

# *Deep Learning with R*

SECOND EDITION

FRANÇOIS CHOLLET  
WITH TOMASZ KALINOWSKI  
AND J.J. ALLAIRE



MANNING  
SHELTER ISLAND

# contents

---

*preface* xii  
*acknowledgments* xiv  
*about this book* xv  
*about the authors* xviii

## **What is deep learning?** 1

### 1.1 Artificial intelligence, machine learning, and deep learning 2

*Artificial intelligence* 2 ▪ *Machine learning* 3 ▪ *Learning rules and representations from data* 4 ▪ *The “deep” in “deep learning”* 7 ▪ *Understanding how deep learning works, in three figures* 8 ▪ *What deep learning has achieved so far* 10 ▪ *Don’t believe the short-term hype* 11 ▪ *The promise of AI* 12

### 1.2 Before deep learning: A brief history of machine learning 13

*Probabilistic modeling* 13 ▪ *Early neural networks* 13 ▪ *Kernel methods* 14 ▪ *Decision trees, random forests, and gradient-boosting machines* 15 ▪ *Back to neural networks* 16 ▪ *What makes deep learning different?* 17 ▪ *The modern machine learning landscape* 17

- 1.3 Why deep learning? Why now? 20  
*Hardware* 20 ▪ *Data* 21 ▪ *Algorithms* 22 ▪ *A new wave of investment* 22 ▪ *The democratization of deep learning* 23  
*Will it last?* 24

## 2 *The mathematical building blocks of neural networks* 26

- 2.1 A first look at a neural network 27
- 2.2 Data representations for neural networks 31  
*Scalars (rank 0 tensors)* 31 ▪ *Vectors (rank 1 tensors)* 31  
*Matrices (rank 2 tensors)* 32 ▪ *Rank 3 and higher-rank tensors* 32 ▪ *Key attributes* 33 ▪ *Manipulating tensors in R* 34 ▪ *The notion of data batches* 35 ▪ *Real-world examples of data tensors* 35 ▪ *Vector data* 35 ▪ *Time-series data or sequence data* 36 ▪ *Image data* 36 ▪ *Video data* 37
- 2.3 The gears of neural networks: Tensor operations 37  
*Element-wise operations* 38 ▪ *Broadcasting* 40 ▪ *Tensor product* 41 ▪ *Tensor reshaping* 43 ▪ *Geometric interpretation of tensor operations* 44 ▪ *A geometric interpretation of deep learning* 47
- 2.4 The engine of neural networks: Gradient-based optimization 48  
*What's a derivative?* 49 ▪ *Derivative of a tensor operation: The gradient* 50 ▪ *Stochastic gradient descent* 51 ▪ *Chaining derivatives: The backpropagation algorithm* 54
- 2.5 Looking back at our first example 59  
*Reimplementing our first example from scratch in TensorFlow* 61  
*Running one training step* 63 ▪ *The full training loop* 65  
*Evaluating the model* 66

## 3 *Introduction to Keras and TensorFlow* 68

- 3.1 What's TensorFlow? 69
- 3.2 What's Keras? 69
- 3.3 Keras and TensorFlow: A brief history 71
- 3.4 Python and R interfaces: A brief history 71
- 3.5 Setting up a deep learning workspace 72  
*Installing Keras and TensorFlow* 73
- 3.6 First steps with TensorFlow 74  
*TensorFlow tensors* 74

- 3.7 Tensor attributes 75  
*Tensor shape and reshaping* 77 ▪ *Tensor slicing* 78 ▪ *Tensor broadcasting* 79 ▪ *The tf module* 80 ▪ *Constant tensors and variables* 81 ▪ *Tensor operations: Doing math in TensorFlow* 82  
*A second look at the GradientTape API* 83 ▪ *An end-to-end example: A linear classifier in pure TensorFlow* 84
- 3.8 Anatomy of a neural network: Understanding core Keras APIs 89  
*Layers: The building blocks of deep learning* 89 ▪ *From layers to models* 94 ▪ *The “compile” step: Configuring the learning process* 95 ▪ *Picking a loss function* 98 ▪ *Understanding the fit() method* 99 ▪ *Monitoring loss and metrics on validation data* 99 ▪ *Inference: Using a model after training* 101

