

Hands-On ML ToC

1. Preface
 1. The Machine Learning Tsunami
 2. Machine Learning in Your Projects
 3. Objective and Approach
 4. Prerequisites
 5. Roadmap
 6. Changes in the Second Edition
 7. Other Resources
 8. Conventions Used in This Book
 9. Code Examples
 10. Using Code Examples
 11. O'Reilly Online Learning
 12. How to Contact Us
 13. Acknowledgments
2. I. The Fundamentals of Machine Learning
3. 1. The Machine Learning Landscape
 1. What Is Machine Learning?
 2. Why Use Machine Learning?
 3. Examples of Applications
 4. Types of Machine Learning Systems
 1. Supervised/Unsupervised Learning
 2. Batch and Online Learning
 3. Instance-Based Versus Model-Based Learning
 5. Main Challenges of Machine Learning
 1. Insufficient Quantity of Training Data
 2. Nonrepresentative Training Data
 3. Poor-Quality Data
 4. Irrelevant Features
 5. Overfitting the Training Data
 6. Underfitting the Training Data
 7. Stepping Back
 6. Testing and Validating
 1. Hyperparameter Tuning and Model Selection
 2. Data Mismatch
 7. Exercises
4. 2. End-to-End Machine Learning Project
 1. Working with Real Data
 2. Look at the Big Picture
 1. Frame the Problem
 2. Select a Performance Measure
 3. Check the Assumptions
 3. Get the Data
 1. Create the Workspace
 2. Download the Data
 3. Take a Quick Look at the Data Structure
 4. Create a Test Set
 4. Discover and Visualize the Data to Gain Insights
 1. Visualizing Geographical Data
 2. Looking for Correlations
 3. Experimenting with Attribute Combinations
 5. Prepare the Data for Machine Learning Algorithms
 1. Data Cleaning
 2. Handling Text and Categorical Attributes
 3. Custom Transformers
 4. Feature Scaling

5. Transformation Pipelines
6. Select and Train a Model
 1. Training and Evaluating on the Training Set
 2. Better Evaluation Using Cross-Validation
7. Fine-Tune Your Model
 1. Grid Search
 2. Randomized Search
 3. Ensemble Methods
 4. Analyze the Best Models and Their Errors
 5. Evaluate Your System on the Test Set
8. Launch, Monitor, and Maintain Your System
9. Try It Out!
10. Exercises
5. 3. Classification
 1. MNIST
 2. Training a Binary Classifier
 3. Performance Measures
 1. Measuring Accuracy Using Cross-Validation
 2. Confusion Matrix
 3. Precision and Recall
 4. Precision/Recall Trade-off
 5. The ROC Curve
 4. Multiclass Classification
 5. Error Analysis
 6. Multilabel Classification
 7. Multioutput Classification
 8. Exercises
6. 4. Training Models
 1. Linear Regression
 1. The Normal Equation
 2. Computational Complexity
 2. Gradient Descent
 1. Batch Gradient Descent
 2. Stochastic Gradient Descent
 3. Mini-batch Gradient Descent
 3. Polynomial Regression
 4. Learning Curves
 5. Regularized Linear Models
 1. Ridge Regression
 2. Lasso Regression
 3. Elastic Net
 4. Early Stopping
 6. Logistic Regression
 1. Estimating Probabilities
 2. Training and Cost Function
 3. Decision Boundaries
 4. Softmax Regression
 7. Exercises
7. 5. Support Vector Machines
 1. Linear SVM Classification
 1. Soft Margin Classification
 2. Nonlinear SVM Classification
 1. Polynomial Kernel
 2. Similarity Features
 3. Gaussian RBF Kernel
 4. Computational Complexity
 3. SVM Regression
 4. Under the Hood
 1. Decision Function and Predictions

2. Training Objective
 3. Quadratic Programming
 4. The Dual Problem
 5. Kernelized SVMs
 6. Online SVMs
5. Exercises
8. 6. Decision Trees
 1. Training and Visualizing a Decision Tree
 2. Making Predictions
 3. Estimating Class Probabilities
 4. The CART Training Algorithm
 5. Computational Complexity
 6. Gini Impurity or Entropy?
 7. Regularization Hyperparameters
 8. Regression
 9. Instability
 10. Exercises
9. 7. Ensemble Learning and Random Forests
 1. Voting Classifiers
 2. Bagging and Pasting
 1. Bagging and Pasting in Scikit-Learn
 2. Out-of-Bag Evaluation
 3. Random Patches and Random Subspaces
 4. Random Forests
 1. Extra-Trees
 2. Feature Importance
 5. Boosting
 1. AdaBoost
 2. Gradient Boosting
 6. Stacking
 7. Exercises
10. 8. Dimensionality Reduction
 1. The Curse of Dimensionality
 2. Main Approaches for Dimensionality Reduction
 1. Projection
 2. Manifold Learning
 3. PCA
 1. Preserving the Variance
 2. Principal Components
 3. Projecting Down to d Dimensions
 4. Using Scikit-Learn
 5. Explained Variance Ratio
 6. Choosing the Right Number of Dimensions
 7. PCA for Compression
 8. Randomized PCA
 9. Incremental PCA
 4. Kernel PCA
 1. Selecting a Kernel and Tuning Hyperparameters
 5. LLE
 6. Other Dimensionality Reduction Techniques
 7. Exercises
11. 9. Unsupervised Learning Techniques
 1. Clustering
 1. K-Means
 2. Limits of K-Means
 3. Using Clustering for Image Segmentation
 4. Using Clustering for Preprocessing
 5. Using Clustering for Semi-Supervised Learning
 6. DBSCAN

