Contents

1 Introduction 15

1.1 I Studied Deep Learning the *Wrong Way*... This Is the *Right Way* **15**

1.2 Who This Book Is For **17**

1.2.1 Just Getting Started in Deep Learning? 17

1.2.2 Already a Seasoned Deep Learning Practitioner? 17

1.3 Book Organization **17**

1.3.1 Volume #1: Starter Bundle 17

1.3.2 Volume #2: Practitioner Bundle 18

1.3.3 Volume #3: ImageNet Bundle 18

1.3.4 Need to Upgrade Your Bundle? 18

1.4 Tools of the Trade: Python, Keras, and Mxnet **18**

1.4.1 What About TensorFlow? 18

1.4.2 Do I Need to Know OpenCV? 19

1.5 Developing Our Own Deep Learning Toolset **19**

1.6 Summary **20**

2 What Is Deep Learning? 21

2.1 A Concise History of Neural Networks and Deep Learning **22**

2.2 Hierarchical Feature Learning **24**

2.3 How "Deep" Is Deep? **27**

2.4 Summary **30**

3 Image Fundamentals 31

3.1 Pixels: The Building Blocks of Images **31**

3.1.1 Forming an Image From Channels 34

3.2 The Image Coordinate System 34
3.2.1 Images as NumPy Arrays 35
3.2.2 RGB and BGR Ordering 36
3.3 Scaling and Aspect Ratios 36
3.4 Summary 38

4 Image Classification Basics 39
4.1 What Is Image Classification? 40
4.1.1 A Note on Terminology 40
4.1.2 The Semantic Gap 41
4.1.3 Challenges 42
4.2 Types of Learning 45
4.2.1 Supervised Learning 45
4.2.2 Unsupervised Learning 46
4.2.3 Semi-supervised Learning 47
4.3 The Deep Learning Classification Pipeline 48
4.3.1 A Shift in Mindset 48
4.3.2 Step #1: Gather Your Dataset 50
4.3.3 Step #2: Split Your Dataset 50
4.3.4 Step #3: Train Your Network 51
4.3.5 Step #4: Evaluate 51
4.3.6 Feature-based Learning versus Deep Learning for Image Classification 51
4.3.7 What Happens When my Predictions Are Incorrect? 52
4.4 Summary 52

5 Datasets for Image Classification 53
5.1 MNIST 53
5.2 Animals: Dogs, Cats, and Pandas 54
5.3 CIFAR-10 55
5.4 SMILES 55
5.5 Kaggle: Dogs vs. Cats 56
5.6 Flowers-17 57
5.7 CALTECH-101 57
5.8 Tiny ImageNet 200 58
5.9 Adience 58
5.10 ImageNet 58
5.10.1 What Is ImageNet? 59
5.10.2 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 59
5.11 Kaggle: Facial Expression Recognition Challenge 59
5.12 Indoor CVPR 60
5.13 Stanford Cars 60
5.14 LISA Traffic Signs 60
5.15 Front/Rear View Vehicles 61
5.16 Summary 62

6 Configuring Your Development Environment 63
6.1 Libraries and Packages 63
6.1.1 Python 63
6.1.2 Keras 64
6.1.3 Mxnet 64
6.1.4 OpenCV, scikit-image, scikit-learn, and more 64
6.2 Configuring Your Development Environment? 64
6.3 Preconfigured Virtual Machine 65
6.4 Cloud-based Instances 65
6.5 How to Structure Your Projects 65
6.6 Summary 66
7 Your First Image Classifier 67
7.1 Working with Image Datasets 67
7.1.1 Introducing the "Animals" Dataset 67
7.1.2 The Start to Our Deep Learning Toolkit 68
7.1.3 A Basic Image Preprocessor 69
7.1.4 Building an Image Loader 70
7.2 k-NN: A Simple Classifier 72
7.2.1 A Worked k-NN Example 74
7.2.2 k-NN Hyperparameters 75
7.2.3 Implementing k-NN 75
7.2.4 k-NN Results 78
7.2.5 Pros and Cons of k-NN 79
7.3 Summary 80
8 Parameterized Learning 81
8.1 An Introduction to Linear Classification 82
8.1.1 Four Components of Parameterized Learning 82
8.1.2 Linear Classification: From Images to Labels 83
8.1.3 Advantages of Parameterized Learning and Linear Classification 84
8.1.4 A Simple Linear Classifier With Python 85
8.2 The Role of Loss Functions 88
8.2.1 What Are Loss Functions? 88
8.2.2 Multi-class SVM Loss 89
8.2.3 Cross-entropy Loss and Softmax Classifiers 91
8.3 Summary 94
9 Optimization Methods and Regularization 95
9.1 Gradient Descent 96
9.1.1 The Loss Landscape and Optimization Surface 96
9.1.2 The "Gradient" in Gradient Descent 97
9.1.3 Treat It Like a Convex Problem (Even if It's Not) 98
9.1.4 The Bias Trick 98
9.1.5 Pseudocode for Gradient Descent 99
9.1.6 Implementing Basic Gradient Descent in Python 100

9.1.7 Simple Gradient Descent Results . . . 104
9.2 Stochastic Gradient Descent (SGD) 106
9.2.1 Mini-batch SGD . . . 106
9.2.2 Implementing Mini-batch SGD . . . 107
9.2.3 SGD Results . . . 111
9.3 Extensions to SGD 111
9.3.1 Momentum . . . 112
9.3.2 Nesterov's Acceleration . . . 113
9.3.3 Anecdotal Recommendations . . . 113
9.4 Regularization 114
9.4.1 What Is Regularization and Why Do We Need It? . . . 114
9.4.2 Updating Our Loss and Weight Update To Include Regularization . . . 115
9.4.3 Types of Regularization Techniques . . . 117
9.4.4 Regularization Applied to Image Classification . . . 117
9.5 Summary 119

