contents

preface
acknowledgments
about this book
about the authors

1 **What is deep learning?**

1.1 Artificial intelligence, machine learning, and deep learning

Artificial intelligence ■ *Machine learning* ■ *Learning rules and representations from data* ■ *The "deep" in "deep learning"* ■ *Understanding how deep learning works, in three figures* ■ *What deep learning has achieved so far* ■ *Don't believe the short-term hype* ■ *The promise of AI*

1.2 Before deep learning: A brief history of machine learning

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

1.3 Why deep learning? Why now?

Hardware ■ *Data* ■ *Algorithms* ■ *A new wave of investment* ■ *The democratization of deep learning* ■ *Will it last?*

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

2.1 A first look at a neural network

2.2 Data representations for neural networks

Scalars (rank 0 tensors) ■ *Vectors (rank 1 tensors)* ■ *Matrices (rank 2 tensors)* ■ *Rank 3 and higher-rank tensors* ■ *Key attributes* ■ *Manipulating tensors in R* ■ *The notion of data batches* ■ *Real-world examples of data tensors* ■ *Vector data* ■ *Time-series data or sequence data* ■ *Image data* ■ *Video data*

2.3 The gears of neural networks: Tensor operations

Element-wise operations ■ *Broadcasting* ■ *Tensor product* ■ *Tensor reshaping* ■ *Geometric interpretation of tensor operations* ■ *A geometric interpretation of deep learning*

2.4 The engine of neural networks: Gradient-based optimization

What's a derivative? ■ *Derivative of a tensor operation: The gradient* ■ *Stochastic gradient descent* ■ *Chaining derivatives: The backpropagation algorithm*

2.5 Looking back at our first example

Reimplementing our first example from scratch in TensorFlow ■ *Running one training step* ■ *The full training loop* ■ *Evaluating the model*

3 ***Introduction to Keras and TensorFlow***

3.1 What's TensorFlow?

3.2 What's Keras?

3.3 Keras and TensorFlow: A brief history

3.4 Python and R interfaces: A brief history

3.5 Setting up a deep learning workspace

Installing Keras and TensorFlow

3.6 First steps with TensorFlow

TensorFlow tensors

3.7 Tensor attributes

Tensor shape and reshaping ■ *Tensor slicing* ■ *Tensor broadcasting* ■ *The tf module* ■ *Constant tensors and variables* ■ *Tensor operations: Doing math in TensorFlow* ■ *A second look at the GradientTape API* ■ *An end-to-end example: A linear classifier in pure TensorFlow*

3.8 Anatomy of a neural network: Understanding core Keras APIs

Layers: The building blocks of deep learning ■ *From layers to models* ■ *The "compile" step: Configuring the learning process* ■ *Picking a loss function* ■ *Understanding the fit() method* ■ *Monitoring loss and metrics on validation data* ■ *Inference: Using a model after training*

4 ***Getting started with neural networks: Classification and regression***

4.1 Classifying movie reviews: A binary classification example

The IMDB dataset ■ *Preparing the data* ■ *Building your model* ■ *Validating your approach* ■ *Using a trained model to generate predictions on new data* ■ *Further experiments* ■

Wrapping up

4.2 Classifying newswires: A multiclass classification example

The Reuters dataset ■ *Preparing the data* ■ *Building your model* ■ *Validating your approach* ■ *Generating predictions on new data* ■ *A different way to handle the labels and the loss* ■ *The importance of having sufficiently large intermediate layers* ■ *Further experiments* ■ *Wrapping up*

4.3 Predicting house prices: A regression example

The Boston housing price dataset ■ *Preparing the data* ■ *Building your model* ■ *Validating your approach using K-fold validation* ■ *Generating predictions on new data* ■ *Wrapping up*

5 ***Fundamentals of machine learning***

5.1 Generalization: The goal of machine learning

Underfitting and overfitting ■ *The nature of generalization in deep learning*

5.2 Evaluating machine learning models

Training, validation, and test sets ■ *Beating a common-sense baseline* ■ *Things to keep in mind about model evaluation*

5.3 Improving model fit

Tuning key gradient descent parameters ■ *Leveraging better architecture priors* ■ *Increasing model capacity*

5.4 Improving generalization

Dataset curation ■ *Feature engineering* ■ *Using early stopping* ■ *Regularizing your model*

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

6.1 Define the task

Frame the problem ■ *Collect a dataset* ■ *Understand your data* ■ *Choose a measure of success*

6.2 Develop a model

Prepare the data ■ *Choose an evaluation protocol* ■ *Beat a baseline* ■ *Scale up: Develop a model that overfits* ■ *Regularize and tune your model*

6.3 Deploy the model

Explain your work to stakeholders and set expectations ■ *Ship an inference model* ■ *Monitor your model in the wild* ■ *Maintain your model*

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

7.1 A spectrum of workflows

7.2 Different ways to build Keras models

The Sequential model ■ *The Functional API* ■ *Subclassing the Model class* ■ *Mixing and matching different components* ■ *Remember: Use the right tool for the job*

7.3 Using built-in training and evaluation loops

Writing your own metrics ■ *Using callbacks* ■ *Writing your own callbacks* ■ *Monitoring and visualization with TensorBoard*

