Cognitive Computation and Communication: A Complement Solution to Cloud for IoT

Van Tam Nguyen[1,3], Nhan Nguyen-Thanh[1], Lita Yang[3], Duy H. N Nguyen[2], Chadi Jabbour[1], and Boris Murmann[3]
[1] LTCI, CNRS, Télécom ParisTech, Université Paris Saclay, 75013, Paris, France
[2] Department of Electrical and Computer Engineering, San Diego State University, CA 92182, USA
[3] Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA

(Invited Paper)

***Abstract*—The Internet of Thing (IoT) is experiencing explosive growth in the number of devices and applications. However, the existing cloud-centric architecture of IoT poses serious challenges regarding network latency, privacy, and energy-efficiency. We have presented COGNICOM+ concept, a brain-inspired software-hardware paradigm, to support IoT's future growth and developed 4 research directions – flexible radio, convolutional neural network accelerator, compressed deep learning, and game theory for reasoning and collaboration – within COGNICOM+. The key idea is to bring computing closer to the end-user while focusing on optimal uses of local smart application gateway and cloud computing. COGNICOM+ consists of two key components: Cognitive Engine (CE) and Smart Connectivity (SC). The cognitive engine is powered by deep-learning algorithms integrated with game-theoretic decision analytics, implemented on a low-power application-specific integrated circuit. It provides cognitive functions to smart objects. The smart connectivity integrates neural network inspired designs of cognitive radio, transceivers, and baseband processors. The SC provides flexible and reliable connections to IoT objects and optimally distributes communication resources.**

Keywords—COGNICOM; IoT; cognitive engine; smart connectivity; deep learning; CNN accelerator; compressed CNN; game theory; flexible radio; smart application gateway;

I. INTRODUCTION

In the next decades, Internet of Things (IoT), the interconnected networks of physical objects embedded with electronics, software, sensors, and connectivity will revolutionize how we work, live, exercise, entertain, and travel. IoT is experiencing explosive growth in both quantities (20.8 billion IoT devices by 2020 [1]) and utility with increasingly important applications in healthcare, military operations, transportation, and urban planning [2]. However, IoT faces several major growing challenges. First, incorporating appropriate intelligence and smart connectivity into IoT objects requires a computing paradigm that exceeds the current computing capabilities of smart phones and portables [3]. In particular, the cognitive computation emulating some of the brain's abilities has the potential to transform mobility by spurring innovation around an entirely new class of applications with sensory capabilities at incredibly low power levels [4]. Second, ensuring privacy, security, and safety of IoT applications is critically important, as IoT devices are susceptible to external attacks that can cause either leak of private information or dangers to users. Third, many IoT applications have reliability, robustness, and latency requirements, which exceed the current design of wireless communication, and cloud computing. Fourth, since IoT requires processing of large amounts of data from numerous devices, hardware design for IoT applications needs to be not only flexible and adaptive but also highly energy-efficient.

In the current architecture of IoT, cloud computing provides the virtual infrastructure for data collection, analysis, visualization, and service delivery. With the growing number of billions of IoT devices, there will be a great demand on cloud Data Centers (DCs), resulting in massive energy expenditure and emission of CO_2. Today, the DCs are already responsible for about 2% of global greenhouse gas emissions, a similar share to aviation. In 2007, the DCs consumed on the order of 330bn kWh, equivalent to the entire electricity demand of the UK. This demand is projected to triple or quadruple by 2020 and accounts for 1.5-2% of all global electricity demand at a growing rate of 12% per year [5].

Many IoT applications, such as smart vehicular traffic management system, smart driving, and smart grid require real-time and low-latency services. If the processing, computation, and storage of the enormous amount of data are performed only within DCs, the massive data traffic generated from IoT devices will result in network bottlenecks and affect the performance of all IoT applications. With respect to the last challenge, recently Cisco proposed the concept of Fog computing [6] to bring cloud computing closer to the end user by transforming as much as possible data into action at the network edge. The recent work in [7] shows that in the context of IoT with a high number of latency-sensitive applications Fog computing outperforms traditional cloud computing paradigms.

In this paper, we present research directions to address those challenges, providing a complement solution to cloud computing. The proposed concept, COGNICOM+, is a hybrid architecture powered by Cognitive Engine (CE) and Smart Connectivity (SC) that facilitates optimal use of both local smart application gateways and cloud computing. COGNICOM+ shares some common features with the Fog computing concept in terms of bringing computing closer to the end user, however with one key difference: where computation occurs will be decided by the CE to maximize

978-1-5090-2710-1/16/$31.00 ©2016 IEEE

utility, reliability, and privacy, while minimizing latency and energy expenditures of the entire IoT networks.

II. COGNICOM+ CONCEPT

The COGNICOM+ concept, briefly presented in [8], is depicted in Fig. 1. At the heart of the CE are deep-learning algorithms organically integrated with advanced game-theoretic decision analytics to supply cognitive functions for selective smart objects as well as the entire IoT application. An equally important feature of COGNICOM+ is the SC, which enables seamless, energy-efficient and reliable connection to the cloud, smart-objects, and other IoT devices and sensors.

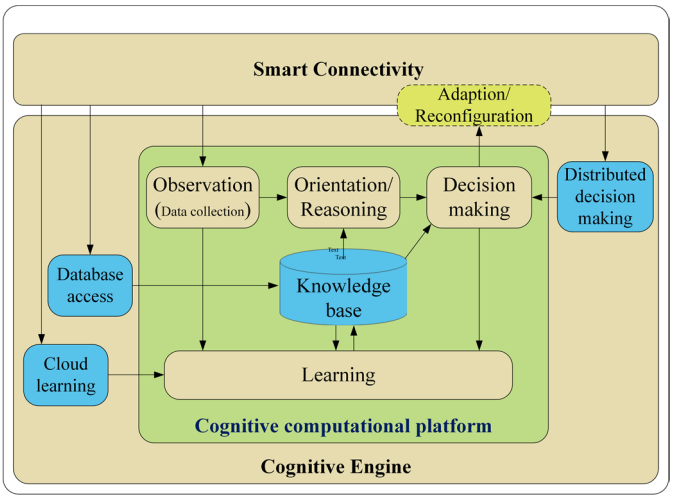

Fig. 1. COGNICOM+ concept.

The CE concept is an extension of our previous research on the cognitive radio [9][10] and is defined as an intelligent agent that manages the cognition tasks. Observation (data collection), which extracts information and knowledge regarding the environment or user, is the foundation of learning and reasoning. The CE then reasons in order to analyze and classify the situation, determine the appropriate response, and carry out the adequate decision. Once decisions are made, the CE adapts and reconfigures its parameters with respect to user-defined objectives. Since the learning engine evaluates the outcomes of the decisions, it is responsible for building up knowledge and context awareness to be exploited in future reasoning phases. It should be powerful enough to enrich the knowledge base, to foster the increased efficiency of the reasoning, and to enhance the decision. As a result, there is a close interaction between learning, knowledge, reasoning, and decision, which complement each other to improve the operation of the system as a whole. The SC enables connections with the cloud and other objects to amplify the capabilities and the value of the CE. It is able to connect to every device everywhere and anytime and will leverage the CE to support Dynamic Spectrum Access (DSA). To harvest the full capacity out of the RF spectrum, the SC based on the flexible radio should bring greater intelligence to avoid interference while maximizing utility. These radios will be able to collaborate directly with their peers to derive stable and satisfactory communications for all. Without strict frequency allocations, by taking the advantage of deep learning and game theory algorithms developed in the CE, radio networks should be able to autonomously collaborate to sense the local RF landscape, reason about how to avoid interference and exploit opportunities to achieve efficient use of the available spectrum. In this way, the SC enables scaling of network capacity, low latency, flexibility, spectrum and energy efficiency, and high reliability.

III. COGNICOM+ IMPLEMENTATION

The IoT hybrid architecture shown in Fig. 2 is inspired by a past trend in mobile communication, where base stations became smaller, less expensive, and more capable over time (micro-cells, pico-cells, and femto-cells). The idea is to move away from cloud computing and Fog computing and to leverage local computing whenever possible. This not only reduces costs, boosts capacity, reduces latency, and speeds up network expansion, but also enhances privacy, security, and reliability of the network. It also improves the sustainable development of the IoT ecosystem. In order to do so, the key enabler is the smart application gateway (SAG), which should be able to perform many tasks that are currently relegated to cloud computing. In addition to its traditional functionalities, SAG will also (1) collect, classify and integrate data; (2) interpret data to generate appropriate responses; and (3) perform adequate actions. The majority of data will be stored and processed in local databases. The interpretation of the data will be performed by the CE, whose deep-learning algorithms will be pre-trained using cloud-based computing. The CE will detect abnormal activities and emergency situations then directly provide appropriate responses. Responsible for timely response services, it decides which data should be sent to the cloud platform for further analysis and interpretation while optimizing the distribution of computing and communication resources. It is also capable of learning to adapt its functionalities, capabilities, and behavior to the environment and user in order to achieve predefined objectives.

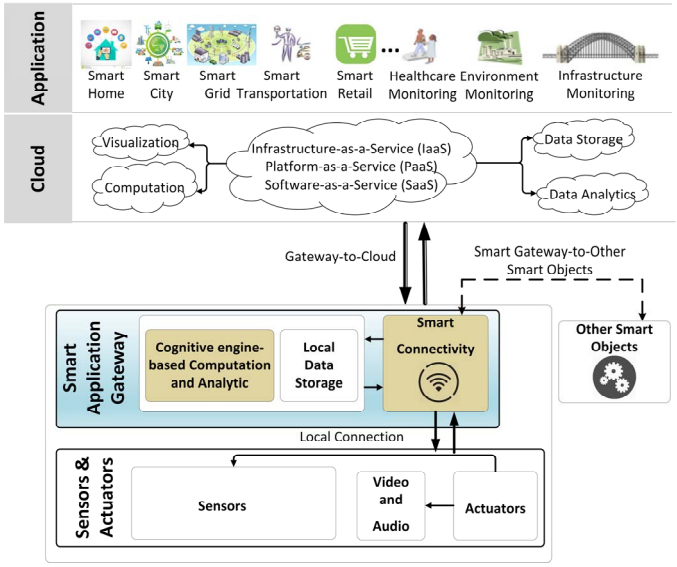

Fig. 2. Local area computing with smart application gateway.

Most notably, the localized storage and processing of the majority of large-scale data will reduce the risk of connection bottleneck. Furthermore, the users can regulate the type and the

amount of information to be processed on the cloud by defining their own level of privacy and information sharing. The SC will provide a reliable and efficient connection through wireless networks, even in disaster situations by implementing DSA. The cognitive radio based on deep learning and game theory is able to establish a connection in any situation and upgrade to future wireless technologies. At the cloud layer, less computation will be executed and at the application layer, services other than timely response service will be provided.

IV. Research Directions

In COGNICOM+, we have identified four main research directions whose objective is to develop the algorithms (compressed deep learning and game theory), architectures, and hardware (flexible radio and deep learning accelerator) to support the CE and the SC.

A. Flexible Radio

The SC should be able to dynamically switch between a multitude of radio communication techniques, which include adaptively changing temporal, spatial, spectral, waveform, protocol, or other aspects of the communication waveform. The radio should therefore be very flexible and support not only multiple standards but also the upcoming ones. A key bottleneck in the flexible radio has always been and continues to be the flexible RF transceiver [9] we will focus on in this section.

The smart connectivity manages the connections to the cloud and to the IoT-connected devices. The connection to the cloud is high-speed in order to transfer a large amount of computing information and/or media data. The connection to the IoT-connected devices can be divided into two categories: high-speed connections for video stream coming from cameras and ultra-low power and low-speed connections with sensors/actuators to transfer measurement and control data.

The high-speed connectivity needs to adopt the wideband standards such as WiFi/LTE/LTE-A/5G and upcoming wideband standards, supporting multi bands and scalable bandwidths. The frequency range starts typically from 400 MHz to profit from the TV white spaces and can go up to 6 GHz to cover the high-frequency band of WiFi. This flexibility is very crucial for the smart application gateway in order to enable opportunistic and dynamic spectrum access for high-speed connections.

The low-speed connectivity to the IoT-connected devices needs to support multi-standard including Zigbee, Bluetooth low energy (BTLE), WiFi (Direct, HaLo), and future M2M and D2D technologies. Therefore, the transceiver should be ultra-low power (sub-mW to 1-2 mW) to low power (several to 10 mW) and multi-mode multi-band (e.g. 450MHz / 900MHz / 2.4GHz / 5.0GHz).

For the transmitter, the most common approach is the direct-conversion architecture [11]. Although it is simple, it suffers from I/Q mismatch, high power consumption due to quadrature LO generator and LO pulling issues. The PLL-based transmitter uses PLL modulation to avoid drift of LO signal [12]. The advantage is its simple baseband processing and non-quadrature LO generator. However, it is only applicable for constant envelope modulation scheme and the modulated signal bandwidth accuracy strongly depends on process, voltage and temperature variations. The polar transmitter uses decomposed phase and envelope signals to control the power amplifier [13], enabling high power efficiency and working with an arbitrarily nonlinear output stage. In contrast, we have to deal with signal mismatch corrupting the output, sensitivity to the linearity of the envelope detector, and limit of the phase detector at high frequency. The PLL-based and polar-based transmitter can take the advantages of the two architectures [14].

Receivers for low power and low data rate connectivity employ Zero-IF or near Zero-IF architectures due to their high flexibility, robustness, and integration [15]. However, they require quadrature LO clock burning a significant amount of power. To overcome this issue, sliding IF has been proposed [16], but very careful frequency planning is needed to avoid imaging issues especially for the multi-band reception. All these architectures still need a high-resolution back-end ADC, i.e., 9-10 bit.

For the high speed and wideband receivers, the blocker tolerant receiver [17] enables 0 dBm blocker tolerant and wide frequency range of 0.4 - 6.0 GHz at the cost of high power consumption. The harmonic rejection receiver decreases blockers at the 3rd and 5th LO harmonics by 30-80 dB rejection [18] at the cost of high power consumption. Passive mixer first receiver gains high linearity by removing the low noise amplifier (LNA) [19]. Recently, noise canceling has been used in this architecture, allowing 1.8-1.9 dB noise figure [20] at the cost of lower IIP3 and extra power consumption. All of these architectures still require a high-performance back-end ADC to complete the receiver chain from RF to digital. To overcome this challenge, a direct RF-to-digital receiver called the delta-sigma receiver [21] has been proposed by integrating the RF stages, i.e., the mixer and the LNA into the delta-sigma loop filter. This results in two main advantages: 1) reduced voltage swing at the RF nodes, leading to linearity improvement and 2) an increased delta-sigma loop order which results in a dynamic range improvement. Continuous-time implementations [22], along with high power consumption due to the fully active loop filter, suffer from the quantization noise-folding problem, which limits the frequency range. In discrete-time implementations [23], the loop filter operates at the LO frequency, which drastically increases the design constraints and power consumption at high LO frequency. Moreover, at low LO speed, the low oversampling ratio causes a significant impact on the noise figure. This limits the frequency range of the architecture to below 4.0 GHz and raises the sensitivity in low-frequency bands. Furthermore, in all the implemented delta-sigma receivers, the channel filtering is insufficient, because of the tight requirement of quantization noise performance.

In COGNICOM+, our research direction on the flexible transceiver is based on an ADPLL-based polar transmitter/RFDAC and a sigma-delta receiver as shown in Fig. 3. The architecture enables flexibility based on a coarsely configurable front-end for matching, filtering, power and low-noise amplification, and a finely configurable signal path for conditioning and down conversion. To fully cover the range of

frequencies mentioned, at least, 4 rows of such elements are included, covering 4 frequency bands: 400MHz-800MHz, 800MHz-1.6GHz, 1.6GHz-3.2GHz, and 3.2GHz-6GHz. This removes the need for custom-designed RF transceivers for each radio system and allows upgrades to future standards.

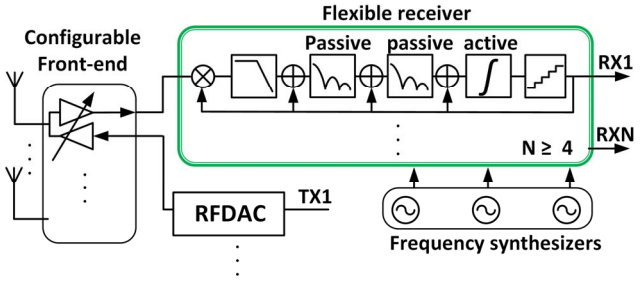

Fig. 3. The architecture of flexible transceiver.

B. CNN Accelerator

A large majority of the data carried in the networks are pixel-based data [24]. Recently, deep learning [25], especially deep convolutional neural networks (CNNs), has achieved state-of-the-art results in computer vision applications ranging from the recognition / detection of objects [26][27][28] to scene understanding [29]. Therefore, deep learning algorithms are relevant for the CE in COGNICOM+ concept. They have, however, steadily grown in computational complexity requiring up to hundreds of megabytes for filter weight storage and billions of operations per second [26]. The large size of these CNNs poses both throughput and energy efficiency challenges to the underlying processing hardware especially in mobile and embedded devices. Current solutions in developing CNN Application-Specific Integrated Circuit (ASIC) accelerators (e.g. [30]) focus on two major problems: 1) the memory bandwidth bottleneck and 2) the high energy cost of data movement [31].

We focus on CNN ASIC accelerators that target ultra-low power applications. To achieve high energy efficiency and reduce the memory bandwidth bottleneck, several recent efforts have proposed specialized architectures for CNN operations, which exploit properties such as data parallelism and parameter reuse to be leveraged at the architecture level and perform accurately even in the presence of circuit errors for significant energy savings.

Techniques for improving CNN ASIC energy efficiency have seen dramatic shifts in design methodologies both at the algorithmic and hardware level. Divergent efforts in machine learning (increasing network sizes) versus hardware (ultra-low power designs) motivated optimizations in reducing memory fetches [32][33]. This lead to efforts in exploring data flows to exploit the inherent data parallelism, reuse, and locality characteristics present in CNNs [34][35] as well as methods for reducing on-chip memory sizes such as compression and coding [36][37]. Error resiliency of CNNs motivated work on quantization and voltage scaling effects on the network [38][39][40][41], eventually reducing precisions down to binary weights and activations [42][43]. We outline key energy-efficient techniques for CNN ASICs: **(1)** reducing memory fetches; **(2)** exploiting data reuse, minimizing partial sum accumulation, and gating operations; **(3)** bit precision quantization and voltage scaling; and **(4)** binarized neural networks.

1. *Reducing memory fetches*: Large CNN models consume large amounts of energy because the model must be stored in external DRAM. The DianNao family of architectures is one of the first post-layout CNN accelerators to address the memory access bottleneck [32]. DianNao, a customized inner-product neural processing unit (NPU), exploits high compute parallelism using direct communication between the NPU array and local buffers. The dominance of off-chip DRAM energy motivated the authors to eliminate off-chip DRAM access by having all weights on-chip (in eDRAM or SRAM). Origami, one of the first silicon CNN accelerators, extends on this idea and further reduces energy by using 12 bits instead of 16 bits while operating at an acceptable lower voltage of 0.8V [33].

2. *Exploiting data reuse, minimizing partial sum accumulations, and gating operations*: Fortunately, CNNs inherently have input data reuse properties which can be exploited in hardware architectural design to optimize data flow. While previous work exploited convolutional and filter reuse [34], or implemented local storage to minimize partial sum accumulation cost [32], Eyeriss introduces a data flow called row-stationary, to exploit both local data reuse of filter weights and feature map pixels while minimizing data movement of partial sum accumulations [35]. The Eyeriss chip is a silicon CNN implementation of the row-stationary dataflow to exploit data reuse and further reduces energy by gating neuron computations to avoid unnecessary reads and computations caused by the sparsity of the ReLu operation.

3. *Bit precision quantization and voltage scaling*: As demonstrated in several notable works [38], quantization and bit precision reduction can be performed in CNNs without significant accuracy degradation due to the error resiliency of the network. The low power CNN processor from [39] expands on methods of data locality and guarding memory fetches and operations from [33] and [35], with precision quantization per layer and voltage scaling methods for significant energy savings. This highly energy-efficient digital CNN silicon accelerator consumes only 25-288mW for LeNet-5 and AlexNet CNN tasks. Beyond conventional digital design, exploiting error resiliency of CNNs in energy-efficient analog/mixed-signal arithmetic [40], low-swing signaling, and low-voltage storage [41] is also an attractive option for aggressively lowering energy consumption. Implications of quantization noise, thermal noise, and bit-flip errors caused by mixed-signal circuit design on the classification accuracy of CNNs have been demonstrated in [38].

4. *Binarized CNN ASIC:* Recently, a binarized CNN with weights and activations constrained to +1 and -1 was introduced with negligible accuracy losses on several CNN benchmarks [42]. This novel algorithm allows for a drastic reduction in on-chip storage and I/O requirements and replaces expensive multiplications with XNORs. YodaNN is a post-place and route convolutional engine based on binarized

CNNs in UMC 65nm using standard cell memory to extend voltage operation down to 0.6V [43]. Currently, there is no published work on silicon BinaryConnect accelerators but it is expected that future implementations will combine aforementioned energy saving techniques for a highly energy-efficient BinaryConnect CNN ASIC.

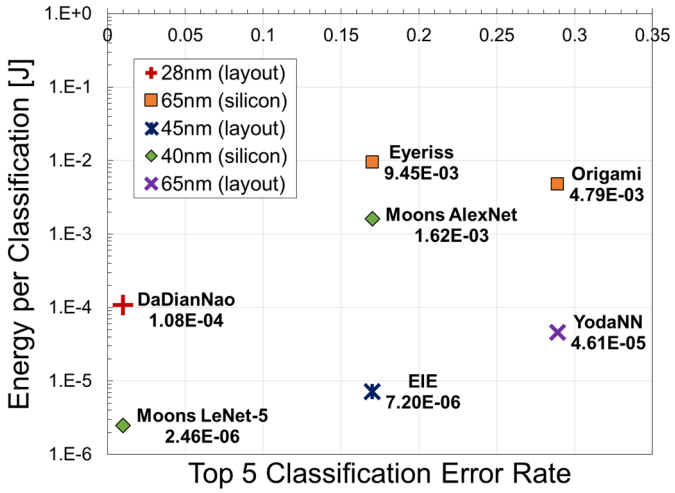

Fig. 4. A literature survey compares energy per classification versus top 5 classification error rate for various tasks including MNIST (error rate:0.01), AlexNet (error rate:0.17), and Stanford Backgrounds Scene Labeling (error rate:0.29).

In an attempt to capture varying CNN sizes and datasets, while highlighting the gains in energy efficiency for each CNN accelerator, we plot energy per classification task versus prediction error rate in Fig. 4. Note that while there are several post-layout simulation-based CNN accelerators, we differentiate between silicon results and post-layout simulations for comparison since post-layout simulation results are known to be overly optimistic. In COGNICOM+, the research direction is to leverage CNN characteristics including parallelism, reuse, locality, and error resilience in energy-efficient architectural and circuit design, and restructure the algorithm to be more conscious of hardware design implications in order to fit to custom hardware. To further reduce energy and silicon area, we can compress the CNN algorithm model and accelerate the sparse matrix-vector multiplication to fit the whole network-on-chip. For example, the Energy-Efficient Inference Engine (EIE) is a post-layout ASIC that implements deep compression with no loss of accuracy on AlexNet and VGG-16 networks [36].

With the combination of algorithm exploitation in both architecture and circuit design, we can achieve the greatest potential in energy savings to enable deployment of CNNs in IoT and mobile platforms.

C. *Compressed Deep CNN*

Deep CNNs provide the ability of automatically detecting features/patterns using inexpensive unlabeled data and have become the state-of-the-art technique for some cognitive tasks [25]. Moreover, they can learn features over multiple modalities where cross-modality feature learning achieves a better result [44]. The state-of-the-art solutions are over-parameterized to facilitate convergence during training [45]. Therefore, current implementations of CNNs demand substantial storage capability, memory bandwidth, and computational resources, making them prohibitive for embedded systems [46]. The convolutional layers mainly consume run time (about 90%) and the densely connected layers take the majority of the storage (>90%) [46]. Another issue is their fixed architecture, which cannot be enhanced during the training phase [47]. To address these problems, we need to reduce the storage and computation requirements without reducing the performance while enhancing the architecture during the training phase. This is motivated by the non-uniform distributions of weights and activations and the many redundant connections in CNNs. We outline key techniques for CNN compression: **(1)** weight/activation compression; **(2)** model compression; and **(3)** CNN computation acceleration, especially in convolution layers.

1. *Weight/activation compression*: A trained CNN is essentially defined by weights. Obviously, compressing the stored weights by quantization and sharing reduces its memory size. In [48][49], the pre-trained weights of deep CNNs are quantized by ternary representation (+1, 0 and -1) and retrained with 3-bit fixed-point activations. Quantizing the pre-trained weights is effective in reducing memory size, however, it still requires high computation of precise pre-training and additional fine-tuning. The binarized CNN goes further in this direction by constraining the weights and activations to +1 and -1, with negligible accuracy losses [42]. In [50][51], the weight vector quantization is adopted during the training process. A shared weight codebook for quantization is built up based on the centroid grouping method and trained during backpropagation process. The gradients of weights are also grouped for training the codebook, which reduces backpropagation computation. The shared weight approach is also found in HashedNets [52]. The compression of HashedNets is implemented in the frequency domain by using a hash function to randomly cluster connection weights. The same weight value is assigned to all connections within the same hash bucket. The hash function parameters are learned through backpropagation training instead of weights. However, the weight values in the codebook in HashedNets are not built upon the distribution of weights as the same as those in [51]. Therefore, HashedNets cannot be further compressed using encoding representation.

Similar to weight compression, input feature maps can be also compressed to reduce required computational resources for operation. Specifically, the substitution of floating-point units with fixed-point comes with significant gains in the energy efficiency and computational throughput. A fixed-point implementation with 8-bit integer (vs 32-bit floating point) activations has been demonstrated in [53]. Deep networks can be trained using only 16-bit wide fixed-point number representation when using stochastic rounding and incur little to no degradation in the classification accuracy [54]. Another work which implements log-based coding on data representation with non-uniformly distributed values reduces bit precision to 3-bits and eliminates expensive multipliers for simple bit shifts [37].

2. *Model compression*: Varying the structure of the network can also reduce its size. For example, the fully connected layer is replaced by global average pooling [55][56]. The network complexity is reduced by pruning unimportant connections [57], resulting in a significant reduction for several state-of-the-art large scale networks. For example, Deep Compression uses pruning to learn only important connections, trained quantization, and Huffman coding to reduce the storage requirement of CNNs from 34-49x while improving energy efficiency by 3-7x [51]. A recent work on model compression, SqueezeNet, has discovered an architecture with fewer parameters but equivalent accuracy compared to a well-known model [58]. The essence of SqueezeNet remains even with the decrease of the number of input channels and the free use of smaller convolution filters (1x1 filters) constructing CNN microarchitectures called Fire modules. The configuration of those microarchitectures is adjusted by changing combinations of small filters (1x1 and 3x3). These smaller modules build up to construct the whole CNN macroarchitecture. In GoogLeNet [56][59], a similar idea of dividing a large neural network into smaller modules called Inception modules is also adopted. SqueezeNet achieves AlexNet-level accuracy on ImageNet with 50x fewer parameters. Additionally, it is compressed to less than 0.5MB (510x smaller than AlexNet).

3. *CNN computation acceleration*: By investigating the computation in convolutional layers, we can speed up CNN computation, because these layers dominate CNN run time [46]. In [60], the redundancy present within the convolutional filters is exploited with linear compression techniques to derive approximations that significantly reduce the required computation. The proposed solution consists of compressing each convolutional layer by finding an appropriate low-rank approximation and then fine-tuning the upper layers until the prediction performance is restored. Several elementary tensor decompositions based on singular value decompositions, as well as filter clustering methods are considered to take advantage of similarities between learned features. Similar low-rank decomposition of convolutional kernel tensor was proposed in [61][62]. Because of significant redundancy between different filters and feature channels, a learned full rank filter bank can be approximated as combinations of a rank-1 filter basis. These approximations require significantly fewer operations to compute, resulting in large speedups [61]. In [63], the decomposition of DxD convolution into Dx1, 1xD and 1x1 convolutions to accelerate convolutional layers combined with an optimal rank selection based on accumulated principal component analysis was proposed. Tucker decomposition is also used to compress the entire convolutional and fully connected layers [64]. Convolution can be efficiently computed in the Fourier domain, where it becomes element-wise multiplication [65][66]. Most importantly, the FFT method can be used jointly with most of the techniques presented in this section.

The research direction in COGNICOM+ is to explore the broad range of possibilities in the design space of CNN architectures. We will investigate the design choices in microarchitecture, macroarchitecture, weight/activation and model compressions, and computation acceleration.

D. Reasoning and Collaboration with Game Theory

After collecting data, analyzing the situation, and deciding an appropriate response to maximize certain user-defined objectives, the CE helps the SC adapt and reconfigure its wireless access parameters. Through spectrum sensing, learning, reasoning, and dynamic spectrum access, the SC is capable of performing three main functionalities: 1) autonomously collaborate to sense the local RF landscape and detect available spectrum holes unoccupied by primary spectrum users; 2) reason about how to avoid interference and exploit opportunities to achieve efficient use of the available spectrum; and 3) adaptively and dynamically transmit and receive data in a changing radio environment.

Game theory, an extremely powerful tool to analyze the interactions between competing and rational agents [67], is adopted to model collaboration and derive reasoning in COGNICOM+. Specifically, we consider each SAG as an agent in a multi-agent game. In noncooperative strategic games (NSG), each agent is a rational player who selfishly maximizes his or her own user-defined utility, regardless of the actions of other players. We denote Ω as the set of players, s_i as the strategy of player-i within a set of admissible strategies S_i, and s_{-i} as the strategy of all other players except player-i. The strategy s_i of a player is specified accordingly to the user-defined utility. Examples of a player strategy include channel allocations for multiple end users connected with the SAG, and power allocations and/or rate allocations for the end users. Effectively, a player yields a user-defined utility $U_i(s_i, s_{-i})$, which is dependent on the strategies of all players due to their mutual interactions. Examples of user-defined utility include achievable throughput, packet delivery rate, and energy efficiency. Mathematically, a NSG can be now modeled as $\mathcal{G} = \{\Omega, \{S_i\}_{i\in\Omega}, \{U_i(s_i, s_{-i})\}_{i\in\Omega}\}$. Note that each player has to function within a user-specified strategy set, which may include constraints on their transmit power, data rate, selectable channels and interference induced the primary networks.

When acting selfishly, each player searches for their optimal strategy s_i^* by solving

$$\max_{s_i \in S_i} U_i(s_i, s_{-i}).$$

Typically, $U_i(s_i, s_{-i})$ is modeled as a concave function on s_i but not necessarily concave on s_{-i}. An example of such an utility function is the multi-channel sum data rates in water-filling games [68]. A key concept in NSG is the Nash Equilibrium (NE) [69], where it is defined as

$$\mathrm{U_i}(\mathrm{s_i^*}, \mathrm{s_{-i}^*}) \geq \mathrm{U_i}(s_i, s_{-i}^*), \forall s_i \in S_i, \forall i \in \Omega.$$

Effectively, NE - (s_i, s_{-i}) - is a stable operating point of the system where no player has the incentive to literally deviate from it. Therefore, we analyze the existence and uniqueness of the NE. If $U_i(s_i, s_{-i})$ is quasi-concave on s_i, the existence of the game's NE is always guaranteed [70]. In a NSG, each player can adjust its allocation strategy in fully distributed fashion without any coordination until the game converges to a NE. However, it is well known that the NE need not be Pareto-efficient [71]. In addition, all the players may collectively exceed the amount of allowable interference induced to the primary networks.

A joint optimization approach can be used to solve the resource allocation problem in COGNICOM+, all APs aim to maximize a common utility function $U(s_i, s_{-i}) = \sum_i w_i U_i(s_i, s_{-i})$, where $w_i \geq 0$ is the weight of agent-i. The achievable utility, usually being Pareto-optimal, then serves as the benchmark for the efficiency of the NE in the game with the corresponding utility functions $U_i(s_i, s_{-i})$'s. The main advantage of the joint optimization approach is that it allows all the users to collaboratively optimize their strategies. In addition, the approach enables a dynamic allocation of the interference temperature budget among SAGs. Unfortunately, the joint optimization approach may demand a centralized unit to find the optimal resource allocation strategy for all SAGs.

To circumvent the centralized approach, we will investigate a distributed optimization strategy in COGNICOM+, where each SAG individually optimizes its strategy in a more cooperative manner through a modified utility $\widetilde{U}_i(s_i, s_{-i})$. One approach composing such utility is to approximate the common utility $U(s_i, s_{-i})$ into a concave function in s_i, where

$$U(s_i, s_{-i}) = w_i U_i(s_i, s_{-i}) + \sum_{j \neq i} U_j(s_i, s_{-i}).$$

The new utility $\widetilde{U}_i(s_i, s_{-i})$ usually comprises of two components: one is agent-i's own utility $U_i(s_i, s_{-i})$ and the other is the approximation of other agent utilities $\sum_{j \neq i} U_j(s_i, s_{-i})$. Since $U_j(s_i, s_{-i})$ is typically non-concave in s_i, taking the Taylor series expansion and retaining only the linear term, $\sum_{j \neq i} U_j(s_i, s_{-i})$ can be approximated into $-p_i f_i(s_i, s_{-i})$, which is linear in s_i. The common utility is then approximated as

$$U(s_i, s_{-i}) \approx w_i U_i(s_i, s_{-i}) - p_i f_i(s_i, s_{-i}) \triangleq \widetilde{U}_i(s_i, s_{-i}).$$

With certain utility functions U_j's, $f_i(s_i, s_{-i})$ resembles the interference induced by SAG-i to the users at other SAGs, while p_i represents the pricing factor on the induced interference [72][73]. Thus, by adopting the new utility $\widetilde{U}_i(s_i, s_{-i})$, we introduce collaboration between the SAGs. In particular, SAG-i now sets the balance between the selfish maximization of its own utility $U_i(s_i, s_{-i})$ and the cooperative minimization of the interference to nearby SAGs. With the right pricing factors, the SAGs can collectively drive their strategy to a locally optimal solution of network-wide sum utility $U(s_i, s_{-i})$ maximization [73]. In COGNICOM+, we will define a controller, which computes and distributes the pricing factors to these SAGs. The majority of computations on updating the allocation strategy are distributively performed at the SAGs.

V. Conclusion

The existing cloud-centric architecture of IoT poses serious challenges regarding cognitive capacity, connectivity, safety, privacy, flexibility, latency, and energy-efficiency. We have presented the COGNICOM+ concept, a brain-inspired software-hardware paradigm, to support IoT's future growth. Consisting of the cognitive engine and the smart connectivity, COGNICOM+ brings computing closer to end-user and focuses on optimal uses of local SAG and cloud computing. The CE is powered by deep learning algorithms integrated with game-theoretic decision analytics, implemented on low power ASICs. It provides cognitive functions to smart objects. The SC integrates neural network inspired designs of the cognitive radio (CR), transceiver, and baseband processor. It provides flexible and reliable connections to IoT objects and optimally distributes communication resources. We have also shown the SAG based IoT hybrid architecture to leverage local computing whenever possible. The SAG performs many tasks that are currently relegated to cloud computing (e. g. collecting, classifying, and integrating data from the sensing layer, interpreting data, performing appropriate responses). It is also responsible for timely response services and the decision of which data should be sent to the cloud platform for further analytic and interpretation.

We have also presented state of the art works and research directions related to 4 topics in COGNICOM+: flexible radio, CNN accelerator, compressed deep learning, and game theory based collaboration and reasoning.

Acknowledgment

The research leading to these results has received funding from the CATRENE project CORITF and the Senior Marie Curie Fellowship CORPA.

References

[1] Available: http://www.gartner.com/newsroom/id/3165317

[2] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, "Vision and challenges for realising the Internet of Things," *vol. 20, no. 10. CERP-IoT* – Cluster of European Research Projects on the Internet of Things, 2010.

[3] J. Gubbi, R. Buyyaa, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[4] P. A. Merolla *et al.,* "A million spiking-neuron integrated circuit with a scalable communication network and interface", *Science*, vol. 345, no. 6197, pp. 668-673, 2014.

[5] G. Cook and J. Van Horn, *"How dirty s your data? A look at the Energy Choices That Power Cloud Computing,"* Greenpeace International. http://www.greenpeace.org/international/Global/international/publications/climate/2011/Cool%20IT/dirty-data-report-greenpeace.pdf

[6] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 13–16.

[7] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the Suitability of Fog Computing in the Context of Internet of Things," *Cloud Computing, IEEE Transactions on*, no. 99, pp. 1–1, 2015.

[8] V. T. Nguyen, "Cognitive computation and communication for IoT," in *Spring 2016 European Energy Innovation COMMUNICATION,* 2016, pp. 44–45.

[9] V. T. Nguyen, F. Villain, and Y. Le Guillou, "Cognitive radio RF: overview and challenges," *VLSI Design*, vol. 2012, p. 1, 2012.

[10] N. Nguyen-Thanh, T. A. Pham, and V.-T. Nguyen, "Medium access control design for cognitive radio networks: a survey," *IEICE Transactions on Communications*, vol. 97, no. 2, pp. 359–374, 2014.

[11] A. He, J. Gaeddert, K. K. Bae, T. R. Newman, J. H. Reed, L. Morales, and C.-H. Park, "Development of a case-based reasoning cognitive engine for ieee 802.22 wran applications," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 13, no. 2, pp. 37–48, 2009.

[12] Y.-H. Liu and T.-H. Lin, "A wideband PLL-based G/FSK transmitter in 0.18 m CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 9, pp. 2452–2462, 2009.

[13] M. Youssef, A. Zolfaghari, B. Mohammadi, H. Darabi, A. Abidi, and others, "A low-power GSM/EDGE/WCDMA polar transmitter in 65-nm

CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 46, no. 12, pp. 3061–3074, 2011.

[14] J. Chen, L. Rong, F. Jonsson, G. Yang, and L.-R. Zheng, "The design of all-digital polar transmitter based on ADPLL and phase synchronized ΔΣ modulator," *Solid-State Circuits, IEEE Journal of*, vol. 47, no. 5, pp. 1154–1164, 2012.

[15] M. Vidojkovic, X. Huang, X. Wang, C. Zhou, A. Ba, M. Lont, Y.-H. Liu, P. Harpe, M. Ding, B. Busze, and others, "9.7 A 0.33 nJ/b IEEE802. 15.6/proprietary-MICS/ISM-band transceiver with scalable data-rate from 11kb/s to 4.5 Mb/s for medical applications," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, 2014, pp. 170–171.

[16] L. Zhang, H. Jiang, J. Wei, J. Dong, F. Li, W. Li, J. Gao, J. Cui, B. Chi, C. Zhang, and others, "A reconfigurable sliding-IF transceiver for 400 MHz/2.4 GHz IEEE 802.15. 6/ZigBee WBAN hubs with only 21% tuning range VCO," *Solid-State Circuits, IEEE Journal of*, vol. 48, no. 11, pp. 2705–2716, 2013.

[17] J. Zhou, P. Kinget, and H. Krishnaswamy, "A blocker-resilient wideband receiver with low-noise active two-point cancellation of 0 dBm TX leakage and TX noise in RX band for FDD/co-existence," *IEEE ISSCC Dig. Tech. Papers*, pp. 352–353, 2014.

[18] D. Murphy, H. Darabi, and H. Xu, "A Noise-Cancelling Receiver Resilient to Large Harmonic Blockers," *IEEE JOURNAL OF SOLID-STATE CIRCUITS,*, vol. 50, no. 6, pp. 1336–1350, Jun. 2015.

[19] M. Soer, E. A. Klumperink, Z. Ru, F. E. van Vliet, and B. Nauta, "A 0.2-to-2.0 GHz 65nm CMOS receiver without LNA achieving 11dBm IIP3 and 6.5 dB NF," in *Solid-State Circuits Conference-Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, 2009, pp. 222–223.

[20] H. Hedayati, W.-F. A. Lau, N. Kim, V. Aparin, and K. Entesari, "A 1.8 dB NF Blocker-Filtering Noise-Canceling Wideband Receiver With Shared TIA in 40 nm CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 50, no. 5, pp. 1148–1164, May 2015.

[21] M. T. Nguyen, C. Jabbour, M. Homayouni, D. Dupperay, P. Triaire, and V. T. Nguyen, "System Design for Direct RF-to-Digital ΔΣ Receiver," *IEEE Transaction on Circuit and Systems - I,* in press.

[22] M. Englund, K. B. Ostman, O. Viitala, M. Kaltiokallio, K. Stadius, K. Koli, and J. Ryynanen, "A Programmable 0.7–2.7 GHz Direct Receiver in 40 nm CMOS," *Solid-State Circuits, IEEE Journal of*, vol. 50, no. 3, pp. 644–655, 2015.

[23] C. Wu, E. Alon, and B. Nikolic, "A wideband 400 MHz-to-4 GHz direct RF-to-digital multimode receiver," *Solid-State Circuits, IEEE Journal of*, vol. 49, no. 7, pp. 1639–1652, 2014.

[24] Available:http://www.bsa.org/~/media/Files/StudiesDownload/bsadatastudy_en.pdf

[25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, 2015.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE CVPR*, 2016.

[28] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *IEEE CVPR*, 2014.

[29] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning Deep Features for Scene Recognition using Places Database," in *NIPS*, 2014.

[30] F. Conti and L. Benini, "A ultra-low-energy convolution engine for fast brain-inspired vision in multicore clusters," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, 2015, pp. 683–688.

[31] M. Horowitz, "Computing's energy problem (and what we can do about it)," in *IEEE ISSCC*, 2014.

[32] Z. Du, R. Fasthuber, T. Chen et al., "ShiDianNao: Shifting Vision Processing Closer to the Sensor," in *Proc. ACM/IEEE Int. Symp. Comput. Architecture*, 2015.

[33] L. Cavigelli, D. Gschwend, C. Mayer et al., "Origami: A Convolutional Network Accelerator," in *Proc. ACM Gt. Lakes Symp. VLSI. ACM Press*, 2015, pp. 199–204.

[34] S. Park, K. Bong, D. Shin, J. Lee, S. Choi, and H.-J. Yoo, "A 1.93TOPS/W scalable deep learning/inference processor with tetra-parallel MIMD architecture for big-data applications," in *IEEE ISSCC*, 2015.

[35] Y. H. Chen, T. Krishna, J. Emer, and V. Sze, "14.5 Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," in *ISSCC-16*, Jan 2016, pp. 262–263.

[36] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. Horowitz, and W. Dally, "EIE: Efficient inference engine on compressed deep neural network," in *ISCA*, 2016.

[37] D. Miyashita, E.H. Lee, and B. Murmann, "Convolutional Neural Networks using Logarithmic Data Representation," *arXiv:1603.01025 [cs.NE]*, March 2016.

[38] B. Murmann, D. Bankman, E. Chai, D. Miyashita, and L. Yang, "Mixed-Signal Circuits for Embedded Machine-Learning Applications," *Asilomar Conference on Signals, Systems and Computers*, Asilomar, CA, Nov. 2015.

[39] B. Moons, M. Verhelst, "A 0.3-2.6 TOPS/W Precision-Scalable Processor for Real-Time Large-Scale ConvNets," in *Symp. On VLSI Circuits*, Jun. 2016.

[40] D. Bankman and B. Murmann, "Passive charge redistribution digital-to-analogue multiplier," *Electronics Letters*, vol. 51, no. 5, pp. 386-388, March 5 2015.

[41] B. Reagen, P. Whatmough, R. Adolf, S. Rama, H. Lee, S. K. Lee, J. M. Hernandez-Lobato, G.-Y. Wei, and D. Brooks, "Minerva: Enabling Low-Power, Highly-Accurate Deep Neural Network Accelerators," in *ISCA*, 2016.

[42] M. Courbariaux and Y. Bengio, "BinaryNet: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1," *arXiv*, 2016.

[43] R. Andri, L. Cavigelli, D. Rossi, and L. Benini, "YodaNN: An Ultra-Low Power Convolutional Neural Network Accelerator Based on Binary Weights," *arXiv*, 2016.

[44] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.

[45] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and others, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[46] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014*, Springer, 2014, pp. 818–833.

[47] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both Weights and Connections for Efficient Neural Network," in *Advances in Neural Information Processing Systems*, 2015, pp. 1135–1143.

[48] K. Hwang and W. Sung, "Fixed-point feedforward deep neural network design using weights -1, 0, and 1," in *2014 IEEE Workshop on Signal Processing Systems (SiPS)*, Oct 2014, pp. 1–6.

[49] S. Anwar, K. Hwang, and W. Sung, "Fixed point optimization of deep convolutional neural networks for object recognition," in *2015 IEEE ICASSP*, 2015, pp. 1131–1135.

[50] Y. Gong, L. Liu, M. Yang, and L. D. Bourdev, "Compressing deep convolutional networks using vector quantization," *CoRR*, vol. abs/1412.6115, 2014.

[51] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and Huffman coding," *CoRR*, vol. abs/1510.00149, 2015.

[52] W. Chen, J. Wilson, S. Tyree, K. Q. Weinberger, and Y. Chen, "Compressing convolutional neural networks in the frequency domain," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD '16*. New York, NY, USA: ACM, 2016, pp. 1475–1484.

[53] V. Vanhoucke, A. Senior, and M. Z. Mao, "Improving the speed of neural networks on cpus," in *Proc. Deep Learning and Unsupervised Feature Learning NIPS Workshop*, 2011.

[54] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," *CoRR*, abs/1502.02551, vol. 392, 2015.

[55] M. Lin, Q. Chen, and S. Yan, "Network in network," *CoRR*, vol. abs/1312.4400, 2013.

[56] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[57] Y. LeCun, J. S. Denker, S. A. Solla, R. E. Howard, and L. D. Jackel, "Optimal brain damage." in NIPs, vol. 2, 1989, pp. 598–605.

[58] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size," *CoRR*, vol. abs/1602.07360, 2016.

[59] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015.

[60] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, "Exploiting linear structure within convolutional networks for efficient evaluation," in Advances in Neural Information Processing Systems, 2014, pp. 1269–1277.

[61] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Speeding up convolutional neural networks with low rank expansions," *arXiv preprint arXiv:1405.3866*, 2014.

[62] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky, "Speeding-up convolutional neural networks using fine-tuned CP decomposition," *arXiv preprint arXiv:1412.6553*, 2014.

[63] X. Zhang, J. Zou, X. Ming, K. He, and J. Sun, "Efficient and accurate approximations of nonlinear convolutional networks," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1984–1992.

[64] Y. Kim, E. Park, S. Yoo, T. Choi, L. Yang, and D. Shin, "Compression of deep convolutional neural networks for fast and low power mobile applications," *CoRR*, vol. abs/1511.06530, 2015.

[65] M. Mathieu, M. Henaff, and Y. LeCun, "Fast training of convolutional networks through ffts," *arXiv preprint arXiv:1312.5851*, 2013.

[66] N. Nguyen-Thanh, H. Le-Duc, D.-T. Ta, and V.-T. Nguyen, "Energy efficient techniques using fft for deep convolutional neural networks," in *IEEE ATC'16*, 2016

[67] D. Fudenberg and J. Tirole, *Game Theory*. MA, USA: MIT Press, 1991.

[68] W. Yu, G. Ginis, and J. M. Cioffi, "Distributed multiuser power control for digital subscriber lines," *IEEE J. Select. Areas in Commun*., vol. 20, no. 5, pp. 1105–1115, Jun. 2002.

[69] J. F. Nash, "Equilibrium points in n-person games," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 36, no. 1, pp. 48–49, 1950.

[70] J.B.Rosen,"Existence and uniqueness of equilibrium points for concave N-person games," *Econometrica*, vol. 33, no. 3, pp. 520–534, Jul. 1965.

[71] P. Dubey, "Inefficiency of Nash equilibria," Math. Oper. Res., vol. 11, no. 1, pp. 1–8, Feb. 1986.

[72] D. H. N. Nguyen and T. Le-Ngoc, "Multiuser downlink beamforming in multicell wireless systems: A game theoretical approach," *IEEE Trans. Signal Process*., vol. 59, no. 7, pp. 3326–3338, Jul. 2011.

[73] D. H. N. Nguyen, H. Nguyen-Le, and T. Le-Ngoc, "Block-diagonalization precoding in a multiuser multicell MIMO system: Competition and coordination," *IEEE Trans. Wireless Commun*., vol. 13, no. 2, pp. 968–981, Feb. 2014.