DeepX: A Software Accelerator for Embedded Deep Learning

Nicholas D. Lane[‡], Sourav Bhattacharya[‡], Petko Georgiev[†]
Claudio Forlivesi[‡], and Fahim Kawsar[‡]
[‡]Bell Labs, [†]University of Cambridge

***Abstract*—Deep learning has revolutionized the way sensor measurements are interpreted and application of deep learning has seen a great leap in inference accuracies in a number of fields. However, significant requirement of memory and computational power have been the main bottlenecks in wide scale adoption of these novel computational techniques on resource constrained wearable and mobile platforms. In this demonstration we present DeepX, a software accelerator for efficiently running deep neural networks and convolutional neural networks on resource constrained embedded platforms, e.g., Nvidia Tegra K1 and Qualcomm Snapdragon 800.**

I. Introduction

Novel breakthroughs from the field of deep learning are radically changing the way sensor measurments are interpreted and applied to extract high-level information needed by mobile apps [1]. It is critical that the gains in inference accuracy that deep models afford become embedded in future generations of mobile apps. However, deep learning-based models are yet to become mainstream on embedded platforms, where inference tasks are often challenging due to high measurements noise. In this demonstration, we present *DeepX*, a software accelerator for deep learning models that allows efficiently running *deep neural network* (DNN) and *convolutional neural network* (CNN) models on resource constrained mobile platforms. DeepX significantly lowers the device resources (viz. memory, computation, energy) required by deep learning that currently act as a severe bottleneck to mobile adoption.

The foundation of DeepX is a pair of resource control algorithms, designed for the inference stage of deep learning, that: (1) decompose monolithic deep model network architectures into unit-blocks of various types, that are then more efficiently executed by heterogeneous local device processors (e.g., GPUs, CPUs); and (2), perform principled resource scaling that adjusts the architecture of deep models to shape the overhead each unit-blocks introduces.

II. Design and Operation

DeepX aims to radically reduce mobile resource use, in addition to the execution time, of performing inference with large-scale deep learning models by exploiting a mix of network-based computation and heterogeneous local processors. Towards this goal, we propose two novel techniques:

- **Runtime Layer Compression (RLC):** A building block to optimizing mobile resource usage for deep learning is an ability to shape and control them. But existing approaches, such as those of model compression, focus on the training phase of deep learning models, rather than the inference. RLC provides runtime control of the memory and computation (along with energy as a side-effect) consumed during the inference phase by extending model compression principles, e.g., SVD.
- **Deep Architecture Decomposition (DAD):** A typical deep model is comprised of an architecture of many layers and thousands of units. DAD efficiently identifies unit-blocks of this architecture and creates a "decomposition plan" that allocates blocks to local and remote processors; such plans maximize resource utilization and seek to satisfy user performance goals.

Overview of DeepX architecture is given in Fig. 1.

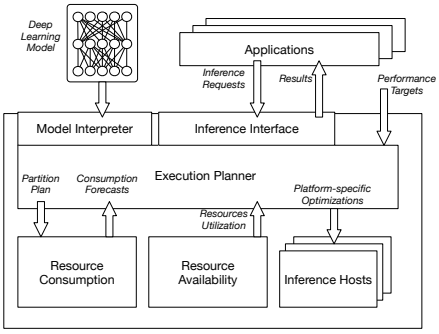
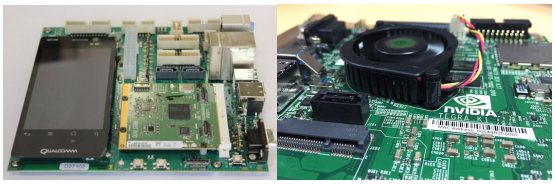


Fig. 1: DeepX Proof-of-Concept System



(a) Snapdragon 800 **(b)** Tegra K1

Fig. 2: Developer Boards for SoCs used for DeepX Prototype

III. Prototype Performance

We briefly highlight representative performance benefits of DeepX under two large-scale deep models that were originally conceived for the cloud. The first model, *AlexNet* [2], performs object recognition and supports more than 1,000 object classes. The second model, *SpeakerId*, is used for speech recognition. We find when running on the Tegra K1(see Fig. 2b), DeepX improves the energy efficiency of AlexNet by factors of 22.1×, 1.8× and 13.2× compared to benchmarks that use cloud computation, and GPU- or CPU-only solutions, respectively. Under

(a) Speaker identification task

(b) Speaker recognition results under DeepX running on Tegra and Snapdragon 400

Fig. 3: Text independent speaker recognition model (SpeakerId) running on Tegra and Snapdragon SoCs under DeepX Prototype

(a) Image recognition task

(b) Image recognition results under DeepX running on Tegra and Snapdragon 400

Fig. 4: Image recognition (AlexNet) running on Tegra and Snapdragon SoCs under DeepX Prototype

SpeakerId, these numbers are: 29.7×, 1.4× and 7.8×. DeepX on the Snapdragon 800 (see Fig. 2a), for the same models, has similar energy benefits: AlexNet, 11.2× (cloud) and 2.1× (CPU); and SpeakerId, 8.9× (cloud) and 8.1× (CPU). DeepX also benefits memory and execution time bottlenecks, with execution tightly coupled to energy gains and reductions of model memory footprint of 2.5× typical for AlexNet and SpeakerId.

IV. Demo: Graphical User Interface

The demonstration showcases an end-to-end prototype of DeepX running on two latest SoCs: Qualcomm Snapdragon 800 and Nvidia Tegra K1. Snapdragon 800 is widely used in modern mobile and wearable devices (e.g., Nexus 5 and Galaxy Gear), whereas the Tegra K1 is aimed at high performance IoT and embedded devices (e.g., Microwave ovens and automobiles). The demonstration includes a front-end user interface powered by HTML5 and Node.js, which accepts recognition tasks, delegates them to two SoCs and visualizes the recognition results together with three performance metrics: (i) execution time, (ii) memory usage, and (iii) battery life. We have selected image and audio recognition tasks with established deep models, e.g., AlexNet and 2-hidden layer DNN, and borrowing two challenge datasets: ImageNet Challenge Dataset 2012 [3], and Automatic Speaker Verification Spoofing and Countermeasures Challenge Dataset [4]. Fig. 3 and 4 respectively show the GUI for audio and image recognition tasks and their performances under DeepX. During the demonstration, visitors will be welcomed to interact with DeepX GUI by selecting and running audio and image recognition tasks running on the mobile SoCs.

References

[1] Y. Bengio, I. J. Goodfellow, and A. Courville, "Deep learning," 2015, book in preparation for MIT Press.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, 2015.

[4] Z. Wu, T. Kinnunen, N. Evans, and J. Yamagishi, "Automatic speaker verification spoofing and countermeasures challenge (asvspoof 2015) database," University of Edinburgh. The Centre for Speech Technology Research (CSTR), Tech. Rep., 2015.