7.4 Writing your own training and evaluation loops

Training vs. inference ■ *Low-level usage of metrics* ■ *A complete training and evaluation loop* ■ *Make it fast with tf_function()* ■ *Leveraging fit() with a custom training loop*

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

8.1 Introduction to convnets

The convolution operation ■ *The max-pooling operation*

8.2 Training a convnet from scratch on a small dataset

The relevance of deep learning for small data problems ■ *Downloading the data* ■ *Building the model* ■ *Data preprocessing* ■ *Using data augmentation*

8.3 Leveraging a pretrained model

Feature extraction with a pretrained model ■ *Fine-tuning a pretrained model*

9 **Advanced deep learning for computer vision**

9.1 Three essential computer vision tasks

9.2 An image segmentation example

9.3 Modern convnet architecture patterns

Modularity, hierarchy, and reuse ■ *Residual connections* ■ *Batch normalization* ■ *Depthwise separable convolutions* ■ *Putting it together: A mini Xception-like model*

9.4 Interpreting what convnets learn

Visualizing intermediate activations ■ *Visualizing convnet filters* ■ *Visualizing heatmaps of class activation*

10 **Deep learning for time series**

10.1 Different kinds of time-series tasks

10.2 A temperature-forecasting example

Preparing the data ■ *A common-sense, non–machine learning baseline* ■ *Let's try a basic machine learning model* ■ *Let's try a 1D convolutional model* ■ *A first recurrent baseline*

10.3 Understanding recurrent neural networks

A recurrent layer in Keras

10.4 Advanced use of recurrent neural networks

Using recurrent dropout to fight overfitting ■ *Stacking recurrent layers* ■ *Using bidirectional RNNs* ■ *Going even further*

11 **Deep learning for text**

11.1 Natural language processing: The bird's-eye view

11.2 Preparing text data

Text standardization ■ *Text splitting (tokenization)* ■ *Vocabulary indexing* ■ *Using layer_text_vectorization*

11.3 Two approaches for representing groups of words: Sets and sequences

Preparing the IMDB movie reviews data ■ *Processing words as a set: The bag-of-words approach* ■ *Processing words as a sequence: The sequence model approach*

11.4 The Transformer architecture

Understanding self-attention ■ *Multi-head attention* ■ *The Transformer encoder* ■ *When to use sequence models over bag-of-words models*

11.5 Beyond text classification: Sequence-to-sequence learning

A machine translation example ■ *Sequence-to-sequence learning with RNNs* ■ *Sequence-to-sequence learning with Transformer*

12 **Generative deep learning**

12.1 Text generation

A brief history of generative deep learning for sequence generation ■ *How do you generate sequence data?* ■ *The importance of the sampling strategy* ■ *Implementing text generation with Keras* ■ *A text-generation callback with variable-temperature sampling* ■ *Wrapping up*

12.2 DeepDream

Implementing DeepDream in Keras ■ *Wrapping up*

12.3 Neural style transfer

The content loss ■ *The style loss* ■ *Neural style transfer in Keras* ■ *Wrapping up*

12.4 Generating images with variational autoencoders

Sampling from latent spaces of images ■ *Concept vectors for image editing* ■ *Variational autoencoders* ■ *Implementing a VAE with Keras* ■ *Wrapping up*

12.5 Introduction to generative adversarial networks

A schematic GAN implementation ■ *A bag of tricks* ■ *Getting our hands on the CelebA dataset* ■ *The discriminator* ■ *The generator* ■ *The adversarial network* ■ *Wrapping up*

13 ***Best practices for the real world***

13.1 Getting the most out of your models

Hyperparameter optimization ■ *Model ensembling*

13.2 Scaling-up model training

Speeding up training on GPU with mixed precision ■ *Multi-GPU training* ■ *TPU training*

14 ***Conclusions***

14.1 Key concepts in review

Various approaches to AI ■ *What makes deep learning special within the field of machine learning* ■ *How to think about deep learning* ■ *Key enabling technologies* ■ *The universal machine learning workflow* ■ *Key network architectures* ■ *The space of possibilities*

14.2 The limitations of deep learning

The risk of anthropomorphizing machine learning models ■ *Automatons vs. intelligent agents* ■ *Local generalization vs. extreme generalization* ■ *The purpose of intelligence* ■ *Climbing the spectrum of generalization*

14.3 Setting the course toward greater generality in AI

On the importance of setting the right objective: The shortcut rule ■ *A new target*

14.4 Implementing intelligence: The missing ingredients

Intelligence as sensitivity to abstract analogies ■ *The two poles of abstraction* ■ *The two poles of abstraction* ■ *The missing half of the picture*

14.5 The future of deep learning

Models as programs ■ *Machine learning vs. program synthesis* ■ *Blending together deep learning and program synthesis* ■ *Lifelong learning and modular subroutine reuse* ■ *The long-term vision*

14.6 Staying up-to-date in a fast-moving field

Practice on real-world problems using Kaggle ■ *Read about the latest developments on arXiv* ■ *Explore the Keras ecosystem*

14.7 Final words

appendix: Python primer for R users

index