

1. Record Nr.	UNINA9910799497203321
Titolo	Artificial Intelligence for Edge Computing // Mudhakar Srivatsa, Tarek Abdelzaher, and Ting He, editors
Pubbl/distr/stampa	Cham, Switzerland : , : Springer, , [2023] ©2023
ISBN	3-031-40787-3
Edizione	[First edition.]
Descrizione fisica	1 online resource (373 pages)
Disciplina	006.3
Soggetti	Artificial intelligence Edge computing
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references.
Nota di contenuto	Intro -- Preface -- Part 1: Core Problems -- Part 2: Distributed Problems -- Part 3: Cross-Cutting Thoughts -- Contents -- Contributors -- Part I Core Problems -- 1 Neural Network Models for Time Series Data -- 1 Introduction -- 2 DeepSense Framework -- 2.1 Convolutional Layers -- 2.2 Recurrent Layers -- 2.3 Output Layer -- 3 Task-Specific Customization -- 3.1 General Customization Process -- 3.2 Customize Mobile Sensing Tasks -- 4 Evaluation -- 4.1 Data Collection and Datasets -- 4.2 Evaluation Platforms -- 4.3 Algorithms in Comparison -- 4.4 Effectiveness -- 4.4.1 CarTrack -- 4.4.2 HHAR -- 4.4.3 UserID -- 4.5 Latency and Energy -- References -- 2 Self-Supervised Learning from Unlabeled IoT Data -- 1 Introduction -- 1.1 Time-Domain Self-Supervised Contrastive Learning -- 1.2 Frequency-Domain Self-Supervised Contrastive Learning -- 1.3 Semi-Supervised Contrastive Learning -- 1.4 Spectrogram Masked Autoencoder for IoT Applications -- 1.5 A Case Study: Self-Supervised Learning on IoBT-OS -- 1.6 Chapter Organization -- 2 Time-Domain Self-Supervised Contrastive Learning Framework for IoT -- 2.1 Overview -- 2.2 Signal Model -- 2.3 Architecture of SemiAMC -- 2.4 Self-Supervised Contrastive Pre-Training -- 2.4.1 Data Augmentation -- 2.4.2 Encoder -- 2.4.3 Projection Head -- 2.4.4 Contrastive Loss -- 2.5 Evaluation -- 2.5.1 Dataset -- 2.5.2 Experimental Setup -- 2.5.3 Comparison with Supervised Frameworks -- 2.5.4 Performance under Different Amount

of Labeled Data -- 2.5.5 Performance under Different Amount of Unlabeled Data -- 3 Frequency-Domain Self-Supervised Contrastive Learning Framework for IoT -- 3.1 Overview -- 3.2 Background and Related Work -- 3.2.1 Deep Neural Network for IoT Applications -- 3.2.2 Self-Supervised Learning -- 3.2.3 Representation Learning -- 3.3 Design of STFNet -- 3.3.1 STFNet Overview -- 3.3.2 STFNet Block Fundamentals. 3.3.3 STFNet Hologram Interleaving -- 3.3.4 STFNet-Filtering Operation -- 3.3.5 STFNet-Convolution Operation -- 3.3.6 STFNet-Pooling Operation -- 3.4 Design of STF-CLS -- 3.4.1 Overview -- 3.4.2 Contrastive Self-Supervised Learning Framework -- 3.4.3 Data Augmentation -- 3.4.4 Design of the STFNet-Based Encoder -- 3.5 Evaluation -- 3.5.1 Datasets -- 3.5.2 Experiment Setup -- 3.5.3 Results -- 3.6 Discussion and Limitations -- 4 Frequency-Domain Semi-Supervised Contrastive Learning Framework for IoT -- 4.1 Overview -- 4.2 Preliminary and Motivation -- 4.2.1 Self-Supervised Contrastive Learning -- 4.2.2 Supervised Contrastive Learning -- 4.2.3 Motivation -- 4.3 Design of SemiC-HAR -- 4.3.1 Overview -- 4.3.2 Supervised Training -- 4.3.3 Self-Labeling -- 4.3.4 Semi-Supervised Contrastive Pre-Training -- 4.3.5 Downstream HAR Task -- 4.4 Evaluation -- 4.4.1 Experiment Setup -- 4.4.2 Results -- 5 Spectrogram Masked Autoencoder for IoT -- 5.1 Overview -- 5.2 Self-Supervised Learning for Sensing Data -- 5.3 Design of SMAE -- 5.3.1 Overview of SMAE -- 5.3.2 Masking -- 5.3.3 SMAE Encoder -- 5.3.4 SMAE Decoder -- 5.3.5 SMAE Loss Function -- 5.4 Evaluation -- 5.4.1 Datasets -- 5.4.2 Experiment Setup -- 5.4.3 Comparison with Previous Self-Supervised Approaches -- 5.4.4 Performance Under Different Number of Training Data -- 5.4.5 Performance Under Different Augmentation Strategies -- 6 A Case Study: Self-Supervised Learning on IoBT-OS -- 6.1 Overview -- 6.2 Background: The Decision Loop -- 6.3 IoBT-OS -- 6.4 The Case Study -- 6.4.1 Hardware Set-Up and Execution Loop -- 6.4.2 Experimentation Results -- 7 Chapter Summary and Future Work -- 7.1 Summary -- 7.1.1 Time-Domain Self-Supervised Contrastive Learning for IoT -- 7.1.2 Frequency-Domain Self-Supervised Contrastive Learning for IoT -- 7.1.3 Semi-Supervised Contrastive Learning for IoT. 7.1.4 Spectrogram Masked AutoEncoder for IoT -- 7.1.5 A Case Study: Self-Supervised Learning on IoBT-OS -- 7.2 Lessons -- 7.2.1 Self-Supervised Contrastive Learning -- 7.2.2 Masked Autoencoding -- 7.3 Future Work -- 7.3.1 Self-Supervised Learning Frameworks For Multi-Modality Inputs -- 7.3.2 Training/Inferencing Deep Neural Network Models with Noisy Data -- 7.3.3 Model Compression for Self-Supervised Learning Frameworks -- References -- 3 On the Generalization Power of Overfitted Two-Layer Neural Tangent Kernel Models -- 1 Introduction -- 2 Problem Setup -- 3 Learnable Functions and Generalization Performance -- 4 What Exactly Are the Functions in the Learnable Set? -- 4.1 A Special Case: When $d=2$ -- 5 Proof Sketch of Theorem 1 -- 6 Conclusions -- References -- 4 Out of Distribution Detection -- 1 Introduction -- 2 Related Work -- 3 Our Method: NeuralFP -- 3.1 Problem Statement -- 3.2 Motivating Example -- 3.3 Design Details -- 3.3.1 Fingerprinting on the Cloud -- 3.3.2 OOD Detection in the Edge -- 4 Experiments -- 4.1 Experimental Setup -- 4.1.1 Dataset and Model Architectures -- 4.1.2 Metrics -- 4.2 Detection Effectiveness -- 4.2.1 Detecting Statistical OOD Data -- 4.2.2 Detecting Adversarial OOD Data -- 4.2.3 Effectiveness of One-Out Integration Strategy -- 4.3 Advantageous over Previous State-of-the-Arts -- 4.4 Guidelines for Parameter Selection -- 4.5 Fingerprinting-Based Model Ranking -- 5 Conclusion -- References -- 5 Model

Compression for Edge Computing -- 1 Introduction -- 2 The Design of DeepIoT Framework -- 2.1 Dropout Operations in the Original Neural Network -- 2.2 Compressor Neural Network -- 2.3 Compressor-Critic Framework -- 3 The Evaluation of DeepIoT -- 3.1 Evaluation Platforms -- 3.2 Baseline Algorithms -- 3.2.1 Handwritten Digits Recognition with LeNet5 -- 3.3 Image Recognition with VGGNet. 3.4 Speech Recognition with Deep Bidirectional LSTM -- 3.5 Supporting Human-Centric Context Sensing -- 4 The Design of FastDeepIoT -- 4.1 Nonlinearities: Evidence and Exploitation -- 4.2 Profiling Module -- 4.2.1 Neural Network Profiling -- 4.2.2 Execution Time Model Building -- 4.2.3 Execution Time Model with Statistical Analysis -- 4.3 Compression Steering Module -- 5 The Evaluation of FastDeepIoT -- 5.1 Implementation -- 5.2 Execution Time Model -- 5.3 Compression Steering Module -- 5.3.1 Image