFF-THE-SHELF MODEL TO OUR PROJECT 367 13.5 UPDATING THE DATASET FOR SEGMENTATION 369 U-NET HAS VERY SPECIFIC INPUT SIZE REQUIREMENTS 370 - U-NET TRADE YINU PUNOU8 OYP BUIPING - OLEDYPP AZ 'SA AE IOF SFFO DATA 371 : IMPLEMENTING LUNA2DSEGMENTATIONDATASET 378 DESIGNING OUR TRAINING AND VALIDATION DATA 382 - IMPLEMENTING TRAININGLUNA2DSEGMENTATIONDATASET 383 - AUGMENTING ON THE GPU 384 13.6 UPDATING THE TRAINING SCRIPT FOR SEGMENTATION 386 INITIALIZING OUR SEGMENTATION AND AUGMENTATION MODELS 387 USING THE ADAM OPTIMIZER 388 - DICE LOSS 389 - GETTING IMAGES INTO TENSORBOARD 392 : UPDATING OUR METRICS LOGGING 396 SAVING OUR MODEL 397 13.7 RESULTS 399 13.8 CONCLUSION 401 13.9 EXERCISES 402 13.10 SUMMARY 402 END-TO-END NODULE ANALYSIS, AND WHERE TO GO NEXT 404 14.1 TOWARDS THE FINISH LINE 405 14.2 INDEPENDENCE OF THE VALIDATION SET 407 14.3 BRIDGING CT SEGMENTATION AND NODULE CANDIDATE CLASSIFICATION 408 SEGMENTATION 410 - GROUPING VOXELS INTO NODULE CANDIDATES 411 DID WE FIND A NODULE? CLASSIFICATION TO REDUCE FALSE POSITIVES 412 14.4 QUANTITATIVE VALIDATION 416 14.5 PREDICTING MALIGNANCY 417 GETTING MALIGNANCY INFORMATION 417 - AN AREA UNDER THE CURVE BASELINE: CLASSIFYING BY DIAMETER 419 - REUSING PREEXISTING WEIGHTS: FINE-TUNING 422 - MORE OUTPUT IN TENSORBOARD 428 14.6 WHAT WE SEE WHEN WE DIAGNOSE 432 TRAINING, VALIDATION, AND TEST SETS 433 14.7 WHAT NEXT? ADDITIONAL SOURCES OF INSPIRATION (AND DATA) 434 PREVENTING OVERFTTING: BETTER REGULARIZATION 434 - REFINED TRAINING DATA 437 : COMPETITION RESULTS AND RESEARCH PAPERS 438 XIII CONTENTS 14.8 CONCLUSION 439 BEHIND THE CURTAIN 439 14.9 EXERCISES 441 14.10 SUMMARY 441 PART 3 DEPLOYMENT : 443 DEPLOYING TO PRODUCTION445 15.1 SERVING PYTORCH MODELS 446 OUR MODEL BEHIND A FLASK SERVER 446 - WHAT WE WANT FROM DEPLOYMENT 448 : REQUEST BATCHING 449 15.2 EXPORTING MODELS 455 INTEROPERABILITY BEYOND PYTORCH WITH ONNX 455 - PYTORCH'S OWN EXPORT: TRACING 456 - OUR SERVER WITH A TRACED MODEL 458 15.3 INTERACTING WITH THE PYTORCH JIT 458 WHAT TO EXPECT FROM MOVING BEYOND CLASSIC PYTHON/PYTORCH 458 THE DUAL NATURE OF PYTORCH AS INTERFACE AND BACKEND 460 TORCHSCRIPT 461 - SCRIPTING THE GAPS OF TRACEABILITY 464 15.4 LIBTORCH: PYTORCH IN C++ 465 RUNNING JITED MODELS FROM C++ 465 : C++ FROM THE START: THE C++ API 468 15.5 GOING MOBILE 472 IMPROVING EFFICIENCY: MODEL DESIGN AND QUANTIZATION 475 15.6 EMERGING TECHNOLOGY: ENTERPRISE SERVING OF PYTORCH MODELS 476 15.7 CONCLUSION 477 15.8 EXERCISES 477 15.9 SUMMARY 477 INDEX 479DEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING UNTRAINED CLOUD MULTIPROCESS MODEL DATA LOADING SAMPLE BATCH TRAINED TENSORS TENSOR PRODUCTION MODEL DATA (ONNX,JIT SOURCE TRAINING TORCHSCRIPT) LOOP DISTRIBUTED TRAINING PRODUCTION ON MULTIPLE SERVERS/GPUS SERVER DEEP LEARNING WITH PYTORCH ELI STEVENS,LUCA ANTIGA AND THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING SHELTER ISLAND FOR ONLINE INFORMATION AND ORDERING OF THIS AND OTHER MANNING BOOKS, PLEASE VISIT WWW.MANNING.COM. THE PUBLISHER OFFERS DISCOUNTS ON THIS BOOK WHEN ORDERED IN QUANTITY FOR MORE INFORMATION, PLEASE CONTACT SPECIAL SALES DEPARTMENT MANNING PUBLICATIONS CO. 20 BALDWIN ROAD PO BOX 761 SHELTER ISLAND, NY 11964 EMAIL: ORDERS@MANNING.COM @2020 BY MANNING PUBLICATIONS CO. ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED, STORED IN A RETRIEVAL SYSTEM, OR TRANSMITTED, IN ANY FORM OR BY MEANS ELECTRONIC, MECHANICAL, PHOTOCOPYING, OR OTHERWISE, WITHOUT PRIOR WRITTEN PERMISSION OF THE PUBLISHER. MANY OF THE DESIGNATIONS USED BY MANUFACTURERS AND SELLERS TO DISTINGUISH THEIR PRODUCTS ARE CLAIMED AS TRADEMARKS. WHERE THOSE DESIGNATIONS APPEAR IN THE BOOK, AND MANNING PUBLICATIONS WAS AWARE OF A TRADEMARK CLAIM, THE DESIGNATIONS HAVE BEEN PRINTED IN INITIAL CAPS OR ALL CAPS. @ RECOGNIZING THE IMPORTANCE OF PRESERVING WHAT HAS BEEN WRITTEN, IT IS MANNING'S POLICY TO HAVE THE BOOKS WE PUBLISH PRINTED ON ACID-FREE PAPER, AND WE EXERT OUR BEST EFFORTS TO THAT END. RECOGNIZING ALSO OUR RESPONSIBILITY TO CONSERVE THE RESOURCES OF OUR PLANET, MANNING BOOKS ARE PRINTED ON PAPER THAT IS AT LEAST 15 PERCENT RECYCLED AND PROCESSED WITHOUT THE USE OF ELEMENTAL CHLORINE. MANNING PUBLICATIONS CO. DEVELOPMENT EDITOR: FRANCES LEFKOWITZ 20 BALDWIN ROAD TECHNICAL DEVELOPMENT EDITOR: ARTHUR ZUBAREV PO BOX 761 REVIEW EDITOR: IVAN MARTINOVIC SHELTER ISLAND, NY 11964 PRODUCTION EDITOR: DEIRDRE HIAM COPYEDITOR: TIFFANY TAYLOR PROOFREADER: KATIE TENNANT TECHNICAL PROOFREADER: KOSTAS PASSADIS TYPESETTER: GORDAN SALINOVIC COVER DESIGNER: MARIJA TUDOR ISBN 9781617295263 PRINTED IN THE UNITED STATES OF AMERICA TO MY WIFE (THIS BOOK WOULD NOT HAVE HAPPENED WITHOUT HER INVALUABLE SUPPORT AND PARTNERSHIP), MY PARENTS (I WOULD NOT HAVE HAPPENED WITHOUT THEM), AND MY CHILDREN (THIS BOOK WOULD HAVE HAPPENED A LOT SOONER BUT FOR THEM) THANK YOU FOR BEING MY HOME, MY FOUNDATION, AND MY JOY ELI STEVENS SAME :-) BUT, REALLY, THIS IS FOR YOU, ALICE AND LUIGI LUCA ANTIGA TO EVA, REBEKKA, JONATHAN, AND DAVID. THOMAS VIEHMANN CONTENTS FOREWORD XV PREFACE XVII ACKNOWLEDGMENTS XIX ABOUT THIS BOOK XXI ABOUT THE AUTHORS XXVII ABOUT THE COVER ILLUSTRATION XXVIII PART 1 CORE PYTORCH. INTRODUCING DEEP LEARNING AND THE PYTORCH LIBRARY3 1.1 THE DEEP LEARNING REVOLUTION 4 1.2 PYTORCH FOR DEEP LEARNING 6 1.3 WHY PYTORCH?7 THE DEEP LEARNING COMPETITIVE LANDSCAPE8 1.4 AN OVERVIEW OF HOW PYTORCH SUPPORTS DEEP LEARNING PROJECTS 10 1.5 HARDWARE AND SOFTWARE REQUIREMENTS 13 USING JUPYTER NOTEBOOKS 14 1.6 EXERCISES 15 1.7 SUMMARY 15 VI CONTENTS PRETRAINED NETWORKS 16 2.1 A PRETRAINED NETWORK THAT RECOGNIZES THE SUBJECT OF AN IMAGE 17 OBTAINING A PRETRAINED NETWORK FOR IMAGE RECOGNITION 19 ALEXNET 20 - RESNET 22 - READY, SET, ALMOST RUN 22 RUN! 25 2.2 A PRETRAINED MODEL THAT FAKES IT UNTIL IT MAKES IT 27 THE GAN GAME 28 : CYCLEGAN 29 - A NETWORK THAT TURNS HORSES INTO ZEBRAS 30 2.3 A PRETRAINED NETWORK THAT DESCRIBES SCENES 33 NEURALTALK2 34 2.4 TORCH HUB 35 2.5 CONCLUSION 37 2.6 EXERCISES 38 2.7 SUMMARY 38 IT STARTS WITH A TENSOR 39 3.1 THE WORLD AS FLOATING-POINT NUMBERS 40 3.2 TENSORS: MULTIDIMENSIONAL ARRAYS 42 FROM PYTHON LISTS TO PYTORCH TENSORS 42 - CONSTRUCTING OUR FIRST TENSORS 43 - THE ESSENCE OF TENSORS 43 3.3 INDEXING TENSORS 46 3.4 NAMED TENSORS 46 3.5 TENSOR ELEMENT TYPES 50 SPECIFYING THE NUMERIC TYPE WITH DTYPE 50 - A DTYPE FOR EVERY OCCASION 51 - MANAGING A TENSOR'S DTYPE ATTRIBUTE 51 3.6 THE TENSOR API 52 3.7 TENSORS: SCENIC VIEWS OF STORAGE 53 INDEXING INTO STORAGE 54 - MODIFYING STORED VALUES: IN-PLACE OPERATIONS 55 3.8 TENSOR METADATA: SIZE, OFFSET, AND STRIDE 55 VIEWS OF ANOTHER TENSOR'S STORAGE 56 - TRANSPOSING WITHOUT COPYING 58 : TRANSPOSING IN HIGHER DIMENSIONS 60 CONTIGUOUS TENSORS 60 3.9 MOVING TENSORS TO THE GPU 62 MANAGING A TENSOR'S DEVICE ATTRIBUTE 63 VII CONTENTS 3.10 NUMPY INTEROPERABILITY64 3.11 GENERALIZED TENSORS ARE TENSORS, TOO 65 3.12 SERIALIZING TENSORS 66 SERIALIZING TO HDF5 WITH H5PY 67 3.13 CONCLUSION 68 3.14 EXERCISES 68 3.15 SUMMARY 68 REAL-WORLD DATA REPRESENTATION USING TENSORS 70 4.1 WORKING WITH IMAGES 71 ADDING COLOR CHANNELS 72 - LOADING AN IMAGE FILE 72 CHANGING THE LAYOUT 73 : NORMALIZING THE DATA 74 4.2 3D IMAGES: VOLUMETRIC DATA 75 LOADING A SPECIALIZED FORMAT 76 4.3 REPRESENTING TABULAR DATA 77 USING A REAL-WORLD DATASET 77 - LOADING A WINE DATA TENSOR 78 REPRESENTING SCORES 81 : ONE-HOT ENCODING 81 : WHEN TO CATEGORIZE 83 - FINDING THRESHOLDS 84 4.4 WORKING WITH TIME SERIES 87 ADDING A TIME DIMENSION 88 - SHAPING THE DATA BY TIME PERIOD 89 - READY FOR TRAINING 90 4.5 REPRESENTING TEXT 93 CONVERTING TEXT TO NUMBERS94 - ONE-HOT-ENCODING CHARACTERS94 ONE-HOT ENCODING WHOLE WORDS 96 : TEXT EMBEDDINGS 98 TEXT EMBEDDINGS AS A BLUEPRINT 100 4.6 CONCLUSION 101 4.7 EXERCISES 101 4.8 SUMMARY 102 THE MECHANICS OF LEARNING 103 5.1 A TIMELESS LESSON IN MODELING 104 5.2 LEARNING IS JUST PARAMETER ESTIMATION 106 A HOT PROBLEM 107 : GATHERING SOME DATA 107 : VISUALIZING THE DATA 108 : CHOOSING A LINEAR MODEL AS A FIRST TRY 108 5.3 LESS LOSS IS WHAT WE WANT 109 FROM PROBLEM BACK TO PYTORCH 110 VIII CONTENTS 5.4 DOWN ALONG THE GRADIENT 113 DECREASING LOSS 113 - GETTING ANALYTICAL 114 - ITERATING TO FIT THE MODEL 116 - NORMALIZING INPUTS 119 : VISUALIZING (AGAIN) 122 5.5 PYTORCH'S AUTOGRAD: BACKPROPAGATING ALL THINGS 123 COMPUTING THE GRADIENT AUTOMATICALLY 123 - OPTIMIZERS A LA CARTE 127 : TRAINING, VALIDATION, AND OVERFITTING 131 AUTOGRAD NITS AND SWITCHING IT OFF. 137 5.6 CONCLUSION 139 5.7 EXERCISE 139 5.8 SUMMARY 139 USING A NEURAL NETWORK TO FIT THE DATA 141 6.1 ARTIFICIAL NEURONS142 COMPOSING A MULTILAYER NETWORK 144 - UNDERSTANDING THE ERROR FUNCTION 144 : ALL WE NEED IS ACTIVATION 145 : MORE ACTIVATION FUNCTIONS 147 : CHOOSING THE BEST ACTIVATION FUNCTION 148 WHAT LEARNING MEANS FOR A NEURAL NETWORK 149 6.2 THE PYTORCH NN MODULE 151 USING--CALL- RATHER THAN FORWARD 152 - RETURNING TO THE LINEAR MODEL 153 6.3 FINALLY A NEURAL NETWORK 158 REPLACING THE LINEAR MODEL 158 - INSPECTING THE PARAMETERS 159 COMPARING TO THE LINEAR MODEL 161 6.4 CONCLUSION 162 6.5 EXERCISES 162 6.6 SUMMARY 163 TELLING BIRDS FROM AIRPLANES: LEARNING FROM IMAGES 164 7.1 A DATASET OF TINY IMAGES 165 DOWNLOADING CIFAR-10 166 : THE DATASET CLASS 166 DATASET TRANSFORMS 168 - NORMALIZING DATA 170 7.2 DISTINGUISHING BIRDS FROM AIRPLANES 172 BUILDING THE DATASET 173 - A FULLY CONNECTED MODEL 174 OUTPUT OF A CLASSIFIER 175 : REPRESENTING THE OUTPUT AS PROBABILITIES 176 - A LOSS FOR CLASSIFYING 180 : TRAINING THE CLASSIFIER 182 - THE LIMITS OF GOING FULLY CONNECTED 189 7.3 CONCLUSION 191 IX CONTENTS 7.4 EXERCISES 191 7.5 SUMMARY 192 USING CONVOLUTIONS TO GENERALIZE 193 8.1 THE CASE FOR CONVOLUTIONS 194 WHAT CONVOLUTIONS DO 194 8.2 CONVOLUTIONS IN ACTION 196 PADDING THE BOUNDARY 198 - DETECTING FEATURES WITH CONVOLUTIONS 200 - LOOKING FURTHER WITH DEPTH AND POOLING 202 PUTTING IT ALL TOGETHER FOR OUR NETWORK 205 8.3 SUBCLASSING NN.MODULE 207 OUR NETWORK AS AN NN.MODULE 208 - HOW PYTORCH KEEPS TRACK OF PARAMETERS AND SUBMODULES 209 : THE FUNCTIONAL API 210 8.4 TRAINING OUR CONVNET 212 MEASURING ACCURACY 214 : SAVING AND LOADING OUR MODEL 214 TRAINING ON THE GPU 215 8.5 MODEL DESIGN 217 ADDING MEMORY CAPACITY: WIDTH 218 - HELPING OUR MODEL TO CONVERGE AND GENERALIZE: REGULARIZATION 219 - GOING DEEPER TO LEARN MORE COMPLEX STRUCTURES: DEPTH223 - COMPARING THE DESIGNS FROM THIS SECTION 228 - IT'S ALREADY OUTDATED 229 8.6 CONCLUSION 229 8.7 EXERCISES 230 8.8 SUMMARY 231 PART 2 LEARNING FROM IMAGES IN THE REAL WORLD: EARLY DETECTION OF LUNG CANCER. 