

1. Record Nr.	UNINA9910556881503321
Autore	Michelucci Umberto
Titolo	Applied deep learning with TensorFlow 2 : learn to implement advanced deep learning techniques with Python // Umberto Michelucci
Pubbl/distr/stampa	New York, NY : , : Apress, , [2022] ©2022
ISBN	1-5231-5107-2 1-4842-8020-2
Edizione	[2nd ed.]
Descrizione fisica	1 online resource (397 pages)
Collana	ITpro collection
Disciplina	006.32
Soggetti	Python (Computer program language) Machine learning Neural networks (Computer science)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Intro -- Table of Contents -- About the Author -- About the Contributing Author -- About the Technical Reviewer -- Acknowledgments -- Foreword -- Introduction -- Chapter 1: Optimization and Neural Networks -- A Basic Understanding of Neural Networks -- The Problem of Learning -- A First Definition of Learning -- [Advanced Section] Assumption in the Formulation -- A Definition of Learning for Neural Networks -- Constrained vs. Unconstrained Optimization -- [Advanced Section] Reducing a Constrained Problem to an Unconstrained Optimization Problem -- Absolute and Local Minima of a Function -- Optimization Algorithms -- Line Search and Trust Region -- Steepest Descent -- The Gradient Descent Algorithm -- Choosing the Right Learning Rate -- Variations of GD -- Mini-Batch GD -- Stochastic GD -- How to Choose the Right Mini-Batch Size -- [Advanced Section] SGD and Fractals -- Exercises -- Conclusion -- Chapter 2: Hands-on with a Single Neuron -- A Short Overview of a Neuron's Structure -- A Short Introduction to Matrix Notation -- An Overview of the Most Common Activation Functions -- Identity Function -- Sigmoid Function -- Tanh (Hyperbolic Tangent) Activation Function -- ReLU (Rectified Linear Unit) Activation Function -- Leaky

ReLU -- The Swish Activation Function -- Other Activation Functions --
How to Implement a Neuron in Keras -- Python Implementation Tips:
Loops and NumPy -- Linear Regression with a Single Neuron -- The
Dataset for the Real-World Example -- Dataset Splitting -- Linear
Regression Model -- Keras Implementation -- The Model's Learning
Phase -- Model's Performance Evaluation on Unseen Data -- Logistic
Regression with a Single Neuron -- The Dataset for the Classification
Problem -- Dataset Splitting -- The Logistic Regression Model -- Keras
Implementation -- The Model's Learning Phase -- The Model's
Performance Evaluation.
Conclusion -- Exercises -- References -- Chapter 3: Feed-Forward
Neural Networks -- A Short Review of Network's Architecture
and Matrix Notation -- Output of Neurons -- A Short Summary
of Matrix Dimensions -- Example: Equations for a Network with Three
Layers -- Hyper-Parameters in Fully Connected Networks -- A Short
Review of the Softmax Activation Function for Multiclass Classifications
-- A Brief Digression: Overfitting -- A Practical Example of Overfitting
-- Basic Error Analysis -- Implementing a Feed-Forward Neural
Network in Keras -- Multiclass Classification with Feed-Forward Neural
Networks -- The Zalando Dataset for the Real-World Example --
Modifying Labels for the Softmax Function: One-Hot Encoding -- The
Feed-Forward Network Model -- Keras Implementation -- Gradient
Descent Variations Performances -- Comparing the Variations --
Examples of Wrong Predictions -- Weight Initialization -- Adding Many
Layers Efficiently -- Advantages of Additional Hidden Layers --
Comparing Different Networks -- Tips for Choosing the Right Network
-- Estimating the Memory Requirements of Models -- General Formula
for the Memory Footprint -- Exercises -- References -- Chapter 4:
Regularization -- Complex Networks and Overfitting -- What Is
Regularization -- About Network Complexity -- p Norm -- 2
Regularization -- Theory of 2 Regularization -- Keras Implementation
-- 1 Regularization -- Theory of 1 Regularization and Keras
Implementation -- Are the Weights Really Going to Zero? -- Dropout --
Early Stopping -- Additional Methods -- Exercises -- References --
Chapter 5: Advanced Optimizers -- Available Optimizers in Keras
in TensorFlow 2.5 -- Advanced Optimizers -- Exponentially Weighted
Averages -- Momentum -- RMSProp -- Adam -- Comparison
of the Optimizers' Performance -- Small Coding Digression -- Which
Optimizer Should You Use?.
Chapter 6: Hyper-Parameter Tuning -- Black-Box Optimization --
Notes on Black-Box Functions -- The Problem of Hyper-Parameter
Tuning -- Sample Black-Box Problem -- Grid Search -- Random Search
-- Coarse to Fine Optimization -- Bayesian Optimization -- Nadaraya-
Watson Regression -- Gaussian Process -- Stationary Process --
Prediction with Gaussian Processes -- Acquisition Function -- Upper
Confidence Bound (UCB) -- Example -- Sampling on a Logarithmic
Scale -- Hyper-Parameter Tuning with the Zalando Dataset -- A Quick
Note about the Radial Basis Function -- Exercises -- References --
Chapter 7: Convolutional Neural Networks -- Kernels and Filters --
Convolution -- Examples of Convolution -- Pooling -- Padding --
Building Blocks of a CNN -- Convolutional Layers -- Pooling Layers --
Stacking Layers Together -- An Example of a CNN -- Conclusion --
Exercises -- References -- Chapter 8: A Brief Introduction to Recurrent
Neural Networks -- Introduction to RNNs -- Notation -- The Basic Idea
of RNNs -- Why the Name Recurrent -- Learning to Count --
Conclusion -- Further Readings -- Chapter 9: Autoencoders --
Introduction -- Regularization in Autoencoders -- Feed-Forward
Autoencoders -- Activation Function of the Output Layer -- ReLU --

