contents

preface *xiii*
acknowledgments *xv*
about this book *xvi*
about the author *xix*
about the cover illustration *xx*

PART 1 DEEP LEARNING FOUNDATION 1

1 ***Welcome to computer vision*** ***3***

1.1 Computer vision 4

What is visual perception? *5* ▪ *Vision systems* *5*
Sensing devices *7* ▪ *Interpreting devices* *8*

1.2 Applications of computer vision 10

Image classification *10* ▪ *Object detection and localization* *12*
Generating art (style transfer) *12* ▪ *Creating images* *13*
Face recognition *15* ▪ *Image recommendation system* *15*

1.3 Computer vision pipeline: The big picture 17

1.4 Image input 19

Image as functions *19* ▪ *How computers see images* *21*
Color images *21*

1.5 Image preprocessing 23

Converting color images to grayscale to reduce computation complexity 23

1.6 Feature extraction 27

What is a feature in computer vision? 27 ▪ What makes a good (useful) feature? 28 ▪ Extracting features (handcrafted vs. automatic extracting) 31

1.7 Classifier learning algorithm 33

2 *Deep learning and neural networks 36*

2.1 Understanding perceptrons 37

What is a perceptron? 38 ▪ How does the perceptron learn? 43 Is one neuron enough to solve complex problems? 43

2.2 Multilayer perceptrons 45

Multilayer perceptron architecture 46 ▪ What are hidden layers? 47 ▪ How many layers, and how many nodes in each layer? 47 ▪ Some takeaways from this section 50

2.3 Activation functions 51

Linear transfer function 53 ▪ Heaviside step function (binary classifier) 54 ▪ Sigmoid/logistic function 55 ▪ Softmax function 57 ▪ Hyperbolic tangent function (tanh) 58 Rectified linear unit 58 ▪ Leaky ReLU 59

2.4 The feedforward process 62

Feedforward calculations 64 ▪ Feature learning 65

2.5 Error functions 68

What is the error function? 69 ▪ Why do we need an error function? 69 ▪ Error is always positive 69 ▪ Mean square error 70 ▪ Cross-entropy 71 ▪ A final note on errors and weights 72

2.6 Optimization algorithms 74

What is optimization? 74 ▪ Batch gradient descent 77 Stochastic gradient descent 83 ▪ Mini-batch gradient descent 84 Gradient descent takeaways 85

2.7 Backpropagation 86

What is backpropagation? 87 ▪ Backpropagation takeaways 90

3 *Convolutional neural networks 92*

3.1 Image classification using MLP 93

Input layer 94 ▪ Hidden layers 96 ▪ Output layer 96 Putting it all together 97 ▪ Drawbacks of MLPs for processing images 99

3.2 CNN architecture 102

The big picture 102 ▪ A closer look at feature extraction 104 A closer look at classification 105

3.3 Basic components of a CNN 106

Convolutional layers 107 ▪ Pooling layers or subsampling 114 Fully connected layers 119

3.4 Image classification using CNNs 121

Building the model architecture 121 ▪ Number of parameters (weights) 123

3.5 Adding dropout layers to avoid overfitting 124

What is overfitting? 125 ▪ What is a dropout layer? 125 Why do we need dropout layers? 126 ▪ Where does the dropout layer go in the CNN architecture? 127

3.6 Convolution over color images (3D images) 128

How do we perform a convolution on a color image? 129 What happens to the computational complexity? 130

3.7 Project: Image classification for color images 133

4 *Structuring DL projects and hyperparameter tuning 145*

4.1 Defining performance metrics 146

Is accuracy the best metric for evaluating a model? 147 Confusion matrix 147 ▪ Precision and recall 148 F-score 149

4.2 Designing a baseline model 149

4.3 Getting your data ready for training 151

Splitting your data for train/validation/test 151 Data preprocessing 153

4.4 Evaluating the model and interpreting its performance 156

Diagnosing overfitting and underfitting 156 ▪ Plotting the learning curves 158 ▪ Exercise: Building, training, and evaluating a network 159

4.5 Improving the network and tuning hyperparameters 162

Collecting more data vs. tuning hyperparameters 162 ▪ Parameters vs. hyperparameters 163 ▪ Neural network hyperparameters 163 ▪ Network architecture 164

4.6 Learning and optimization 166

Learning rate and decay schedule 166 ▪ A systematic approach to find the optimal learning rate 169 ▪ Learning rate decay and adaptive learning 170 ▪ Mini-batch size 171

4.7 Optimization algorithms 174

Gradient descent with momentum 174 ▪ Adam 175 ▪ Number of epochs and early stopping criteria 175 ▪ Early stopping 177

4.8 Regularization techniques to avoid overfitting 177

L2 regularization 177 ▪ Dropout layers 179 ▪ Data augmentation 180

4.9 Batch normalization 181

The covariate shift problem 181 ▪ Covariate shift in neural networks 182 ▪ How does batch normalization work? 183 ▪ Batch normalization implementation in Keras 184 ▪ Batch normalization recap 185

4.10 Project: Achieve high accuracy on image classification 185

PART 2 IMAGE CLASSIFICATION AND DETECTION..........193

5 *Advanced CNN architectures 195*

5.1 CNN design patterns 197

5.2 LeNet-5 199

LeNet architecture 199 ▪ LeNet-5 implementation in Keras 200 ▪ Setting up the learning hyperparameters 202 ▪ LeNet performance on the MNIST dataset 203