7. Other Clustering Algorithms
2. Gaussian Mixtures
 1. Anomaly Detection Using Gaussian Mixtures
 2. Selecting the Number of Clusters
 3. Bayesian Gaussian Mixture Models
 4. Other Algorithms for Anomaly and Novelty Detection
3. Exercises
12. II. Neural Networks and Deep Learning
13. 10. Introduction to Artificial Neural Networks with Keras
 1. From Biological to Artificial Neurons
 1. Biological Neurons
 2. Logical Computations with Neurons
 3. The Perceptron
 4. The Multilayer Perceptron and Backpropagation
 5. Regression MLPs
 6. Classification MLPs
 2. Implementing MLPs with Keras
 1. Installing TensorFlow 2
 2. Building an Image Classifier Using the Sequential API
 3. Building a Regression MLP Using the Sequential API
 4. Building Complex Models Using the Functional API
 5. Using the Subclassing API to Build Dynamic Models
 6. Saving and Restoring a Model
 7. Using Callbacks
 8. Using TensorBoard for Visualization
 3. Fine-Tuning Neural Network Hyperparameters
 1. Number of Hidden Layers
 2. Number of Neurons per Hidden Layer
 3. Learning Rate, Batch Size, and Other Hyperparameters
 4. Exercises
14. 11. Training Deep Neural Networks
 1. The Vanishing/Exploding Gradients Problems
 1. Glorot and He Initialization
 2. Nonsaturating Activation Functions
 3. Batch Normalization
 4. Gradient Clipping
 2. Reusing Pretrained Layers
 1. Transfer Learning with Keras
 2. Unsupervised Pretraining
 3. Pretraining on an Auxiliary Task
 3. Faster Optimizers
 1. Momentum Optimization
 2. Nesterov Accelerated Gradient
 3. AdaGrad
 4. RMSProp
 5. Adam and Nadam Optimization
 6. Learning Rate Scheduling
 4. Avoiding Overfitting Through Regularization
 1. ℓ_1 and ℓ_2 Regularization
 2. Dropout
 3. Monte Carlo (MC) Dropout
 4. Max-Norm Regularization
 5. Summary and Practical Guidelines
 6. Exercises
15. 12. Custom Models and Training with TensorFlow
 1. A Quick Tour of TensorFlow
 2. Using TensorFlow like NumPy
 1. Tensors and Operations
 2. Tensors and NumPy

- 3. Type Conversions
 - 4. Variables
 - 5. Other Data Structures
- 3. Customizing Models and Training Algorithms
 - 1. Custom Loss Functions
 - 2. Saving and Loading Models That Contain Custom Components
 - 3. Custom Activation Functions, Initializers, Regularizers, and Constraints
 - 4. Custom Metrics
 - 5. Custom Layers
 - 6. Custom Models
 - 7. Losses and Metrics Based on Model Internals
 - 8. Computing Gradients Using Autodiff
 - 9. Custom Training Loops
- 4. TensorFlow Functions and Graphs
 - 1. AutoGraph and Tracing
 - 2. TF Function Rules
- 5. Exercises
- 16. 13. Loading and Preprocessing Data with TensorFlow
 - 1. The Data API
 - 1. Chaining Transformations
 - 2. Shuffling the Data
 - 3. Preprocessing the Data
 - 4. Putting Everything Together
 - 5. Prefetching
 - 6. Using the Dataset with tf.keras
 - 2. The TFRecord Format
 - 1. Compressed TFRecord Files
 - 2. A Brief Introduction to Protocol Buffers
 - 3. TensorFlow Protobufs
 - 4. Loading and Parsing Examples
 - 5. Handling Lists of Lists Using the SequenceExample Protobuf
 - 3. Preprocessing the Input Features
 - 1. Encoding Categorical Features Using One-Hot Vectors
 - 2. Encoding Categorical Features Using Embeddings
 - 3. Keras Preprocessing Layers
 - 4. TF Transform
 - 5. The TensorFlow Datasets (TFDS) Project
 - 6. Exercises
- 17. 14. Deep Computer Vision Using Convolutional Neural Networks
 - 1. The Architecture of the Visual Cortex
 - 2. Convolutional Layers
 - 1. Filters
 - 2. Stacking Multiple Feature Maps
 - 3. TensorFlow Implementation
 - 4. Memory Requirements
 - 3. Pooling Layers
 - 1. TensorFlow Implementation
 - 4. CNN Architectures
 - 1. LeNet-5
 - 2. AlexNet
 - 3. GoogLeNet
 - 4. VGGNet
 - 5. ResNet
 - 6. Xception
 - 7. SENet
 - 5. Implementing a ResNet-34 CNN Using Keras
 - 6. Using Pretrained Models from Keras
 - 7. Pretrained Models for Transfer Learning
 - 8. Classification and Localization