233 USING PYTORCH TO FIGHT CANCER 235 9.1 INTRODUCTION TO THE USE CASE 236 9.2 PREPARING FOR A LARGE-SCALE PROJECT237 9.3 WHAT IS A CT SCAN, EXACTLY?238 9.4 THE PROJECT: AN END-TO-END DETECTOR FOR LUNG CANCER 241 WHY CAN'T WE JUST THROW DATA AT A NEURAL NETWORK UNTIL IT WORKS? 245 - WHAT IS A NODULE? 249 - OUR DATA SOURCE: THE LUNA GRAND CHALLENGE 251 - DOWNLOADING THE LUNA DATA 251 CONTENTS X 9.5 CONCLUSION 252 9.6 SUMMARY 253 COMBINING DATA SOURCES INTO A UNIFIED DATASET 254 1 10.1 RAW CT DATA FILES 256 10.2 PARSING LUNA'S ANNOTATION DATA 256 TRAINING AND VALIDATION SETS 258 : UNIFYING OUR ANNOTATION AND CANDIDATE DATA 259 10.3 LOADING INDIVIDUAL CT SCANS 262 HOUNSFIELD UNITS 264 10.4 LOCATING A NODULE USING THE PATIENT COORDINATE SYSTEM 265 THE PATIENT COORDINATE SYSTEM 265 - CT SCAN SHAPE AND VOXEL SIZES 267 - CONVERTING BETWEEN MILLIMETERS AND VOXEL ADDRESSES 268 - EXTRACTING A NODULE FROM A CT SCAN 270 10.5 A STRAIGHTFORWARD DATASET IMPLEMENTATION 271 CACHING CANDIDATE ARRAYS WITH THE GETCTRAWCANDIDATE FUNCTION 274 : CONSTRUCTING OUR DATASET IN LUNADATASET \_INIT\_- 275 - A TRAINING/VALIDATION SPLIT 275 - RENDERING THE DATA 277 10.6 CONCLUSION 277 10.7 EXERCISES 278 10.8 SUMMARY 278 TRAINING A CLASSIFICATION MODEL TO DETECT SUSPECTED TUMORS 279 11.1 A FOUNDATIONAL MODEL AND TRAINING LOOP 280 11.2 THE MAIN ENTRY POINT FOR OUR APPLICATION 282 11.3 PRETRAINING SETUP AND INITIALIZATION 284 INITIALIZING THE MODEL AND OPTIMIZER 285 : CARE AND FEEDING OF DATA LOADERS 287 11.4 OUR FIRST-PASS NEURAL NETWORK DESIGN 289 THE CORE CONVOLUTIONS 290 : THE FULL MODEL 293 11.5 TRAINING AND VALIDATING THE MODEL 295 THE COMPUTEBATCHLOSS FUNCTION 297 - THE VALIDATION LOOP IS SIMILAR 299 11.6 OUTPUTTING PERFORMANCE METRICS 300 THE LOGMETRICS FUNCTION 301 XI CONTENTS 11.7 RUNNING THE TRAINING SCRIPT 304 NEEDED DATA FOR TRAINING 305 - INTERLUDE: THE ENUMERATEWITHESTIMATE FUNCTION 306 11.8 EVALUATING THE MODEL: GETTING 99.7% CORRECT MEANS WE'RE DONE, RIGHT? 308 11.9 GRAPHING TRAINING METRICS WITH TENSORBOARD 309 RUNNING TENSORBOARD 309 - ADDING TENSORBOARD SUPPORT TO THE METRICS LOGGING FUNCTION 313 11.10 WHY ISN'T THE MODEL LEARNING TO DETECT NODULES? 315 11.11 CONCLUSION 316 11.12 EXERCISES 316 11.13 SUMMARY 316 12 IMPROVING TRAINING WITH METRICS AND AUGMENTATION 318 12.1 HIGH-LEVEL PLAN FOR IMPROVEMENT 319 12.2 GOOD DOGS VS. BAD GUYS: FALSE POSITIVES AND FALSE NEGATIVES 320 12.3 GRAPHING THE POSITIVES AND NEGATIVES322 RECALL IS ROXIE'S STRENGTH 324 - PRECISION IS PRESTON'S FORTE 326 IMPLEMENTING PRECISION AND RECALL IN LOGMETRICS 327 : OUR ULTIMATE PERFORMANCE METRIC: THE F1 SCORE 328 - HOW DOES OUR MODEL PERFORM WITH OUR NEW METRICS? 332 12.4 WHAT DOES AN IDEAL DATASET LOOK LIKE? 334 MAKING THE DATA LOOK LESS LIKE THE ACTUAL AND MORE LIKE THE "IDEAL" 336 CONTRASTING TRAINING WITH A BALANCED LUNADATASET TO PREVIOUS RUNS 341 - RECOGNIZING THE SYMPTOMS OF OVERFITTING 343 12.5 REVISITING THE PROBLEM OF OVERFITTING 345 AN OVERFIT FACE-TO-AGE PREDICTION MODEL 345 12.6 PREVENTING OVERFITTING WITH DATA AUGMENTATION 346 SPECIFIC DATA AUGMENTATION TECHNIQUES 347 : SEEING THE IMPROVEMENT FROM DATA AUGMENTATION 352 12.7 CONCLUSION 354 12.8 EXERCISES 355 12.9 SUMMARY 356 USING SEGMENTATION TO FIND SUSPECTED NODULES 357 13.1 ADDING A SECOND MODEL TO OUR PROJECT 358 13.2 VARIOUS TYPES OF SEGMENTATION 360 XII CONTENTS 13.3 SEMANTIC SEGMENTATION: PER-PIXEL CLASSIFICATION 361 THE U-NET ARCHITECTURE 364 13.4 UPDATING THE MODEL FOR SEGMENTATION 366 ADAPTING AN OFF-THE-SHELF MODEL TO OUR PROJECT 367 13.5 UPDATING THE DATASET FOR SEGMENTATION 369 U-NET HAS VERY SPECIFIC INPUT SIZE REQUIREMENTS 370 - U-NET TRADE YINU PUNOU8 OYP BUIPING - OLEDYPP AZ 'SA AE IOF SFFO DATA 371 : IMPLEMENTING LUNA2DSEGMENTATIONDATASET 378 DESIGNING OUR TRAINING AND VALIDATION DATA 382 - IMPLEMENTING TRAININGLUNA2DSEGMENTATIONDATASET 383 - AUGMENTING ON THE GPU 384 13.6 UPDATING THE TRAINING SCRIPT FOR SEGMENTATION 386 INITIALIZING OUR SEGMENTATION AND AUGMENTATION MODELS 387 USING THE ADAM OPTIMIZER 388 - DICE LOSS 389 - GETTING IMAGES INTO TENSORBOARD 392 : UPDATING OUR METRICS LOGGING 396 SAVING OUR MODEL 397 13.7 RESULTS 399 13.8 CONCLUSION 401 13.9 EXERCISES 402 13.10 SUMMARY 402 END-TO-END NODULE ANALYSIS, AND WHERE TO GO NEXT 404 14.1 TOWARDS THE FINISH LINE 405 14.2 INDEPENDENCE OF THE VALIDATION SET 407 14.3 BRIDGING CT SEGMENTATION AND NODULE CANDIDATE CLASSIFICATION 408 SEGMENTATION 410 - GROUPING VOXELS INTO NODULE CANDIDATES 411 DID WE FIND A NODULE? CLASSIFICATION TO REDUCE FALSE POSITIVES 412 14.4 QUANTITATIVE VALIDATION 416 14.5 PREDICTING MALIGNANCY 417 GETTING MALIGNANCY INFORMATION 417 - AN AREA UNDER THE CURVE BASELINE: CLASSIFYING BY DIAMETER 419 - REUSING PREEXISTING WEIGHTS: FINE-TUNING 422 - MORE OUTPUT IN TENSORBOARD 428 14.6 WHAT WE SEE WHEN WE DIAGNOSE 432 TRAINING, VALIDATION, AND TEST SETS 433 14.7 WHAT NEXT? ADDITIONAL SOURCES OF INSPIRATION (AND DATA) 434 PREVENTING OVERFTTING: BETTER REGULARIZATION 434 - REFINED TRAINING DATA 437 : COMPETITION RESULTS AND RESEARCH PAPERS 438 XIII CONTENTS 14.8 CONCLUSION 439 BEHIND THE CURTAIN 439 14.9 EXERCISES 441 14.10 SUMMARY 441 PART 3 DEPLOYMENT : 443 DEPLOYING TO PRODUCTION445 15.1 SERVING PYTORCH MODELS 446 OUR MODEL BEHIND A FLASK SERVER 446 - WHAT WE WANT FROM DEPLOYMENT 448 : REQUEST BATCHING 449 15.2 EXPORTING MODELS 455 INTEROPERABILITY BEYOND PYTORCH WITH ONNX 455 - PYTORCH'S OWN EXPORT: TRACING 456 - OUR SERVER WITH A TRACED MODEL 458 15.3 INTERACTING WITH THE PYTORCH JIT 458 WHAT TO EXPECT FROM MOVING BEYOND CLASSIC PYTHON/PYTORCH 458 THE DUAL NATURE OF PYTORCH AS INTERFACE AND BACKEND 460 TORCHSCRIPT 461 - SCRIPTING THE GAPS OF TRACEABILITY 464 15.4 LIBTORCH: PYTORCH IN C++ 465 RUNNING JITED MODELS FROM C++ 465 : C++ FROM THE START: THE C++ API 468 15.5 GOING MOBILE 472 IMPROVING EFFICIENCY: MODEL DESIGN AND QUANTIZATION 475 15.6 EMERGING TECHNOLOGY: ENTERPRISE SERVING OF PYTORCH MODELS 476 15.7 CONCLUSION 477 15.8 EXERCISES 477 15.9 SUMMARY 477 INDEX 479 FOREWORD WHEN WE STARTED THE PYTORCH PROJECT IN MID-2016, WE WERE A BAND OF OPEN SOURCE HACKERS WHO MET ONLINE AND WANTED TO WRITE BETTER DEEP LEARNING SOFTWARE. TWO OF THE THREE AUTHORS OF THIS BOOK, LUCA ANTIGA AND THOMAS VIEHMANN, WERE INSTRUMENTAL IN DEVELOPING PYTORCH AND MAKING IT THE SUCCESS THAT IT IS TODAY. OUR GOAL WITH PYTORCH WAS TO BUILD THE MOST FLEXIBLE FRAMEWORK POSSIBLE TO EXPRESS DEEP LEARNING ALGORITHMS. WE EXECUTED WITH FOCUS AND HAD A RELATIVELY SHORT DEVELOP MENT TIME TO BUILD A POLISHED PRODUCT FOR THE DEVELOPER MARKET. THIS WOULDN'T HAVE BEEN POSSIBLE IF WE HADN'T BEEN STANDING ON THE SHOULDERS OF GIANTS. PYTORCH DERIVES A SIGNIFICANT PART OF ITS CODEBASE FROM THE TORCH7 PROJECT STARTED IN 2007 BY RONAN COL- LOBERT AND OTHERS, WHICH HAS ROOTS IN THE LUSH PROGRAMMING LANGUAGE PIONEERED BY YANN LECUN AND LEON BOTTOU. THIS RICH HISTORY HELPED US FOCUS ON WHAT NEEDED TO CHANGE, RATHER THAN CONCEPTUALLY STARTING FROM SCRATCH. IT IS HARD TO ATTRIBUTE THE SUCCESS OF PYTORCH TO A SINGLE FACTOR. THE PROJECT OFFERS A GOOD USER EXPERIENCE AND ENHANCED DEBUGGABILITY AND FLEXIBILITY, ULTIMATELY MAKING USERS MORE PRODUCTIVE. THE HUGE ADOPTION OF PYTORCH HAS RESULTED IN A BEAUTIFUL ECO. SYSTEM OF SOFTWARE AND RESEARCH BUILT ON TOP OF IT, MAKING PYTORCH EVEN RICHER IN ITS EXPERIENCE. SEVERAL COURSES AND UNIVERSITY CURRICULA, AS WELL AS A HUGE NUMBER OF ONLINE BLOGS AND TUTORIALS, HAVE BEEN OFFERED TO MAKE PYTORCH EASIER TO LEARN. HOWEVER, WE HAVE SEEN VERY FEW BOOKS. IN 2017, WHEN SOMEONE ASKED ME, "WHEN IS THE PYTORCH BOOK GOING TO BE WRITTEN?" I RESPONDED, "IF IT GETS WRITTEN NOW, I CAN GUARANTEE THAT IT WILL BE OUTDATED BY THE TIME IT IS COMPLETED." XVDEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING UNTRAINED CLOUD MULTIPROCESS MODEL DATA LOADING SAMPLE BATCH TRAINED TENSORS TENSOR PRODUCTION MODEL DATA (ONNX,JIT SOURCE TRAINING TORCHSCRIPT) LOOP DISTRIBUTED TRAINING PRODUCTION ON MULTIPLE SERVERS/GPUS SERVER DEEP LEARNING WITH PYTORCH ELI STEVENS,LUCA ANTIGA AND THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING SHELTER ISLAND FOR ONLINE INFORMATION AND ORDERING OF THIS AND OTHER MANNING BOOKS, PLEASE VISIT WWW.MANNING.COM. THE PUBLISHER OFFERS DISCOUNTS ON THIS BOOK WHEN ORDERED IN QUANTITY FOR MORE INFORMATION, PLEASE CONTACT SPECIAL SALES DEPARTMENT MANNING PUBLICATIONS CO. 20 BALDWIN ROAD PO BOX 761 SHELTER ISLAND, NY 11964 EMAIL: ORDERS@MANNING.COM @2020 BY MANNING PUBLICATIONS CO. ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED, STORED IN A RETRIEVAL SYSTEM, OR TRANSMITTED, IN ANY FORM OR BY MEANS ELECTRONIC, MECHANICAL, PHOTOCOPYING, OR OTHERWISE, WITHOUT PRIOR WRITTEN PERMISSION OF THE PUBLISHER. MANY OF THE DESIGNATIONS USED BY MANUFACTURERS AND SELLERS TO DISTINGUISH THEIR PRODUCTS ARE CLAIMED AS TRADEMARKS. WHERE THOSE DESIGNATIONS APPEAR IN THE BOOK, AND MANNING PUBLICATIONS WAS AWARE OF A TRADEMARK CLAIM, THE DESIGNATIONS HAVE BEEN PRINTED IN INITIAL CAPS OR ALL CAPS. @ RECOGNIZING THE IMPORTANCE OF PRESERVING WHAT HAS BEEN WRITTEN, IT IS MANNING'S POLICY TO HAVE THE BOOKS WE PUBLISH PRINTED ON ACID-FREE PAPER, AND WE EXERT OUR BEST EFFORTS TO THAT END. RECOGNIZING ALSO OUR RESPONSIBILITY TO CONSERVE THE RESOURCES OF OUR PLANET, MANNING BOOKS ARE PRINTED ON PAPER THAT IS AT LEAST 15 PERCENT RECYCLED AND PROCESSED WITHOUT THE USE OF ELEMENTAL CHLORINE. MANNING PUBLICATIONS CO. DEVELOPMENT EDITOR: FRANCES LEFKOWITZ 20 BALDWIN ROAD TECHNICAL DEVELOPMENT EDITOR: ARTHUR ZUBAREV PO BOX 761 REVIEW EDITOR: IVAN MARTINOVIC SHELTER ISLAND, NY 11964 PRODUCTION EDITOR: DEIRDRE HIAM COPYEDITOR: TIFFANY TAYLOR PROOFREADER: KATIE TENNANT TECHNICAL PROOFREADER: KOSTAS PASSADIS TYPESETTER: GORDAN SALINOVIC COVER DESIGNER: MARIJA TUDOR ISBN 9781617295263 PRINTED IN THE UNITED STATES OF AMERICA TO MY WIFE (THIS BOOK WOULD NOT HAVE HAPPENED WITHOUT HER INVALUABLE SUPPORT AND PARTNERSHIP), MY PARENTS (I WOULD NOT HAVE HAPPENED WITHOUT THEM), AND MY CHILDREN (THIS BOOK WOULD HAVE HAPPENED A LOT SOONER BUT FOR THEM) THANK YOU FOR BEING MY HOME, MY FOUNDATION, AND MY JOY ELI STEVENS SAME :-) BUT, REALLY, THIS IS FOR YOU, ALICE AND LUIGI LUCA ANTIGA TO EVA, REBEKKA, JONATHAN, AND DAVID. THOMAS VIEHMANN CONTENTS FOREWORD XV PREFACE XVII ACKNOWLEDGMENTS XIX ABOUT THIS BOOK XXI ABOUT THE AUTHORS XXVII ABOUT THE COVER ILLUSTRATION XXVIII PART 1 CORE PYTORCH. INTRODUCING DEEP LEARNING AND THE PYTORCH LIBRARY3 1.1 THE DEEP LEARNING REVOLUTION 4 1.2 PYTORCH FOR DEEP LEARNING 6 1.3 WHY PYTORCH?7 THE DEEP LEARNING COMPETITIVE LANDSCAPE8 1.4 AN OVERVIEW OF HOW PYTORCH SUPPORTS DEEP LEARNING PROJECTS 10 1.5 HARDWARE AND SOFTWARE REQUIREMENTS 13 USING JUPYTER NOTEBOOKS 14 1.6 EXERCISES 15 1.7 SUMMARY 15 VI CONTENTS PRETRAINED NETWORKS 16 2.1 A PRETRAINED NETWORK THAT RECOGNIZES THE SUBJECT OF AN IMAGE 17 OBTAINING A PRETRAINED NETWORK FOR IMAGE RECOGNITION 19 ALEXNET 20 - RESNET 22 - READY, SET, ALMOST RUN 22 RUN! 25 2.2 A PRETRAINED MODEL THAT FAKES IT UNTIL IT MAKES IT 27 THE GAN GAME 28 : CYCLEGAN 29 - A NETWORK THAT TURNS HORSES INTO ZEBRAS 30 2.3 A PRETRAINED NETWORK THAT DESCRIBES SCENES 33 NEURALTALK2 34 2.4 TORCH HUB 35 2.5 CONCLUSION 37 2.6 EXERCISES 38 2.7 SUMMARY 38 IT STARTS WITH A TENSOR 39 3.1 THE WORLD AS FLOATING-POINT NUMBERS 40 3.2 TENSORS: MULTIDIMENSIONAL ARRAYS 42 FROM PYTHON LISTS TO PYTORCH TENSORS 42 - CONSTRUCTING OUR FIRST TENSORS 43 - THE ESSENCE OF TENSORS 43 3.3 INDEXING TENSORS 46 3.4 NAMED TENSORS 46 3.5 TENSOR ELEMENT TYPES 50 SPECIFYING THE NUMERIC TYPE WITH DTYPE 50 - A DTYPE FOR EVERY OCCASION 51 - MANAGING A TENSOR'S DTYPE ATTRIBUTE 51 3.6 THE TENSOR API 52 3.7 TENSORS: SCENIC VIEWS OF STORAGE 53 INDEXING INTO STORAGE 54 - MODIFYING STORED VALUES: IN-PLACE OPERATIONS 55 3.8 TENSOR METADATA: SIZE, OFFSET, AND STRIDE 55 VIEWS OF ANOTHER TENSOR'S STORAGE 56 - TRANSPOSING WITHOUT COPYING 58 : TRANSPOSING IN HIGHER DIMENSIONS 60 CONTIGUOUS TENSORS 60 3.9 MOVING TENSORS TO THE GPU 62 MANAGING A TENSOR'S DEVICE ATTRIBUTE 63 VII CONTENTS 3.10 NUMPY INTEROPERABILITY64 3.11 GENERALIZED TENSORS ARE TENSORS, TOO 65 3.12 SERIALIZING TENSORS 66 SERIALIZING TO HDF5 WITH H5PY 67 3.13 CONCLUSION 68 3.14 EXERCISES 68 3.15 SUMMARY 68 REAL-WORLD DATA REPRESENTATION USING TENSORS 70 4.1 WORKING WITH IMAGES 71 ADDING COLOR CHANNELS 72 - LOADING AN IMAGE FILE 72 CHANGING THE LAYOUT 73 : NORMALIZING THE DATA 74 4.2 3D IMAGES: VOLUMETRIC DATA 75 LOADING A SPECIALIZED FORMAT 76 4.3 REPRESENTING TABULAR DATA 77 USING A REAL-WORLD DATASET 77 - LOADING A WINE DATA TENSOR 78 REPRESENTING SCORES 81 : ONE-HOT ENCODING 81 : WHEN TO CATEGORIZE 83 - FINDING THRESHOLDS 84 4.4 WORKING WITH TIME SERIES 87 ADDING A TIME DIMENSION 88 - SHAPING THE DATA BY TIME PERIOD 89 - READY FOR TRAINING 90 4.5 REPRESENTING TEXT 93 CONVERTING TEXT TO NUMBERS94 - ONE-HOT-ENCODING CHARACTERS94 ONE-HOT ENCODING WHOLE WORDS 96 : TEXT EMBEDDINGS 98 TEXT EMBEDDINGS AS A BLUEPRINT 100 4.6 CONCLUSION 101 4.7 EXERCISES 101 4.8 SUMMARY 102 THE MECHANICS OF LEARNING 103 5.1 A TIMELESS LESSON IN MODELING 104 5.2 LEARNING IS JUST PARAMETER ESTIMATION 106 A HOT PROBLEM 107 : GATHERING SOME DATA 107 : VISUALIZING THE DATA 108 : CHOOSING A LINEAR MODEL AS A FIRST TRY 108 5.3 LESS LOSS IS WHAT WE WANT 109 FROM PROBLEM BACK TO PYTORCH 110 VIII CONTENTS 5.4 DOWN ALONG THE GRADIENT 113 DECREASING LOSS 113 - GETTING ANALYTICAL 114 - ITERATING TO FIT THE MODEL 116 - NORMALIZING INPUTS 119 : VISUALIZING (AGAIN) 122 5.5 PYTORCH'S AUTOGRAD: BACKPROPAGATING ALL THINGS 123 COMPUTING THE GRADIENT AUTOMATICALLY 123 - OPTIMIZERS A LA CARTE 127 : TRAINING, VALIDATION, AND OVERFITTING 131 AUTOGRAD NITS AND SWITCHING IT OFF. 137 5.6 CONCLUSION 139 5.7 EXERCISE 139 5.8 SUMMARY 139 USING A NEURAL NETWORK TO FIT THE DATA 141 6.1 ARTIFICIAL NEURONS142 COMPOSING A MULTILAYER NETWORK 144 - UNDERSTANDING THE ERROR FUNCTION 144 : ALL WE NEED IS ACTIVATION 145 : MORE ACTIVATION FUNCTIONS 147 : CHOOSING THE BEST ACTIVATION FUNCTION 148 WHAT LEARNING MEANS FOR A NEURAL NETWORK 149 6.2 THE PYTORCH NN MODULE 151 USING--CALL- RATHER THAN FORWARD 152 - RETURNING TO THE LINEAR MODEL 153 6.3 FINALLY A NEURAL NETWORK 158 REPLACING THE LINEAR MODEL 158 - INSPECTING THE PARAMETERS 159 COMPARING TO THE LINEAR MODEL 161 6.4 CONCLUSION 162 6.5 EXERCISES 162 6.6 SUMMARY 163 TELLING BIRDS FROM AIRPLANES: LEARNING FROM IMAGES 164 7.1 A DATASET OF TINY IMAGES 165 DOWNLOADING CIFAR-10 166 : THE DATASET CLASS 166 DATASET TRANSFORMS 168 - NORMALIZING DATA 170 7.2 DISTINGUISHING BIRDS FROM AIRPLANES 172 BUILDING THE DATASET 173 - A FULLY CONNECTED MODEL 174 OUTPUT OF A CLASSIFIER 175 : REPRESENTING THE OUTPUT AS PROBABILITIES 176 - A LOSS FOR CLASSIFYING 180 : TRAINING THE CLASSIFIER 182 - THE LIMITS OF GOING FULLY CONNECTED 189 7.3 CONCLUSION 191 IX CONTENTS 7.4 EXERCISES 191 7.5 SUMMARY 192 USING CONVOLUTIONS TO GENERALIZE 193 8.1 THE CASE FOR CONVOLUTIONS 194 WHAT CONVOLUTIONS DO 194 8.2 CONVOLUTIONS IN ACTION 196 PADDING THE BOUNDARY 198 - DETECTING FEATURES WITH CONVOLUTIONS 200 - LOOKING FURTHER WITH DEPTH AND POOLING 202 PUTTING IT ALL TOGETHER FOR OUR NETWORK 205 8.3 SUBCLASSING NN.MODULE 207 OUR NETWORK AS AN NN.MODULE 208 - HOW PYTORCH KEEPS TRACK OF PARAMETERS AND SUBMODULES 209 : THE FUNCTIONAL API 210 8.4 TRAINING OUR CONVNET 212 MEASURING ACCURACY 214 : SAVING AND LOADING OUR MODEL 214 TRAINING ON THE GPU 215 8.5 MODEL DESIGN 217 ADDING MEMORY CAPACITY: WIDTH 218 - HELPING OUR MODEL TO CONVERGE AND GENERALIZE: REGULARIZATION 219 - GOING DEEPER TO LEARN MORE COMPLEX STRUCTURES: DEPTH223 - COMPARING THE DESIGNS FROM THIS SECTION 228 - IT'S ALREADY OUTDATED 229 8.6 CONCLUSION 229 8.7 EXERCISES 230 8.8 SUMMARY 231 PART 2 LEARNING FROM IMAGES IN THE REAL WORLD: EARLY DETECTION OF LUNG CANCER. 233 USING PYTORCH TO FIGHT CANCER 235 9.1 INTRODUCTION TO THE USE CASE 236 9.2 PREPARING FOR A LARGE-SCALE PROJECT237 9.3 WHAT IS A CT SCAN, EXACTLY?238 9.4 THE PROJECT: AN END-TO-END DETECTOR FOR LUNG CANCER 241 WHY CAN'T WE JUST THROW DATA AT A NEURAL NETWORK UNTIL IT WORKS? 245 - WHAT IS A NODULE? 249 - OUR DATA SOURCE: THE LUNA GRAND CHALLENGE 251 - DOWNLOADING THE LUNA DATA 251 CONTENTS X 9.5 CONCLUSION 252 9.6 SUMMARY 253 COMBINING DATA SOURCES INTO A UNIFIED DATASET 254 1 10.1 RAW CT DATA FILES 256 10.2 PARSING LUNA'S ANNOTATION DATA 256 TRAINING AND VALIDATION SETS 258 : UNIFYING OUR ANNOTATION AND CANDIDATE DATA 259 10.3 LOADING INDIVIDUAL CT SCANS 262 HOUNSFIELD UNITS 264 10.4 LOCATING A NODULE USING THE PATIENT COORDINATE SYSTEM 265 THE PATIENT COORDINATE SYSTEM 265 - CT SCAN SHAPE AND VOXEL SIZES 267 - CONVERTING BETWEEN MILLIMETERS AND VOXEL ADDRESSES 268 - EXTRACTING A NODULE FROM A CT SCAN 270 10.5 A STRAIGHTFORWARD DATASET IMPLEMENTATION 271 CACHING CANDIDATE ARRAYS WITH THE GETCTRAWCANDIDATE FUNCTION 274 : CONSTRUCTING OUR DATASET IN LUNADATASET \_INIT\_- 275 - A TRAINING/VALIDATION SPLIT 275 - RENDERING THE DATA 277 10.6 CONCLUSION 277 10.7 EXERCISES 278 10.8 SUMMARY 278 TRAINING A CLASSIFICATION MODEL TO DETECT SUSPECTED TUMORS 279 11.1 A FOUNDATIONAL MODEL AND TRAINING LOOP 280 11.2 THE MAIN ENTRY POINT FOR OUR APPLICATION 282 11.3 PRETRAINING SETUP AND INITIALIZATION 284 INITIALIZING THE MODEL AND OPTIMIZER 285 : CARE AND FEEDING OF DATA LOADERS 287 11.4 OUR FIRST-PASS NEURAL NETWORK DESIGN 289 THE CORE CONVOLUTIONS 290 : THE FULL MODEL 293 11.5 TRAINING AND VALIDATING THE MODEL 295 THE COMPUTEBATCHLOSS FUNCTION 297 - THE VALIDATION LOOP IS SIMILAR 299 11.6 OUTPUTTING PERFORMANCE METRICS 300 THE LOGMETRICS FUNCTION 301 XI CONTENTS 11.7 RUNNING THE TRAINING SCRIPT 304 NEEDED DATA FOR TRAINING 305 - INTERLUDE: THE ENUMERATEWITHESTIMATE FUNCTION 306 11.8 EVALUATING THE MODEL: GETTING 99.7% CORRECT MEANS WE'RE DONE, RIGHT? 