5.3 AlexNet 203

AlexNet architecture 205 ▪ Novel features of AlexNet 205 ▪ AlexNet implementation in Keras 207 ▪ Setting up the learning hyperparameters 210 ▪ AlexNet performance 211

5.4 VGGNet 212

Novel features of VGGNet 212 ▪ VGGNet configurations 213 ▪ Learning hyperparameters 216 ▪ VGGNet performance 216

5.5 Inception and GoogLeNet 217

Novel features of Inception 217 ▪ Inception module: Naive version 218 ▪ Inception module with dimensionality reduction 220 ▪ Inception architecture 223 ▪ GoogLeNet in Keras 225 ▪ Learning hyperparameters 229 ▪ Inception performance on the CIFAR dataset 229

5.6 ResNet 230

Novel features of ResNet 230 ▪ Residual blocks 233 ▪ ResNet implementation in Keras 235 ▪ Learning hyperparameters 238 ResNet performance on the CIFAR dataset 238

6 ***Transfer learning 240***

6.1 What problems does transfer learning solve? 241

6.2 What is transfer learning? 243

6.3 How transfer learning works 250

How do neural networks learn features? 252 ▪ Transferability of features extracted at later layers 254

6.4 Transfer learning approaches 254

Using a pretrained network as a classifier 254 ▪ Using a pretrained network as a feature extractor 256 ▪ Fine-tuning 258

6.5 Choosing the appropriate level of transfer learning 260

Scenario 1: Target dataset is small and similar to the source dataset 260 ▪ Scenario 2: Target dataset is large and similar to the source dataset 261 ▪ Scenario 3: Target dataset is small and different from the source dataset 261 ▪ Scenario 4: Target dataset is large and different from the source dataset 261 ▪ Recap of the transfer learning scenarios 262

6.6 Open source datasets 262

MNIST 263 ▪ Fashion-MNIST 264 ▪ CIFAR 264 ImageNet 265 ▪ MS COCO 266 ▪ Google Open Images 267 Kaggle 267

6.7 Project 1: A pretrained network as a feature extractor 268

6.8 Project 2: Fine-tuning 274

7 ***Object detection with R-CNN, SSD, and YOLO 283***

7.1 General object detection framework 285

Region proposals 286 ▪ Network predictions 287 Non-maximum suppression (NMS) 288 ▪ Object-detector evaluation metrics 289

7.2 Region-based convolutional neural networks (R-CNNs) 292

R-CNN 293 ▪ Fast R-CNN 297 ▪ Faster R-CNN 300 Recap of the R-CNN family 308

7.3 Single-shot detector (SSD) 310

High-level SSD architecture 311 ▪ Base network 313 Multi-scale feature layers 315 ▪ Non-maximum suppression 319

7.4 You only look once (YOLO) 320

How YOLOv3 works 321 ▪ YOLOv3 architecture 324

7.5 Project: Train an SSD network in a self-driving car application 326

Step 1: Build the model 328 ▪ Step 2: Model configuration 329 Step 3: Create the model 330 ▪ Step 4: Load the data 331 Step 5: Train the model 333 ▪ Step 6: Visualize the loss 334 Step 7: Make predictions 335

PART 3 GENERATIVE MODELS AND VISUAL EMBEDDINGS... 339

8 *Generative adversarial networks (GANs) 341*

8.1 GAN architecture 343

Deep convolutional GANs (DCGANs) 345 ▪ The discriminator model 345 ▪ The generator model 348 ▪ Training the GAN 351 ▪ GAN minimax function 354

8.2 Evaluating GAN models 357

Inception score 358 ▪ Fréchet inception distance (FID) 358 Which evaluation scheme to use 358

8.3 Popular GAN applications 359

Text-to-photo synthesis 359 ▪ Image-to-image translation (Pix2Pix GAN) 360 ▪ Image super-resolution GAN (SRGAN) 361 Ready to get your hands dirty? 362

8.4 Project: Building your own GAN 362

9 *DeepDream and neural style transfer 374*

9.1 How convolutional neural networks see the world 375

Revisiting how neural networks work 376 ▪ Visualizing CNN features 377 ▪ Implementing a feature visualizer 381

9.2 DeepDream 384

How the DeepDream algorithm works 385 ▪ DeepDream implementation in Keras 387

9.3 Neural style transfer 392

Content loss 393 ▪ Style loss 396 ▪ Total variance loss 397 Network training 397

10 ***Visual embeddings 400***

10.1 Applications of visual embeddings 402

Face recognition 402 ▪ Image recommendation systems 403 Object re-identification 405

10.2 Learning embedding 406

10.3 Loss functions 407

Problem setup and formalization 408 ▪ Cross-entropy loss 409 Contrastive loss 410 ▪ Triplet loss 411 ▪ Naive implementation and runtime analysis of losses 412

10.4 Mining informative data 414

Dataloader 414 ▪ Informative data mining: Finding useful triplets 416 ▪ Batch all (BA) 419 ▪ Batch hard (BH) 419 Batch weighted (BW) 421 ▪ Batch sample (BS) 421

10.5 Project: Train an embedding network 423

Fashion: Get me items similar to this 424 ▪ Vehicle re-identification 424 ▪ Implementation 426 ▪ Testing a trained model 427

10.6 Pushing the boundaries of current accuracy 431

appendix A Getting set up 437

index 445