## 4 Getting started with neural networks: Classification and regression 103

- 4.1 Classifying movie reviews: A binary classification example 105  
*The IMDB dataset* 105 ▪ *Preparing the data* 107 ▪ *Building your model* 108 ▪ *Validating your approach* 110 ▪ *Using a trained model to generate predictions on new data* 113 ▪ *Further experiments* 113 ▪ *Wrapping up* 113
- 4.2 Classifying newswires: A multiclass classification example 114  
*The Reuters dataset* 114 ▪ *Preparing the data* 116 ▪ *Building your model* 116 ▪ *Validating your approach* 117 ▪ *Generating predictions on new data* 119 ▪ *A different way to handle the labels and the loss* 120 ▪ *The importance of having sufficiently large intermediate layers* 120 ▪ *Further experiments* 121  
*Wrapping up* 121
- 4.3 Predicting house prices: A regression example 122  
*The Boston housing price dataset* 122 ▪ *Preparing the data* 123  
*Building your model* 123 ▪ *Validating your approach using K-fold validation* 124 ▪ *Generating predictions on new data* 128  
*Wrapping up* 128

## 5 Fundamentals of machine learning 130

- 5.1 Generalization: The goal of machine learning 130  
*Underfitting and overfitting* 131 ▪ *The nature of generalization in deep learning* 136
- 5.2 Evaluating machine learning models 142  
*Training, validation, and test sets* 142 ▪ *Beating a common-sense baseline* 145 ▪ *Things to keep in mind about model evaluation* 146

### 5.3 Improving model fit 146

*Tuning key gradient descent parameters 147* ▪ *Leveraging better architecture priors 149* ▪ *Increasing model capacity 150*

### 5.4 Improving generalization 152

*Dataset curation 152* ▪ *Feature engineering 153* ▪ *Using early stopping 154* ▪ *Regularizing your model 155*

## 6 *The universal workflow of machine learning 166*

### 6.1 Define the task 168

*Frame the problem 168* ▪ *Collect a dataset 169* ▪ *Understand your data 173* ▪ *Choose a measure of success 173*

### 6.2 Develop a model 174

*Prepare the data 174* ▪ *Choose an evaluation protocol 175*  
*Beat a baseline 176* ▪ *Scale up: Develop a model that overfits 177* ▪ *Regularize and tune your model 177*

### 6.3 Deploy the model 178

*Explain your work to stakeholders and set expectations 178* ▪ *Ship an inference model 179* ▪ *Monitor your model in the wild 182*  
*Maintain your model 183*

## 7 *Working with Keras: A deep dive 185*

### 7.1 A spectrum of workflows 186

### 7.2 Different ways to build Keras models 186

*The Sequential model 187* ▪ *The Functional API 189*  
*Subclassing the Model class 196* ▪ *Mixing and matching different components 199* ▪ *Remember: Use the right tool for the job 200*

### 7.3 Using built-in training and evaluation loops 201

*Writing your own metrics 202* ▪ *Using callbacks 204* ▪ *Writing your own callbacks 205* ▪ *Monitoring and visualization with TensorBoard 208*

### 7.4 Writing your own training and evaluation loops 210

*Training vs. inference 210* ▪ *Low-level usage of metrics 211*  
*A complete training and evaluation loop 212* ▪ *Make it fast with tf.function() 215* ▪ *Leveraging fit() with a custom training loop 216*

## 8 *Introduction to deep learning for computer vision 220*

### 8.1 Introduction to convnets 221

*The convolution operation 223* ▪ *The max-pooling operation 228*

- 8.2 Training a convnet from scratch on a small dataset 230
  - The relevance of deep learning for small data problems* 230
  - Downloading the data* 231 ▪ *Building the model* 234 ▪ *Data preprocessing* 235 ▪ *Using data augmentation* 241
- 8.3 Leveraging a pretrained model 245
  - Feature extraction with a pretrained model* 246 ▪ *Fine-tuning a pretrained model* 254

## 9 *Advanced deep learning for computer vision* 258

- 9.1 Three essential computer vision tasks 259
- 9.2 An image segmentation example 260
- 9.3 Modern convnet architecture patterns 269
  - Modularity, hierarchy, and reuse* 269 ▪ *Residual connections* 272 ▪ *Batch normalization* 275 ▪ *Depthwise separable convolutions* 278 ▪ *Putting it together: A mini Xception-like model* 280
- 9.4 Interpreting what convnets learn 282
  - Visualizing intermediate activations* 283 ▪ *Visualizing convnet filters* 289 ▪ *Visualizing heatmaps of class activation* 294

## 10 *Deep learning for time series* 301

- 10.1 Different kinds of time-series tasks 301
- 10.2 A temperature-forecasting example 302
  - Preparing the data* 306 ▪ *A common-sense, non-machine learning baseline* 310 ▪ *Let's try a basic machine learning model* 311
  - Let's try a 1D convolutional model* 314 ▪ *A first recurrent baseline* 316
- 10.3 Understanding recurrent neural networks 317
  - A recurrent layer in Keras* 320
- 10.4 Advanced use of recurrent neural networks 324
  - Using recurrent dropout to fight overfitting* 324 ▪ *Stacking recurrent layers* 327 ▪ *Using bidirectional RNNs* 329
  - Going even further* 332

