Queen Jane Approximately*: Enabling Efficient Neural Network Inference with Context-Adaptivity

Octavian Machidon
Faculty of Computer and Information Science,
University of Ljubljana, Slovenia
Octavian.Machidon@fri.uni-lj.si

Davor Sluga
Faculty of Computer and Information Science,
University of Ljubljana, Slovenia
Davor.Sluga@fri.uni-lj.si

Veljko Pejović
Faculty of Computer and Information Science,
University of Ljubljana, Slovenia
Veljko.Pejovic@fri.uni-lj.si

ABSTRACT

Recent advances in deep learning allow on-demand reduction of model complexity, without a need for re-training, thus enabling a dynamic trade-off between the inference accuracy and the energy savings. Approximate mobile computing, on the other hand, adapts the computation approximation level as the context of usage, and consequently the computation needs or result accuracy needs, vary. In this work, we propose a synergy between the two directions and develop a context-aware method for dynamically adjusting the width of an on-device neural network based on the input and context-dependent classification confidence. We implement our method on a human activity recognition neural network and through measurements on a real-world embedded device demonstrate that such a network would save up to 37.8% energy and induce only 1% loss of accuracy, if used for continuous activity monitoring in the field of elderly care.

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Computing methodologies** → **Neural networks**.

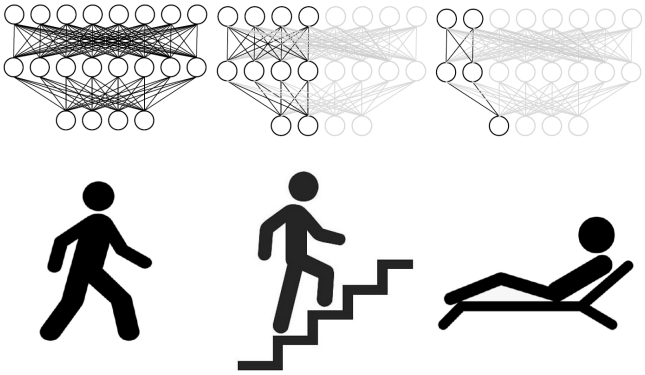

Figure 1: Context-dependent computing requirements in case of a convolutional neural network performing human activity recognition. Activity states are depicted in the lower, while the corresponding slimmed CNNs required for high-quality inference are depicted in the upper part of the figure. In many activity states, only a fraction of the network widths are necessary to achieve maximum accuracy, fostering a context-adaptive, energy-efficient neural network inference.

1 INTRODUCTION

The ages-old illusion of a centralised AI-overlord personified in the Space Odyssey's HAL 9000 supercomputer is giving way to a new concept of edge intelligence where decentralised ubiquitous computers provide individual services according to their local view of the world around them. Such a paradigm shift ameliorates many of the issues associated with centralised processing, including data security concerns and the query-processing-transmission delay. Nevertheless, edge intelligence remains severely limited by the restrictions of the hardware it runs on: in terms of floating-point arithmetic, the performance of the fastest smartphones remain three orders of magnitude poorer than those of high-end servers, while the limited capacity confines a smartphone to no more than a day of active use before the battery is depleted. Whereas in the past the developers could rely on the Moore's law to make their applications future-proof, the breakdown of that one, as well as the Dennard's scaling law [6], leaves us with only one option to sustain the proliferation of edge intelligence – *be as frugal with the available resources as possible.*

* The title of the paper borrows from Bob Dylan's song of the same name. In the song the subject is warned about the unsustainability of her lavish lifestyle and is advised towards more frugal, down-to-earth existence.

EuroMLSys '21, April 26, 2021, CyberSpace

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

The last decade witnessed deep learning (DL) revolutionising areas as diverse as computer vision, machine translation, autonomous driving, and natural language processing. The power of DL lies in its complexity – a large number of numeric parameters are adapted to model the behaviour of the target domain and several thousands or even million of computations are executed to extract higher-level knowledge from low-level inputs. Due to an array of sensors they are often equipped with, ubiquitous computing devices are ideally positioned to provide such low-level inputs and stand to benefit the most from the inclusion of DL in their edge intelligence repertoire. Regrettably, the complexity makes DL ill-suited for execution on resource-constrained hardware, rendering traditional DL models incompatible with mobile and ubiquitous computing, thus pushing researchers into devising custom models suitable for these platforms.

The harsh restrictions put in front of the edge intelligence have their silver lining in a strong research movement towards extracting the actual intelligence from DL models. Approaches have been developed to do away with unnecessary parameters through network weight pruning [8], to reduce the network complexity through weight matrix decomposition [25], to reduce the number of bits used for describing individual weights [8], and to transfer the knowledge embedded in a complex network to a smaller one [10], to name but a few. Nevertheless, none of these complexity reduction methods come for free, as the accuracy of DL inference drops, albeit unproportionally, as the network becomes lighter. Furthermore, as the compression often gets performed only once leaving the device with a potentially permanently impaired network, the edge intelligence remains disadvantaged in competition with its centralised counterpart.

In this work we harness recent advances in adaptable DL where model complexity reductions can be performed on demand, directly on a target device, and without a need for network re-training. We embrace the Slimmable Neural Network (SNN) [28] concept, that enables us to reduce the number of active network parameters on the fly – with consequences possibly felt at the output where the accuracy of DL inference may drop. Building on the philosophy of approximate mobile computing [20], we harness the key property of ubiquitous computing – its context awareness – *to dynamically adapt the SNN reduction according to the context of use*. We observe that the accuracy of inference of the same network varies depending on the usage scenario (as illustrated by Figure 1). For instance, the same human activity recognition network may perform well when a user is walking or sitting, yet stops short of detecting less frequent activities, such as walking down the stairs. A more complex network, on the other hand, might perform well across the range of scenarios, yet would use significantly more computational, memory, and battery resources. We therefore develop a method, based on the reported classification *confidence*, to adjust the SNN slimming ratio so that the desired accuracy of inference is preserved with the least amount of computation used, irrespective of the potentially unpredictable situations in which the mobile device hosting the computation is found.

We implement and evaluate our solution on a real-world embedded device and experimentally quantify the resource savings enabled by context-aware SNN adaptation. In case of continuous activity recognition, we observe a potential 37.8% energy use reduction in scenarios reflecting real-world human behaviour. Crucially, the savings come with negligible sacrifice of inference accuracy (just 1% drop), as the network inference power remains perfectly matched to the contextual computation needs.

2 RELATED WORK

Following the recent DL breakthroughs, a question of the necessity of such a large number of parameters and layers of depth appeared [3]. It was demonstrated that, theoretically, a smaller network can successfully capture the knowledge of a larger one, if coached by it [10]. The practicality of such knowledge distillation is limited, thus research has focused on compressing the existing networks instead. DeepX, for instance, enables DL execution on resource-constrained devices by reducing the amount of computation through singular value decomposition-based layer compression [13]. Solutions based on intelligent weight pruning [17], weight virtualisation [15], and quantisation [24] followed, all demonstrating that complex neural networks can be trimmed to work on edge devices. Yet, the issue of the lost inference accuracy, which in case of DeepX amounts to $\approx 5\%$, remained.

The model's compression-performance relationship is seldom straightforward. Frameworks such as FastDeepIoT examine the said relationship and guide the compression algorithm towards the execution time minimisation [27]. A common issue here is that the compressed network represents merely a single option along the accuracy–resource usage trade-off front. Should the circumstances change, we would have to prepare (and retrain) a new network and deploy it on the device. MCDNN tackles this issue by constructing a catalogue of compressed models, storing them in the cloud, and downloading the most appropriate model according to the changing system requirements [9].

Transferring models from the cloud as the requirements vary leads to both unreliable inference and additional energy spent on model transfer. To tackle these issues, a cascade of increasingly more complex neural network classifiers is proposed by Bolukbasi et al [5]. In case of unreliable inference, reflected in a low classifier confidence value, a more powerful classifier is employed at the succeeding step. Our work,

too, relies on the last layer softmax confidence for assessing the quality of classification. However, our approach does not require multiple models to be trained and stored on a device, but dynamically adapts the weights of a single model.

Dynamic adaptation to *the context* of use has received surprisingly little attention. SPINN [14], a recently published distributed synergistic device-cloud inference system, implements progressive inference based on a scheduler that adapts the execution to dynamic contextual conditions. While close to our work, SPINN focuses on network conditions and device/server load, while we take a much broader definition of context. Furthermore, SPINN uses early network exiting as an approximation strategy, which limits the approximation levels to the number of layers after whicch an early exit can be mamde. SNNs used in our work enable much higher granularity of approximation. Finally, our idea of seamless adaptation to the context of use is close to cDeepArch [26], a method that decomposes the inference task on two subproblems: context recognition and context-dependent target recognition. cDeepArch relies on a lightweight network for the former and a suite of pre-built networks for the latter task. Consequently, cDeepArch requires different training sets separated by the context of use, as well as either pre-loading or real-time distribution of different models. Our approach, on the other hand, requires that a single (complete) neural network is trained, loaded on the device only once, and any further adaptation is conducted on this network.

3 CONTEXT-ADAPTIVE COMPRESSION

3.1 Preliminaries – SNN

Experiments on network weight pruning and knowledge distillation demonstrate that, in practice, expanding a neural network with more weights disproportionally impacts the classification accuracy – doubling the number of parameters will unlikely provide double the accuracy rates. Slimmable Neural Networks (SNNs) utilise this observation, albeit in the opposite direction – reducing the number of weights should lead to a graceful degradation of classification performance [28]. The issue of the re-training, which is usually necessary after a configuration change, is in SNNs solved via swappable batch normalization layers. Such an approach allows a preloaded network to use a varying fraction (e.g. 100%, 75%, 50%, 25%) of the weights at runtime.

3.2 Adapting to the context

In a static setting, slimming a neural network leads to reduced accuracy and increased energy efficiency, as fewer computations are executed. However, mobile devices operate in highly dynamic environments, where the input to the neural network varies (e.g. in case of a smart assistant – microphone data from a speaking user is rather different than the data from a sleeping user), the user's expectation vary (e.g. a user might need high accuracy from an elderly care app when a user is alone, but can tolerate a lower accuracy when a caretaker is around), and the need for efficient operation varies with the context of use [20].

We focus on the case where the need for a more complex model stems from the nature of the input data. We investigate the case of human activity recognition and demonstrate that different activities can be reliably recognised with SNNs using different fractions of widths. To guide the SNN adaptation, however, we cannot rely on knowing the activity class in advance. Instead, we examine the reliability of the classification proxied through the softmax-based confidence and conclude that this confidence can indeed be used to drive the network slimming in the absence of the actual class label.

4 IMPLEMENTATION & EXPERIMENTS

We implement a human activity recognition (HAR) classifier in the form of a SNN based on MobileNet-V1 [11]. We choose the MobileNet-V1, a small, low-latency, low-power model that on one hand is suitable for energy-efficient mobile tasks, and on the other has been previously successfully trained using the SNN approach (however, for a different task – image classification) [28]. While MobileNet is well-established for image classification and mobile vision applications [11, 16], we are among the first to show its viability on classifying time-domain signals, more specifically in the field of HAR [18]. Our SNN implementation allows operation under four width fraction levels: with 25%, 50%, 75% or 100% width parameters kept. Note that in each case, the network has to be preloaded to a device only once and the number of parameters used can be modified at runtime.

We trained our network on the publicly available UCI HAR dataset containing recordings of 30 volunteers performing 6 different activities (walking, walking upstairs, walking downstairs, sitting, standing and lying) while carrying a waist-mounted smartphone with embedded inertial sensors [2]. The collected data was de-noised and sampled in fixed-width sliding windows of 2.56s with 50% overlap (128 readings/window). We used the dataset's predefined train and test partitioning (random partitioning where 70% of the volunteers were selected for generating the training data and 30% the test data), and 20% of the train data was randomly selected as validation data.

We train the network on a high performance computing cluster and evaluate the network inference accuracy and energy consumption separately on an embedded platform: the NVIDIA Jetson Nano, a single-board computer with a computing architecture very similar to the modern mobile phones. For energy measurements, we used the Monsoon power monitor tool [1], a high sampling frequency platform

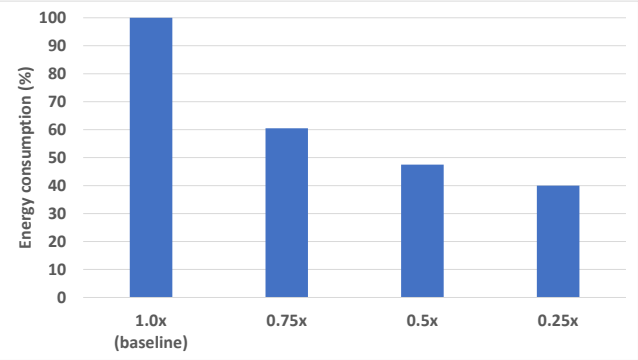

Figure 2: Relative energy consumption with varying SNN slimming ratios on NVIDIA Jetson Nano board.

commonly used for power measurements in mobile and embedded computing [22].

5 RESULTS & DISCUSSION

5.1 Accuracy and energy usage analysis

The original SNN paper reports more than an order of magnitude reduction in the number of floating point operations (FLOPs) needed for inference on a 25%-wide network, as opposed to the inference on the full-width network. Indeed, we observe the same level of reduction in the case of a MobileNet-V1 trained for human activity recognition (11.6M FLOPs for 1.0× vs. 834K for 0.25×). Nevertheless, this reduction does not necessarily translate to system-level energy savings, as the overhead of neural network weight loading and the coordination among threads may impact the total energy spent for the inference. In Figure 2 we observe that the energy consumption is reduced by up to 60% when a quarter-width network is used. However, unlike the reduction in FLOPs, the reduction in energy consumption experiences a more substantial drop during the initial slimming (i.e. from 100% to 75% width) and appears to be plateauing off as the network is slimmed further. Consequently, unless the impact on the inference accuracy remains negligible, practical benefits of extreme-level slimming and fine-grain slimming are unlikely.

While the energy expenditure does not depend on the input, but merely on the slimming ratio of the SNN, the inference accuracy actually varies with the context of use, as some activities may be easier to detect than others. Indeed, Figure 3 demonstrates that network slimming affects different classes of activities differently. While three mobility states (walking, downstairs and sitting) require a 100%-wide network to achieve the maximum classification accuracy, for the remaining three states this is can be achieved also by using lower network widths (e.g. 75%-wide network when standing, 50%-wide network when walking upstairs, and 25%-wide network when lying).

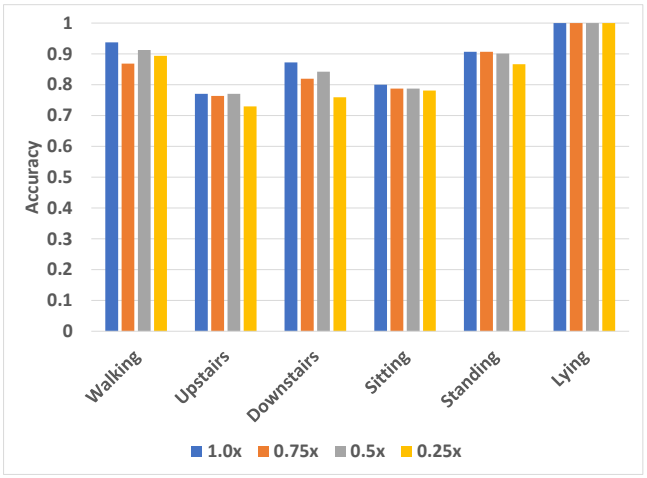

Figure 3: Per-class accuracy for different SNN slimming ratios.

Table 1: Min. slimming ratio for each class while maintaining the max. prediction accuracy.

Class	Width	Accuracy	Confidence
Walking	1.0×	0.93	0.97
Upstairs	0.5×	0.77	0.96
Downstairs	1.0×	0.87	0.97
Sitting	1.0×	0.80	0.97
Standing	0.75×	0.90	0.97
Lying	0.25×	1.0	0.95

5.2 Towards context-adaptive slimming

The above results indicate that the energy-optimal operation is reached when the slimmest network yielding the highest inference accuracy is used. In real-world situations, however, the classifier does not have the ground truth information, thus cannot assess its accuracy. Instead, we focus on classifier *confidence* as a proxy to accuracy. In a DL classifier, the final layer's softmax function generates normalised classification scores, often termed confidence[2]. The confidence value pertaining to the predicted class can be used to guide dynamic inference adjustment, for instance in the so-called *early-exit* [23], *multistage* [29] or *cascaded* [4] networks.

We show the relationship between accuracy and confidence in Figure 4. The overall trend is for the confidence to monotonically increase with accuracy, thus we can indeed use it to drive the network slimming. Some activities exhibit

[2]Gal and Ghahramani argue that a model can be uncertain in its predictions even with a high softmax output, thus argue against the term "confidence" [7]. Our analysis demonstrates a correlation between the accuracy and the top softmax score value, thus, we use the term "confidence" acknowledging that the actual interpretation might be subject to debate.