308 11.9 GRAPHING TRAINING METRICS WITH TENSORBOARD 309 RUNNING TENSORBOARD 309 - ADDING TENSORBOARD SUPPORT TO THE METRICS LOGGING FUNCTION 313 11.10 WHY ISN'T THE MODEL LEARNING TO DETECT NODULES? 315 11.11 CONCLUSION 316 11.12 EXERCISES 316 11.13 SUMMARY 316 12 IMPROVING TRAINING WITH METRICS AND AUGMENTATION 318 12.1 HIGH-LEVEL PLAN FOR IMPROVEMENT 319 12.2 GOOD DOGS VS. BAD GUYS: FALSE POSITIVES AND FALSE NEGATIVES 320 12.3 GRAPHING THE POSITIVES AND NEGATIVES322 RECALL IS ROXIE'S STRENGTH 324 - PRECISION IS PRESTON'S FORTE 326 IMPLEMENTING PRECISION AND RECALL IN LOGMETRICS 327 : OUR ULTIMATE PERFORMANCE METRIC: THE F1 SCORE 328 - HOW DOES OUR MODEL PERFORM WITH OUR NEW METRICS? 332 12.4 WHAT DOES AN IDEAL DATASET LOOK LIKE? 334 MAKING THE DATA LOOK LESS LIKE THE ACTUAL AND MORE LIKE THE "IDEAL" 336 CONTRASTING TRAINING WITH A BALANCED LUNADATASET TO PREVIOUS RUNS 341 - RECOGNIZING THE SYMPTOMS OF OVERFITTING 343 12.5 REVISITING THE PROBLEM OF OVERFITTING 345 AN OVERFIT FACE-TO-AGE PREDICTION MODEL 345 12.6 PREVENTING OVERFITTING WITH DATA AUGMENTATION 346 SPECIFIC DATA AUGMENTATION TECHNIQUES 347 : SEEING THE IMPROVEMENT FROM DATA AUGMENTATION 352 12.7 CONCLUSION 354 12.8 EXERCISES 355 12.9 SUMMARY 356 USING SEGMENTATION TO FIND SUSPECTED NODULES 357 13.1 ADDING A SECOND MODEL TO OUR PROJECT 358 13.2 VARIOUS TYPES OF SEGMENTATION 360 XII CONTENTS 13.3 SEMANTIC SEGMENTATION: PER-PIXEL CLASSIFICATION 361 THE U-NET ARCHITECTURE 364 13.4 UPDATING THE MODEL FOR SEGMENTATION 366 ADAPTING AN OFF-THE-SHELF MODEL TO OUR PROJECT 367 13.5 UPDATING THE DATASET FOR SEGMENTATION 369 U-NET HAS VERY SPECIFIC INPUT SIZE REQUIREMENTS 370 - U-NET TRADE YINU PUNOU8 OYP BUIPING - OLEDYPP AZ 'SA AE IOF SFFO DATA 371 : IMPLEMENTING LUNA2DSEGMENTATIONDATASET 378 DESIGNING OUR TRAINING AND VALIDATION DATA 382 - IMPLEMENTING TRAININGLUNA2DSEGMENTATIONDATASET 383 - AUGMENTING ON THE GPU 384 13.6 UPDATING THE TRAINING SCRIPT FOR SEGMENTATION 386 INITIALIZING OUR SEGMENTATION AND AUGMENTATION MODELS 387 USING THE ADAM OPTIMIZER 388 - DICE LOSS 389 - GETTING IMAGES INTO TENSORBOARD 392 : UPDATING OUR METRICS LOGGING 396 SAVING OUR MODEL 397 13.7 RESULTS 399 13.8 CONCLUSION 401 13.9 EXERCISES 402 13.10 SUMMARY 402 END-TO-END NODULE ANALYSIS, AND WHERE TO GO NEXT 404 14.1 TOWARDS THE FINISH LINE 405 14.2 INDEPENDENCE OF THE VALIDATION SET 407 14.3 BRIDGING CT SEGMENTATION AND NODULE CANDIDATE CLASSIFICATION 408 SEGMENTATION 410 - GROUPING VOXELS INTO NODULE CANDIDATES 411 DID WE FIND A NODULE? CLASSIFICATION TO REDUCE FALSE POSITIVES 412 14.4 QUANTITATIVE VALIDATION 416 14.5 PREDICTING MALIGNANCY 417 GETTING MALIGNANCY INFORMATION 417 - AN AREA UNDER THE CURVE BASELINE: CLASSIFYING BY DIAMETER 419 - REUSING PREEXISTING WEIGHTS: FINE-TUNING 422 - MORE OUTPUT IN TENSORBOARD 428 14.6 WHAT WE SEE WHEN WE DIAGNOSE 432 TRAINING, VALIDATION, AND TEST SETS 433 14.7 WHAT NEXT? ADDITIONAL SOURCES OF INSPIRATION (AND DATA) 434 PREVENTING OVERFTTING: BETTER REGULARIZATION 434 - REFINED TRAINING DATA 437 : COMPETITION RESULTS AND RESEARCH PAPERS 438 XIII CONTENTS 14.8 CONCLUSION 439 BEHIND THE CURTAIN 439 14.9 EXERCISES 441 14.10 SUMMARY 441 PART 3 DEPLOYMENT : 443 DEPLOYING TO PRODUCTION445 15.1 SERVING PYTORCH MODELS 446 OUR MODEL BEHIND A FLASK SERVER 446 - WHAT WE WANT FROM DEPLOYMENT 448 : REQUEST BATCHING 449 15.2 EXPORTING MODELS 455 INTEROPERABILITY BEYOND PYTORCH WITH ONNX 455 - PYTORCH'S OWN EXPORT: TRACING 456 - OUR SERVER WITH A TRACED MODEL 458 15.3 INTERACTING WITH THE PYTORCH JIT 458 WHAT TO EXPECT FROM MOVING BEYOND CLASSIC PYTHON/PYTORCH 458 THE DUAL NATURE OF PYTORCH AS INTERFACE AND BACKEND 460 TORCHSCRIPT 461 - SCRIPTING THE GAPS OF TRACEABILITY 464 15.4 LIBTORCH: PYTORCH IN C++ 465 RUNNING JITED MODELS FROM C++ 465 : C++ FROM THE START: THE C++ API 468 15.5 GOING MOBILE 472 IMPROVING EFFICIENCY: MODEL DESIGN AND QUANTIZATION 475 15.6 EMERGING TECHNOLOGY: ENTERPRISE SERVING OF PYTORCH MODELS 476 15.7 CONCLUSION 477 15.8 EXERCISES 477 15.9 SUMMARY 477 INDEX 479 FOREWORD WHEN WE STARTED THE PYTORCH PROJECT IN MID-2016, WE WERE A BAND OF OPEN SOURCE HACKERS WHO MET ONLINE AND WANTED TO WRITE BETTER DEEP LEARNING SOFTWARE. TWO OF THE THREE AUTHORS OF THIS BOOK, LUCA ANTIGA AND THOMAS VIEHMANN, WERE INSTRUMENTAL IN DEVELOPING PYTORCH AND MAKING IT THE SUCCESS THAT IT IS TODAY. OUR GOAL WITH PYTORCH WAS TO BUILD THE MOST FLEXIBLE FRAMEWORK POSSIBLE TO EXPRESS DEEP LEARNING ALGORITHMS. WE EXECUTED WITH FOCUS AND HAD A RELATIVELY SHORT DEVELOP MENT TIME TO BUILD A POLISHED PRODUCT FOR THE DEVELOPER MARKET. THIS WOULDN'T HAVE BEEN POSSIBLE IF WE HADN'T BEEN STANDING ON THE SHOULDERS OF GIANTS. PYTORCH DERIVES A SIGNIFICANT PART OF ITS CODEBASE FROM THE TORCH7 PROJECT STARTED IN 2007 BY RONAN COL- LOBERT AND OTHERS, WHICH HAS ROOTS IN THE LUSH PROGRAMMING LANGUAGE PIONEERED BY YANN LECUN AND LEON BOTTOU. THIS RICH HISTORY HELPED US FOCUS ON WHAT NEEDED TO CHANGE, RATHER THAN CONCEPTUALLY STARTING FROM SCRATCH. IT IS HARD TO ATTRIBUTE THE SUCCESS OF PYTORCH TO A SINGLE FACTOR. THE PROJECT OFFERS A GOOD USER EXPERIENCE AND ENHANCED DEBUGGABILITY AND FLEXIBILITY, ULTIMATELY MAKING USERS MORE PRODUCTIVE. THE HUGE ADOPTION OF PYTORCH HAS RESULTED IN A BEAUTIFUL ECO. SYSTEM OF SOFTWARE AND RESEARCH BUILT ON TOP OF IT, MAKING PYTORCH EVEN RICHER IN ITS EXPERIENCE. SEVERAL COURSES AND UNIVERSITY CURRICULA, AS WELL AS A HUGE NUMBER OF ONLINE BLOGS AND TUTORIALS, HAVE BEEN OFFERED TO MAKE PYTORCH EASIER TO LEARN. HOWEVER, WE HAVE SEEN VERY FEW BOOKS. IN 2017, WHEN SOMEONE ASKED ME, "WHEN IS THE PYTORCH BOOK GOING TO BE WRITTEN?" I RESPONDED, "IF IT GETS WRITTEN NOW, I CAN GUARANTEE THAT IT WILL BE OUTDATED BY THE TIME IT IS COMPLETED." XV XVI FOREWORD WITH THE PUBLICATION OF DEEP LEARNING WITH PYTORCH, WE FINALLY HAVE A DEFINITIVE TREA- TISE ON PYTORCH. IT COVERS THE BASICS AND ABSTRACTIONS IN GREAT DETAIL, TEARING APART THE UNDERPINNINGS OF DATA STRUCTURES LIKE TENSORS AND NEURAL NETWORKS AND MAKING SURE YOU UNDERSTAND THEIR IMPLEMENTATION. ADDITIONALLY, IT COVERS ADVANCED SUBJECTS SUCH AS JIT AND DEPLOYMENT TO PRODUCTION (AN ASPECT OF PYTORCH THAT NO OTHER BOOK CUR- RENTLY COVERS) . NEURAL NETWORKS TO HELP SOLVE A COMPLEX AND IMPORTANT MEDICAL PROBLEM. WITH LUCA'S DEEP EXPERTISE IN BIOENGINEERING AND MEDICAL IMAGING, ELI'S PRACTICAL EXPERIENCE CRE ATING SOFTWARE FOR MEDICAL DEVICES AND DETECTION, AND THOMAS'S BACKGROUND AS A PYTORCH CORE DEVELOPER, THIS JOURNEY IS TREATED CAREFULLY, AS IT SHOULD BE. ALL IN ALL, I HOPE THIS BOOK BECOMES YOUR "EXTENDED" REFERENCE DOCUMENT AND AN IMPORTANT PART OF YOUR LIBRARY OR WORKSHOP SOUMITH CHINTALA COCREATOR OF PYTORCHDEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING UNTRAINED CLOUD MULTIPROCESS MODEL DATA LOADING SAMPLE BATCH TRAINED TENSORS TENSOR PRODUCTION MODEL DATA (ONNX,JIT SOURCE TRAINING TORCHSCRIPT) LOOP DISTRIBUTED TRAINING PRODUCTION ON MULTIPLE SERVERS/GPUS SERVER DEEP LEARNING WITH PYTORCH ELI STEVENS,LUCA ANTIGA AND THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING SHELTER ISLAND FOR ONLINE INFORMATION AND ORDERING OF THIS AND OTHER MANNING BOOKS, PLEASE VISIT WWW.MANNING.COM. THE PUBLISHER OFFERS DISCOUNTS ON THIS BOOK WHEN ORDERED IN QUANTITY FOR MORE INFORMATION, PLEASE CONTACT SPECIAL SALES DEPARTMENT MANNING PUBLICATIONS CO. 20 BALDWIN ROAD PO BOX 761 SHELTER ISLAND, NY 11964 EMAIL: ORDERS@MANNING.COM @2020 BY MANNING PUBLICATIONS CO. ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED, STORED IN A RETRIEVAL SYSTEM, OR TRANSMITTED, IN ANY FORM OR BY MEANS ELECTRONIC, MECHANICAL, PHOTOCOPYING, OR OTHERWISE, WITHOUT PRIOR WRITTEN PERMISSION OF THE PUBLISHER. MANY OF THE DESIGNATIONS USED BY MANUFACTURERS AND SELLERS TO DISTINGUISH THEIR PRODUCTS ARE CLAIMED AS TRADEMARKS. WHERE THOSE DESIGNATIONS APPEAR IN THE BOOK, AND MANNING PUBLICATIONS WAS AWARE OF A TRADEMARK CLAIM, THE DESIGNATIONS HAVE BEEN PRINTED IN INITIAL CAPS OR ALL CAPS. @ RECOGNIZING THE IMPORTANCE OF PRESERVING WHAT HAS BEEN WRITTEN, IT IS MANNING'S POLICY TO HAVE THE BOOKS WE PUBLISH PRINTED ON ACID-FREE PAPER, AND WE EXERT OUR BEST EFFORTS TO THAT END. RECOGNIZING ALSO OUR RESPONSIBILITY TO CONSERVE THE RESOURCES OF OUR PLANET, MANNING BOOKS ARE PRINTED ON PAPER THAT IS AT LEAST 15 PERCENT RECYCLED AND PROCESSED WITHOUT THE USE OF ELEMENTAL CHLORINE. MANNING PUBLICATIONS CO. DEVELOPMENT EDITOR: FRANCES LEFKOWITZ 20 BALDWIN ROAD TECHNICAL DEVELOPMENT EDITOR: ARTHUR ZUBAREV PO BOX 761 REVIEW EDITOR: IVAN MARTINOVIC SHELTER ISLAND, NY 11964 PRODUCTION EDITOR: DEIRDRE HIAM COPYEDITOR: TIFFANY TAYLOR PROOFREADER: KATIE TENNANT TECHNICAL PROOFREADER: KOSTAS PASSADIS TYPESETTER: GORDAN SALINOVIC COVER DESIGNER: MARIJA TUDOR ISBN 9781617295263 PRINTED IN THE UNITED STATES OF AMERICA TO MY WIFE (THIS BOOK WOULD NOT HAVE HAPPENED WITHOUT HER INVALUABLE SUPPORT AND PARTNERSHIP), MY PARENTS (I WOULD NOT HAVE HAPPENED WITHOUT THEM), AND MY CHILDREN (THIS BOOK WOULD HAVE HAPPENED A LOT SOONER BUT FOR THEM) THANK YOU FOR BEING MY HOME, MY FOUNDATION, AND MY JOY ELI STEVENS SAME :-) BUT, REALLY, THIS IS FOR YOU, ALICE AND LUIGI LUCA ANTIGA TO EVA, REBEKKA, JONATHAN, AND DAVID. THOMAS VIEHMANN CONTENTS FOREWORD XV PREFACE XVII ACKNOWLEDGMENTS XIX ABOUT THIS BOOK XXI ABOUT THE AUTHORS XXVII ABOUT THE COVER ILLUSTRATION XXVIII PART 1 CORE PYTORCH. INTRODUCING DEEP LEARNING AND THE PYTORCH LIBRARY3 1.1 THE DEEP LEARNING REVOLUTION 4 1.2 PYTORCH FOR DEEP LEARNING 6 1.3 WHY PYTORCH?7 THE DEEP LEARNING COMPETITIVE LANDSCAPE8 1.4 AN OVERVIEW OF HOW PYTORCH SUPPORTS DEEP LEARNING PROJECTS 10 1.5 HARDWARE AND SOFTWARE REQUIREMENTS 13 USING JUPYTER NOTEBOOKS 14 1.6 EXERCISES 15 1.7 SUMMARY 15 VI CONTENTS PRETRAINED NETWORKS 16 2.1 A PRETRAINED NETWORK THAT RECOGNIZES THE SUBJECT OF AN IMAGE 17 OBTAINING A PRETRAINED NETWORK FOR IMAGE RECOGNITION 19 ALEXNET 20 - RESNET 22 - READY, SET, ALMOST RUN 22 RUN! 25 2.2 A PRETRAINED MODEL THAT FAKES IT UNTIL IT MAKES IT 27 THE GAN GAME 28 : CYCLEGAN 29 - A NETWORK THAT TURNS HORSES INTO ZEBRAS 30 2.3 A PRETRAINED NETWORK THAT DESCRIBES SCENES 33 NEURALTALK2 34 2.4 TORCH HUB 35 2.5 CONCLUSION 37 2.6 EXERCISES 38 2.7 SUMMARY 38 IT STARTS WITH A TENSOR 39 3.1 THE WORLD AS FLOATING-POINT NUMBERS 40 3.2 TENSORS: MULTIDIMENSIONAL ARRAYS 42 FROM PYTHON LISTS TO PYTORCH TENSORS 42 - CONSTRUCTING OUR FIRST TENSORS 43 - THE ESSENCE OF TENSORS 43 3.3 INDEXING TENSORS 46 3.4 NAMED TENSORS 46 3.5 TENSOR ELEMENT TYPES 50 SPECIFYING THE NUMERIC TYPE WITH DTYPE 50 - A DTYPE FOR EVERY OCCASION 51 - MANAGING A TENSOR'S DTYPE ATTRIBUTE 51 3.6 THE TENSOR API 52 3.7 TENSORS: SCENIC VIEWS OF STORAGE 53 INDEXING INTO STORAGE 54 - MODIFYING STORED VALUES: IN-PLACE OPERATIONS 55 3.8 TENSOR METADATA: SIZE, OFFSET, AND STRIDE 55 VIEWS OF ANOTHER TENSOR'S STORAGE 56 - TRANSPOSING WITHOUT COPYING 58 : TRANSPOSING IN HIGHER DIMENSIONS 60 CONTIGUOUS TENSORS 60 3.9 MOVING TENSORS TO THE GPU 62 MANAGING A TENSOR'S DEVICE ATTRIBUTE 63 VII CONTENTS 3.10 NUMPY INTEROPERABILITY64 3.11 GENERALIZED TENSORS ARE TENSORS, TOO 65 3.12 SERIALIZING TENSORS 66 SERIALIZING TO HDF5 WITH H5PY 67 3.13 CONCLUSION 68 3.14 EXERCISES 68 3.15 SUMMARY 68 REAL-WORLD DATA REPRESENTATION USING TENSORS 70 4.1 WORKING WITH IMAGES 71 ADDING COLOR CHANNELS 72 - LOADING AN IMAGE FILE 72 CHANGING THE LAYOUT 73 : NORMALIZING THE DATA 74 4.2 3D IMAGES: VOLUMETRIC DATA 75 LOADING A SPECIALIZED FORMAT 76 4.3 REPRESENTING TABULAR DATA 77 USING A REAL-WORLD DATASET 77 - LOADING A WINE DATA TENSOR 78 REPRESENTING SCORES 81 : ONE-HOT ENCODING 81 : WHEN TO CATEGORIZE 83 - FINDING THRESHOLDS 84 4.4 WORKING WITH TIME SERIES 87 ADDING A TIME DIMENSION 88 - SHAPING THE DATA BY TIME PERIOD 89 - READY FOR TRAINING 90 4.5 REPRESENTING TEXT 93 CONVERTING TEXT TO NUMBERS94 - ONE-HOT-ENCODING CHARACTERS94 ONE-HOT ENCODING WHOLE WORDS 96 : TEXT EMBEDDINGS 98 TEXT EMBEDDINGS AS A BLUEPRINT 100 4.6 CONCLUSION 101 4.7 EXERCISES 101 4.8 SUMMARY 102 THE MECHANICS OF LEARNING 103 5.1 A TIMELESS LESSON IN MODELING 104 5.2 LEARNING IS JUST PARAMETER ESTIMATION 106 A HOT PROBLEM 107 : GATHERING SOME DATA 107 : VISUALIZING THE DATA 108 : CHOOSING A LINEAR MODEL AS A FIRST TRY 108 5.3 LESS LOSS IS WHAT WE WANT 109 FROM PROBLEM BACK TO PYTORCH 110 VIII CONTENTS 5.4 DOWN ALONG THE GRADIENT 113 DECREASING LOSS 113 - GETTING ANALYTICAL 114 - ITERATING TO FIT THE MODEL 116 - NORMALIZING INPUTS 119 : VISUALIZING (AGAIN) 122 5.5 PYTORCH'S AUTOGRAD: BACKPROPAGATING ALL THINGS 123 COMPUTING THE GRADIENT AUTOMATICALLY 123 - OPTIMIZERS A LA CARTE 127 : TRAINING, VALIDATION, AND OVERFITTING 131 AUTOGRAD NITS AND SWITCHING IT OFF. 137 5.6 CONCLUSION 139 5.7 EXERCISE 139 5.8 SUMMARY 139 USING A NEURAL NETWORK TO FIT THE DATA 141 6.1 ARTIFICIAL NEURONS142 COMPOSING A MULTILAYER NETWORK 144 - UNDERSTANDING THE ERROR FUNCTION 144 : ALL WE NEED IS ACTIVATION 145 : MORE ACTIVATION FUNCTIONS 147 : CHOOSING THE BEST ACTIVATION FUNCTION 148 WHAT LEARNING MEANS FOR A NEURAL NETWORK 149 6.2 THE PYTORCH NN MODULE 151 USING--CALL- RATHER THAN FORWARD 152 - RETURNING TO THE LINEAR MODEL 153 6.3 FINALLY A NEURAL NETWORK 158 REPLACING THE LINEAR MODEL 158 - INSPECTING THE PARAMETERS 159 COMPARING TO THE LINEAR MODEL 161 6.4 CONCLUSION 162 6.5 EXERCISES 162 6.6 SUMMARY 163 TELLING BIRDS FROM AIRPLANES: LEARNING FROM IMAGES 164 7.1 A DATASET OF TINY IMAGES 165 DOWNLOADING CIFAR-10 166 : THE DATASET CLASS 166 DATASET TRANSFORMS 168 - NORMALIZING DATA 170 7.2 DISTINGUISHING BIRDS FROM AIRPLANES 172 BUILDING THE DATASET 173 - A FULLY CONNECTED MODEL 174 OUTPUT OF A CLASSIFIER 175 : REPRESENTING THE OUTPUT AS PROBABILITIES 176 - A LOSS FOR CLASSIFYING 180 : TRAINING THE CLASSIFIER 182 - THE LIMITS OF GOING FULLY CONNECTED 189 7.3 CONCLUSION 191 IX CONTENTS 7.4 EXERCISES 191 7.5 SUMMARY 192 USING CONVOLUTIONS TO GENERALIZE 193 8.1 THE CASE FOR CONVOLUTIONS 194 WHAT CONVOLUTIONS DO 194 8.2 CONVOLUTIONS IN ACTION 196 PADDING THE BOUNDARY 198 - DETECTING FEATURES WITH CONVOLUTIONS 200 - LOOKING FURTHER WITH DEPTH AND POOLING 202 PUTTING IT ALL TOGETHER FOR OUR NETWORK 205 8.3 SUBCLASSING NN.MODULE 207 OUR NETWORK AS AN NN.MODULE 208 - HOW PYTORCH KEEPS TRACK OF PARAMETERS AND SUBMODULES 209 : THE FUNCTIONAL API 210 8.4 TRAINING OUR CONVNET 212 MEASURING ACCURACY 214 : SAVING AND LOADING OUR MODEL 214 TRAINING ON THE GPU 215 8.5 MODEL DESIGN 217 ADDING MEMORY CAPACITY: WIDTH 218 - HELPING OUR MODEL TO CONVERGE AND GENERALIZE: REGULARIZATION 219 - GOING DEEPER TO LEARN MORE COMPLEX STRUCTURES: DEPTH223 - COMPARING THE DESIGNS FROM THIS SECTION 228 - IT'S ALREADY OUTDATED 229 8.6 CONCLUSION 229 8.7 EXERCISES 230 8.8 SUMMARY 231 PART 2 LEARNING FROM IMAGES IN THE REAL WORLD: EARLY DETECTION OF LUNG CANCER. 233 USING PYTORCH TO FIGHT CANCER 235 9.1 INTRODUCTION TO THE USE CASE 236 9.2 PREPARING FOR A LARGE-SCALE PROJECT237 9.3 WHAT IS A CT SCAN, EXACTLY?238 9.4 THE PROJECT: AN END-TO-END DETECTOR FOR LUNG CANCER 241 WHY CAN'T WE JUST THROW DATA AT A NEURAL NETWORK UNTIL IT WORKS? 245 - WHAT IS A NODULE? 249 - OUR DATA SOURCE: THE LUNA GRAND CHALLENGE 251 - DOWNLOADING THE LUNA DATA 251 CONTENTS X 9.5 CONCLUSION 252 9.6 SUMMARY 253 COMBINING DATA SOURCES INTO A UNIFIED DATASET 254 1 10.1 RAW CT DATA FILES 256 10.2 PARSING LUNA'S ANNOTATION DATA 256 TRAINING AND VALIDATION SETS 258 : UNIFYING OUR ANNOTATION AND CANDIDATE DATA 259 10.3 LOADING INDIVIDUAL CT SCANS 262 HOUNSFIELD UNITS 264 10.4 LOCATING A NODULE USING THE PATIENT COORDINATE SYSTEM 265 THE PATIENT COORDINATE SYSTEM 265 - CT SCAN SHAPE AND VOXEL SIZES 267 - CONVERTING BETWEEN MILLIMETERS AND VOXEL ADDRESSES 268 - EXTRACTING A NODULE FROM A CT SCAN 270 10.5 A STRAIGHTFORWARD DATASET IMPLEMENTATION 271 CACHING CANDIDATE ARRAYS WITH THE GETCTRAWCANDIDATE FUNCTION 274 : CONSTRUCTING OUR DATASET IN LUNADATASET \_INIT\_- 275 - A TRAINING/VALIDATION SPLIT 275 - RENDERING THE DATA 277 10.6 CONCLUSION 277 10.7 EXERCISES 278 10.8 SUMMARY 278 TRAINING A CLASSIFICATION MODEL TO DETECT SUSPECTED TUMORS 279 11.1 A FOUNDATIONAL MODEL AND TRAINING LOOP 280 11.2 THE MAIN ENTRY POINT FOR OUR APPLICATION 282 11.3 PRETRAINING SETUP AND INITIALIZATION 284 INITIALIZING THE MODEL AND OPTIMIZER 285 : CARE AND FEEDING OF DATA LOADERS 287 11.4 OUR FIRST-PASS NEURAL NETWORK DESIGN 289 THE CORE CONVOLUTIONS 290 : THE FULL MODEL 293 11.5 TRAINING AND VALIDATING THE MODEL 295 THE COMPUTEBATCHLOSS FUNCTION 297 - THE VALIDATION LOOP IS SIMILAR 299 11.6 OUTPUTTING PERFORMANCE METRICS 300 THE LOGMETRICS FUNCTION 301 XI CONTENTS 11.7 RUNNING THE TRAINING SCRIPT 304 NEEDED DATA FOR TRAINING 305 - INTERLUDE: THE ENUMERATEWITHESTIMATE FUNCTION 306 11.8 EVALUATING THE MODEL: GETTING 99.7% CORRECT MEANS WE'RE DONE, RIGHT? 308 11.9 GRAPHING TRAINING METRICS WITH TENSORBOARD 309 RUNNING TENSORBOARD 309 - ADDING TENSORBOARD SUPPORT TO THE METRICS LOGGING FUNCTION 313 11.10 WHY ISN'T THE MODEL LEARNING TO DETECT NODULES? 315 11.11 CONCLUSION 316 11.12 EXERCISES 316 11.13 SUMMARY 316 12 IMPROVING TRAINING WITH METRICS AND AUGMENTATION 318 12.1 HIGH-LEVEL PLAN FOR IMPROVEMENT 319 12.2 GOOD DOGS VS. BAD GUYS: FALSE POSITIVES AND FALSE NEGATIVES 320 12.3 GRAPHING THE POSITIVES AND NEGATIVES322 RECALL IS ROXIE'S STRENGTH 324 - PRECISION IS PRESTON'S FORTE 326 IMPLEMENTING PRECISION AND RECALL IN LOGMETRICS 327 : OUR ULTIMATE PERFORMANCE METRIC: THE F1 SCORE 328 - HOW DOES OUR MODEL PERFORM WITH OUR NEW METRICS? 