Energy Optimization for Deep Learning in Edge Computing: An Overview

# INTRODUCTION

Edge computing

Deep learning

Energy concerns, optimization and challenges

# BACKGROUND

## Background on Edge computing

## Background on Deep Learning

# Industrial Applications (Energy-efficient DEEP LEARNING in Edge)

There are many existing studies on applying energy-efficient deep learning techniques in edge computing. They mainly focus on the following industrial applications.

## Internet of vehicles (6)

Vehicular edge computing (VEC) is the application of mobile edge computing (MEC) in vehicular scenarios, and it has received great attention.

Wang *et al.* [1] consider an increasingly prominent challenge of meeting communication and computational needs of vehicles with the emergence of vehicular applications. Fog computing improves the performance of vehicular services by using computational offloading at the network edge. They design a fog-cloud computational offloading method in Internet of Vehicles (IoVs) for minimizing both the power consumption of vehicles and that of computational facilities. An offloading problem is formulated as an NP-hard one, and solved by a heuristic algorithm gradually. Specifically, a predictive combination transmission mode is designed for vehicles, and a deep learning model is establishes for computational facilities for determining the optimal allocation of workload.

Ning *et al*. [75] consider the challenge of how to meet quality of experiences of users in intelligent networks with limited computing abilities of vehicular fog nodes. Fog computing infrastructure is deployed closely to terminals, and provides updated communication and computing platforms to emerging vehicular services. They develop a three-layer offloading framework for Internet of Vehicles (IoV) to minimize their total energy consumption while meeting delay constraints of users. Considering the high computational complexity, an optimization problem is formulated and decomposed into two parts including flow redirection and offloading decision. Then, a deep reinforcement learning-based mechanism is proposed to solve it. Real-world data-based evaluations show that the average energy consumption is reduced by 60% compared with the baseline algorithm.

Jiang *et al*. [19] consider a hybrid mobile edge computing (HMEC) platform including ground vehicles (GVs), ground stations (GSs) and unmanned aerial vehicle (UAVs). They are connected with mobile edge cloud that enables Internet of thing (IoT) devices or user equipments (UEs) to offload intensive computing tasks. It aims to develop an online offloading algorithm to achieve the energy consumption minimization for all UEs by optimizing positions of UAVs and GVs, user association and resource allocation in a dynamic environment. Then, a hybrid deep learning-based online offloading framework is designed and it applies a large-scale path-loss fuzzy c-means algorithm to predict the optimal positions of GVs and UAVs. A fuzzy membership matrix U-based particle swarm optimization algorithm is adopted to produce sample datasets for a deep neural network (DNN). Then, a DNN with a scheduling layer is developed to obtain computing resource allocation and user association while meeting practical latency needs of tasks with limited computing resources and energy of HMEC.

Lammie et al. [76] consider the challenging problem of the robust and efficient detection of weed species in robotic weed control technologies. Deep neural networks (DNNs) have demonstrated remarkable performance for plant classification. Training DNNs on graphics processing units (GPUs) provides higher levels of performance; however, GPUs consume large power. The field programmable gate array (FPGA)-based DNNs have many advantages in energy efficiency compared with traditional GPU- and CPU-accelerated networks. DNNs networks that are properly designed and customized on GPUs and FPGAs, are ideal candidates for inference and learning in resource-constrained and power-limited portable devices, e.g., robots and edge devices in Internet of Things (IoT). They adopt GPU- and FPGA-accelerated DNNs that are deterministically binarized for weed species classification in robotic weed control. Their results show that the FPGA-accelerated binarized networks dramatically outperform the GPU-accelerated ones in terms of power consumption reduction and weed image inference time. It provides a significant step for deep learning and inference on IoT edge devices, and portable machines like agricultural robots.

The fast increase of data processing needs from users in MEC, the traditional mobile edge servers (MESs) fail to provide effective and timely services. To solve it, Li *et al.* [37] tries to adopt unmanned aerial vehicle (UAV) as an MES that provides offloading of computational tasks for users. They aim to maximize the migration throughput for users’ tasks with UAV that only has limited energy. A maximization problem is formulated as a semi-Markov decision process without transition probability. Then, a deep reinforcement learning (DRL)-based scheme is proposed to maximize the migration throughput of user tasks. It achieves a maximum autonomic migration throughput for users’ tasks with limited UAV energy and improves quality of service of MEC.

Zhan *et al.* [81] consider a computation offloading scheduling problem in a VEC scenario, where a vehicular terminal (VT) travels along an expressway, and makes the scheduling decision for tasks waiting in their task queue. These tasks are independently produced by different applications, and therefore they have heterogeneous characteristics in terms of data size and computation-to-volume ratio. They lead to dynamical changes of data transmission time, energy consumption and transmission failures. Servers in MEC are equipped in roadside units (RSUs) are executed to perform computation for the VT. The resource-demanding tasks can be offloaded to MEC servers in RSUs for significantly reducing the energy consumption and execution latency of in-vehicle applications. They consider a key challenge of how to achieve a trade-off between energy consumption and task execution latency. To tackle it, a DRL-based offloading approach is designed to minimize the long-term cost in terms of a trade-off between task latency and energy consumption. Their DRL implementation is proposed according to a proximal policy optimization algorithm. It combines a parameter-shared network architecture with a convolutional neural network to approximate both value and policy functions for effective representative feature extraction.

## Microgrid (5)

Munir *et al.* [3] consider a problem of risk-sensitive microgrid energy profiling for a MEC network. It considers conditional value-at-risk and determines the predicted energy shortfall risk by using the coordination with uncertainties of both supply and demand. A multi-agent system is designed to specify an optimal scheduling strategy for the agents. Then, a multi-agent deep reinforcement learning based on asynchronous advantage actor-critic is adopted to mitigate the dimensionality curse and design the optimal energy profile among agents. Their results show that the proposed model realizes high-accuracy energy profiling than a single agent solution. Munir *et al.* [39] propose an energy supply plan for MEC networks supported by microgrid. An energy consumption minimization problem is formulated for microgrid-enabled MEC networks. It is a mixed integer nonlinear optimization one with tasks’ latency and computational constraints, and it also considers the uncertainty of both energy generation and consumption. It is decomposed into two subproblems including energy-efficient tasks allocation and energy supply plan. In addition, a density-based spatial application clustering is applied to solve the first one for each base station, and a model-based deep reinforcement learning is adopted to the second one. Naderializadeh and Hashemi [66] consider a computation offloading problem in a MEC architecture, and many energy-constrained users simultaneously offload their tasks to servers with a shared wireless medium. A multi-agent deep reinforcement learning method is proposed where an agent is designed for each server and it observes its associated users’ status and chooses the optimal offloaded user in each step. The task completion time and system lifetime are selected as two key performance factors, and their results prove that the proposed method achieves better performance than baseline algorithms. Khan *et al.* [95] introduce the factors that affect the selection of microgrids as major electrical grids, and gives benefits of microgrids. In addition, they consider the problems that hinder benefits brought by distributed energy production in microgrids, and then design an architecture based on artificial intelligence (AI) to address these challenges. Furthermore, a simulation framework is designed and useful data is adopted to build AI capabilities within energy utilities. In addition, a scalable framework that applies deep learning techniques is also implemented. Based on it, the AI inference at nodes and sensors in edge is realized to optimize the benefits brought by microgrids at different scenarios including community, enterprise and campus levels in smart cities.

## Computer Vision

Luo *et al.* [10] jointly optimize quality of experience (QoE) and energy consumption for video streaming in software-defined mobile networks. Specifically, a mechanism is proposed to jointly consider video quality adaption, buffer dynamics, video transcoding, edge caching and transmission. The time-varying channel is assumed as a discrete-time Markov chain, based on which two optimization problems are formulated as a Markov decision process (MDP) and a constrained MDP. A Lyapunov technique is adopted to transform a constrained MDP into regular one, which is further solved by an asynchronous advantage actor-critic algorithm. Then, the energy saving is achieved while QoE is also enhanced. Xu *et al*. [55] present an efficient CNN to reconstruct speckle image in cloud-edge computing for better image resolution with fewer inputs. A self-back stacked efficient residual factorized network is designed to reconstruct image through scattering medium. It includes two training stages, and the model is used for analyzing speckle image from low resolution to high one. Results show that a high resolution is achieved even if there are a small input samples. Lim *et al*. [38] propose an energy-efficient communication method in edge computing with deep learning, and it decreases power consumed by image transmission using edge computing. An energy-efficient IoT camera called CamThings is implemented by using the proposed communication and periodic on-off scheduling. CamThings performs better than the method that only adopts periodic on-off scheduling with respect to lifetime and power consumption. Zhang *et al*. [54] propose an offloading prediction algorithm to minimize real-time transmission of images. It predicts the future need for deep learning of each unmanned aerial vehicle (UAV), and transmits images only when necessary. Holistic allocation of resources is determined at edge according to the likelihood analysis of offloading for multiple UAVs. Monburinon *et al*. [68] present a hierarchical image recognition system based on edge computing, and its major processing is implemented at Raspberry Pi. A dynamic learning method is implemented and a convolutional neural network is trained to achieve recognition of animals in a specific environment. The recognition module is deployed in edge servers on gateway devices for performing offline image classification.

## Mobile Edge Computing (7)

Jin *et al*. [17] investigate a multi-user MEC system and propose computation offloading and resource allocation policies with the objective of minimization of energy consumption and service delay in a dynamic environment. An optimization framework is proposed based on deep reinforcement learning to maximize long-term cumulative rewards. Zhu *et al*. [44] propose a computation offloading mechanism to decrease completion time of applications and energy consumed by user devices. The formulated computation offloading problem is transformed into a time and energy optimization one. The optimal cost strategy is obtained with deep Q-learning. It outperforms local execution and random offloading with respect to energy consumption and completion time of service workflows.

Wang *et al*. [18] propose a DRL-based offloading framework to solve problems of task adaption and dependency in dynamic scenarios. It well learns an offloading policy represented by a sequence-to-sequence neural network. The offloading policy is inferred by automatically finding common patterns in different applications in various scenarios. Li *et al*. [23] investigate a multi-user MEC system in which many user equipments (UEs) realize computation offloading through wireless channels connected to an MEC server. The weighted cost of energy consumption and delay for UEs is formulated as an optimization objective. The offloading and allocation of computational resources are jointly optimized in an MEC system. A reinforcement Learning-based optimization framework is proposed by adopting Q-learning schemes. Zhang *et al*. [72] design an offloading framework for a network architecture including an MEC server and a mobile user based on deep reinforcement learning. The task flow offloading process is modeled as a Markov decision process. It aims to minimize the weighted sum of power consumption and offloading latency, which is transformed into the reward in each time slot. Yang *et al*. [30] consider a problem of joint minimization of energy and latency for hierarchical machine learning task distribution in mobile edge computing. The shallow neural network models are embedded in mobile devices in the framework. The computing-intensive and latency-sensitive tasks are offloaded to a nearby MEC server, which supports a deep neural network model. A piecewise convex optimization problem is formulated to minimize the weighted-sum of energy and latency. Then, a closed-form solution for an optimal strategy of partial offloading is obtained analytically. Dong *et al*. [21] investigate a mobile edge computing system supporting both low-latency and ultra-reliable communication services and delay tolerant ones. The normalized energy consumption is minimized by optimizing resource allocation, user association and offloading probabilities while meeting quality-of-service needs. A deep learning (DL) architecture is proposed and it is trained in a central server. An optimization algorithm is proposed to obtain the optimal offloading and resource allocation.

## Smart Grid (5)

Cheung *et al*. [16] design an unsupervised disaggregation model for the disaggregation of solar production from measurements of advanced metering infrastructures (AMI) without training data. The model only needs the inputs of AMI measurements from users in a regional area and solar irradiance. Then, users’ consumption is modeled by neighboring households and it does not need rooftop photovoltaics for achieving the disaggregation. Li *et al*. [69] develop a home energy management system in a demand side management (DSM) program and it is equipped with an edge computing server. It maximizes the expected total reward of a home owner, which is obtained by calculating the difference between reward of edge computing tasks and the sum of electricity cost, the computation offloading cost and the violating penalty of DSM needs. Then, the deep deterministic policy gradient is adopted to solve long-term temporal interdependency and high-dimensional state space in their formulated MDPs. Sirojan *et al*. [77] present a deep learning-based sustainable method for an edge device and it can be used on top of a power pole for detecting high impedance faults in a real-time manner. In the embedded edge node, larger throughput, less latency and network traffic offloading are achieved by using feature extraction, data acquisition and deep learning based fault identification. In addition, hardware pipelining and parallelism are used to realize the real-time fault identification on edge nodes, and guarantee efficient usage of resources. Zhang *et al*. [85] adopt a deep learning method named deep stacked autoencoder (SAE) to discover anomalies in physical system measurements. The proposed unsupervised method is evaluated to discover anomalies and investigate root cause analysis with an end-to-end transactive energy system testbed. It provides a transactive control mechanism for energy production of a large number of devices in edge computing and Internet of Things (IoT). Wang *et al*. [2] propose a model based on MobileNets convolutional neural network to discover the patterns of gas-insulated switchgear partial discharge (PD). The PD pattern datasets for recognition and classification are constructed by using FDTD simulation. After the MCNN model is obtained, an inverse residual structure and depthwise separable convolutions are used to address the vanishing gradient of deep convolutional neural networks in the recognition of PD patterns. The model improves the recognition performance of the gas-insulated switchgear PD, and it can also be incorporated into Ubiquitous Power Internet of Things where intelligent terminals are enabled by edge computing in embedded systems.

## Healthcare system (5)

There have been a number of research efforts devoted on this topic. Hartmann et al. [58] introduce classification techniques of healthcare applications, specifically for electrocardiogram (ECG) beat, in emerging edge computing. These classification techniques are implemented on a platform based on Raspberry Pi. A performance comparison of classification techniques is given in terms of three key performance indicators including latency, energy efficiency and accuracy, for health care applications. Reddy *et al*. [59] implement a voice disorder detection system with a deep learning method. A patient collects his/her voice samples apprehended by smart sensors, and these samples are sent to edge computing for initial transformation. Then, the produced knowledge is transmitted through edge to core cloud for further transformation. The analysis is realized by a service provider running in a cloud manager. Finally, automatic analysis is given and the result is sent back to a consultant, who gives the optimal therapy to patients. Granados *et al*. [52] investigate the balance between performance, latency and power consumption among gateway, cloud, fog and edge layers in an IoT medical platform. An IoT architecture is implemented to classify multichannel electrocardiogram (ECG) signals into abnormal or normal states with deep learning models. Then, a clinically relevant condition is represented by combining users’ embedded devices with current machine learning packages, *e.g.*, TensorFlow. Different hardware platforms are evaluated to obtain the optimal compromise with respect to power consumption, latency and convenience.

In addition, Naeini *et al*. [96] propose a self-aware and smart system for the continuous monitoring of pain intensity at an edge layer, and it can dynamically make decisions at run-time for the pain level changes. Nased on the dataset of BioVid heat pain, the proposed method significantly reduces energy consumption with a negligible loss of accuracy compared to its non-adaptive peer. It can further provide objective and accurate pain assessment, which is a key for effective management pain. In this way, the energy consumption is minimized by adaptively offloading tasks to gateway devices in the edge layer. Tuli *et al*. [97] present a framework named HealthFog to apply ensemble deep learning in devices of edge computing, and run it for a real-world application to realize the automatic analysis of heart disease. It provides healthcare as a service by using IoT nodes and efficiently manages the heart data of patients. A Fog-supported cloud framework is adopted to evaluate its performance with respect to network bandwidth, power consumption, jitter, accuracy, execution time and latency.

## Smart Cities (5)

Ke *et al*. [43] propose a renewable energy aware and data offloading model and it investigates an MEC server in a dynamic IoT environment. The server executes different stochastic tasks and involves many time-changed wireless channels. A joint method is proposed for optimizing bandwidth allocation, data transmission delay, data offloading and consumption of renewable energy for IoT nodes according to a deep reinforcement learning method. It can cope with a continuous action space and avoid the dimensionality curse due to the action space complexity. Liu *et al*. [46] adopt deep reinforcement learning to realize an energy management system for an IoT-based edge computing infrastructure. Then, they give an introduction of IoT-based management of energy in smart cities. Besides, the software model and framework designed for the edge computing system are described. Furthermore, an energy scheduling method based on deep reinforcement learning is presented for the framework and evaluated. It can efficiently use green energy management systems in explosively developed smart cities. Zhang *et al*. [47] design an assessment method for urban street cleanliness by using deep learning and mobile edge computing. High-resolution cameras run in vehicles and obtain images of the street. The servers in mobile edge are adopted to store the information of street image. Then, the street data is delivered to a cloud data center for further processing and analysis by using city networks. In addition, the faster region convolutional neural network is adopted to determine the categories of street garbage and calculate the number of garbage. Finally, the proposed method is integrated into a framework of the calculation of street cleanliness. It visualizes the levels of street cleanliness, and helps city managers to effectively schedule clean-up personnel. Luong *et al*. [82] adopt deep learning and propose an optimal auction for the allocation of edge resources. A multi-layer neural network architecture is proposed according to the analytical solution for the obtained optimal auction. The monotone transformations of bids of the miners are performed by neural networks. Then, the conditional payment rules are calculated for the miners. Then, the valuations of the miners are used as the training data to change parameters of neural networks to achieve the loss function minimization. It derives the optimal auction that leads to high revenue for mobile blockchain. Lee *et al*. [98] design a system to predict the consumption of energy by using a deep learning algorithm in an edge environment. The proposed system is applied in an office environment by creating a testbed. The long short-term memory network is adopted to demonstrate its high prediction accuracy for time series energy data in each day. It greatly improves the sustainability and sophistication of smart cities for governments around the world.

# DNN-based mechanisms for energy optimization

## DNN compression (5)

DNN compression is an emerging method to decrease the network complexity. Han *et al*. [99] propose a 3-step algorithm including pruning, quantization and encoding to decrease the memory consumption of a DNN. Structured pruning [100] adopt constraints on some parameters of DNNs to keep a certain structure. Another method aims to prune the least significant and redundant weight parameters of DNNs [101], and weights are shared to decrease the dimensionality. Other compression approaches are designed by using knowledge transfer [102], low-rank approximations [103] and variational dropout [104]. In addition, Yan and Pei [56] develop a framework to produce a robust compressed model based on deep convolutional neural network in an edge environment. The obtained model is then partitioned and further trained in mobile devices and an edge server, respectively. The compressed model is robust and it is mainly obtained by using model robustness and compression. First, a defensive mechanism is designed in model robustness, and it improves the robustness of the compressed model compared with adversarial examples. In addition, its weight distribution is further considered to improve its accuracy. The compressed model is small-scale yet robust, and it can be deployed in mobile devices for recognition tasks. Then, the collaborative inference between devices and servers is achieved while providing strong robustness and high accuracy. Lu *et al*. [62] adopt the predictive ability of machine learning, and implement a device-level engine for the selection of DNN models and quality-of-experience-optimal inference in edge. A problem of DNN model selection is formulated in a multi-armed contextual bandit framework. The DNN models and features of edge nodes are contexts and pre-trained models are arms selected according to the historic information of actions and users' quality of experience. An efficient online learning algorithm is developed to achieve a balance between exploration and exploitation. Wu *et al*. [80] present an energy-efficient accelerator for sparse compressed convolutional neural networks (CNNs) by decreasing accesses of DRAM and removing zero-operand computation. The compression of weights is adopted for sparse compressed CNNs to decrease the needed and memory. Therefore, zero-valued activations are produced by ReLU functions. In addition, workloads are allocated according to channels to enhance the task parallelism degree, and all-row-to-all-row multiplication of non-zero elements is utlized for avoiding redundant computation.

Young *et al*. [84] show that the automated design of deep neural networks is attracting a growing amount of attention, and it creates the different networks that contain significantly different characteristics of computations, *e.g.*, model size, energy usage and inference time, especially as the number of deep learning applications increases. Then, it provides a chance to utilize this hyperparameter optimization process to make deep neural networks energy efficient, and they are easy to be deployed in smaller devices. Marchisio *et al*. [20] point that deep neural networks have unmatchable performance in different applications, *e.g.*, computer vision, natural language processing and image processing. However, their energy consumption is becoming a difficult problem as deep neural networks increase in complexity because the edge devices are usually resource-limited and energy- limited. Consequently, specialized optimizations, *e.g.*, compression techniques, are highly needed for deep learning and have to be implemented in both hardware and software. The current trends of these optimization algorithms are comprehensively surveyed and important open research challenges are also discussed. Zhang *et al*. [4] give a performance comparison of many widely used machine learning packages designed for devices in edge computing, *e.g.*, TensorFlow, Caffe2, MXNet, PyTorch, and TensorFlow Lite. They mainly evaluate memory footprint, latency and energy of these packages by using two types of neural networks on different devices in edge. The evaluation gives a reference to choose suitable combinations of software and hardware packages for end users, and also show potential future directions to further optimize packages.

## DNN partitioning (5)

Shi *et al*. [35] discuss the impact of DNN partitioning on the latency of inference and the risks of privacy in edge computing. According to the obtained insights, an offloading method is designed to partition the DNN adaptively in different network environments. In this way, an optimal tradeoff between privacy and performance for battery-powered mobile devices is achieved. The strategy is obtained with a learning-based Lyapunov optimization framework and it provides a provable guarantee of performance. A small-scale testbed is built to prove the efficacy of the designed offloading strategy. Teerapittayanon *et al*. [36] present distributed deep neural networks (DDNNs) for distributed hierarchies of computing. They include end devices, edge (fog) and cloud. DDNN enables localized and fast inference by using its shallow portions at end devices and the edge. The sections of a DNN are mapped onto a distributed computing hierarchy. Then, they are jointly trained to minimize resource and communication usage for end devices and maximize usage of obtained features utilized in the cloud. In this way, a DDNN exploits geographical difference of sensors to increase accuracy of object recognition and decrease communication cost. Ko et al. [92] applies partitioning of a deep neural network between a host platform and an edge in an IoT environment. A DNN is presented as an encoding pipeline, and the space of output feature for an intermediate layer is transmitted to the host platform. Then, the encoding of the feature space is designed to improve the maximum input rate for the edge platform and decrease its energy consumption. In addition, energy-efficiency and throughput of the edge and the host are improved. Yang et al. [93] consider a joint energy and latency minimization problem for distribution of hierarchical machine learning tasks in mobile edge computing. Then, a framework is proposed to enable end devices embedded with shallow neural network models. The computing-intensive and latency-sensitive tasks are offloaded to a nearby server where a powerful deep neural network model is deployed. The offloading strategy is formulated as a piecewise convex optimization problem. An optimal partial offloading method for tasks is obtained analytically for minimizing the weighted sum of energy and latency. Zientara *et al*. [94] investigate the tradeoff between energy consumption and the performance for deep neural networks. A scheme is presented to exploit QoS tolerance for DNNs with decreased mobile device power consumption and transmission fidelity. Then, an adaptive method is developed for the run-time selection of a suitable model, offload selection, and transmission power setting for given noisy environments.

## Power/Battery Management (7)

Kang *et al*. [22] design a joint optimization approach for distributed energy-efficient data centers. It uses a long short-term memory (LSTM) algorithm to increase the prediction accuracy of green energy for a long period, and an unsupervised deep learning method to optimize the coordinated frequency scaling and right sizing. In addition, a macro/micro time scale-based management method for data center management is presented to decrease wake-up transition overhead and improve high energy efficiency. Xu *et al*. [24] adopt an energy harvesting method and study a dynamic access control problem for MEC. It aims to maximize the long-term average rate of uplink transmission, and minimize the energy consumption of transmission for green IoT networks. Each IoT device is powered by a battery, which utilizes energy from its surrounding environments. The problem is formulated as a Markov decision process with unknown system dynamics. An LSTM-based deep Q-network is proposed to achieve the optimal access control for the IoT network. Nguyen *et al*. [26] adopt a gradient algorithm of deep deterministic policy, and propose two methods for a multi-agent problem of power allocation for vehicle-to-vehicle communications in device-to-device environments. Their results demonstrate that the proposed models perform better than other deep reinforcement learning methods with respect to network energy flexibility and efficiency. Chen *et al*. [33] consider a representative MEC scenario for mobile users in an ultra-dense network, where many base stations are available for computation offloading. A Markov decision process is modelled and an optimal computation offloading policy is achieved to minimize the long-term cost. The offloading decision is obtained by considering channel qualities between BSs and mobile users, states of energy queue and task queue. To address high dimensionality curse in state space, a deep Q-network-based computation offloading method is proposed to find the optimal policy without priori dynamic statistics.

Meng *et al*. [34] propose a nonvolatile memories-friendly training method for neural networks. The update of weights is redesigned to decrease bit flips in cells of nonvolatile memories. In addition, two methods including bitwise rotation and filter exchange are adopted to balance writes to different bits of each weight and different weights. The methods are implemented in Caffe and results demonstrate they significantly reduce power consumption and improve endurance while keeping high accuracy of inference. Zhang *et al*. [42] propose a double deep Q-learning model for energy-efficient edge scheduling. It includes a new network for obtaining a Q-value for each dynamic voltage and frequency scaling algorithm, and a target network for obtaining the target Q-values for the training of the parameters. Besides, a rectified linear units function is adopted as the activation function in a double deep Q-learning model to solve the problem of gradient vanishing. Then, an experience replay-based learning algorithm is proposed to train the model parameters, and achieve energy reduction and training time. Zhu *et al*. [45] aim to achieve an efficient mechanism of computation offloading for service workflows in mobile edge computing. It decreases completion time of applications and energy consumption of user devices in offloading. The computation offloading problem is formulated as a time and energy optimization one. Then, the optimal cost strategy is obtained by using deep Q-learning according to users’ experience.

## Resource Allocation (9)

Yang *et al*. [27] propose an energy-efficient edge processing method to execute tasks of deep learning inference at edge nodes whose connections to mobile devices are subject to uncertainties in channels. A joint downlink beamforming and inference tasking problem is formulated by considering characteristics of a group sparse objective function. A statistical learning method is provided by using robust optimization to approximate intractable nonconvex quadratic constraints. He *et al*. [31] design a tandem queueing model to obtain the end-to-end inference latency of deep learning tasks in multiple partitions of deep neural networks. A joint optimization problem of resource allocation and partition deployment is proposed to minimize the end-to-end latency. It is formulated as a mixed integer nonlinear Programming (MINLP) one, which is decomposed into two problems including computing resource allocation and DNN partition deployment. The the near-optimal solution is obtained by using a designed computing resource allocation algorithm and a low-complexity DNN partition deployment algorithm. Lei *et al*. [48] design time and energy-efficient methods for content delivery at the edge of network. Two resource allocation problems are formulated to minimize the transmission time and energy in content delivery, respectively. They are formulated as mixed-integer linear programs, and solved by learning-based methods including fully-connected deep neural network and convolutional neural network to provide a computationally-efficient and high-quality solution.

Dong *et al*. [61] consider a mobile edge computing system that provides both low-latency and ultra-reliable services and delay tolerant services. It aims to achieve the normalized energy consumption minimization by optimizing resource allocation, user association, and offloading probabilities subject to quality-of-service constraints. A deep learning architecture is proposed by using a digital twin of the real network environment to train the deep learning algorithm offline in a central server. An optimization algorithm is proposed to obtain the optimal offloading probabilities and resource allocation and at each access point. Dai *et al*. [71] adopt deep reinforcement learning (DRL) to obtain an optimal resource allocation and computation offloading strategy for minimizing the energy consumption of system by using network information including computation resources, available bandwidth and wireless channel state. A multi-user edge computing architecture in heterogeneous networks is presented by considering strong relations among devices on application needs or radio access. The joint resource allocation and computation offloading problem is formulated as a DRL form, and a new DRL-inspired algorithm is proposed to minimize energy consumption of system.

Gu *et al*. [74] propose a framework for efficient training of distributed deep neural network (DNN) that affects the best configuration of a training cluster with heterogeneous computing resources. It adopts pre-training with a limited number of training steps and predicts training time, energy, and energy-delay product for each configuration of training cluster. It performs training of DNN models for remaining steps by the selected best cluster configuration according to preferences, e.g., energy efficiency and training time, of DNN service providers. Wan *et al*. [88] design an online three-layer data processing network by using a MEC technique. The raw data is produced by distributed sensors with local information. Then, unmanned aerial vehicle base stations are deployed as MEC servers that collect data and process it. Then, a centralized cloud obtains processed results. Edge nodes stabilize delay to guarantee freshness of data and processing capability under limited onboard energy constraints. Then, an algorithm for online edge processing scheduling is proposed with Lyapunov optimization, and deep reinforcement learning is adopted to propose online path planning. Cui *et al*. [49] formulate the energy and latency optimization for satellite-based IoT as a mixed-integer programming problem, which is difficult to solve. It is decomposed into two sub-problems including communication and computing resource allocation, and joint offloading and user association with the optimal resource allocation. Especially, deep reinforcement learning is adopted to solve the second sub-problem that is formulated as a Markov decision process. Heydari *et al*. [51] considers a system consisting of multiple non-cooperative mobile devices and edge devices. Several offloading policies are designed to minimize the drop rate of tasks and the execution delay without information about task arrival rates and channel models at mobile devices. This problem of non-cooperative resource allocation is partially observable for each mobile device. A deep reinforcement learning-based method is applied to progressively learn dynamic information of the environment and the long-term results of decisions.

## Computational offloading (8)

In order to achieve lower average task’s latency and energy consumption, we minimize the weighted summation of the average task’s delay and energy consumption by optimizing the task’s offloading decision [91]. Ren *et al*. [28] . In this way, computational offload decisions for management of federation and complex resources are specified in a real-time manner by considering dynamic radio environments and workloads. Multiple DRL agents are deployed in multiple edge nodes. In addition, federated learning is adopted to train the agents in in a distributed way with the objective of decreasing transmission cost between edge nodes and IoT devices. Ali *et al*. [40] design an energy-efficient deep learning based offloading method to obtain a smart decision-making algorithm. It chooses an optimal set of components according to remaining energy of user equipments, energy consumption, network conditions, workload, data transfer, and communication latency. A cost function is formulated by considering the factors and obtain the cost of all combinations of component offloading methods. In addition, it determines the optimal methods based on a dataset, and a deep learning network is trained accordingly. Dey *et al*. [57] implement a deep learning inference offloading system with a Raspberry Pi 3-based robot vehicle in an Intel hardware accelerator. A workload partitioning method is proposed and experimental performance results are presented. It is shown that the optimal partitioning can be obtained by considering dynamic system load of edge devices in the deep learning profiling method. Lu *et al*. [65] propose a quality of experience model for the computation offloading. It capture three major factors including service latency, energy consumption, and task success rate. Moreover, a deep deterministic policy gradients algorithm is proposed to improve the quality of experience. To address the slow convergence and poor stability in the computation offloading, double Q-learning and dueling networks are used to improve the critic network. In this way, the quality of experience in edge-enabled IoT is improved.

Naderializadeh and Hashemi [67] investigate the computation offloading problem in a mobile edge computing environment. Multiple energy-limited users compete to offload tasks to available servers with a common wireless medium. A multi-agent deep reinforcement learning algorithm is proposed and an agent is designed for each server. The agents monitor the status of its users and choose the best user for offloading in each step. Two key performance factors including system lifetime and task completion time are considered. Gaballo *et al*. [78] provide an architecture for managing traffic that adopts deep learning to enable forwarding during task offloading for mobile and IoT devices. It is shown that deep learning is adopted to forward traffic of microservices and offloadi requests for mobile and IoT devices. Each router individually learns a lossy path and achieves high throughput without coordination by using signals from performance-unaware protocols, *e.g.*, OSPF. Chen *et al*. [89] consider mobile-edge computing for a mobile user in an ultradense sliced radio access network, where several base stations are selected for computation offloading. The problem is formulated as a Markov decision process and an optimal computation offloading method is proposed to maximize long-term utility performance. The optimal offloading is obtained according to the state of task queue and energy, and channel information between base stations and mobile users. To address a high dimensionality problem, a double deep Q-network-based computation offloading algorithm is proposed without needing a priori network knowledge. Liu *et al*. [90] investigate a three-tier architecture for mobile computing network. It includes edge computing servers, a user equipment and a cloud server where tasks can be run locally, in edge servers or in cloud server, respectively. The weighted sum of the average delay of tasks and energy consumption is obtained by optimizing task offloading. An optimization framework is proposed based on deep reinforcement learning to solve the computation offloading problem. Specifically, an actor-critic framework is adopted to solve it.

## DNN hardware architecture (9)

To provide efficient deep neural networks, it is highly essential to optimize them at both algorithmic and hardware levels. Xiang *et al*. [86] propose a hardware implementation of analog deep neural network based on NOR Flash Computing Array. It removes additional analog-to-digital/digital-to-analog transformation between adjacent layers. It is high-speed, energy efficient and promising with respect to artificial intelligence at the edge. Chen *et al*. [87] propose a compute in memory method that is paired with high-density nonvolatile memory to improve operations of deep neural networks for artificial intelligence edge processors. Specifically, a binary-input hardware-driven ternary-weighted network is proposed to achieve a smaller energy-hardware cost by using the pseudo-binary nonvolatile memory macros and a two-macro deep neural network. Verhelst and Moons [83] introduce tightly interwoven methods for hardware and software processing for energy efficiency. It also shows the details of implementation of algorithmic innovations by using flexible processing architectures. It also points out the implementation challenges and chances for deep neural network algorithms.

Ganguly *et al*. [79] give an analysis of deep learning architectures and how they benefit from modeling its algorithms as dataflows that are represented as neurons. It is shown that accuracy and performance of workloads are drastically improved when it is modeled as non-von Neumann architectures. A landscape of implementations of different convolutional neural networks across von Neumann and non-von Neumann models. Several factors that affect energy efficiency are discussed and they include energy models, estimation methods, benchmarks and standards. Asadikouhanjani and Ko [53] develop a novel architecture in which the activation function is merged with the prior computational layer. Our proposed dataflow can reduce the computations needed to generate a specific output feature in convolutional and fully connected layers by reordering the computation to achieve early detection of the sign of the output feature. In addition, our computation engine supports zero computation skipping, which further accelerates the layer computation. In this work, we build on a state-of-the-art DNN accelerator and modify it to perform early detection of negative output features.

Deep Neural Networks (DNNs) perform billions of computations in applications such as image classification, object detection, and image segmentation. In most well-known DNNs, an activation function follows a convolutional or a fully connected layer. Several popular activation functions involve setting all negative inputs to zero.

When compared to the original design, our method achieves a speedup of ×2.19 and reduces energy consumption by ×1.94. The average reduction in the number of multiply-accumulate (MAC) operations is 10.64% and the average reduction in the number of the load operations is 3.86%. These improvements are achieved while maintaining classification accuracy in two popular benchmark networks.

[50]

[70]

[63]

[15]

Hardware Accelerators: The optimizations at the software level should be supported by specialized hardware accelerators in a codesign fashion [47] [19]. Recent advances in the datacenter computing deep learning [27] have inspired accelerators for edge devices. Specialized accelerators like [5] [28] exploit the concurrency and the parallelism available in the processing of the DNNs, especially for convolutional leyers, while [20] takes care also of the fully-connected layers. These architectures, however, accelerate dense DNNs, and cannot exploit the sparsity introduced by pruning. Therefore, specialized accelerators for sparse DNNs are required [13][52]. Challenging aspects of these accelerators are flexibility, reconfigurability and data reuse [35][39][65]. Moreover, particular types of DNNs, like CapsuleNets [60] and GANs [91] present several differences in the computation patterns, as compared to traditional DNNs. These challenges are addressed by their specialized accelerators. For example, CapsAcc [46] adopts a data reuse policy to efficiently process the routing-by-agreement algorithm on a systolyc array-based accelerator for CapsuleNets, and GANAX [76] propose a unified MIMD-SIMD design for concurrent execution of GANs. The software-level optimizations mainly include network pruning (Step-1 in Fig. 3) and quantization (Step-2 in Fig. 3) of the parameters.

## DNN packages and tools (11)

[11]

[12]

[60]

[64]

[73]

[29]

Neurosurgeon [5] is a lightweight scheduler that can automatically partition DNN computation between mobile devices and datacenters at the granularity of neural network layers. By effectively leveraging the resources in the cloud and at the edge, neurosurgeon achieves low computing latency, **low energy consumption**, and high traffic throughput.

FPGA-based accelerators can achieve high-performance computing with low energy, high parallelism, high flexibility, and high security [6].

An efficient speech recognition engine (ESE) is designed to speed up the predictions and save energy when applying the deep learning model of LSTM. ESE is implemented in a Xilinx XCKU060 FPGA operating at 200 MHz. For the sparse LSTM network, it can achieve 282 GOPS, corresponding to 2.52 tera operations per second (TOPS) on the dense LSTM network. In addition, energy efficiency improvements of 40× and 11.5× are achieved, respectively, compared with the CPU- and GPU-based solution [7].

NVIDIA Jetson TX2 is an embedded AI computing device, which is designed to achieve low latency and high power efficiency. It is built upon an NVIDIA Pascal GPU with 256 CUDA cores, an HMP Dual Denver CPU, and a Qualcomm ARM CPU. It is loaded with 8 GB of memory and 59.7 GB/s of memory bandwidth and the power is about 7.5 W. The GPU is used to execute the deep learning task, and CPUs are used to maintain general tasks. It also supports the NVIDIA Jetpack SDK that includes libraries for deep learning, computer vision, GPU computing, and multimedia processing [8].

Edge TPU is Google’s purpose-built ASIC for edge computing. It augments Google’s Cloud TPU and Cloud IoT to provide an end-to-end infrastructure and facilitates the deployment of customers’ AI-based solutions. In addition, Edge TPU can combine the custom hardware, open software, and state-of-the-art AI algorithms to achieve high performance with a small physical area and low power consumption [9].

Prevalence of internet of things (IoT) enabled applications provide a new opportunity to low-cost FPGA devices to act as edge computing neural network nodes. Although FPGA vendors provide neural network development environments, they often target high-end devices. At the same time these development platforms are not as user friendly as their software counterparts. In this work we introduce ZyNet, a Python package, which enables faster implementation of deep neural networks (DNNs) targeting low-cost hybrid FPGA platforms such as the Xilinx Zynq. Based on hardware-software co-design approach, this platform supports pre-trained or on-board trained networks with development environment very similar to the popular TensorFlow. Implementation results show that the DNNs generated by the platform achieve accuracy very close to software implementations at the same time gives throughput by an order of magnitude compared to other edge computing devices at lower energy footprint. The platform is integrated with Xilinx development tools and is distributed as open source.

List a table.

# NEW TRENDS AND OPEN CHALLENGES (Duplicated papers)

OPEN RESEARCH CHALLENGES

**Hardware Software Co-Design**: A common trend is to optimize the DNN for achieving high accuracy, without caring much about the underlying hardware complexity and energy consumption of a computing device. On the other hand, hardware designers have to implement a-posteriori architectures to exploit the software-level optimizations. However, hardware-aware software-level optimizations, e.g., for DNN architecture exploration [69] or compression [43] are promising and need further efforts to succeed. [20].

**In-Memory Computing**: It seems to be a promising paradigm for developing accelerators that can offer orders of magnitude of energy-efficiency gains compared to the conventional CPU and GPU based systems. However, the high variation characteristics associated with ReRAM and other non-volatile memories limit the accelerators which are based on them to offer precise functionality. Towards this, the multi-level cell (MLC) ReRAM technology has to be mature enough to offer reasonable precision while offering high data density. Also, a significant amount of work is required to develop methods which can be used to train networks such that they can offer high accuracy even when operated on NVM-based in-memory computing devices. [20].

**Hardware-Aware Hyperparameter Tuning and DNN Architectural Exploration**: Several software-level optimization techniques have been proposed which highlight that sparse DNNs, i.e., having lesser number of parameters, can also offer nearly the same level of output accuracy as dense DNNs. Systematic methodologies are required which, while being aware of the underlying hardware architecture and the system, can tune the network such that it offers near-optimal energy and performance efficiency while maintaining the baseline accuracy. [20].

Event-based Spiking Neural Networks: They have the potential to be much more energy-efficient, as compared to digital-based DNNs, because the power is only consumed when a spike is firing. Such event-driven processing are promising. Therefore, companies like IBM and Intel are investing into their respective neuromorphic architecture chips and its accelerators [48] [9]. [20].

## Distributed and Collaborative DNN

[13].

## Relationship to SDN and NFV Technologies

[10]

## Management and Scheduling of Edge Compute Resources

## Emerging Technologies

[13].

## Deep Learning Benchmarks on Edge Devices

# CONCLUSION

References

1. X. Wang, X. Wei and L. Wang, "A deep learning based energy-efficient computational offloading method in Internet of vehicles," in China Communications, vol. 16, no. 3, pp. 81-91, March 2019.
2. A MobileNets Convolutional Neural Network for GIS Partial Discharge Pattern Recognition in the Ubiquitous Power Internet of Things Context Optimization- Comparison- and Application
3. A Multi-Agent System toward the Green Edge Computing with Microgrid-- M. S. Munir, S. F. Abedin, D. H. Kim, N. H. Tran, Z. Han and C. S. Hong, "A Multi-Agent System toward the Green Edge Computing with Microgrid," 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 2019, pp. 1-7.
4. X. Zhang, Y. Wang, and W. Shi, “pCAMP: Performance comparison of machine learning packages on the edges,” in Proc. USENIX Workshop Hot Topics Edge Comput. (HotEdge), Boston, MA, USA, 2018. [Online]. Available: https://www. usenix.org/conference/hotedge18/presentation/zhang
5. Y. Kang et al., “Neurosurgeon: Collaborative intelligence between the cloud and mobile edge,” ACM SIGPLAN Notices, vol. 52, no. 4, pp. 615–629, 2017.
6. T.Wang, C. Wang, X. Zhou, and H. Chen, “A survey of FPGA based deep learning accelerators: Challenges and opportunities,” 2018, arXiv:1901.04988. [Online]. Available: <https://arxiv.org/abs/1901.04988>
7. J. Qiu et al., “Going deeper with embedded FPGA platform for convolutional neural network,” in Proc. ACM/SIGDA Int. Symp. Field-Program. Gate Arrays, 2016, pp. 26–35.
8. (2019). NVIDIA Jetson TX2 Module. [Online]. Available: <https://developer>. nvidia.com/embedded/buy/jetson-tx2
9. Z. Du et al., “Shidiannao: Shifting vision processing closer to the sensor,” ACM Sigarch Comput. Archit. News, vol. 43, no. 3, pp. 92–104, 2015.
10. Adaptive Video Streaming With Edge Caching and Video Transcoding Over Software-Defined Mobile Networks A Deep Reinforcement Learning Approach
11. An Associativity-Agnostic in-Cache Computing Architecture Optimized for Multiplication
12. An overview of next-generation architectures for machine learning Roadmap- opportunities and challenges in the IoT era
13. M. M. Sabry Aly et al., "Energy-Efficient Abundant-Data Computing: The N3XT 1,000x," in Computer, vol. 48, no. 12, pp. 24-33, Dec. 2015.
14. Architecture-aware design and implementation of CNN algorithms for embedded inference the ALOHA project
15. Area and Energy Optimization for Bit-Serial Log-Quantized DNN Accelerator with Shared Accumulators
16. Behind-the-Meter Solar Generation Disaggregation using Consumer Mixture Models
17. Computation Offloading and Resource Allocation for MEC in C-RAN A Deep Reinforcement Learning Approach
18. Computation Offloading in Multi-Access Edge Computing Using a Deep Sequential Model Based on Reinforcement Learning
19. F. Jiang, K. Wang, L. Dong, C. Pan, W. Xu and K. Yang, "Deep Learning Based Joint Resource Scheduling Algorithms for Hybrid MEC Networks," in IEEE Internet of Things Journal.
20. Deep Learning for Edge Computing Current Trends- Cross-Layer Optimizations- and Open Research Challenges
21. Deep Learning for Hybrid 5G Services in Mobile Edge Computing Systems Learn From a Digital Twin
22. Deep Learning-Based Sustainable Data Center Energy Cost Minimization With Temporal MACRO-MICRO Scale Management
23. Deep reinforcement learning based computation offloading and resource allocation for MEC
24. Deep Reinforcement Learning for Dynamic Access Control with Battery Prediction for Mobile-Edge Computing in Green IoT Networks
25. Deep Reinforcement Learning-Based Computation Offloading in Vehicular Edge Computing
26. Distributed Deep Deterministic Policy Gradient for Power Allocation Control in D2D-Based V2V Communications
27. Energy-Efficient Processing and Robust Wireless Cooperative Transmission for Edge Inference
28. Federated Learning-Based Computation Offloading Optimization in Edge Computing-Supported Internet of Things
29. GeneSys Enabling Continuous Learning through Neural Network Evolution in Hardware
30. Joint Communication and Computing Optimization for Hierarchical Machine Learning Tasks Distribution
31. Joint DNN Partition Deployment and Resource Allocation for Delay-Sensitive Deep Learning Inference in IoT
32. Joint Optimization of Networking and Computing Resources for Green M2M Communications Based on DRL
33. Performance Optimization in Mobile-Edge Computing via Deep Reinforcement Learning
34. Power- and Endurance-Aware Neural Network Training in NVM-Based Platforms
35. Privacy-Aware Edge Computing Based on Adaptive DNN Partitioning
36. S. Teerapittayanon, B. McDanel, and H. Kung, “Distributed deep neural networks over the cloud, the edge and end devices,” in Distributed Computing Systems, 2017 IEEE 37th Int’l Conf. on, pp. 328–339.
37. J. Li, Q. Liu, P. Wu, F. Shu and S. Jin, "Task Offloading for UAV-based Mobile Edge Computing via Deep Reinforcement Learning," 2018 IEEE/CIC International Conference on Communications in China (ICCC), Beijing, China, 2018, pp. 798-802.
38. J. Lim, J. Seo and Y. Baek, "CamThings: IoT Camera with Energy-Efficient Communication by Edge Computing based on Deep Learning," 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), Sydney, NSW, 2018, pp. 1-6.
39. M. S. Munir, S. F. Abedin, N. H. Tran and C. S. Hong, "When Edge Computing Meets Microgrid: A Deep Reinforcement Learning Approach," in IEEE Internet of Things Journal, vol. 6, no. 5, pp. 7360-7374, Oct. 2019.
40. Z. Ali, L. Jiao, T. Baker, G. Abbas, Z. H. Abbas and S. Khaf, "A Deep Learning Approach for Energy Efficient Computational Offloading in Mobile Edge Computing," in IEEE Access, vol. 7, pp. 149623-149633, 2019.
41. S. H. Lee, T. Lee, S. Kim and S. Park, "Energy Consumption Prediction System Based on Deep Learning with Edge Computing," 2019 IEEE 2nd International Conference on Electronics Technology (ICET), Chengdu, China, 2019, pp. 473-477.
42. Q. Zhang, M. Lin, L. T. Yang, Z. Chen, S. U. Khan and P. Li, "A Double Deep Q-Learning Model for Energy-Efficient Edge Scheduling," in IEEE Transactions on Services Computing, vol. 12, no. 5, pp. 739-749, 1 Sept.-Oct. 2019.
43. H. Ke, J. Wang, H. Wang and Y. Ge, "Joint Optimization of Data Offloading and Resource Allocation With Renewable Energy Aware for IoT Devices: A Deep Reinforcement Learning Approach," in IEEE Access, vol. 7, pp. 179349-179363, 2019.
44. A. Zhu et al., "Computation Offloading for Workflow in Mobile Edge Computing Based on Deep Q-Learning," 2019 28th Wireless and Optical Communications Conference (WOCC), Beijing, China, 2019, pp. 1-5.
45. M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu and W. Zhuang, "Learning-Based Computation Offloading for IoT Devices With Energy Harvesting," in IEEE Transactions on Vehicular Technology, vol. 68, no. 2, pp. 1930-1941, Feb. 2019.
46. Y. Liu, C. Yang, L. Jiang, S. Xie and Y. Zhang, "Intelligent Edge Computing for IoT-Based Energy Management in Smart Cities," in IEEE Network, vol. 33, no. 2, pp. 111-117, March/April 2019.
47. P. Zhang, Q. Zhao, J. Gao, W. Li and J. Lu, "Urban Street Cleanliness Assessment Using Mobile Edge Computing and Deep Learning," in IEEE Access, vol. 7, pp. 63550-63563, 2019.
48. L. Lei, Y. Yuan, T. X. Vu, S. Chatzinotas and B. Ottersten, "Learning-Based Resource Allocation: Efficient Content Delivery Enabled by Convolutional Neural Network," 2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Cannes, France, 2019, pp. 1-5.
49. G. Cui, X. Li, L. Xu and W. Wang, "Latency and Energy Optimization for MEC Enhanced SAT-IoT Networks," in IEEE Access, vol. 8, pp. 55915-55926, 2020.
50. M. Ahn et al., "AIX: A high performance and energy efficient inference accelerator on FPGA for a DNN-based commercial speech recognition," 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), Florence, Italy, 2019, pp. 1495-1500.
51. J. Heydari, V. Ganapathy and M. Shah, "Dynamic Task Offloading in Multi-Agent Mobile Edge Computing Networks," 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 2019, pp. 1-6.
52. J. Granados, H. Chu, Z. Zou and L. Zheng, "Towards Workload-Balanced, Live Deep Learning Analytics for Confidentiality-Aware IoT Medical Platforms," 2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), Hsinchu, Taiwan, 2019, pp. 62-66.
53. M. Asadikouhanjani and S. Ko, "A Novel Architecture for Early Detection of Negative Output Features in Deep Neural Network Accelerators," in IEEE Transactions on Circuits and Systems II: Express Briefs.
54. Z. Zhang, L. L. Njilla, S. Yu and J. Yuan, "Edge-Assisted Learning for Real-Time UAV Imagery via Predictive Offloading," 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 2019, pp. 1-6.
55. L. Xu, J. Han, T. Wang and L. Bai, "An Efficient CNN to Realize Speckle Correlation Imaging Based on Cloud-Edge for Cyber-Physical-Social-System," in IEEE Access, vol. 8, pp. 54154-54163, 2020.
56. Y. Yan and Q. Pei, "A Robust Deep-Neural-Network-Based Compressed Model for Mobile Device Assisted by Edge Server," in IEEE Access, vol. 7, pp. 179104-179117, 2019.
57. S. Dey, J. Mondal and A. Mukherjee, "Offloaded Execution of Deep Learning Inference at Edge: Challenges and Insights," 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kyoto, Japan, 2019, pp. 855-861.
58. M. Hartmann, H. Farooq and A. Imran, "Distilled Deep Learning based Classification of Abnormal Heartbeat Using ECG Data through a Low Cost Edge Device," 2019 IEEE Symposium on Computers and Communications (ISCC), Barcelona, Spain, 2019, pp. 1068-1071.
59. C. R. T, G. Sirisha and A. M. Reddy, "Smart Healthcare Analysis and Therapy for Voice Disorder using Cloud and Edge Computing," 2018 4th International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Mangalore, India, 2018, pp. 103-106.
60. C. Gong, Z. Jiang, D. Wang, Y. Lin, Q. Liu and D. Z. Pan, "Mixed Precision Neural Architecture Search for Energy Efficient Deep Learning," 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Westminster, CO, USA, 2019, pp. 1-7.
61. R. Dong, C. She, W. Hardjawana, Y. Li and B. Vucetic, "Deep Learning for Hybrid 5G Services in Mobile Edge Computing Systems: Learn From a Digital Twin," in IEEE Transactions on Wireless Communications, vol. 18, no. 10, pp. 4692-4707, Oct. 2019.
62. B. Lu, J. Yang, L. Y. Chen and S. Ren, "Automating Deep Neural Network Model Selection for Edge Inference," 2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI), Los Angeles, CA, USA, 2019, pp. 184-193.
63. J. Lee, J. Lee, D. Han, J. Lee, G. Park and H. Yoo, "An Energy-Efficient Sparse Deep-Neural-Network Learning Accelerator With Fine-Grained Mixed Precision of FP8–FP16," in IEEE Solid-State Circuits Letters, vol. 2, no. 11, pp. 232-235, Nov. 2019.
64. J. Edstrom, H. Das, Y. Xu and N. Gong, "Memory Optimization for Energy-Efficient Differentially Private Deep Learning," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 2, pp. 307-316, Feb. 2020.
65. H. Lu, X. He, M. Du, X. Ruan, Y. Sun and K. Wang, "Edge QoE: Computation Offloading with Deep Reinforcement Learning for Internet of Things," in IEEE Internet of Things Journal.
66. G. Stamatescu, R. Entezari, K. Römer and O. Saukh, "Deep and Efficient Impact Models for Edge Characterization and Control of Energy Events," 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), Tianjin, China, 2019, pp. 639-646.
67. N. Naderializadeh and M. Hashemi, "Energy-Aware Multi-Server Mobile Edge Computing: A Deep Reinforcement Learning Approach," 2019 53rd Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 2019, pp. 383-387.
68. N. Monburinon, S. M. S. Zabir, N. Vechprasit, S. Utsumi and N. Shiratori, "A Novel Hierarchical Edge Computing Solution Based on Deep Learning for Distributed Image Recognition in IoT Systems," 2019 4th International Conference on Information Technology (InCIT), Bangkok, Thailand, 2019, pp. 294-299.
69. T. Li, Y. Xiao and L. Song, "Deep Reinforcement Learning Based Residential Demand Side Management With Edge Computing," 2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), Beijing, China, 2019, pp. 1-6.
70. G. Ascia, V. Catania, S. Monteleone, M. Palesi, D. Patti and J. Jose, "Networks-on-Chip based Deep Neural Networks Accelerators for IoT Edge Devices," 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), Granada, Spain, 2019, pp. 227-234.
71. Y. Dai, D. Xu, K. Zhang, Y. Lu, S. Maharjan and Y. Zhang, "Deep Reinforcement Learning for Edge Computing and Resource Allocation in 5G Beyond," 2019 IEEE 19th International Conference on Communication Technology (ICCT), Xi'an, China, 2019, pp. 866-870.
72. H. Zhang, W. Wu, C. Wang, M. Li and R. Yang, "Deep Reinforcement Learning-Based Offloading Decision Optimization in Mobile Edge Computing," 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 2019, pp. 1-7.
73. K. Vipin, "ZyNet: Automating Deep Neural Network Implementation on Low-Cost Reconfigurable Edge Computing Platforms," 2019 International Conference on Field-Programmable Technology (ICFPT), Tianjin, China, 2019, pp. 323-326.
74. B. Gu, J. Kong, A. Munir and Y. G. Kim, "A Framework for Distributed Deep Neural Network Training with Heterogeneous Computing Platforms," 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), Tianjin, China, 2019, pp. 430-437.
75. Z. Ning et al., "Deep Reinforcement Learning for Intelligent Internet of Vehicles: An Energy-Efficient Computational Offloading Scheme," in IEEE Transactions on Cognitive Communications and Networking, vol. 5, no. 4, pp. 1060-1072, Dec. 2019.
76. C. Lammie, A. Olsen, T. Carrick and M. Rahimi Azghadi, "Low-Power and High-Speed Deep FPGA Inference Engines for Weed Classification at the Edge," in IEEE Access, vol. 7, pp. 51171-51184, 2019.
77. T. Sirojan, S. Lu, B. T. Phung, D. Zhang and E. Ambikairajah, "Sustainable Deep Learning at Grid Edge for Real-time High Impedance Fault Detection," in IEEE Transactions on Sustainable Computing.
78. A. Gaballo, M. Flocco, F. Esposito and G. Marchetto, "ADELE: An Architecture for Steering Traffic and Computations via Deep Learning in Challenged Edge Networks," 2019 4th International Conference on Computing, Communications and Security (ICCCS), Rome, Italy, 2019, pp. 1-8.
79. A. Ganguly, R. Muralidhar and V. Singh, "Towards Energy Efficient non-von Neumann Architectures for Deep Learning," 20th International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, USA, 2019, pp. 335-342.
80. I. Wu, P. Huang, C. Lo and W. Hwang, "An Energy-Efficient Accelerator with Relative- Indexing Memory for Sparse Compressed Convolutional Neural Network," 2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), Hsinchu, Taiwan, 2019, pp. 42-45.
81. W. Zhan et al., "Deep Reinforcement Learning-Based Offloading Scheduling for Vehicular Edge Computing," in IEEE Internet of Things Journal.
82. N. C. Luong, Z. Xiong, P. Wang and D. Niyato, "Optimal Auction for Edge Computing Resource Management in Mobile Blockchain Networks: A Deep Learning Approach," 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, 2018, pp. 1-6.
83. M. Verhelst and B. Moons, "Embedded Deep Neural Network Processing: Algorithmic and Processor Techniques Bring Deep Learning to IoT and Edge Devices," in IEEE Solid-State Circuits Magazine, vol. 9, no. 4, pp. 55-65, Fall 2017.
84. S. R. Young et al., "Evolving Energy Efficient Convolutional Neural Networks," 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 2019, pp. 4479-4485.
85. Y. Zhang et al., "Cyber Physical Security Analytics for Transactive Energy Systems," in IEEE Transactions on Smart Grid, vol. 11, no. 2, pp. 931-941, March 2020.
86. Y. C. Xiang et al., "Analog Deep Neural Network Based on NOR Flash Computing Array for High Speed/Energy Efficiency Computation," 2019 IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, 2019, pp. 1-4.
87. W. Chen et al., "A 65nm 1Mb nonvolatile computing-in-memory ReRAM macro with sub-16ns multiply-and-accumulate for binary DNN AI edge processors," 2018 IEEE International Solid - State Circuits Conference - (ISSCC), San Francisco, CA, 2018, pp. 494-496.
88. S. Wan, J. Lu, P. Fan and K. B. Letaief, "Towards Big data processing in IoT: Path Planning and Resource Management of UAV Base Stations in Mobile-Edge Computing System," in IEEE Internet of Things Journal.
89. X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji and M. Bennis, "Optimized Computation Offloading Performance in Virtual Edge Computing Systems Via Deep Reinforcement Learning," in IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4005-4018, June 2019.
90. Y. Liu, Q. Cui, J. Zhang, Y. Chen and Y. Hou, "An Actor-Critic Deep Reinforcement Learning Based Computation Offloading for Three-Tier Mobile Computing Networks," 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP), Xi'an, China, 2019, pp. 1-6.
91. S. Jiang et al., "Accelerating Mobile Applications at the Network Edge with Software-Programmable FPGAs," IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, Honolulu, HI, 2018, pp. 55-62.
92. J. H. Ko, T. Na, M. F. Amir and S. Mukhopadhyay, "Edge-Host Partitioning of Deep Neural Networks with Feature Space Encoding for Resource-Constrained Internet-of-Things Platforms," 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 2018, pp. 1-6.
93. B. Yang, X. Cao, X. Li, T. Kroecker and L. Qian, "Joint Communication and Computing Optimization for Hierarchical Machine Learning Tasks Distribution," 2019 IEEE Symposium on Computers and Communications (ISCC), Barcelona, Spain, 2019, pp. 1-6.
94. P. A. Zientara, J. Sampson and V. Narayanan, "Noise Aware Power Adaptive Partitioned Deep Networks for Mobile Visual Assist Platforms," 2018 31st IEEE International System-on-Chip Conference (SOCC), Arlington, VA, 2018, pp. 186-191.
95. S. Khan, D. Paul, P. Momtahan and M. Aloqaily, "Artificial intelligence framework for smart city microgrids: State of the art, challenges, and opportunities," 2018 Third International Conference on Fog and Mobile Edge Computing (FMEC), Barcelona, 2018, pp. 283-288.
96. E. Kasaeyan Naeini, S. Shahhosseini, A. Subramanian, T. Yin, A. M. Rahmani and N. Dutt, "An Edge-Assisted and Smart System for Real-Time Pain Monitoring," 2019 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), Arlington, VA, USA, 2019, pp. 47-52.
97. Tuli S, Basumatary N, Gill SS, Kahani M, Arya RC, Wander GS, Buyya R. HealthFog: An ensemble deep learning based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in integrated IoT and fog computing environments. Future Generation Computer Systems. 2020 Mar 1;104:187-200.
98. S. H. Lee, T. Lee, S. Kim and S. Park, "Energy Consumption Prediction System Based on Deep Learning with Edge Computing," 2019 IEEE 2nd International Conference on Electronics Technology (ICET), Chengdu, China, 2019, pp. 473-477.
99. S. Han et al. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In ICLR, 2016.
100. W. Wen et al. Learning structured sparsity in deep neural networks. In NIPS, 2016.
101. A. Marchisio et al. PruNet: Class-Blind Pruning Method For Deep Neural Networks. In IJCNN, 2018.
102. G. E. Hinton et al. Distilling the knowledge in a neural network. In NIPS, 2015.
103. C. Tai et al. Convolutional neural networks with low-rank regularization. vol. abs/1511.06067, 2015.
104. C. Louizos et al. Bayesian compression for deep learning. In NIPS, 2017.