Sigmoid -- The Loss Function -- Mean Square Error -- Binary Cross-Entropy -- The Reconstruction Error -- Example: Reconstructing Handwritten Digits -- Autoencoder Applications -- Dimensionality Reduction -- Equivalence with PCA -- Classification -- Classification with Latent Features -- The Curse of Dimensionality: A Small Detour -- Anomaly Detection -- Model Stability: A Short Note -- Denoising Autoencoders -- Beyond FFA: Autoencoders with Convolutional Layers -- Implementation in Keras -- Exercises -- Further Readings -- Chapter 10: Metric Analysis -- Human-Level Performance and Bayes Error. A Short Story About Human-Level Performance -- Human-Level Performance on MNIST -- Bias -- Metric Analysis Diagram -- Training Set Overfitting -- Test Set -- How to Split Your Dataset -- Unbalanced Class Distribution: What Can Happen -- Datasets with Different Distributions -- k-fold Cross Validation -- Manual Metric Analysis: An Example -- Exercises -- References -- Chapter 11: Generative Adversarial Networks (GANs) -- Introduction to GANs -- Training Algorithm for GANs -- A Practical Example with Keras and MNIST -- A Note on Training -- Conditional GANs -- Conclusion -- Appendix A: Introduction to Keras -- Some History -- Understanding the Sequential Model -- Understanding Keras Layers -- Setting the Activation Function -- Using Functional APIs -- Specifying Loss Functions and Metrics -- Putting It All Together and Training -- Modeling `evaluate()` and `predict()` -- Using Callback Functions -- Saving and Loading Models -- Saving Your Weights Manually -- Saving the Entire Model -- Conclusion -- Appendix B: Customizing Keras -- Customizing Callback Classes -- Example of a Custom Callback Class -- Custom Training Loops -- Calculating Gradients -- Custom Training Loop for a Neural Network -- Index.

Sommario/riassunto

Understand how neural networks work and learn how to implement them using TensorFlow 2.0 and Keras. This new edition focuses on the fundamental concepts and at the same time on practical aspects of implementing neural networks and deep learning for your research projects. This book is designed so that you can focus on the parts you are interested in. You will explore topics as regularization, optimizers, optimization, metric analysis, and hyper-parameter tuning. In addition, you will learn the fundamentals ideas behind autoencoders and generative adversarial networks. All the code presented in the book will be available in the form of Jupyter notebooks which would allow you to try out all examples and extend them in interesting ways. A companion online book is available with the complete code for all examples discussed in the book and additional material more related to TensorFlow and Keras. All the code will be available in Jupyter notebook format and can be opened directly in Google Colab (no need to install anything locally) or downloaded on your own machine and tested locally.

2. Record Nr.	UNISA996247880703316
Autore	White Lynn Townsend <1907-1987.>
Titolo	Medieval technology and social change [[electronic resource] /] / Lynn White, Jr
Pubbl/distr/stampa	London, : Oxford University Press, c1962
Descrizione fisica	194 p. : ill. ; ; 21 cm
Soggetti	Civilization, Medieval Technology and civilization
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia