Distributed Deep Neural Networks Over The Cloud The Edge

Distributed Computing and Artificial Intelligence, 15th International Conference
Deep Learning with PyTorch
Deep Learning with Keras
Deep Learning at Scale
Distributed Deep Learning with Apache Spark
Scalable and Distributed Machine Learning and Deep Learning Patterns
Deep Learning at Scale
ADVANCED TOPICS IN NEURAL NETWORKS WITH MATLAB. PARALLEL COMPUTING, OPTIMIZE AND TRAINING
Deep Learning with Keras
Pièces de Nivière-Chol, ci-devant maire de Lyon. Exposition et justification de sa conduite, particulièrement dans les événements de février 1793. En marge de la première pièce : Renvoyé au Comité de sûreté générale par celui des pétitions et correspondances Signé : St-Prix, président
Apache Spark Deep Learning Cookbook
Elastic Synchronization for Efficient and Effective Distributed Deep Learning
Distributed Deep Neural Networks
Hands-On Deep Learning with Apache Spark
Deep Learning for Search
Efficient Processing of Deep Neural Networks
Deep Learning with Hadoop
Collaborative Distributed Deep Learning Systems on the Edges
Next-Generation Machine Learning with Spark
Advenced Neural Networks With Matlab
Strengthening Deep Neural Networks
Learning Over and for Networks with Efficiency and Security Guarantees
Ultra-Low-Latency Distributed Deep Neural Network Over Hierarchical Mobile Networks
Advances in Distributed Computing and Machine Learning
Introduction to Deep Learning
Deep Learning with PyTorch Quick Start Guide
Communication Optimizations for Distributed Deep Learning
Advances in Distributed Computing and Machine Learning
Mastering TensorFlow 1.x
Applying Distributed Learning of Deep Neural Networks to Improve Their Classification Accuracy on Radio-Frequency Datasets
Deep Learning with R Cookbook
TensorFlow for Deep Learning
Distributed Deep Neural Networks Training for Brain Imaging Applications
Artificial Intelligence in the Age of Neural Networks and Brain Computing

Deep Learning with MATLAB
Handbook of Neuroevolution Through Erlang
Java Deep Learning Cookbook
Distributed Training of Very Large Neural Networks
Scaling up Machine Learning
Edge Computing for Distributed Deep Neural Network Inference

OCONNOR EDWARD

Distributed Computing and Artificial Intelligence, 15th International Conference Packt Publishing Ltd
Get to grips with the basics of Keras to implement fast and efficient deep-learning modelsAbout This Book* Implement various deep-learning algorithms in Keras and see how deep-learning can be used in games* See how various deep-learning models and practical use-cases can be implemented using Keras* A practical, hands-on guide with real-world examples to give you a strong foundation in KerasWho This Book Is ForIf you are a data scientist with experience in machine learning or an AI programmer with some exposure to neural networks, you will find this book a useful entry point to deep-learning with Keras. A knowledge of Python is required for this book.What You Will Learn* Optimize step-by-step functions on a large neural network using the Backpropagation Algorithm* Fine-tune a neural network to improve the quality of results* Use deep learning for image and audio processing* Use Recursive Neural Tensor Networks (RNTNs) to outperform standard word embedding in special cases* Identify problems for which Recurrent Neural Network (RNN) solutions are suitable* Explore the process required to implement Autoencoders* Evolve a deep neural network using reinforcement learningIn DetailThis book starts by introducing you to supervised learning algorithms such as simple linear regression, the classical multilayer perceptron and more sophisticated deep convolutional networks. You will also explore image processing with recognition of hand written digit images, classification of images into different categories, and advanced objects recognition with related image annotations. An example of identification of salient points for face detection is also provided. Next you will be introduced to Recurrent Networks, which are optimized for processing sequence data such as text, audio or time series. Following that, you will learn about unsupervised learning algorithms such as Autoencoders and the very popular Generative Adversarial Networks (GAN). You will also explore non-traditional uses of neural networks as Style Transfer.Finally, you will look at Reinforcement Learning and its application to AI game playing, another popular direction of research and application of neural networks.Style and approachThis book is an easy-to-follow guide full of examples and real-world applications to help you gain an in-depth understanding of Keras. This book will showcase more than twenty working Deep Neural Networks

coded in Python using Keras.