## 11 *Deep learning for text* 334

- 11.1 Natural language processing: The bird's-eye view 334
- 11.2 Preparing text data 336
  - Text standardization* 337 ▪ *Text splitting (tokenization)* 338
  - Vocabulary indexing* 339 ▪ *Using layer\_text\_vectorization* 340

- 11.3 Two approaches for representing groups of words:  
Sets and sequences 344
  - Preparing the IMDB movie reviews data* 345 ▪ *Processing words as a set: The bag-of-words approach* 347 ▪ *Processing words as a sequence: The sequence model approach* 355
- 11.4 The Transformer architecture 366
  - Understanding self-attention* 366 ▪ *Multi-head attention* 371
  - The Transformer encoder* 372 ▪ *When to use sequence models over bag-of-words models* 381
- 11.5 Beyond text classification: Sequence-to-sequence learning 382
  - A machine translation example* 383 ▪ *Sequence-to-sequence learning with RNNs* 387 ▪ *Sequence-to-sequence learning with Transformer* 392

## 12 Generative deep learning 399

- 12.1 Text generation 401
  - A brief history of generative deep learning for sequence generation* 401 ▪ *How do you generate sequence data?* 402
  - The importance of the sampling strategy* 402 ▪ *Implementing text generation with Keras* 404 ▪ *A text-generation callback with variable-temperature sampling* 408 ▪ *Wrapping up* 413
- 12.2 DeepDream 414
  - Implementing DeepDream in Keras* 415 ▪ *Wrapping up* 421
- 12.3 Neural style transfer 422
  - The content loss* 423 ▪ *The style loss* 424 ▪ *Neural style transfer in Keras* 424 ▪ *Wrapping up* 431
- 12.4 Generating images with variational autoencoders 432
  - Sampling from latent spaces of images* 432 ▪ *Concept vectors for image editing* 433 ▪ *Variational autoencoders* 434
  - Implementing a VAE with Keras* 436 ▪ *Wrapping up* 442
- 12.5 Introduction to generative adversarial networks 442
  - A schematic GAN implementation* 443 ▪ *A bag of tricks* 444 ▪ *Getting our hands on the CelebA dataset* 445
  - The discriminator* 447 ▪ *The generator* 447 ▪ *The adversarial network* 448 ▪ *Wrapping up* 452

## 13 Best practices for the real world 454

- 13.1 Getting the most out of your models 455
  - Hyperparameter optimization* 455 ▪ *Model ensembling* 462

- 13.2 Scaling-up model training 464  
*Speeding up training on GPU with mixed precision* 465  
*Multi-GPU training* 467 ▪ *TPU training* 471

## 14 Conclusions 473

- 14.1 Key concepts in review 474  
*Various approaches to AI* 474 ▪ *What makes deep learning special within the field of machine learning* 474 ▪ *How to think about deep learning* 475 ▪ *Key enabling technologies* 476 ▪ *The universal machine learning workflow* 477 ▪ *Key network architectures* 478 ▪ *The space of possibilities* 482
- 14.2 The limitations of deep learning 484  
*The risk of anthropomorphizing machine learning models* 485  
*Automatons vs. intelligent agents* 487 ▪ *Local generalization vs. extreme generalization* 488 ▪ *The purpose of intelligence* 490  
*Climbing the spectrum of generalization* 491
- 14.3 Setting the course toward greater generality in AI 492  
*On the importance of setting the right objective: The shortcut rule* 492 ▪ *A new target* 494
- 14.4 Implementing intelligence: The missing ingredients 495  
*Intelligence as sensitivity to abstract analogies* 496 ▪ *The two poles of abstraction* 497 ▪ *The two poles of abstraction* 500 ▪ *The missing half of the picture* 500
- 14.5 The future of deep learning 501  
*Models as programs* 502 ▪ *Machine learning vs. program synthesis* 503 ▪ *Blending together deep learning and program synthesis* 503 ▪ *Lifelong learning and modular subroutine reuse* 505 ▪ *The long-term vision* 506
- 14.6 Staying up-to-date in a fast-moving field 507  
*Practice on real-world problems using Kaggle* 508 ▪ *Read about the latest developments on arXiv* 508 ▪ *Explore the Keras ecosystem* 508
- 14.7 Final words 509
- appendix Python primer for R users* 511  
*index* 535