10 Neural Network Fundamentals 121

10.1 Neural Network Basics 121
10.1.1 Introduction to Neural Networks . . . 122
10.1.2 The Perceptron Algorithm . . . 129
10.1.3 Backpropagation and Multi-layer Networks . . . 137
10.1.4 Multi-layer Networks with Keras . . . 153
10.1.5 The Four Ingredients in a Neural Network Recipe . . . 163
10.1.6 Weight Initialization . . . 165
10.1.7 Constant Initialization . . . 165
10.1.8 Uniform and Normal Distributions . . . 165
10.1.9 LeCun Uniform and Normal . . . 166
10.1.10 Glorot/Xavier Uniform and Normal . . . 166
10.1.11 He et al./Kaiming/MSRA Uniform and Normal . . . 167
10.1.12 Differences in Initialization Implementation . . . 167
10.2 Summary 168

11 Convolutional Neural Networks 169

11.1 Understanding Convolutions 170
11.1.1 Convolutions versus Cross-correlation . . . 170
11.1.2 The "Big Matrix" and "Tiny Matrix" Analogy . . . 171
11.1.3 Kernels . . . 171
11.1.4 A Hand Computation Example of Convolution . . . 172
11.1.5 Implementing Convolutions with Python . . . 173
11.1.6 The Role of Convolutions in Deep Learning . . . 179
11.2 CNN Building Blocks 179
11.2.1 Layer Types . . . 181
11.2.2 Convolutional Layers . . . 181
11.2.3 Activation Layers . . . 186
11.2.4 Pooling Layers . . . 186
11.2.5 Fully-connected Layers . . . 188
11.2.6 Batch Normalization . . . 189
11.2.7 Dropout . . . 190

11.3 Common Architectures and Training Patterns 191
11.3.1 Layer Patterns 191
11.3.2 Rules of Thumb 192
11.4 Are CNNs Invariant to Translation, Rotation, and Scaling? 194
11.5 Summary 195

12 Training Your First CNN 197
12.1 Keras Configurations and Converting Images to Arrays 197
12.1.1 Understanding the keras.json Configuration File 197
12.1.2 The Image to Array Preprocessor 198
12.2 ShallowNet 200
12.2.1 Implementing ShallowNet 200
12.2.2 ShallowNet on Animals 202
12.2.3 ShallowNet on CIFAR-10 206
12.3 Summary 209

13 Saving and Loading Your Models 211
13.1 Serializing a Model to Disk 211
13.2 Loading a Pre-trained Model from Disk 214
13.3 Summary 217

14 LeNet: Recognizing Handwritten Digits 219
14.1 The LeNet Architecture 219
14.2 Implementing LeNet 220
14.3 LeNet on MNIST 222
14.4 Summary 227

15 MiniVGGNet: Going Deeper with CNNs 229
15.1 The VGG Family of Networks 229
15.1.1 The (Mini) VGGNet Architecture 230
15.2 Implementing MiniVGGNet 230
15.3 MiniVGGNet on CIFAR-10 234
15.3.1 With Batch Normalization 236
15.3.2 Without Batch Normalization 237
15.4 Summary 238

16 Learning Rate Schedulers 241
16.1 Dropping Our Learning Rate 241
16.1.1 The Standard Decay Schedule in Keras 242
16.1.2 Step-based Decay 244
16.1.3 Implementing Custom Learning Rate Schedules in Keras 245
16.2 Summary 249

17 Spotting Underfitting and Overfitting 251
17.1 What Are Underfitting and Overfitting? 251
17.1.1 Effects of Learning Rates 253
17.1.2 Pay Attention to Your Training Curves 254
17.1.3 What if Validation Loss Is Lower than Training Loss? 254
17.2 Monitoring the Training Process 255
17.2.1 Creating a Training Monitor 255
17.2.2 Babysitting Training 257
17.3 Summary 260

18 Checkpointing Models 263
18.1 Checkpointing Neural Network Model Improvements 263
18.2 Checkpointing Best Neural Network Only 267
18.3 Summary 269

19 Visualizing Network Architectures 271
19.1 The Importance of Architecture Visualization 271
19.1.1 Installing graphviz and pydot 272
19.1.2 Visualizing Keras Networks 272
19.2 Summary 275

20 Out-of-the-box CNNs for Classification 277
20.1 State-of-the-art CNNs in Keras 277
20.1.1 VGG16 and VGG19 278
20.1.2 ResNet 279
20.1.3 Inception V3 280
20.1.4 Xception 280
20.1.5 Can We Go Smaller? 280
20.2 Classifying Images with Pre-trained ImageNet CNNs 281
20.2.1 Classification Results 284
20.3 Summary 286

21 Case Study: Breaking Captchas 287
21.1 Breaking Captchas with a CNN 288
21.1.1 A Note on Responsible Disclosure 288
21.1.2 The Captcha Breaker Directory Structure 290
21.1.3 Automatically Downloading Example Images 291
21.1.4 Annotating and Creating Our Dataset 292
21.1.5 Preprocessing the Digits 297
21.1.6 Training the Captcha Breaker 299
21.1.7 Testing the Captcha Breaker 303
21.2 Summary 305

22 Case Study: Smile Detection 307
22.1 The SMILES Dataset 307

22.2 Training the Smile CNN 308
22.3 Running the Smile CNN in Real-time 313
22.4 Summary 316

23 Your Next Steps 319
23.1 So, What's Next? 319