EMBEDDED DEEP LEARNING IN IOT

Uma Sai Chaitanya Khandavalli[1], Ginnaram Nishanth Goud[2], Shruti Bhargava Choubey[3]
[1, 2]Department of Electronics and Communication Engineering, Sreenidhi Institute of Science and Technology, 501301, Telangana, India
[3]Associate Professor, Department of Electronics and Communication Engineering, Sreenidhi Institute of Science and Technology, 501301, Telangana, India
[1]chaitusai17@gmail.com
[2]ginnaramnishanth@gmail.com
[3]shruthibhargava@sreenidhi.edu.in

***Abstract*— The proliferation of IoT devices heralds the emergence of intelligent embedded ecosystems that can collectively learn and that interact with humans in a human-like fashion. Recent advances in deep learning revolutionized related fields, such as vision and speech recognition, but the existing techniques remain far from efficient for resource-constrained embedded systems. This dissertation pioneers a broad research agenda on Deep Learning for IoT. By bridging state-of-the-art IoT and deep learning concepts, I hope to enable a future sensor-rich world that is smarter, more dependable, and friendlier, drawing on foundations borrowed from areas as diverse as sensing, embedded systems, machine learning, data mining, and real-time computing. Collectively, this dissertation addresses five research questions related to architecture, performance, predictability and implementation. First, are current deep neural networks fundamentally well-suited for learning from time-series data collected from physical processes, characteristic to IoT applications? If not, what architectural solutions and foundational building blocks are needed? Second, how to reduce the resource consumption of deep learning models such that they can be efficiently deployed on IoT devices or edge servers? Third, how to minimize the human cost of employing deep learning (namely, the cost of data labeling in IoT applications)? Fourth, how to predict uncertainty in deep learning outputs? Finally, how to design deep learning services that meet responsiveness and quality needed for IoT systems? This dissertation elaborates on these core problems and their emerging solutions to help lay a foundation for building IoT systems enriched with effective, efficient, and reliable deep learning models.**

Keywords— Embedded ecosystems, Machine learning, IoT, deep neural networks

I. INTRODUCTION

Deep learning has recently become immensely popular for image recognition, as well as for other recognition and pattern matching tasks in, e.g., speech processing, natural language processing, and so forth. The online evaluation of deep neural networks, however, comes with significant computational complexity, making it, until recently, feasible only on power-hungry server platforms in the cloud. In recent years, we see an emerging trend toward embedded processing of deep learning networks in edge devices: mobiles, wearables, and Internet of Things (IoT) nodes. This would enable us to analyze data locally in real time, which is not only favorable in terms of latency but also mitigates privacy issues. Yet evaluating the powerful but large deep neural networks with power budgets in the milliwatt or even microwatt range requires a significant improvement in processing energy efficiency. To enable such efficient evaluation of deep neural networks, optimizations at both the algorithmic and hardware level are required. This article surveys such tightly interwoven hardware-software processing techniques for energy efficiency and shows how implementation driven algorithmic innovations, together with customized yet flexible processing architectures, can be true game changers. To fully understand the implementation challenges as well as opportunities for deep neural network algorithms, this paper briefly summarizes the basic concept of deep neural networks.

Extracting user behavior and ambient context from sensor data is a key enabler for mobile and Internet-of-Things (IoT) applications. Increasingly, emerging networked appliances (e.g., [4, 3]) monitor user activities (such as, speech, occupancy, motion) to provide an improved user experience. Similarly, for wearables and phones the tracking of the user (e.g., [2]) and surrounding conditions (e.g., [5]) has long been a core building block. Even though sensor applications and systems are highly diverse, a prominent unifying element is their need to make these types of sensor inferences. Reliably mining real-world sensor data for this type of information remains an open problem. The world is dynamic and complex; such conditions often confuse the signal processing and machine learning techniques employed for sensor inference. The most promising approach for coping with this challenge today is deep learning [9, 14]. Advances in this field of machine learning have transformed how many inference tasks related to IoT and mobile applications are performed (e.g., speech [7] and face [6] recognition). Exploration of deep learning for these systems is now underway (e.g., [11, 12, 13]), with promising early results.

Despite its benefits, the adoption of deep learning within IoT and mobile hardware face significant barriers due to the resource requirements of these algorithms. Demands on memory, computation and energy make it impractical or most models to directly execute on target hardware. As a result, prominent examples of deep learning seen on phones (e.g., speech recognition) are largely cloud assisted. This introduces important negative side-effects: first, it exposes users to privacy dangers [8] as sensitive data (e.g., audio) is processed o_-device by a third party; and second, the inference execution becomes coupled to fluctuating and unpredictable network quality (e.g., latency, throughput). Enabling wide-spread device-side deep learning inference will require a range of brand-new techniques for optimized resource sensitive execution. Our existing knowledge of deep learning algorithm behavior on constrained devices is largely limited to one-one task specific experiences (e.g., [1, 10]). Such examples only offer a proof-by-example that forms of local execution are possible, while providing a few pointers for potential directions. What is needed is the development of techniques like o_-line model optimization and runtime execution environments that shape inference-time requirements to match the resources available on target wearable, mobile or embedded platforms. A cornerstone of such efforts will be a detailed understanding of how existing algorithms perform on these platforms. Furthermore, systematic observations of deep learning runtime behavior (e.g., data/control flow) will be pivotal for understanding how to best use upcoming hardware accelerators (e.g., [11]) that perform key phases of these algorithms (e.g., convolution layers).

II. Deep Learning In Sensor Rich IoT Systems

A key research challenge towards the realization of learning-enabled IoT systems lies in the design of deep neural network structures and basic building blocks that can effectively estimate outputs of interest from noisy time-series multi-sensor measurements. Despite the large variety of embedded and mobile computing tasks in IoT contexts, one can generally categorize them into two common subtypes: estimation tasks and classification tasks, depending on whether prediction results are continuous or categorical, respectively. The question therefore becomes whether or not a general neural network architecture exists that can effectively learn the structure of models needed for estimation and classification tasks from sensor data. Such general deep learning neural network architecture would, in principle, overcome disadvantages of today's approaches that are based on analytical model simplifications or the use of hand-crafted engineered features.

Traditionally, for estimation-oriented problems, such as tracking and localization, sensor inputs are processed based on the physical models of the phenomena involved. Sensors generate measurements of physical quantities such as acceleration and angular velocity. From these measurements, other physical quantities are derived (such as displacement through double integration of acceleration over time). However, measurements of commodity sensors are noisy. The noise in measurements is nonlinear and may be correlated over time, which makes it hard to model. It is therefore challenging to separate signal from noise, leading to estimation errors and bias. For classification-oriented problems, such as activity and context recognition, a typical approach is to compute appropriate features derived from raw sensor data. These handcrafted features are then fed into a classifier for training. Designing good hand-crafted features can be time consuming; it requires extensive experiments to generalize well to diverse settings such as different sensor noise patterns and heterogeneous user behaviors.

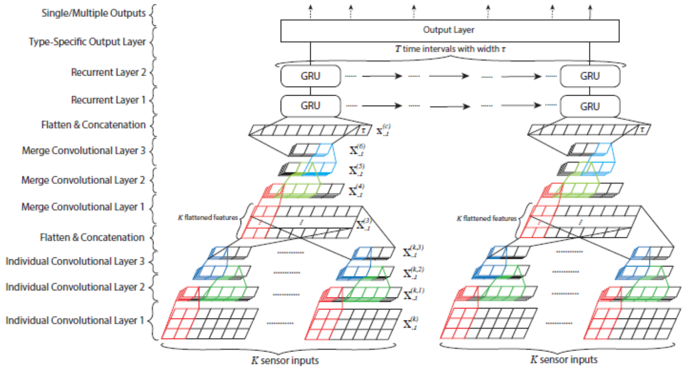

Figure i. Main Architecture of the deep sense network

This architecture solves the general problem of learning multi-sensor fusion tasks for purposes of estimation or classification from time-series data. For estimation-oriented problems, DeepSense learns the physical system and noise models to yield outputs from noisy sensor data directly. The neural network acts as an approximate transfer function. For classification-oriented problems, the neural network acts as an automatic feature extractor encoding local, global, and temporal information. As a unified model, DeepSense can be easily customized for a specific IoT application. The application designer needs only to decide on the number of sensory inputs, input/output dimensions, and the training objective function.

III. Deep Learning In Resource-Constrained IoT Systems

Resource constraints of IoT devices remain an important impediment towards deploying deep learning models. A key question is therefore whether or not it is possible to compress deep neural networks, such as those described in the previous section, to a point where they fit comfortably on low-end embedded devives, enabling real-time \intelligent" interactions with their environment. Can a unified approach compress commonly used deep learning structures, including fully-connected, convolutional, and recurrent neural networks, as well as their combinations? To what degree does the resulting compression reduce energy, execution time, and memory needs in practice? We prosed such a compression framework, called DeepIoT, which compresses commonly used deep neural network structures for sensing applications through deciding the minimum number of elements in each layer. Previous illuminating studies on neural network compression sparsely large dense parameter matrices into large sparse matrices [15]. In contrast, DeepIoT minimizes the number of elements in each layer, which results in converting parameters into a set of small dense matrices. A small dense matrix does not require additional storage for element indices and is efficiently optimized for processing. DeepIoT greatly reduces the effort of designing efficient neural structures for sensing applications by deciding the number of elements in each layer in a manner informed by the topology of the neural network. DeepIoT borrows the idea of dropping hidden elements from a widely-used deep learning regularization method called dropout. The dropout operation gives each hidden element a dropout probability. During the dropout process, hidden elements can be pruned according to their dropout probabilities.

A thinned" network structure can thus be generated. The challenge is to set these dropout probabilities in an informed manner to generate the optimal slim network structure that preserves the accuracy of sensing applications while maximally reducing their resource consumption. An important purpose of DeepIoT is thus to find the optimal dropout probability for each hidden element in the neural network.

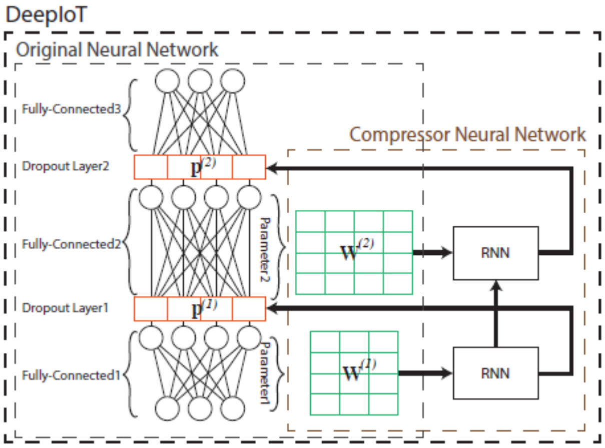

Figure ii. Overall DeepIoT system framework. Orange boxes represent dropout operations. Green boxes represent parameters of the original neural network.

DeepIoT greatly reduces the size of model parameters, and speeds up the execution time by getting rid of the inefficient sparse matrix multiplication. However, a formal way to explore the neural network structure design and underlying system efficiency is still unclear. Most manually designed time-efficient neural network structures for mobile devices use parameter size or FLOPs (oating point operations) as the indicator of model execution time [16-18]. Even the offcial TensorFlow website recommends to use the total number of floating number operations (FLOPs) of neural networks \to make rule-of-thumb estimates of how fast they will run on different devices".1 However, in practice, counting the number of neural network parameters and the total FLOPs does not lead to good estimates of execution time because the relation between these predictors and execution time is not proportional. We therefore design FastDeepIoT [19], showing how a better understanding of the non-linearrelation between neural network structure and performance can further improve execution time and energy consumption without impacting accuracy.

IV. Deep Learning Lable Limited IoT Systems

Labeling data is always time-consuming. This laborious process has become one key factor that hinders researchers and engineers from applying neural networks to sensing and recognition tasks on IoT devices. IoT applications with a large amount of sensing data therefore call for a semi-supervised deep learning framework to solve the challenge of limited labeled data.

In attacking this problem, we propose SenseGAN, a semi-supervised deep learning framework for IoT applications [20]. One core feature of SenseGAN is its capability to leverage unlabeled data for training deep learning networks. SenseGAN can run on resource-constrained IoT devices without additional time or energy consumption compared with its supervised counterpart after training on workstations. Specifically, we adopt the idea of enabling a discriminator to differentiate the joint data/label distributions between the real data/label samples and the partially generated data/label samples made by either the generator or the classifier. Such design can easily decouple the functionalities of discriminator and classifier into two separate neural networks. For an IoT application, users can design their own neural network structure for classification and replace the classifier in the SenseGAN framework with users' own design for the purpose of semi-supervised learning. The adversarial game among the discriminator, generator, and classifier mutually enhances the performance of all automatically.

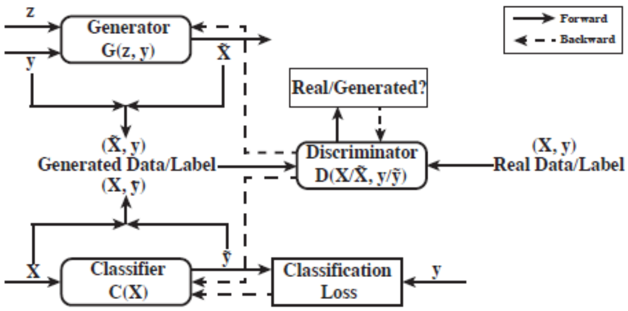

Figure iii. The illustration of SenseGAN components

The intuition of how our SenseGAN framework is able to leverage unlabelled data to enhance its predictive power is as follows. The discriminator tries to discriminate real data/label samples and the partially generated data/label samples; the generator attempts to generate and recover the real sensing inputs based on the categorical information that can fool the discriminator; and the classifier tries to predict the label of sensing inputs that can both fit the supervision and fool the discriminator. During the adversarial game among the three components under the training process, the resulting three improved components can mutually boost performance. When the training reaches the optimality, the discriminator will have learnt the true joint probability distribution of the input sensing data and their corresponding labels for both the labeled and unlabeled samples. The classifier will have learnt the true conditional probability of labels given the sensing input.

All three components are represented by neural networks. We will discuss their specific structures for dealing with the multimodal sensing inputs in detail in the following subsections. In addition, the structure of the classifier can be task-specific or unified with diverse IoT applications [21]. We therefore will not introduce the detailed structure for the classifiers but only discuss its output representation. We treat the classifier as an modular and customizable component for IoT applications when using SenseGAN for semi-supervised learning.

V. Challenges for Deep Embedded Inference

Both the training of a deep network and its own inferences to perform new classifications are now typically executed on power-hungry servers and GPUs [Figure iv]. There is, however, a strong demand to move the inference step, in particular, out of the cloud and into mobiles and wearables to improve latency and privacy issues [Figure iv]. However, current devices lack the capabilities to enable deep inferences for real-life applications.

Recent neural networks for image or speech processing easily require more than 100 giga-operations (GOP)/s to 1 tera-operations (TOP)/s, as well as the ability to fetch millions of network parameters (kernel weights and biases) per network evaluation. The energy consumed in these numerous operations and data fetches is the main bottleneck for embedded inference in energy-scarce milliwatt or microwatt devices. Currently, microcontrollers and embedded GPUs are limited to efficiencies of a few tens to hundreds of GOP/W, while embedded inference will only be fully enabled with efficiencies well beyond 1 TOP/W. Overcoming this bottleneck is possible yet requires a tight interplay between algorithmic optimization (modifying the network topology) and hardware optimization (modifying the processing architectures).

VI. Conclusion and Discussion

We have only scratched the surface of the research landscape on Deep Learning for IoT. Fundamentally, interest in deep learning will evolve as a means to bridge the ever-growing gap between the exponentially increasing planet-wide data generation rate on one hand (thanks to the proliferation of IoT devices), and the at human ability to consume the data, on the other (since our cognitive capacity and population do not increase at the same exponential rate). Deep learning empowers automation that takes the human out of the data processing loop and more to a supervisory capacity.

The past decade witnessed a reemergence of interest in deep learning with significant contributions to human-like perception modalities including computer vision, natural language processing, and speech processing. In the next decade, however, growth of IoT device- sourced data will significantly outpace the growth of human-sources data, due to the proliferation of such devices at rates that far outpace human population growth on the planet. As a consequence, I envision that a growing research interest will shift to modeling and analyzing IoT big data" using deep neural networks. This is not only due to the sheer volume of data created by the growing number of IoT devices, but also due to the unique problem space that IoT data offers. IoT data are generated by physical, social, and spatio-temporal processes that have different dynamics, correlations, and internal structure compared to bits in a video, or words in an article. Researchers have gained much experience designing neural networks for human-like perception tasks, inspired by the way our brain processes information.

DeepIoT and FastDeepIoT are frameworks for understanding and minimizing neural network execution time on mobile and embedded devices. We proposed a tree- structured linear regression model to figure out the causes of execution-time nonlinearity and to interpret execution time through explanatory variables. Furthermore, we utilized the execution time model to rebalance the focus of existing structure compression algorithms to reduce the overall execution time properly.

SenseGAN separates the functionalities of discriminator and classifier into two neural networks, designs specific generator and discriminator structures for handling multimodal sensing inputs, and stabilizes and enhances the adversarial training process by WGAN with gradient penalty as well as Gumbel-Softmax for categorical representations. The evaluation empirically shows that SenseGAN can efficiently leverage both labeled and unlabeled data to effectively improve the predictive power of the classifier without additional time and energy consumption during the inference. Several improvement opportunities remain.

REFERENCES:

[1] How Google Translate Squeezes Deep Learning onto a Phone. http://googleresearch:blogspot:co:uk/2015/07/how-google-translate-squeezes-deep:html.

[2] M. Rabbi, et al. Passive and In-situ Assessment of Mental and Physical Well-being using Mobile Sensors. UbiComp '11.

[3] June Oven. http://juneoven:com/.

[4] Nest Themostat. http://nest:com/thermostat/meet-nest-thermostat.

[5] S. Rallapalli, et al. Enabling Physical Analytics in Retail Stores using Smart Glasses. MobiCom '14.

[6] Y. Taigman, et al. Deepface: Closing the Gap to Human-level Performance in Face Verification. CVPR '14.

[7] G. Hinton, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition. Signal Processing Magazine, 2012.

[8] Your Samsung SmartTV Is Spying on You, Basically. http://www:thedailybeast:com/articles/2015/02/05/your-samsung-smarttv-is-spying-on-you-basically:html.

[9] Y. Bengio, et al. Deep Learning. MIT Press, 2015.

[10] G. Chen, et al. Small-footprint Keyword Spotting using Deep Neural Networks. ICASSP '14.

[11] T. Chen, et al. Diannao: A Small-footprint High-throughput Accelerator for Ubiquitous Machine-learning. ASPLOS '14.

[12] N. Hammerla, et al. PD Disease State Assessment in Naturalistic Environments using Deep Learning. AAAI '15.

[13] N. D. Lane, et al. Can Deep Learning Revolutionize Mobile Sensing? HotMobile '15.

[14] L. Deng and D. Yu. Deep Learning: Methods and Applications. Now Publishers, 2014.

[15] Y. Guo, A. Yao, and Y. Chen, \Dynamic network surgery for efficient dnns," in Advances In Neural Information Processing Systems, 2016, pp. 1379{1387.

[16] X. Zhang, X. Zhou, M. Lin, and J. Sun, “Shufflenet: An extremely efficient convolutional neural network for mobile devices," CoRR, vol. abs/1707.01083, 2017.

[17] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications," CoRR, vol. abs/1704.04861, 2017.

[18] 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.

[19] R. C. Gonzalez, R. E. Woods et al., “Digital image processing," 2002.

[20] S. Yao, Y. Zhao, H. Shao, C. Zhang, A. Zhang, S. Hu, D. Liu, S. Liu, L. Su, and T. Abdelzaher, “Sensegan: Enabling deep learning for internet of things with a semi-supervised framework," Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 2, no. 3, p. 144, 2018.

[21] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, “Deepsense: A unified deep learning framework for time-series mobile sensing data processing," in Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2017, pp. 351{360.