9. Object Detection
 1. Fully Convolutional Networks
 2. You Only Look Once (YOLO)
10. Semantic Segmentation
11. Exercises
18. 15. Processing Sequences Using RNNs and CNNs
 1. Recurrent Neurons and Layers
 1. Memory Cells
 2. Input and Output Sequences
 2. Training RNNs
 3. Forecasting a Time Series
 1. Baseline Metrics
 2. Implementing a Simple RNN
 3. Deep RNNs
 4. Forecasting Several Time Steps Ahead
 4. Handling Long Sequences
 1. Fighting the Unstable Gradients Problem
 2. Tackling the Short-Term Memory Problem
 5. Exercises
19. 16. Natural Language Processing with RNNs and Attention
 1. Generating Shakespearean Text Using a Character RNN
 1. Creating the Training Dataset
 2. How to Split a Sequential Dataset
 3. Chopping the Sequential Dataset into Multiple Windows
 4. Building and Training the Char-RNN Model
 5. Using the Char-RNN Model
 6. Generating Fake Shakespearean Text
 7. Stateful RNN
 2. Sentiment Analysis
 1. Masking
 2. Reusing Pretrained Embeddings
 3. An Encoder–Decoder Network for Neural Machine Translation
 1. Bidirectional RNNs
 2. Beam Search
 4. Attention Mechanisms
 1. Visual Attention
 2. Attention Is All You Need: The Transformer Architecture
 5. Recent Innovations in Language Models
 6. Exercises
20. 17. Representation Learning and Generative Learning Using Autoencoders and GANs
 1. Efficient Data Representations
 2. Performing PCA with an Undercomplete Linear Autoencoder
 3. Stacked Autoencoders
 1. Implementing a Stacked Autoencoder Using Keras
 2. Visualizing the Reconstructions
 3. Visualizing the Fashion MNIST Dataset
 4. Unsupervised Pretraining Using Stacked Autoencoders
 5. Tying Weights
 6. Training One Autoencoder at a Time
 4. Convolutional Autoencoders
 5. Recurrent Autoencoders
 6. Denoising Autoencoders
 7. Sparse Autoencoders
 8. Variational Autoencoders
 1. Generating Fashion MNIST Images
 9. Generative Adversarial Networks
 1. The Difficulties of Training GANs
 2. Deep Convolutional GANs
 3. Progressive Growing of GANs

- 4. [StyleGANs](#)
- 10. [Exercises](#)
- 21. [18. Reinforcement Learning](#)
 - 1. [Learning to Optimize Rewards](#)
 - 2. [Policy Search](#)
 - 3. [Introduction to OpenAI Gym](#)
 - 4. [Neural Network Policies](#)
 - 5. [Evaluating Actions: The Credit Assignment Problem](#)
 - 6. [Policy Gradients](#)
 - 7. [Markov Decision Processes](#)
 - 8. [Temporal Difference Learning](#)
 - 9. [Q-Learning](#)
 - 1. [Exploration Policies](#)
 - 2. [Approximate Q-Learning and Deep Q-Learning](#)
 - 10. [Implementing Deep Q-Learning](#)
 - 11. [Deep Q-Learning Variants](#)
 - 1. [Fixed Q-Value Targets](#)
 - 2. [Double DQN](#)
 - 3. [Prioritized Experience Replay](#)
 - 4. [Dueling DQN](#)
 - 12. [The TF-Agents Library](#)
 - 1. [Installing TF-Agents](#)
 - 2. [TF-Agents Environments](#)
 - 3. [Environment Specifications](#)
 - 4. [Environment Wrappers and Atari Preprocessing](#)
 - 5. [Training Architecture](#)
 - 6. [Creating the Deep Q-Network](#)
 - 7. [Creating the DQN Agent](#)
 - 8. [Creating the Replay Buffer and the Corresponding Observer](#)
 - 9. [Creating Training Metrics](#)
 - 10. [Creating the Collect Driver](#)
 - 11. [Creating the Dataset](#)
 - 12. [Creating the Training Loop](#)
 - 13. [Overview of Some Popular RL Algorithms](#)
 - 14. [Exercises](#)
- 22. [19. Training and Deploying TensorFlow Models at Scale](#)
 - 1. [Serving a TensorFlow Model](#)
 - 1. [Using TensorFlow Serving](#)
 - 2. [Creating a Prediction Service on GCP AI Platform](#)
 - 3. [Using the Prediction Service](#)
 - 2. [Deploying a Model to a Mobile or Embedded Device](#)
 - 3. [Using GPUs to Speed Up Computations](#)
 - 1. [Getting Your Own GPU](#)
 - 2. [Using a GPU-Equipped Virtual Machine](#)
 - 3. [Colaboratory](#)
 - 4. [Managing the GPU RAM](#)
 - 5. [Placing Operations and Variables on Devices](#)
 - 6. [Parallel Execution Across Multiple Devices](#)
 - 4. [Training Models Across Multiple Devices](#)
 - 1. [Model Parallelism](#)
 - 2. [Data Parallelism](#)
 - 3. [Training at Scale Using the Distribution Strategies API](#)
 - 4. [Training a Model on a TensorFlow Cluster](#)
 - 5. [Running Large Training Jobs on Google Cloud AI Platform](#)
 - 6. [Black Box Hyperparameter Tuning on AI Platform](#)
 - 5. [Exercises](#)
 - 6. [Thank You!](#)
- 23. [A. Exercise Solutions](#)
 - 1. [Chapter 1: The Machine Learning Landscape](#)

2. [Chapter 2: End-to-End Machine Learning Project](#)
3. [Chapter 3: Classification](#)
4. [Chapter 4: Training Models](#)
5. [Chapter 5: Support Vector Machines](#)
6. [Chapter 6: Decision Trees](#)
7. [Chapter 7: Ensemble Learning and Random Forests](#)
8. [Chapter 8: Dimensionality Reduction](#)
9. [Chapter 9: Unsupervised Learning Techniques](#)
10. [Chapter 10: Introduction to Artificial Neural Networks with Keras](#)
11. [Chapter 11: Training Deep Neural Networks](#)
12. [Chapter 12: Custom Models and Training with TensorFlow](#)
13. [Chapter 13: Loading and Preprocessing Data with TensorFlow](#)
14. [Chapter 14: Deep Computer Vision Using Convolutional Neural Networks](#)
15. [Chapter 15: Processing Sequences Using RNNs and CNNs](#)
16. [Chapter 16: Natural Language Processing with RNNs and Attention](#)
17. [Chapter 17: Representation Learning and Generative Learning Using Autoencoders and GANs](#)
18. [Chapter 18: Reinforcement Learning](#)
19. [Chapter 19: Training and Deploying TensorFlow Models at Scale](#)
24. [B. Machine Learning Project Checklist](#)
 1. [Frame the Problem and Look at the Big Picture](#)
 2. [Get the Data](#)
 3. [Explore the Data](#)
 4. [Prepare the Data](#)
 5. [Shortlist Promising Models](#)
 6. [Fine-Tune the System](#)
 7. [Present Your Solution](#)
 8. [Launch!](#)
25. [C. SVM Dual Problem](#)
26. [D. Autodiff](#)
 1. [Manual Differentiation](#)
 2. [Finite Difference Approximation](#)
 3. [Forward-Mode Autodiff](#)
 4. [Reverse-Mode Autodiff](#)
27. [E. Other Popular ANN Architectures](#)
 1. [Hopfield Networks](#)
 2. [Boltzmann Machines](#)
 3. [Restricted Boltzmann Machines](#)
 4. [Deep Belief Nets](#)
 5. [Self-Organizing Maps](#)
28. [F. Special Data Structures](#)
 1. [Strings](#)
 2. [Ragged Tensors](#)
 3. [Sparse Tensors](#)
 4. [Tensor Arrays](#)
 5. [Sets](#)
 6. [Queues](#)
29. [G. TensorFlow Graphs](#)
 1. [TF Functions and Concrete Functions](#)
 2. [Exploring Function Definitions and Graphs](#)
 3. [A Closer Look at Tracing](#)
 4. [Using AutoGraph to Capture Control Flow](#)
 5. [Handling Variables and Other Resources in TF Functions](#)
 6. [Using TF Functions with tf.keras \(or Not\)](#)
30. [Index](#)