332 12.4 WHAT DOES AN IDEAL DATASET LOOK LIKE? 334 MAKING THE DATA LOOK LESS LIKE THE ACTUAL AND MORE LIKE THE "IDEAL" 336 CONTRASTING TRAINING WITH A BALANCED LUNADATASET TO PREVIOUS RUNS 341 - RECOGNIZING THE SYMPTOMS OF OVERFITTING 343 12.5 REVISITING THE PROBLEM OF OVERFITTING 345 AN OVERFIT FACE-TO-AGE PREDICTION MODEL 345 12.6 PREVENTING OVERFITTING WITH DATA AUGMENTATION 346 SPECIFIC DATA AUGMENTATION TECHNIQUES 347 : SEEING THE IMPROVEMENT FROM DATA AUGMENTATION 352 12.7 CONCLUSION 354 12.8 EXERCISES 355 12.9 SUMMARY 356 USING SEGMENTATION TO FIND SUSPECTED NODULES 357 13.1 ADDING A SECOND MODEL TO OUR PROJECT 358 13.2 VARIOUS TYPES OF SEGMENTATION 360 XII CONTENTS 13.3 SEMANTIC SEGMENTATION: PER-PIXEL CLASSIFICATION 361 THE U-NET ARCHITECTURE 364 13.4 UPDATING THE MODEL FOR SEGMENTATION 366 ADAPTING AN OFF-THE-SHELF MODEL TO OUR PROJECT 367 13.5 UPDATING THE DATASET FOR SEGMENTATION 369 U-NET HAS VERY SPECIFIC INPUT SIZE REQUIREMENTS 370 - U-NET TRADE YINU PUNOU8 OYP BUIPING - OLEDYPP AZ 'SA AE IOF SFFO DATA 371 : IMPLEMENTING LUNA2DSEGMENTATIONDATASET 378 DESIGNING OUR TRAINING AND VALIDATION DATA 382 - IMPLEMENTING TRAININGLUNA2DSEGMENTATIONDATASET 383 - AUGMENTING ON THE GPU 384 13.6 UPDATING THE TRAINING SCRIPT FOR SEGMENTATION 386 INITIALIZING OUR SEGMENTATION AND AUGMENTATION MODELS 387 USING THE ADAM OPTIMIZER 388 - DICE LOSS 389 - GETTING IMAGES INTO TENSORBOARD 392 : UPDATING OUR METRICS LOGGING 396 SAVING OUR MODEL 397 13.7 RESULTS 399 13.8 CONCLUSION 401 13.9 EXERCISES 402 13.10 SUMMARY 402 END-TO-END NODULE ANALYSIS, AND WHERE TO GO NEXT 404 14.1 TOWARDS THE FINISH LINE 405 14.2 INDEPENDENCE OF THE VALIDATION SET 407 14.3 BRIDGING CT SEGMENTATION AND NODULE CANDIDATE CLASSIFICATION 408 SEGMENTATION 410 - GROUPING VOXELS INTO NODULE CANDIDATES 411 DID WE FIND A NODULE? CLASSIFICATION TO REDUCE FALSE POSITIVES 412 14.4 QUANTITATIVE VALIDATION 416 14.5 PREDICTING MALIGNANCY 417 GETTING MALIGNANCY INFORMATION 417 - AN AREA UNDER THE CURVE BASELINE: CLASSIFYING BY DIAMETER 419 - REUSING PREEXISTING WEIGHTS: FINE-TUNING 422 - MORE OUTPUT IN TENSORBOARD 428 14.6 WHAT WE SEE WHEN WE DIAGNOSE 432 TRAINING, VALIDATION, AND TEST SETS 433 14.7 WHAT NEXT? ADDITIONAL SOURCES OF INSPIRATION (AND DATA) 434 PREVENTING OVERFTTING: BETTER REGULARIZATION 434 - REFINED TRAINING DATA 437 : COMPETITION RESULTS AND RESEARCH PAPERS 438 XIII CONTENTS 14.8 CONCLUSION 439 BEHIND THE CURTAIN 439 14.9 EXERCISES 441 14.10 SUMMARY 441 PART 3 DEPLOYMENT : 443 DEPLOYING TO PRODUCTION445 15.1 SERVING PYTORCH MODELS 446 OUR MODEL BEHIND A FLASK SERVER 446 - WHAT WE WANT FROM DEPLOYMENT 448 : REQUEST BATCHING 449 15.2 EXPORTING MODELS 455 INTEROPERABILITY BEYOND PYTORCH WITH ONNX 455 - PYTORCH'S OWN EXPORT: TRACING 456 - OUR SERVER WITH A TRACED MODEL 458 15.3 INTERACTING WITH THE PYTORCH JIT 458 WHAT TO EXPECT FROM MOVING BEYOND CLASSIC PYTHON/PYTORCH 458 THE DUAL NATURE OF PYTORCH AS INTERFACE AND BACKEND 460 TORCHSCRIPT 461 - SCRIPTING THE GAPS OF TRACEABILITY 464 15.4 LIBTORCH: PYTORCH IN C++ 465 RUNNING JITED MODELS FROM C++ 465 : C++ FROM THE START: THE C++ API 468 15.5 GOING MOBILE 472 IMPROVING EFFICIENCY: MODEL DESIGN AND QUANTIZATION 475 15.6 EMERGING TECHNOLOGY: ENTERPRISE SERVING OF PYTORCH MODELS 476 15.7 CONCLUSION 477 15.8 EXERCISES 477 15.9 SUMMARY 477 INDEX 479 FOREWORD WHEN WE STARTED THE PYTORCH PROJECT IN MID-2016, WE WERE A BAND OF OPEN SOURCE HACKERS WHO MET ONLINE AND WANTED TO WRITE BETTER DEEP LEARNING SOFTWARE. TWO OF THE THREE AUTHORS OF THIS BOOK, LUCA ANTIGA AND THOMAS VIEHMANN, WERE INSTRUMENTAL IN DEVELOPING PYTORCH AND MAKING IT THE SUCCESS THAT IT IS TODAY. OUR GOAL WITH PYTORCH WAS TO BUILD THE MOST FLEXIBLE FRAMEWORK POSSIBLE TO EXPRESS DEEP LEARNING ALGORITHMS. WE EXECUTED WITH FOCUS AND HAD A RELATIVELY SHORT DEVELOP MENT TIME TO BUILD A POLISHED PRODUCT FOR THE DEVELOPER MARKET. THIS WOULDN'T HAVE BEEN POSSIBLE IF WE HADN'T BEEN STANDING ON THE SHOULDERS OF GIANTS. PYTORCH DERIVES A SIGNIFICANT PART OF ITS CODEBASE FROM THE TORCH7 PROJECT STARTED IN 2007 BY RONAN COL- LOBERT AND OTHERS, WHICH HAS ROOTS IN THE LUSH PROGRAMMING LANGUAGE PIONEERED BY YANN LECUN AND LEON BOTTOU. THIS RICH HISTORY HELPED US FOCUS ON WHAT NEEDED TO CHANGE, RATHER THAN CONCEPTUALLY STARTING FROM SCRATCH. IT IS HARD TO ATTRIBUTE THE SUCCESS OF PYTORCH TO A SINGLE FACTOR. THE PROJECT OFFERS A GOOD USER EXPERIENCE AND ENHANCED DEBUGGABILITY AND FLEXIBILITY, ULTIMATELY MAKING USERS MORE PRODUCTIVE. THE HUGE ADOPTION OF PYTORCH HAS RESULTED IN A BEAUTIFUL ECO. SYSTEM OF SOFTWARE AND RESEARCH BUILT ON TOP OF IT, MAKING PYTORCH EVEN RICHER IN ITS EXPERIENCE. SEVERAL COURSES AND UNIVERSITY CURRICULA, AS WELL AS A HUGE NUMBER OF ONLINE BLOGS AND TUTORIALS, HAVE BEEN OFFERED TO MAKE PYTORCH EASIER TO LEARN. HOWEVER, WE HAVE SEEN VERY FEW BOOKS. IN 2017, WHEN SOMEONE ASKED ME, "WHEN IS THE PYTORCH BOOK GOING TO BE WRITTEN?" I RESPONDED, "IF IT GETS WRITTEN NOW, I CAN GUARANTEE THAT IT WILL BE OUTDATED BY THE TIME IT IS COMPLETED." XV XVI FOREWORD WITH THE PUBLICATION OF DEEP LEARNING WITH PYTORCH, WE FINALLY HAVE A DEFINITIVE TREA- TISE ON PYTORCH. IT COVERS THE BASICS AND ABSTRACTIONS IN GREAT DETAIL, TEARING APART THE UNDERPINNINGS OF DATA STRUCTURES LIKE TENSORS AND NEURAL NETWORKS AND MAKING SURE YOU UNDERSTAND THEIR IMPLEMENTATION. ADDITIONALLY, IT COVERS ADVANCED SUBJECTS SUCH AS JIT AND DEPLOYMENT TO PRODUCTION (AN ASPECT OF PYTORCH THAT NO OTHER BOOK CUR- RENTLY COVERS) . NEURAL NETWORKS TO HELP SOLVE A COMPLEX AND IMPORTANT MEDICAL PROBLEM. WITH LUCA'S DEEP EXPERTISE IN BIOENGINEERING AND MEDICAL IMAGING, ELI'S PRACTICAL EXPERIENCE CRE ATING SOFTWARE FOR MEDICAL DEVICES AND DETECTION, AND THOMAS'S BACKGROUND AS A PYTORCH CORE DEVELOPER, THIS JOURNEY IS TREATED CAREFULLY, AS IT SHOULD BE. ALL IN ALL, I HOPE THIS BOOK BECOMES YOUR "EXTENDED" REFERENCE DOCUMENT AND AN IMPORTANT PART OF YOUR LIBRARY OR WORKSHOP SOUMITH CHINTALA COCREATOR OF PYTORCH ACE AS KIDS IN THE 1980S, TAKING OUR FIRST STEPS ON OUR COMMODORE VIC 20 (ELI), THE SIN- CLAIR SPECTRUM 48K (LUCA), AND THE COMMODORE C16 (THOMAS), WE SAW THE DAWN OF PERSONAL COMPUTERS, LEARNED TO CODE AND WRITE ALGORITHMS ON EVER-FASTER MACHINES, AND OFTEN DREAMED ABOUT WHERE COMPUTERS WOULD TAKE US. WE ALSO WERE PAINFULLY AWARE OF THE GAP BETWEEN WHAT COMPUTERS DID IN MOVIES AND WHAT THEY COULD DO IN REAL LIFE, COLLECTIVELY ROLLING OUR EYES WHEN THE MAIN CHARACTER IN A SPY MOVIE SAID, "COMPUTER, ENHANCE." LATER ON, DURING OUR PROFESSIONAL LIVES, TWO OF US, ELI AND LUCA, INDEPENDENTLY CHALLENGED OURSELVES WITH MEDICAL IMAGE ANALYSIS, FACING THE SAME KIND OF STRUGGLE WHEN WRITING ALGORITHMS THAT COULD HANDLE THE NATURAL VARIABILITY OF THE HUMAN BODY. THERE WAS A LOT OF HEURISTICS INVOLVED WHEN CHOOSING THE BEST MIX OF ALGORITHMS THAT COULD MAKE THINGS WORK AND SAVE THE DAY. THOMAS STUDIED NEURAL NETS AND PATTERN RECOGNITION AT THE TURN OF THE CENTURY BUT WENT ON TO GET A PHD IN MATHEMATICS DOING MODELING. WHEN DEEP LEARNING CAME ABOUT AT THE BEGINNING OF THE 2010S, MAKING ITS INITIAL APPEARANCE IN COMPUTER VISION, IT STARTED BEING APPLIED TO MEDICAL IMAGE ANALYSIS TASKS LIKE THE IDENTIFICATION OF STRUCTURES OR LESIONS ON MEDICAL IMAGES. IT WAS AT THAT TIME, IN THE FIRST HALF OF THE DECADE, THAT DEEP LEARNING APPEARED ON OUR INDIVIDUAL RADARS. IT TOOK A BIT TO REALIZE THAT DEEP LEARNING REPRESENTED A WHOLE NEW WAY OF WRIT ING SOFTWARE: A NEW CLASS OF MULTIPURPOSE ALGORITHMS THAT COULD LEARN HOW TO SOLVE COMPLICATED TASKS THROUGH THE OBSERVATION OF DATA. XVIIDEEP LEARNING WITH PYTORCH ELI STEVENS LUCA ANTIGA THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING UNTRAINED CLOUD MULTIPROCESS MODEL DATA LOADING SAMPLE BATCH TRAINED TENSORS TENSOR PRODUCTION MODEL DATA (ONNX,JIT SOURCE TRAINING TORCHSCRIPT) LOOP DISTRIBUTED TRAINING PRODUCTION ON MULTIPLE SERVERS/GPUS SERVER DEEP LEARNING WITH PYTORCH ELI STEVENS,LUCA ANTIGA AND THOMAS VIEHMANN FOREWORD BY SOUMITH CHINTALA MANNING SHELTER ISLAND FOR ONLINE INFORMATION AND ORDERING OF THIS AND OTHER MANNING BOOKS, PLEASE VISIT WWW.MANNING.COM. THE PUBLISHER OFFERS DISCOUNTS ON THIS BOOK WHEN ORDERED IN QUANTITY FOR MORE INFORMATION, PLEASE CONTACT SPECIAL SALES DEPARTMENT MANNING PUBLICATIONS CO. 20 BALDWIN ROAD PO BOX 761 SHELTER ISLAND, NY 11964 EMAIL: ORDERS@MANNING.COM @2020 BY MANNING PUBLICATIONS CO. ALL RIGHTS RESERVED. NO PART OF THIS PUBLICATION MAY BE REPRODUCED, STORED IN A RETRIEVAL SYSTEM, OR TRANSMITTED, IN ANY FORM OR BY MEANS ELECTRONIC, MECHANICAL, PHOTOCOPYING, OR OTHERWISE, WITHOUT PRIOR WRITTEN PERMISSION OF THE PUBLISHER. MANY OF THE DESIGNATIONS USED BY MANUFACTURERS AND SELLERS TO DISTINGUISH THEIR PRODUCTS ARE CLAIMED AS TRADEMARKS. WHERE THOSE DESIGNATIONS APPEAR IN THE BOOK, AND MANNING PUBLICATIONS WAS AWARE OF A TRADEMARK CLAIM, THE DESIGNATIONS HAVE BEEN PRINTED IN INITIAL CAPS OR ALL CAPS. @ RECOGNIZING THE IMPORTANCE OF PRESERVING WHAT HAS BEEN WRITTEN, IT IS MANNING'S POLICY TO HAVE THE BOOKS WE PUBLISH PRINTED ON ACID-FREE PAPER, AND WE EXERT OUR BEST EFFORTS TO THAT END. RECOGNIZING ALSO OUR RESPONSIBILITY TO CONSERVE THE RESOURCES OF OUR PLANET, MANNING BOOKS ARE PRINTED ON PAPER THAT IS AT LEAST 15 PERCENT RECYCLED AND PROCESSED WITHOUT THE USE OF ELEMENTAL CHLORINE. MANNING PUBLICATIONS CO. DEVELOPMENT EDITOR: FRANCES LEFKOWITZ 20 BALDWIN ROAD TECHNICAL DEVELOPMENT EDITOR: ARTHUR ZUBAREV PO BOX 761 REVIEW EDITOR: IVAN MARTINOVIC SHELTER ISLAND, NY 11964 PRODUCTION EDITOR: DEIRDRE HIAM COPYEDITOR: TIFFANY TAYLOR PROOFREADER: KATIE TENNANT TECHNICAL PROOFREADER: KOSTAS PASSADIS TYPESETTER: GORDAN SALINOVIC COVER DESIGNER: MARIJA TUDOR ISBN 9781617295263 PRINTED IN THE UNITED STATES OF AMERICA TO MY WIFE (THIS BOOK WOULD NOT HAVE HAPPENED WITHOUT HER INVALUABLE SUPPORT AND PARTNERSHIP), MY PARENTS (I WOULD NOT HAVE HAPPENED WITHOUT THEM), AND MY CHILDREN (THIS BOOK WOULD HAVE HAPPENED A LOT SOONER BUT FOR THEM) THANK YOU FOR BEING MY HOME, MY FOUNDATION, AND MY JOY ELI STEVENS SAME :-) BUT, REALLY, THIS IS FOR YOU, ALICE AND LUIGI LUCA ANTIGA TO EVA, REBEKKA, JONATHAN, AND DAVID. THOMAS VIEHMANN CONTENTS FOREWORD XV PREFACE XVII ACKNOWLEDGMENTS XIX ABOUT THIS BOOK XXI ABOUT THE AUTHORS XXVII ABOUT THE COVER ILLUSTRATION XXVIII PART 1 CORE PYTORCH. INTRODUCING DEEP LEARNING AND THE PYTORCH LIBRARY3 1.1 THE DEEP LEARNING REVOLUTION 4 1.2 PYTORCH FOR DEEP LEARNING 6 1.3 WHY PYTORCH?7 THE DEEP LEARNING COMPETITIVE LANDSCAPE8 1.4 AN OVERVIEW OF HOW PYTORCH SUPPORTS DEEP LEARNING PROJECTS 10 1.5 HARDWARE AND SOFTWARE REQUIREMENTS 13 USING JUPYTER NOTEBOOKS 14 1.6 EXERCISES 15 1.7 SUMMARY 15 VI CONTENTS PRETRAINED NETWORKS 16 2.1 A PRETRAINED NETWORK THAT RECOGNIZES THE SUBJECT OF AN IMAGE 17 OBTAINING A PRETRAINED NETWORK FOR IMAGE RECOGNITION 19 ALEXNET 20 - RESNET 22 - READY, SET, ALMOST RUN 22 RUN! 25 2.2 A PRETRAINED MODEL THAT FAKES IT UNTIL IT MAKES IT 27 THE GAN GAME 28 : CYCLEGAN 29 - A NETWORK THAT TURNS HORSES INTO ZEBRAS 30 2.3 A PRETRAINED NETWORK THAT DESCRIBES SCENES 33 NEURALTALK2 34 2.4 TORCH HUB 35 2.5 CONCLUSION 37 2.6 EXERCISES 38 2.7 SUMMARY 38 IT STARTS WITH A TENSOR 39 3.1 THE WORLD AS FLOATING-POINT NUMBERS 40 3.2 TENSORS: MULTIDIMENSIONAL ARRAYS 42 FROM PYTHON LISTS TO PYTORCH TENSORS 42 - CONSTRUCTING OUR FIRST TENSORS 43 - THE ESSENCE OF TENSORS 43 3.3 INDEXING TENSORS 46 3.4 NAMED TENSORS 46 3.5 TENSOR ELEMENT TYPES 50 SPECIFYING THE NUMERIC TYPE WITH DTYPE 50 - A DTYPE FOR EVERY OCCASION 51 - MANAGING A TENSOR'S DTYPE ATTRIBUTE 51 3.6 THE TENSOR API 52 3.7 TENSORS: SCENIC VIEWS OF STORAGE 53 INDEXING INTO STORAGE 54 - MODIFYING STORED VALUES: IN-PLACE OPERATIONS 55 3.8 TENSOR METADATA: SIZE, OFFSET, AND STRIDE 55 VIEWS OF ANOTHER TENSOR'S STORAGE 56 - TRANSPOSING WITHOUT COPYING 58 : TRANSPOSING IN HIGHER DIMENSIONS 60 CONTIGUOUS TENSORS 60 3.9 MOVING TENSORS TO THE GPU 62 MANAGING A TENSOR'S DEVICE ATTRIBUTE 63 VII CONTENTS 3.10 NUMPY INTEROPERABILITY64 3.11 GENERALIZED TENSORS ARE TENSORS, TOO 65 3.12 SERIALIZING TENSORS 66 SERIALIZING TO HDF5 WITH H5PY 67 3.13 CONCLUSION 68 3.14 EXERCISES 68 3.15 SUMMARY 68 REAL-WORLD DATA REPRESENTATION USING TENSORS 70 4.1 WORKING WITH IMAGES 71 ADDING COLOR CHANNELS 72 - LOADING AN IMAGE FILE 72 CHANGING THE LAYOUT 73 : NORMALIZING THE DATA 74 4.2 3D IMAGES: VOLUMETRIC DATA 75 LOADING A SPECIALIZED FORMAT 76 4.3 REPRESENTING TABULAR DATA 77 USING A REAL-WORLD DATASET 77 - LOADING A WINE DATA TENSOR 78 REPRESENTING SCORES 81 : ONE-HOT ENCODING 81 : WHEN TO CATEGORIZE 83 - FINDING THRESHOLDS 84 4.4 WORKING WITH TIME SERIES 87 ADDING A TIME DIMENSION 88 - SHAPING THE DATA BY TIME PERIOD 89 - READY FOR TRAINING 90 4.5 REPRESENTING TEXT 93 CONVERTING TEXT TO NUMBERS94 - ONE-HOT-ENCODING CHARACTERS94 ONE-HOT ENCODING WHOLE WORDS 96 : TEXT EMBEDDINGS 98 TEXT EMBEDDINGS AS A BLUEPRINT 100 4.6 CONCLUSION 101 4.7 EXERCISES 101 4.8 SUMMARY 102 THE MECHANICS OF LEARNING 103 5.1 A TIMELESS LESSON IN MODELING 104 5.2 LEARNING IS JUST PARAMETER ESTIMATION 106 A HOT PROBLEM 107 : GATHERING SOME DATA 107 : VISUALIZING THE DATA 108 : CHOOSING A LINEAR MODEL AS A FIRST TRY 108 5.3 LESS LOSS IS WHAT WE WANT 109 FROM PROBLEM BACK TO PYTORCH 110 VIII CONTENTS 5.4 DOWN ALONG THE GRADIENT 113 DECREASING LOSS 113 - GETTING ANALYTICAL 114 - ITERATING TO FIT THE MODEL 116 - NORMALIZING INPUTS 119 : VISUALIZING (AGAIN) 122 5.5 PYTORCH'S AUTOGRAD: BACKPROPAGATING ALL THINGS 123 COMPUTING THE GRADIENT AUTOMATICALLY 123 - OPTIMIZERS A LA CARTE 127 : TRAINING, VALIDATION, AND OVERFITTING 131 AUTOGRAD NITS AND SWITCHING IT OFF. 137 5.6 CONCLUSION 139 5.7 EXERCISE 139 5.8 SUMMARY 139 USING A NEURAL NETWORK TO FIT THE DATA 141 6.1 ARTIFICIAL NEURONS142 COMPOSING A MULTILAYER NETWORK 144 - UNDERSTANDING THE ERROR FUNCTION 144 : ALL WE NEED IS ACTIVATION 145 : MORE ACTIVATION FUNCTIONS 147 : CHOOSING THE BEST ACTIVATION FUNCTION 148 WHAT LEARNING MEANS FOR A NEURAL NETWORK 149 6.2 THE PYTORCH NN MODULE 151 USING--CALL- RATHER THAN FORWARD 152 - RETURNING TO THE LINEAR MODEL 153 6.3 FINALLY A NEURAL NETWORK 158 REPLACING THE LINEAR MODEL 158 - INSPECTING THE PARAMETERS 159 COMPARING TO THE LINEAR MODEL 161 6.4 CONCLUSION 162 6.5 EXERCISES 162 6.6 SUMMARY 163 TELLING BIRDS FROM AIRPLANES: LEARNING FROM IMAGES 164 7.1 A DATASET OF TINY IMAGES 165 DOWNLOADING CIFAR-10 166 : THE DATASET CLASS 166 DATASET TRANSFORMS 168 - NORMALIZING DATA 170 7.2 DISTINGUISHING BIRDS FROM AIRPLANES 172 BUILDING THE DATASET 173 - A FULLY CONNECTED MODEL 174 OUTPUT OF A CLASSIFIER 175 : REPRESENTING THE OUTPUT AS PROBABILITIES 176 - A LOSS FOR CLASSIFYING 180 : TRAINING THE CLASSIFIER 182 - THE LIMITS OF GOING FULLY CONNECTED 189 7.3 CONCLUSION 191 IX CONTENTS 7.4 EXERCISES 191 7.5 SUMMARY 192 USING CONVOLUTIONS TO GENERALIZE 193 8.1 THE CASE FOR CONVOLUTIONS 194 WHAT CONVOLUTIONS DO 194 8.2 CONVOLUTIONS IN ACTION 196 PADDING THE BOUNDARY 198 - DETECTING FEATURES WITH CONVOLUTIONS 200 - LOOKING FURTHER WITH DEPTH AND POOLING 202 PUTTING IT ALL TOGETHER FOR OUR NETWORK 205 8.3 SUBCLASSING NN.MODULE 207 OUR NETWORK AS AN NN.MODULE 208 - HOW PYTORCH KEEPS TRACK OF PARAMETERS AND SUBMODULES 209 : THE FUNCTIONAL API 210 8.4 TRAINING OUR CONVNET 212 MEASURING ACCURACY 214 : SAVING AND LOADING OUR MODEL 214 TRAINING ON THE GPU 215 8.5 MODEL DESIGN 217 ADDING MEMORY CAPACITY: WIDTH 218 - HELPING OUR MODEL TO CONVERGE AND GENERALIZE: REGULARIZATION 219 - GOING DEEPER TO LEARN MORE COMPLEX STRUCTURES: DEPTH223 - COMPARING THE DESIGNS FROM THIS SECTION 228 - IT'S ALREADY OUTDATED 229 8.6 CONCLUSION 229 8.7 EXERCISES 230 8.8 SUMMARY 231 PART 2 LEARNING FROM IMAGES IN THE REAL WORLD: EARLY DETECTION OF LUNG CANCER. 233 USING PYTORCH TO FIGHT CANCER 235 9.1 INTRODUCTION TO THE USE CASE 236 9.2 PREPARING FOR A LARGE-SCALE PROJECT237 9.3 WHAT IS A CT SCAN, EXACTLY?238 9.4 THE PROJECT: AN END-TO-END DETECTOR FOR LUNG CANCER 241 WHY CAN'T WE JUST THROW DATA AT A NEURAL NETWORK UNTIL IT WORKS? 245 - WHAT IS A NODULE? 249 - OUR DATA SOURCE: THE LUNA GRAND CHALLENGE 251 - DOWNLOADING THE LUNA DATA 251 CONTENTS X 9.5 CONCLUSION 252 9.6 SUMMARY 253 COMBINING DATA SOURCES INTO A UNIFIED DATASET 254 1 10.1 RAW CT DATA FILES 256 10.2 PARSING LUNA'S ANNOTATION DATA 256 TRAINING AND VALIDATION SETS 258 : UNIFYING OUR ANNOTATION AND CANDIDATE DATA 259 10.3 LOADING INDIVIDUAL CT SCANS 262 HOUNSFIELD UNITS 264 10.4 LOCATING A NODULE USING THE PATIENT COORDINATE SYSTEM 265 THE PATIENT COORDINATE SYSTEM 265 - CT SCAN SHAPE AND VOXEL SIZES 267 - CONVERTING BETWEEN MILLIMETERS AND VOXEL ADDRESSES 268 - EXTRACTING A NODULE FROM A CT SCAN 270 10.5 A STRAIGHTFORWARD DATASET IMPLEMENTATION 271 CACHING CANDIDATE ARRAYS WITH THE GETCTRAWCANDIDATE FUNCTION 274 : CONSTRUCTING OUR DATASET IN LUNADATASET \_INIT\_- 275 - A TRAINING/VALIDATION SPLIT 275 - RENDERING THE DATA 277 10.6 CONCLUSION 277 10.7 EXERCISES 278 10.8 SUMMARY 278 TRAINING A CLASSIFICATION MODEL TO DETECT SUSPECTED TUMORS 279 11.1 A FOUNDATIONAL MODEL AND TRAINING LOOP 280 11.2 THE MAIN ENTRY POINT FOR OUR APPLICATION 282 11.3 PRETRAINING SETUP AND INITIALIZATION 284 INITIALIZING THE MODEL AND OPTIMIZER 285 : CARE AND FEEDING OF DATA LOADERS 287 11.4 OUR FIRST-PASS NEURAL NETWORK DESIGN 289 THE CORE CONVOLUTIONS 290 : THE FULL MODEL 293 11.5 TRAINING AND VALIDATING THE MODEL 295 THE COMPUTEBATCHLOSS FUNCTION 297 - THE VALIDATION LOOP IS SIMILAR 299 11.6 OUTPUTTING PERFORMANCE METRICS 300 THE LOGMETRICS FUNCTION 301 XI CONTENTS 11.7 RUNNING THE TRAINING SCRIPT 304 NEEDED DATA FOR TRAINING 305 - INTERLUDE: THE ENUMERATEWITHESTIMATE FUNCTION 306 11.8 EVALUATING THE MODEL: GETTING 99.7% CORRECT MEANS WE'RE DONE, RIGHT? 308 11.9 GRAPHING TRAINING METRICS WITH TENSORBOARD 309 RUNNING TENSORBOARD 309 - ADDING TENSORBOARD SUPPORT TO THE METRICS LOGGING FUNCTION 313 11.10 WHY ISN'T THE MODEL LEARNING TO DETECT NODULES? 315 11.11 CONCLUSION 316 11.12 EXERCISES 316 11.13 SUMMARY 316 12 IMPROVING TRAINING WITH METRICS AND AUGMENTATION 318 12.1 HIGH-LEVEL PLAN FOR IMPROVEMENT 319 12.2 GOOD DOGS VS. BAD GUYS: FALSE POSITIVES AND FALSE NEGATIVES 320 12.3 GRAPHING THE POSITIVES AND NEGATIVES322 RECALL IS ROXIE'S STRENGTH 324 - PRECISION IS PRESTON'S FORTE 326 IMPLEMENTING PRECISION AND RECALL IN LOGMETRICS 327 : OUR ULTIMATE PERFORMANCE METRIC: THE F1 SCORE 328 - HOW DOES OUR MODEL PERFORM WITH OUR NEW METRICS? 