1.

Record Nr.	UNINA9910300744803321
Autore	Masters Timothy
Titolo	Deep Belief Nets in C++ and CUDA C: Volume 2 : Autoencoding in the Complex Domain / / by Timothy Masters
Pubbl/distr/stampa	Berkeley, CA : , : Apress : , : Imprint : Apress, , 2018
ISBN	1-4842-3646-7
Edizione	[1st ed. 2018.]
Descrizione fisica	1 online resource (XI, 258 p. 47 illus.)
Disciplina	006
Soggetti	Artificial intelligence Programming languages (Electronic computers) Big data Artificial Intelligence Programming Languages, Compilers, Interpreters Big Data Big Data/Analytics
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	0. Introduction -- 1. Embedded Class Labels -- 2. Signal Preprocessing -- 3. Image Preprocessing -- 4. Autoencoding -- 5. Deep Operating Manual.
Sommario/riassunto	Discover the essential building blocks of a common and powerful form of deep belief net: the autoencoder. You’ll take this topic beyond current usage by extending it to the complex domain for signal and image processing applications. Deep Belief Nets in C++ and CUDA C: Volume 2 also covers several algorithms for preprocessing time series and image data. These algorithms focus on the creation of complex-domain predictors that are suitable for input to a complex-domain autoencoder. Finally, you’ll learn a method for embedding class information in the input layer of a restricted Boltzmann machine. This facilitates generative display of samples from individual classes rather than the entire data distribution. The ability to see the features that the model has learned for each class separately can be invaluable. At each step this book provides you with intuitive motivation, a summary of the most important equations relevant to the topic, and highly commented

code for threaded computation on modern CPUs as well as massive parallel processing on computers with CUDA-capable video display cards. You will: • Code for deep learning, neural networks, and AI using C++ and CUDA C • Carry out signal preprocessing using simple transformations, Fourier transforms, Morlet wavelets, and more • Use the Fourier Transform for image preprocessing • Implement autoencoding via activation in the complex domain • Work with algorithms for CUDA gradient computation • Use the DEEP operating manual.