Recognition on CIFAR-10 -- 5.3.2 Large-Scale Image Recognition on ImageNet -- 5.3.3 Heterogeneous Human Activity Recognition -- References -- Part II Distributed Problems -- 6 Communication Efficient Distributed Learning -- 1 Introduction -- 1.1 Chapter Organization -- 2 Problem Setup and Notation -- 3 Techniques for Communication-Efficient Training -- 3.1 Compression Operation -- 3.1.1 Quantization -- 3.1.2 Sparsification -- 3.1.3 Composition of Quantization and Sparsification -- 3.2 Local Iterations -- 3.3 Triggering Based Updates -- 4 Distributed Training-Qsparse-Local-SGD -- 4.1 Error Compensation -- 4.2 Theoretical Results -- 5 Decentralized Training-SQuARM-SGD -- 5.1 Theoretical Results -- 6 Experimental Results -- 6.1 Distributed Training -- 6.1.1 Setup -- 6.1.2 Results -- 6.2 Decentralized Training -- 6.2.1 Setup -- 6.2.2 Results -- 7 Other Related Works and Discussion -- References -- 7 Coreset-Based Data Reduction for Machine Learning at the Edge -- 1 Introduction -- 2 Background and Preliminaries -- 2.1 General Approaches for Learning over Distributed Data -- 2.2 Cost Function and Coreset -- 2.3 Overview of Coreset Construction Algorithms -- 3 Robust Coreset Construction -- 3.1 Centralized Construction of Robust Coreset -- 3.1.1 Motivating Experiment -- 3.1.2 The k-Clustering Problem -- 3.1.3 Coreset Construction by Optimal k-Clustering. 3.1.4 Coreset Construction by Suboptimal k-Clustering -- 3.1.5 Coreset Construction Algorithm -- 3.2 Distributed Construction of Robust Coreset -- 3.2.1 Algorithm for Distributed Robust Coreset Construction -- 3.2.2 Performance Analysis for Distributed Robust Coreset Construction -- 3.3 Performance Evaluation for Robust Coreset Construction -- 3.3.1 Experiment Setup -- 3.3.2 Experiment Results -- 4 Joint Coreset Construction and Quantization -- 4.1 Background on Coreset and Quantization -- 4.2 Preliminaries -- 4.2.1 Data Representation -- 4.2.2 Coreset Construction -- 4.2.3 Quantization -- 4.3 Budgeted Optimization of Coreset Construction and Quantization -- 4.3.1 Workflow Design -- 4.3.2 Error Bound Analysis -- 4.3.3 Configuration Optimization -- 4.4 Efficient Algorithms for MECB -- 4.4.1 Eigenvalue Decomposition Based Algorithm for MECB (EVD-MECB) -- 4.4.2 Max-Distance Based Algorithm for MECB (MD-MECB) -- 4.4.3 Discussions -- 4.5 Budget Allocation in Distributed Setting -- 4.5.1 Problem Formulation in Distributed Setting -- 4.5.2 Optimal Budget Allocation Algorithm for MECBD (OBA-MECBD) -- 4.6 Performance Evaluation -- 4.6.1 Experiment Setup -- 4.6.2 Experiment Results -- 5 Conclusion -- References -- 8 Lightweight Collaborative Perception at the Edge -- 1 Introduction -- 2 Collaboration Between Sensors and Edge Nodes -- 2.1 Understanding the 2D Scene -- 2.1.1 Opportunities for Collaboration in Multi-Camera Deployments -- 2.1.2 Lightweight State Sharing for Improved Perception -- 2.1.3 Content-Aware Collaboration: Attention and Scheduling -- 2.2 Collaboration for 3D

Sensing -- 2.2.1 V2V Lidar 3D Point Cloud State Sharing -- 2.2.2
Physical Navigation in Virtual Reality -- 2.2.3 Localisation and
Wayfinding in Robotics and Autonomous Vehicles (AV) -- 3 Cross-
Model Collaborative Execution -- 4 Conclusion -- References.
9 Dynamic Placement of Services at the Edge.