332 12.4 WHAT DOES AN IDEAL DATASET LOOK LIKE? 334 MAKING THE DATA LOOK LESS LIKE THE ACTUAL AND MORE LIKE THE "IDEAL" 336 CONTRASTING TRAINING WITH A BALANCED LUNADATASET TO PREVIOUS RUNS 341 - RECOGNIZING THE SYMPTOMS OF OVERFITTING 343 12.5 REVISITING THE PROBLEM OF OVERFITTING 345 AN OVERFIT FACE-TO-AGE PREDICTION MODEL 345 12.6 PREVENTING OVERFITTING WITH DATA AUGMENTATION 346 SPECIFIC DATA AUGMENTATION TECHNIQUES 347 : SEEING THE IMPROVEMENT FROM DATA AUGMENTATION 352 12.7 CONCLUSION 354 12.8 EXERCISES 355 12.9 SUMMARY 356 USING SEGMENTATION TO FIND SUSPECTED NODULES 357 13.1 ADDING A SECOND MODEL TO OUR PROJECT 358 13.2 VARIOUS TYPES OF SEGMENTATION 360 XII CONTENTS 13.3 SEMANTIC SEGMENTATION: PER-PIXEL CLASSIFICATION 361 THE U-NET ARCHITECTURE 364 13.4 UPDATING THE MODEL FOR SEGMENTATION 366 ADAPTING AN OFF-THE-SHELF MODEL TO OUR PROJECT 367 13.5 UPDATING THE DATASET FOR SEGMENTATION 369 U-NET HAS VERY SPECIFIC INPUT SIZE REQUIREMENTS 370 - U-NET TRADE YINU PUNOU8 OYP BUIPING - OLEDYPP AZ 'SA AE IOF SFFO DATA 371 : IMPLEMENTING LUNA2DSEGMENTATIONDATASET 378 DESIGNING OUR TRAINING AND VALIDATION DATA 382 - IMPLEMENTING TRAININGLUNA2DSEGMENTATIONDATASET 383 - AUGMENTING ON THE GPU 384 13.6 UPDATING THE TRAINING SCRIPT FOR SEGMENTATION 386 INITIALIZING OUR SEGMENTATION AND AUGMENTATION MODELS 387 USING THE ADAM OPTIMIZER 388 - DICE LOSS 389 - GETTING IMAGES INTO TENSORBOARD 392 : UPDATING OUR METRICS LOGGING 396 SAVING OUR MODEL 397 13.7 RESULTS 399 13.8 CONCLUSION 401 13.9 EXERCISES 402 13.10 SUMMARY 402 END-TO-END NODULE ANALYSIS, AND WHERE TO GO NEXT 404 14.1 TOWARDS THE FINISH LINE 405 14.2 INDEPENDENCE OF THE VALIDATION SET 407 14.3 BRIDGING CT SEGMENTATION AND NODULE CANDIDATE CLASSIFICATION 408 SEGMENTATION 410 - GROUPING VOXELS INTO NODULE CANDIDATES 411 DID WE FIND A NODULE? CLASSIFICATION TO REDUCE FALSE POSITIVES 412 14.4 QUANTITATIVE VALIDATION 416 14.5 PREDICTING MALIGNANCY 417 GETTING MALIGNANCY INFORMATION 417 - AN AREA UNDER THE CURVE BASELINE: CLASSIFYING BY DIAMETER 419 - REUSING PREEXISTING WEIGHTS: FINE-TUNING 422 - MORE OUTPUT IN TENSORBOARD 428 14.6 WHAT WE SEE WHEN WE DIAGNOSE 432 TRAINING, VALIDATION, AND TEST SETS 433 14.7 WHAT NEXT? ADDITIONAL SOURCES OF INSPIRATION (AND DATA) 434 PREVENTING OVERFTTING: BETTER REGULARIZATION 434 - REFINED TRAINING DATA 437 : COMPETITION RESULTS AND RESEARCH PAPERS 438 XIII CONTENTS 14.8 CONCLUSION 439 BEHIND THE CURTAIN 439 14.9 EXERCISES 441 14.10 SUMMARY 441 PART 3 DEPLOYMENT : 443 DEPLOYING TO PRODUCTION445 15.1 SERVING PYTORCH MODELS 446 OUR MODEL BEHIND A FLASK SERVER 446 - WHAT WE WANT FROM DEPLOYMENT 448 : REQUEST BATCHING 449 15.2 EXPORTING MODELS 455 INTEROPERABILITY BEYOND PYTORCH WITH ONNX 455 - PYTORCH'S OWN EXPORT: TRACING 456 - OUR SERVER WITH A TRACED MODEL 458 15.3 INTERACTING WITH THE PYTORCH JIT 458 WHAT TO EXPECT FROM MOVING BEYOND CLASSIC PYTHON/PYTORCH 458 THE DUAL NATURE OF PYTORCH AS INTERFACE AND BACKEND 460 TORCHSCRIPT 461 - SCRIPTING THE GAPS OF TRACEABILITY 464 15.4 LIBTORCH: PYTORCH IN C++ 465 RUNNING JITED MODELS FROM C++ 465 : C++ FROM THE START: THE C++ API 468 15.5 GOING MOBILE 472 IMPROVING EFFICIENCY: MODEL DESIGN AND QUANTIZATION 475 15.6 EMERGING TECHNOLOGY: ENTERPRISE SERVING OF PYTORCH MODELS 476 15.7 CONCLUSION 477 15.8 EXERCISES 477 15.9 SUMMARY 477 INDEX 479 FOREWORD WHEN WE STARTED THE PYTORCH PROJECT IN MID-2016, WE WERE A BAND OF OPEN SOURCE HACKERS WHO MET ONLINE AND WANTED TO WRITE BETTER DEEP LEARNING SOFTWARE. TWO OF THE THREE AUTHORS OF THIS BOOK, LUCA ANTIGA AND THOMAS VIEHMANN, WERE INSTRUMENTAL IN DEVELOPING PYTORCH AND MAKING IT THE SUCCESS THAT IT IS TODAY. OUR GOAL WITH PYTORCH WAS TO BUILD THE MOST FLEXIBLE FRAMEWORK POSSIBLE TO EXPRESS DEEP LEARNING ALGORITHMS. WE EXECUTED WITH FOCUS AND HAD A RELATIVELY SHORT DEVELOP MENT TIME TO BUILD A POLISHED PRODUCT FOR THE DEVELOPER MARKET. THIS WOULDN'T HAVE BEEN POSSIBLE IF WE HADN'T BEEN STANDING ON THE SHOULDERS OF GIANTS. PYTORCH DERIVES A SIGNIFICANT PART OF ITS CODEBASE FROM THE TORCH7 PROJECT STARTED IN 2007 BY RONAN COL- LOBERT AND OTHERS, WHICH HAS ROOTS IN THE LUSH PROGRAMMING LANGUAGE PIONEERED BY YANN LECUN AND LEON BOTTOU. THIS RICH HISTORY HELPED US FOCUS ON WHAT NEEDED TO CHANGE, RATHER THAN CONCEPTUALLY STARTING FROM SCRATCH. IT IS HARD TO ATTRIBUTE THE SUCCESS OF PYTORCH TO A SINGLE FACTOR. THE PROJECT OFFERS A GOOD USER EXPERIENCE AND ENHANCED DEBUGGABILITY AND FLEXIBILITY, ULTIMATELY MAKING USERS MORE PRODUCTIVE. THE HUGE ADOPTION OF PYTORCH HAS RESULTED IN A BEAUTIFUL ECO. SYSTEM OF SOFTWARE AND RESEARCH BUILT ON TOP OF IT, MAKING PYTORCH EVEN RICHER IN ITS EXPERIENCE. SEVERAL COURSES AND UNIVERSITY CURRICULA, AS WELL AS A HUGE NUMBER OF ONLINE BLOGS AND TUTORIALS, HAVE BEEN OFFERED TO MAKE PYTORCH EASIER TO LEARN. HOWEVER, WE HAVE SEEN VERY FEW BOOKS. IN 2017, WHEN SOMEONE ASKED ME, "WHEN IS THE PYTORCH BOOK GOING TO BE WRITTEN?" I RESPONDED, "IF IT GETS WRITTEN NOW, I CAN GUARANTEE THAT IT WILL BE OUTDATED BY THE TIME IT IS COMPLETED." XV XVI FOREWORD WITH THE PUBLICATION OF DEEP LEARNING WITH PYTORCH, WE FINALLY HAVE A DEFINITIVE TREA- TISE ON PYTORCH. IT COVERS THE BASICS AND ABSTRACTIONS IN GREAT DETAIL, TEARING APART THE UNDERPINNINGS OF DATA STRUCTURES LIKE TENSORS AND NEURAL NETWORKS AND MAKING SURE YOU UNDERSTAND THEIR IMPLEMENTATION. ADDITIONALLY, IT COVERS ADVANCED SUBJECTS SUCH AS JIT AND DEPLOYMENT TO PRODUCTION (AN ASPECT OF PYTORCH THAT NO OTHER BOOK CUR- RENTLY COVERS) . NEURAL NETWORKS TO HELP SOLVE A COMPLEX AND IMPORTANT MEDICAL PROBLEM. WITH LUCA'S DEEP EXPERTISE IN BIOENGINEERING AND MEDICAL IMAGING, ELI'S PRACTICAL EXPERIENCE CRE ATING SOFTWARE FOR MEDICAL DEVICES AND DETECTION, AND THOMAS'S BACKGROUND AS A PYTORCH CORE DEVELOPER, THIS JOURNEY IS TREATED CAREFULLY, AS IT SHOULD BE. ALL IN ALL, I HOPE THIS BOOK BECOMES YOUR "EXTENDED" REFERENCE DOCUMENT AND AN IMPORTANT PART OF YOUR LIBRARY OR WORKSHOP SOUMITH CHINTALA COCREATOR OF PYTORCH ACE AS KIDS IN THE 1980S, TAKING OUR FIRST STEPS ON OUR COMMODORE VIC 20 (ELI), THE SIN- CLAIR SPECTRUM 48K (LUCA), AND THE COMMODORE C16 (THOMAS), WE SAW THE DAWN OF PERSONAL COMPUTERS, LEARNED TO CODE AND WRITE ALGORITHMS ON EVER-FASTER MACHINES, AND OFTEN DREAMED ABOUT WHERE COMPUTERS WOULD TAKE US. WE ALSO WERE PAINFULLY AWARE OF THE GAP BETWEEN WHAT COMPUTERS DID IN MOVIES AND WHAT THEY COULD DO IN REAL LIFE, COLLECTIVELY ROLLING OUR EYES WHEN THE MAIN CHARACTER IN A SPY MOVIE SAID, "COMPUTER, ENHANCE." LATER ON, DURING OUR PROFESSIONAL LIVES, TWO OF US, ELI AND LUCA, INDEPENDENTLY CHALLENGED OURSELVES WITH MEDICAL IMAGE ANALYSIS, FACING THE SAME KIND OF STRUGGLE WHEN WRITING ALGORITHMS THAT COULD HANDLE THE NATURAL VARIABILITY OF THE HUMAN BODY. THERE WAS A LOT OF HEURISTICS INVOLVED WHEN CHOOSING THE BEST MIX OF ALGORITHMS THAT COULD MAKE THINGS WORK AND SAVE THE DAY. THOMAS STUDIED NEURAL NETS AND PATTERN RECOGNITION AT THE TURN OF THE CENTURY BUT WENT ON TO GET A PHD IN MATHEMATICS DOING MODELING. WHEN DEEP LEARNING CAME ABOUT AT THE BEGINNING OF THE 2010S, MAKING ITS INITIAL APPEARANCE IN COMPUTER VISION, IT STARTED BEING APPLIED TO MEDICAL IMAGE ANALYSIS TASKS LIKE THE IDENTIFICATION OF STRUCTURES OR LESIONS ON MEDICAL IMAGES. IT WAS AT THAT TIME, IN THE FIRST HALF OF THE DECADE, THAT DEEP LEARNING APPEARED ON OUR INDIVIDUAL RADARS. IT TOOK A BIT TO REALIZE THAT DEEP LEARNING REPRESENTED A WHOLE NEW WAY OF WRIT ING SOFTWARE: A NEW CLASS OF MULTIPURPOSE ALGORITHMS THAT COULD LEARN HOW TO SOLVE COMPLICATED TASKS THROUGH THE OBSERVATION OF DATA. XVII XVIII PREFACE TO OUR KIDS-OF-THE-80S MINDS, THE HORIZON OF WHAT COMPUTERS COULD DO EXPANDED OVERNIGHT, LIMITED NOT BY THE BRAINS OF THE BEST PROGRAMMERS, BUT BY THE DATA, THE NEU- RAL NETWORK ARCHITECTURE, AND THE TRAINING PROCESS. THE NEXT STEP WAS GETTING OUR HANDS DIRTY. LUCA CHOOSE TORCH 7 (HTTP://TORCH.CH), A VENERABLE PRECURSOR TO PYTORCH; IT'S NIMBLE, LIGHTWEIGHT, AND FAST, WITH APPROACHABLE SOURCE CODE WRITTEN IN LUA AND PLAIN C, A SUPPORTIVE COMMUNITY, AND A LONG HISTORY BEHIND IT. FOR LUCA, IT WAS LOVE AT FIRST SIGHT. THE ONLY REAL DRAWBACK WITH TORCH 7 WAS BEING DETACHED FROM THE EVER-GROWING PYTHON DATA SCIENCE ECOSYSTEM THAT THE OTHER FRAMEWORKS COULD DRAW FROM. ELI HAD BEEN INTERESTED IN AI SINCE COLLEGE,' BUT HIS CAREER POINTED HIM IN OTHER DIRECTIONS, AND HE FOUND OTHER, EARLIER DEEP LEARNING FRAMEWORKS A BIT TOO LABORIOUS TO GET ENTHUSIASTIC ABOUT USING THEM FOR A HOBBY PROJECT. SO WE ALL GOT REALLY EXCITED WHEN THE FIRST PYTORCH RELEASE WAS MADE PUBLIC ON JAN- UARY 18, 2017. LUCA STARTED CONTRIBUTING TO THE CORE, AND ELI WAS PART OF THE COMMU- NITY VERY EARLY ON, SUBMITTING THE ODD BUG FIX, FEATURE, OR DOCUMENTATION UPDATE. THOMAS CONTRIBUTED A TON OF FEATURES AND BUG FIXES TO PYTORCH AND EVENTUALLY BECAME ONE OF THE INDEPENDENT CORE CONTRIBUTORS. THERE WAS THE FEELING THAT SOMETHING BIG WAS STARTING UP, AT THE RIGHT LEVEL OF COMPLEXITY AND WITH A MINIMAL AMOUNT OF COGNI- TIVE OVERHEAD. THE LEAN DESIGN LESSONS LEARNED FROM THE TORCH 7 DAYS WERE BEING CAR- RIED OVER, BUT THIS TIME WITH A MODERN SET OF FEATURES LIKE AUTOMATIC DIFFERENTIATION, DYNAMIC COMPUTATION GRAPHS, AND NUMPY INTEGRATION. GIVEN OUR INVOLVEMENT AND ENTHUSIASM, AND AFTER ORGANIZING A COUPLE OF PYTORCH WORKSHOPS, WRITING A BOOK FELT LIKE A NATURAL NEXT STEP. THE GOAL WAS TO WRITE A BOOK THAT WOULD HAVE BEEN APPEALING TO OUR FORMER SELVES GETTING STARTED JUST A FEW YEARS BACK. PREDICTABLY, WE STARTED WITH GRANDIOSE IDEAS: TEACH THE BASICS, WALK THROUGH END-TO- END PROJECTS, AND DEMONSTRATE THE LATEST AND GREATEST MODELS IN PYTORCH. WE SOON REALIZED THAT WOULD TAKE A LOT MORE THAN A SINGLE BOOK, SO WE DECIDED TO FOCUS ON OUR INITIAL MISSION: DEVOTE TIME AND DEPTH TO COVER THE KEY CONCEPTS UNDERLYING PYTORCH, ASSUMING LITTLE OR NO PRIOR KNOWLEDGE OF DEEP LEARNING, AND GET TO THE POINT WHERE WE COULD WALK OUR READERS THROUGH A COMPLETE PROJECT. FOR THE LATTER, WE WENT BACK TO OUR ROOTS AND CHOSE TO DEMONSTRATE A MEDICAL IMAGE ANALYSIS CHALLENGE. I BACK WHEN "DEEP" NEURAL NETWORKS MEANT THREE HIDDEN LAYERS!