Deep Learning with PyTorch Independently Published

Over the recent years, Deep Neural Networks (DNNs) have surpassed human-level intelligence in recognizing and interpreting complex patterns in data. Ever since the ImageNet competition in 2012, Deep Learning (DL) has become a promising approach for solving numerous problems in the field of Computer Science. However, the neuroscience community is not able to utilize the DL algorithms effectively because the brain imaging datasets are huge in terms of size, and the current sequential training techniques do not scale up well for such big datasets. Without the proper amount of training data, training DNN models to competitive accuracies is quite challenging. Even with powerful GPUs or TPUs, the training performance can still be unsatisfactory if each data sample itself is large, as in the case of the brain imaging datasets. One solution is to parallelize the training process instead of training in a sequential mini-batch fashion. However, the currently available distributed training techniques suffer from several problems like computation bottleneck and model divergence. In this thesis, we discuss a novel training technique that can overcome these problems by distributing the model training across multiple GPUs on different nodes asynchronously and updating the gradients synchronously during the backward pass (backpropagation) in a Ring manner. We explore how to build such systems and train models efficiently using model replication and data parallelism techniques with very minimal changes to the existing code. We perform a comparative performance analysis of the proposed technique, training several Convolutional Neural Network (CNN) models on single-GPU, multi-GPU systems, and a Multi-node Multi-GPU cluster. Our analysis provides conclusive support that the proposed training technique can significantly out-perform the traditional sequential training approach.

Deep Learning with Keras Packt Publishing Ltd

We describe an implementation of deep learning algorithms on shared-nothing machine clusters using a distributed file system and the think-like-a-vertex programming model (the Pregel programming model). While there has been success in distributing very large neural networks across many machines in a cluster, these implementations have been limited in deployability and replication by proprietary technologies and unavailable cluster configurations. Our software, PArallel Neural Distributed Architecture (PANDA), is the first open source implementation that allows training of neural networks with millions or billions of parameters using commodity machine clusters. At its core, the neural network forward- and back-propagation are implemented in a parallel and distributed fashion using neuron centric views and message passing algorithms. This flexible and scalable approach allows for both data and model to be distributed across different machines in a cluster during training and prediction without requiring a centralized parameter server. This implementation uses the

Hadoop Distributed File System and Pregelix, an open source implementation of Pregel. *Deep Learning at Scale* IGI Global

With the increasing amount of data and the growing computing power, deep learning techniques using deep neural networks (DNNs) have been successfully applied in many practical artificial intelligence applications. The mini-batch stochastic gradient descent (SGD) algorithm and its variants are the most widely used algorithms in training deep models. The SGD algorithm is an iterative algorithm that needs to update the model parameters many times by traversing the training data, which is very time-consuming even using the single powerful GPU or TPU. Therefore, it becomes a common practice to exploit multiple processors (e.g., GPUs or TPUs) to accelerate the training process using distributed SGD. However, the iterative nature of distributed SGD requires multiple processors to iteratively communicate with each other to collaboratively update the model parameters. The intensive communication cost easily becomes the system bottleneck and limits the system scalability. In this thesis, we study the communication-efficient techniques for distributed SGD to improve the system scalability and thus accelerate the training process. We identify the performance issues in distributed SGD through benchmarking and modeling and then propose several communication optimization algorithms to address the communication issues. First, we build a performance model with a directed acyclic graph (DAG) to modeling the training process of distributed SGD and verify the model with extensive benchmarks on existing state-of-the-art deep learning frameworks including Caffe, MXNet, TensorFlow, and CNTK. Our benchmarking and modeling point out that existing optimizations for the communication problems are sub-optimal, which we need to address in this thesis. Second, to address the startup problem (due to the high latency of each communication) of layer-wise communications with wait-free backpropagation (WFBP), we propose an optimal gradient merging solution for WFBP, named MG-WFBP, that exploits the layer-wise property to well overlap the communication tasks with the computing tasks and can be adaptive to the training environments. Experiments are conducted on dense-GPU clusters with Ethernet and InfiniBand, and the results show that MG-WFBP can well address the startup problem in distributed training of layer-wise structured DNNs. Third, to make the high computing-intensive training tasks be possible in GPU clusters with low- bandwidth interconnect, we investigate the gradient compression techniques in distributed training. The top-k sparsification can well compress the communication traffic with little impact on the model convergence, but it suffers from a linear communication complexity to the number of workers so that top-k sparsification cannot scale well in large-scale clusters. To address the problem, we propose a global top-k (gTop-k) sparsification algorithm that reduces the communication complexity to be logarithmic to the number of workers. We also provide detailed theoretical analysis for the gTop-k SGD training algorithm, and the theoretical results show that our gTop-k SGD has the same order of convergence rate with SGD. Experiments are conducted on up to 64-GPU cluster to verify that gTop-k SGD significantly improves the system scalability with only a slight impact on the model convergence. Lastly, to enjoy the both benefits of the pipelining technique and the gradient sparsification algorithm, we propose a new distributed training algorithm, layer-wise adaptive gradient sparsification SGD (LAGS-SGD), which supports layer-wise sparsification and communication, and we theoretically and empirically prove that the LAGS-SGD preserves the convergence properties. To further alliterate the impact of the startup problem of layer-wise communications in LAGS-SGD, we also propose the optimal gradient merging solution for LAGS-SGD, named OMGS-SGD, and theoretical prove its optimality. The experimental results on a 16-node GPU cluster connected 1Gbps Ethernet show that OMGS-SGD can always improve the system scalability while the model convergence properties are not affected

Distributed Deep Learning with Apache Spark Packt Publishing Ltd

Speed up the design and implementation of deep learning solutions using Apache Spark Key FeaturesExplore the world of distributed deep learning with Apache SparkTrain neural networks with deep learning libraries such as BigDL and TensorFlowDevelop Spark deep learning applications to intelligently handle large and complex datasetsBook Description Deep learning is a subset of machine learning where datasets with several layers of complexity can be processed. Hands-On Deep Learning with Apache Spark addresses the sheer complexity of technical and analytical parts and the speed at which deep learning solutions can be implemented on Apache Spark. The book starts with the fundamentals of Apache Spark and deep learning. You will set up Spark for deep learning, learn principles of distributed modeling, and understand different types of neural nets. You will then implement deep learning models, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memory (LSTM) on Spark. As you progress through the book, you will gain hands-on experience of what it takes to understand the complex datasets you are dealing with. During the course of this book, you will use popular deep learning frameworks, such as TensorFlow, Deeplearning4j, and Keras to train your distributed models. By the end of this book, you'll have gained experience with the implementation of your models on a variety of use cases. What you will learnUnderstand the basics of deep learningSet up Apache Spark for deep learningUnderstand the principles of distribution modeling and different types of neural networksObtain an understanding of deep learning algorithmsDiscover textual analysis and deep learning with SparkUse popular deep learning frameworks, such as Deeplearning4j, TensorFlow, and KerasExplore popular deep learning algorithms

Who this book is for If you are a Scala developer, data scientist, or data analyst who wants to learn how to use Spark for implementing efficient deep learning models, Hands-On Deep Learning with Apache Spark is for you. Knowledge of the core machine learning concepts and some exposure to Spark will be helpful.

Scalable and Distributed Machine Learning and Deep Learning Patterns Packt Publishing Ltd

Access real-world documentation and examples for the Spark platform for building large-scale, enterprise-grade machine learning applications. The past decade has seen an astonishing series of advances in machine learning. These breakthroughs are disrupting our everyday life and making an impact across every industry. Next-Generation Machine Learning with Spark provides a gentle introduction to Spark and Spark MLlib and advances to more powerful, third-party machine learning algorithms and libraries beyond what is available in the standard Spark MLlib library. By the end of this book, you will be able to apply your knowledge to real-world use cases through dozens of practical examples and insightful explanations. What You Will Learn Be introduced to machine learning, Spark, and Spark MLlib 2.4.xAchieve lightning-fast gradient boosting on Spark with the XGBoost4J-Spark and LightGBM librariesDetect anomalies with the Isolation Forest algorithm for SparkUse the Spark NLP and Stanford CoreNLP libraries that support multiple languagesOptimize your ML workload with the Alluxio in-memory data accelerator for SparkUse GraphX and GraphFrames for Graph AnalysisPerform image recognition using convolutional neural networksUtilize the Keras framework and distributed deep learning libraries with Spark Who This Book Is For Data scientists and machine learning engineers who want to take their knowledge to the next level and use Spark and more powerful, next-generation algorithms and libraries beyond what is available in the standard Spark MLlib library; also serves as a primer for aspiring data scientists and engineers who need an introduction to machine learning, Spark, and Spark MLlib.

Deep Learning at Scale Springer Science & Business Media

Deep Learning for Search teaches readers how to leverage neural networks, NLP, and deep learning techniques to improve search performance. Deep Learning for Search teaches readers how to improve the effectiveness of your search by implementing neural network-based techniques. By the time their finished, they'll be ready to build amazing search engines that deliver the results your users need and get better as time goes on! Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications.

ADVANCED TOPICS IN NEURAL NETWORKS WITH MATLAB. PARALLEL COMPUTING, OPTIMIZE AND TRAINING "O'Reilly Media, Inc."

Artificial Intelligence in the Age of Neural Networks and Brain Computing, Second Edition demonstrates that present disruptive implications and applications of AI is a development of the unique attributes of neural networks, mainly machine learning, distributed architectures, massive parallel processing, black-box inference, intrinsic nonlinearity, and smart autonomous search engines. The book covers the major basic ideas of "brain-like computing" behind AI, provides a framework to deep learning, and launches novel and intriguing paradigms as possible future alternatives. The present success of AI-based commercial products proposed by top industry leaders, such as Google, IBM, Microsoft, Intel, and Amazon, can be interpreted using the perspective presented in this book by viewing the co-existence of a successful synergism among what is referred to as computational intelligence, natural intelligence, brain computing, and neural engineering. The new edition has been updated to include major new advances in the field, including many new chapters. Developed from the 30th anniversary of the International Neural Network Society (INNS) and the 2017 International Joint Conference on Neural Networks (IJCNN Authored by top experts, global field pioneers, and researchers working on cutting-edge applications in signal processing, speech recognition, games, adaptive control and decision-making Edited by high-level academics and researchers in intelligent systems and neural networks Includes all new chapters, including topics such as Frontiers in Recurrent Neural Network Research; Big Science, Team Science, Open Science for Neuroscience; A Model-Based Approach for Bridging Scales of Cortical Activity; A Cognitive Architecture for Object Recognition in Video; How Brain Architecture Leads to Abstract Thought; Deep Learning-Based Speech Separation and Advances in AI, Neural Networks

Deep Learning with Keras

Packt Publishing Ltd

This book presents an integrated collection of representative approaches for scaling up machine learning and data mining methods on parallel and distributed computing platforms. Demand for parallelizing learning algorithms is highly task-specific: in some settings it is driven by the enormous dataset sizes, in others by model complexity or by real-time performance requirements. Making task-appropriate algorithm and platform choices for large-scale machine learning requires understanding the benefits, trade-offs and constraints of the available options. Solutions presented in the book cover a range of parallelization platforms from FPGAs and GPUs to multi-core systems and commodity clusters, concurrent programming frameworks including CUDA, MPI, MapReduce and DryadLINQ, and learning settings (supervised, unsupervised, semi-supervised and online learning). Extensive coverage of parallelization of boosted trees, SVMs, spectral clustering, belief propagation and other popular learning algorithms, and deep dives into several applications, make the book equally useful for researchers, students and practitioners.

Pièces de Nivière-Chol, ci-devant maire de Lyon. Exposition et justification de sa conduite, particulièrement dans les événements de février 1793. En marge de la première pièce : Renvoyé au Comité de sûreté générale par celui des pétitions et

correspondances Signé : St-Prix, président Packt Publishing Ltd
The 15th International Symposium on Distributed Computing and Artificial Intelligence 2018 (DCAI 2018) is a forum to present applications of innovative techniques for studying and solving complex problems. The exchange of ideas between scientists and technicians from both the academic and industrial sector is essential to facilitate the development of systems that can meet the ever-increasing demands of today’s society. The present edition brings together past experience, current work and promising future trends associated with distributed computing, artificial intelligence and their application in order to provide efficient solutions to real problems. This symposium is organized by the University of Castilla-La Mancha, the Osaka Institute of Technology and the University of Salamanca. The present edition was held in Toledo, Spain, from 20th – 22nd June, 2018.

Apache Spark Deep Learning Cookbook

Springer

As deep neural networks (DNNs) become increasingly common in real-world applications, the potential to deliberately "fool" them with data that wouldn’t trick a human presents a new attack vector. This practical book examines real-world scenarios where DNNs—the algorithms intrinsic to much of AI—are used daily to process image, audio, and video data. Author Katy Warr considers attack motivations, the risks posed by this adversarial input, and methods for increasing AI robustness to these attacks. If you’re a data scientist developing DNN algorithms, a security architect interested in how to make AI systems more resilient to attack, or someone fascinated by the differences between artificial and biological perception, this book is for you. Delve into DNNs and discover how they could be tricked by adversarial input Investigate methods used to generate adversarial input capable of fooling DNNs Explore real-world scenarios and model the adversarial threat Evaluate neural network robustness; learn methods to increase resilience of AI systems to adversarial data Examine some ways in which AI might become better at mimicking human perception in years to come

Elastic Synchronization for Efficient and Effective Distributed Deep Learning

Springer Nature

The treatment of large data requires the use of computational structures that implement parallelism and distributed computing. The Big Data structures are responsible for providing these characteristics to computing. The treatment of large data requires the use of computational structures that implement parallelism and distributed computing. The Big Data structures are responsible for providing these characteristics to computing. You can train a convolutional neural network (CNN, ConvNet) or long short-term memory networks (LSTM or BiLSTM networks) using the trainNetwork function. You can choose the execution environment (CPU, GPU, multi-GPU, and parallel) using trainingOptions. Training in parallel, or on a GPU, requires Parallel Computing Toolbox.Neural networks are inherently parallel algorithms. Multicore CPUs, graphical processing units (GPUs), and clusters of computers with multiple CPUs and GPUs can take advantage of this parallelism. Parallel Computing Toolbox, when used in conjunction with Deep Learning Toolbox, enables neural network training and simulation to take advantage of each mode of parallelism. Distributed and GPU computing can be combined to run calculations across multiple CPUs and/or GPUs on a single computer, or on a cluster with MATLAB Distributed Computing Server.Parallel Computing Toolbox allows neural network training and simulation to run acrossmultiple CPU cores on a single PC, or across multiple CPUs on multiple computers on anetwork using MATLAB Distributed Computing Server.Using multiple cores can speed calculations. Using multiple computers can allow you tosolve problems using data sets too big to fi in the RAM of a single computer. The onlylimit to problem size is the total quantity of RAM available across all computers.To manage cluster configurations use the Cluster Profil Manager. You can train a convolutional neural network (CNN, ConvNet) or long short-term memory networks (LSTM or BiLSTM networks) using the trainNetwork function. You can choose the execution environment (CPU, GPU, multi-GPU, and parallel) using trainingOptions. Training in parallel, or on a GPU, requires Parallel Computing Toolbox.Neural networks are inherently parallel algorithms. Multicore CPUs, graphical processing units (GPUs), and clusters of computers with multiple CPUs and GPUs can take advantage of this parallelism. Parallel Computing Toolbox, when used in conjunction with Deep Learning Toolbox, enables neural network training and simulation to take advantage of each mode of parallelism. Distributed and GPU computing can be combined to run calculations across multiple CPUs and/or GPUs on a single computer, or on a cluster with MATLAB Distributed Computing Server. Parallel Computing Toolbox allows neural network training and simulation to run acrossmultiple CPU cores on a single PC, or across multiple CPUs on multiple computers on anetwork using MATLAB Distributed Computing Server.Using multiple cores can speed calculations. Using multiple computers can allow you tosolve problems using data sets too big to fi in the RAM of a single computer. The onlylimit to problem size is the total quantity of RAM available across all computers.To manage cluster configurations use the Cluster Profil Manager.

Distributed Deep Neural Networks Packt Publishing Ltd

Deep learning has revolutionized a wide range of fields. In spite of its success, most deep learning systems are proposed in the cloud, where data are processed in a centralized manner with abundant compute and network resources. This raises a problem when deep learning is deployed on the edge where distributed compute resources are limited. In this dissertation, we propose three distributed systems to enable collaborative deep learning on the edge. These three systems target different scenarios and tasks. The first system dubbed Distream is a distributed live video analytics

system based on the smart camera-edge cluster architecture. Distream fully utilizes the compute resources at both ends to achieve optimized system performance. The second system dubbed Mercury is a system that addresses the key bottleneck of collaborative learning. Mercury enhances the training efficiency of on-device collaborative learning without compromising the accuracies of the trained models. The third system dubbed FedAce is a distributed training system that improves training efficiency under federated learning setting where private on-device data are not allowed to be shared among local devices. Within each participating client, FedAce achieves such improvement by prioritizing important data. In the server where model aggregation is performed, FedAce exploits the client importance and prioritizes important clients to reduce stragglers and reduce the total number of rounds. In addition, FedAce conducts federated model compression to reduce the per-round communication cost and obtains a compact model after training completes. Extensive experiments show that the proposed three systems are able to achieve significant improvements over status-quo systems.
Hands-On Deep Learning with Apache Spark Packt Publishing Ltd
Build, implement and scale distributed deep learning models for large-scale datasets About This Book Get to grips with the deep learning concepts and set up Hadoop to put them to use Implement and parallelize deep learning models on Hadoop's YARN framework A comprehensive tutorial to distributed deep learning with Hadoop Who This Book Is For If you are a data scientist who wants to learn how to perform deep learning on Hadoop, this is the book for you. Knowledge of the basic machine learning concepts and some understanding of Hadoop is required to make the best use of this book. What You Will Learn Explore Deep Learning and various models associated with it Understand the challenges of implementing distributed deep learning with Hadoop and how to overcome it Implement Convolutional Neural Network (CNN) with deeplearning4j Delve into the implementation of Restricted Boltzmann Machines (RBM) Understand the mathematical explanation for implementing Recurrent Neural Networks (RNN) Get hands on practice of deep learning and their implementation with Hadoop. In Detail This book will teach you how to deploy large-scale dataset in deep neural networks with Hadoop for optimal performance. Starting with understanding what deep learning is, and what the various models associated with deep neural networks are, this book will then show you how to set up the Hadoop environment for deep learning. In this book, you will also learn how to overcome the challenges that you face while implementing distributed deep learning with large-scale unstructured datasets. The book will also show you how you can implement and parallelize the widely used deep learning models such as Deep Belief Networks, Convolutional Neural Networks, Recurrent Neural Networks,

Restricted Boltzmann Machines and autoencoder using the popular deep learning library deeplearning4j. Get in-depth mathematical explanations and visual representations to help you understand the design and implementations of Recurrent Neural network and Denoising AutoEncoders with deeplearning4j. To give you a more practical perspective, the book will also teach you the implementation of large-scale video processing, image processing and natural language processing on Hadoop. By the end of this book, you will know how to deploy various deep neural networks in distributed systems using Hadoop. Style and approach This book takes a comprehensive, step-by-step approach to implement efficient deep learning models on Hadoop. It starts from the basics and builds the readers' knowledge as they strengthen their understanding of the concepts. Practical examples are included in every step of the way to supplement the theory.

Deep Learning for Search

O'Reilly Media

Handbook of Neuroevolution Through Erlang presents both the theory behind, and the methodology of, developing a neuroevolutionary-based computational intelligence system using Erlang. With a foreword written by Joe Armstrong, this handbook offers an extensive tutorial for creating a state of the art Topology and Weight Evolving Artificial Neural Network (TWEANN) platform. In a step-by-step format, the reader is guided from a single simulated neuron to a complete system. By following these steps, the reader will be able to use novel technology to build a TWEANN system, which can be applied to Artificial Life simulation, and Forex trading. Because of Erlang's architecture, it perfectly matches that of evolutionary and neurocomptational systems. As a programming language, it is a concurrent, message passing paradigm which allows the developers to make full use of the multi-core & multi-cpu systems. Handbook of Neuroevolution Through Erlang explains how to leverage Erlang's features in the field of machine learning, and the system's real world applications, ranging from algorithmic financial trading to artificial life and robotics.

Efficient Processing of Deep Neural Networks

"O'Reilly Media, Inc."

This thesis aims to improve on the current classification capabilities of deep neural networks on two types of radio-frequency data: radar and OFDM packets. In radar, applying neu- ral networks to Automatic Target Recognition problems is a well-developed field, especially using the MSTAR database. However, existing state-of-the-art methods require precise pre- conditioning of radar data and are unsuited to applications with a large number of radar target classes. Therefore, we asked whether distributed learning can increase the generaliz- ability and scalability of neural networks in these tasks. To test this, we applied distributed learning via Multi-Stage Training and a new network architecture, the Convolutional Multi-Stage Network, to provide a scalable, generalized treatment of radar data for more practical applications. This method was shown to outperform traditional neural network architectures on a new radar dataset. A similar approach was applied to the OFDM data with the goal of identifying specific radio-frequency transmitters for network security purposes. The task of identifying OFDM packet transmitters has previously been performed successfully, though with precise data collection methods. Data collection methods on a live network will likely include imperfect recording times, so we sought to improve network robustness to time- shifted OFDM packets. It was shown that the Convolutional Multi-Stage Network improved robustness to time-shifting of the radio-frequency data over the Multi-Stage Network, which was the previous-best method. Simple preconditioning of the data using variations of the dis- crete wavelet transform further improved robustness to time-shifting of the radio-frequency data using both network architectures. These results are significant, as they provide a new avenue for applying neural networks to radio-frequency in difficult, real-world applications.

Deep Learning with Hadoop Apress

"Deep learning is a subfield of Artificial Intelligence and Machine Learning where a huge amount of data is processed in complex layers of neural networks. It has solved tons of interesting real-world problems in recent years. Distributed deep learning (DL) involves training a deep neural network in parallel across multiple machines. In this course, you will get started with implementing Deep Learning solutions easily with the help of Apache Spark. You will begin with a short introduction on Deep Learning and Apache Spark and the principles of distributed modeling. With the help of real-world examples, you will investigate different types of neural network and work with DL libraries such as BigDL, Deeplearning4j, and the Deep Learning pipelines library to implement DL models and distributed computing on Spark. You will see how you can easily use a large dataset to implement efficient DL solutions to simplify real-world examples. You will also learn how to distribute the computationally heavy parts of DL into processes with the help of Apache Spark. By the end of this course, you'll have gained experience in implementing Distributed Deep Learning for your models at work. Our examples will be based on real-world problems from the banking industry."--Resource description page.

Collaborative Distributed Deep Learning Systems on the Edges Packt Publishing Ltd

Training deep neural networks (DNNs) using a large-scale cluster with an efficient distributed paradigm significantly reduces the training time. However, a distributed paradigm developed only from system engineering perspective is most likely to hindering the model from learning due to the intrinsic optimization properties of machine learning. In this thesis, we present two efficient and effective models in the parameter server setting based on the limitations of the state-of-the-art distributed models such as staleness synchronous parallel (SSP) and bulk synchronous parallel (BSP). We introduce DynamicSSP model that adds smart dynamic communication to SSP, improves its communication efficiency and replaces its fixed staleness threshold with a dynamic threshold. DynamicSSP converges faster and to a higher accuracy than SSP in the heterogeneous environment. Having recognized the importance of bulk synchronization in training, we propose the ElasticBSP model which shares the proprieties of bulk synchronization and elastic synchronization. We develop fast online optimization algorithms with look-ahead mechanisms to materialise ElasticBSP. Empirically, ElasticBSP achieves the convergence speed 1.77 times faster and an overall accuracy 12.6% higher than BSP.

Next-Generation Machine Learning with Spark Cambridge University Press

Build neural network models in text, vision and advanced analytics using PyTorch Key Features Learn PyTorch for implementing cutting-edge deep learning algorithms. Train your neural networks for higher speed and flexibility and learn how to implement them in various scenarios; Cover various advanced neural network architecture such as ResNet, Inception, DenseNet and more with practical examples; Book Description Deep learning powers the most intelligent systems in the world, such as Google Voice, Siri, and Alexa. Advancements in powerful hardware, such as GPUs, software frameworks such as PyTorch, Keras, Tensorflow, and CNTK along with the availability of big data have made it easier to implement solutions to problems in the areas of text, vision, and advanced analytics. This book will get you up and running with one of the most cutting-edge deep learning libraries—PyTorch. PyTorch is grabbing the attention of deep learning researchers and data science professionals due to its accessibility, efficiency and being more native to Python way of development. You'll start off by installing PyTorch, then quickly move on to learn various fundamental blocks that power modern deep learning. You will also learn how to use CNN, RNN, LSTM and other networks to solve real-world problems. This book explains the concepts of various state-of-the-art deep learning architectures, such as ResNet, DenseNet, Inception, and Seq2Seq, without diving deep into the math behind them. You will also learn about GPU computing during the course of the book. You will see how to train a model with PyTorch and dive into complex neural networks such as generative networks for producing text and images. By the end of the book, you'll be able to implement deep learning applications in PyTorch with ease. What you will learn Use PyTorch for GPU-accelerated tensor computations Build custom datasets and data loaders for images and test the models using torchvision and torchtext Build an image classifier by implementing CNN architectures using PyTorch Build systems that do text classification and language modeling using RNN, LSTM, and GRU Learn advanced CNN architectures such as ResNet, Inception, Densenet, and learn how to use them for transfer learning Learn how to mix multiple models for a powerful ensemble model Generate new images using GAN’s and generate artistic images using style transfer Who this book is for This book is for machine learning engineers, data analysts, data scientists interested in deep learning and are looking to explore implementing advanced algorithms in PyTorch. Some knowledge of machine learning is helpful but not a mandatory need. Working knowledge of Python programming is expected.

Advenced Neural Networks With Matlab

Springer Nature

Bringing a deep-learning project into production at scale is quite challenging. To successfully scale your project, a foundational understanding of full stack deep learning, including the knowledge that lies at the intersection of hardware, software, data, and algorithms, is required. This book illustrates complex concepts of full stack deep learning and reinforces them through hands-on exercises to arm you with tools and techniques to scale your project. A scaling effort is only beneficial when it's effective and efficient. To that end, this guide explains the intricate concepts and techniques that will help you scale effectively and efficiently. You'll gain a thorough understanding of: How data flows through the deep-learning network and the role the computation graphs play in building your model How accelerated computing speeds up your training and how best you can utilize the resources at your disposal How to train your model using distributed training paradigms, i.e., data, model, and pipeline parallelism How to leverage PyTorch ecosystems in conjunction with NVIDIA libraries and Triton to scale your model training Debugging, monitoring, and investigating the undesirable bottlenecks that slow down your model training How to expedite the training lifecycle and streamline your feedback loop to iterate model development A set of data tricks and techniques and how to apply them to scale your training model How to select the right tools and techniques for your deep-learning project Options for managing the compute infrastructure when running at scale

Related with Distributed Deep Neural Networks Over The Cloud The Edge:

• The Young Riders Episode Guide : click here