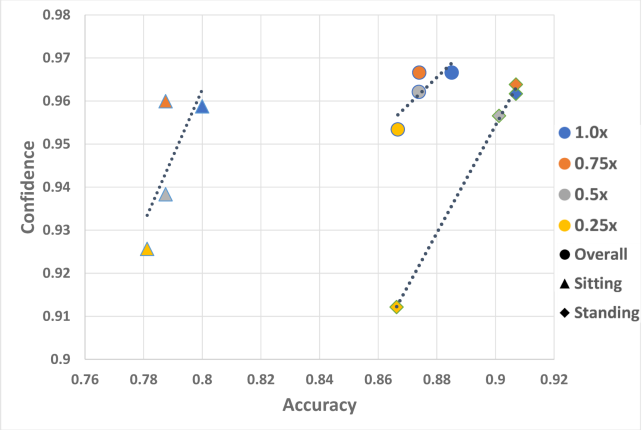

Figure 4: Overall accuracy vs. softmax confidence for different SNN slimming ratios, and class-specific values for Sitting and Standing.

more rapid jumps of the confidence as we approach to high accuracy. We show two such activities, sitting and standing, in the graph. We also note that as the maximum classification accuracy is reached, the relationship between the confidence and accuracy wanes. Nevertheless, as our goal is to identify the lowest width resulting in a certain level of accuracy, as long as the deviation from the monotonic relationship does not lead to rapid accuracy variations (not followed by the corresponding confidence change) among "neighbouring" width ratios, we can use confidence as a proxy for accuracy.

We now develop a practical algorithm for SNN adaptation (Algorithm 1). In a nutshell, we start the inference with the narrowest possible SNN (25% width). At each inference point we assess the confidence level. Should this level drop below the confident inference threshold for the inferred activity, that indicates that classifier's explanatory power might not be sufficient for the current input, thus we then iteratively widen the network to the next available width (in our case 50% width, 75% width, etc.) until the threshold is surpassed or the maximum width is reached. The confident inference thresholds are pre-calculated from the validation data and represent the mean minus two standard deviations of the confidence level of the narrowest network achieving the maximum prediction accuracy on the validation set (Table 1).

Natural human mobility is not characterised by rapid variations [12, 21], thus, the subsequent data point is classified with the starting width corresponding to the final width used at the previous data point. Should this width be insufficient, as indicated by the low confidence, we again perform the iterative width expansion. To minimise the energy use we slim the network at times when the classifier indicates a change in the activity. Then, we re-start the width search process as explained above. Furthermore, we periodically probe slimmer networks even when the detected activity remains the same. The trade-off between the exploration and exploitation is controlled by a parameter α.

Algorithm 1: Dynamic SNN adaptation

```
snn_width=0.25;
alpha=<probing_probability>;
while input do
    set_model_width(snn_width);
    prediction, confidence=model(input);
    if confidence<threshold[prediction] then
        increase(snn_width);
    else
        if prediction!=previous_state then
            snn_width=0.25;
        else if random_choice(alpha) then
            snn_width=0.25;
        end
        previous_state=prediction;
        input=next();
    end
end
```

5.3 Continuous HAR use case

Assisted living and elderly care are areas where continuous monitoring of human activity is bound to be extremely useful, should the energy requirements hampering further proliferation of such systems be reduced. Thus we assess whether our dynamic SNN width adaptation approach could enable continuous activity inference on a dataset reflecting the actual behaviour of two elderly persons throughout the day (32 hours of free-living data) [19]. We observe 37.8% energy savings with our approach, compared to a full width HAR classifier, with just 1% drop in inference accuracy (Figure 5).

The energy savings we enable are *system-wide*, thus may directly translate to more than a third longer battery life, which could in turn make a continuous HAR application viable on a given platform. Our further investigation also reveals that fixing the network at its lowest width possible (0.25×), 60% energy savings are obtained with a 2% drop in overall accuracy. This is explained by the specifics of the dataset – from the total recorded time, 86% consisted of more *static* activities like sitting, standing and lying, with the last one accounting for 46% of the total time. These are all activities for which the overall accuracy gap between the full-width and the lowest-width network is small or even nonexistent: the average accuracy for lying is the same for all network widths. Note, that despite a comparable accuracy drop, such a naive approximation leads to a much higher F1 score drop than our approach (nearly 4%), as those few difficult-to-classify situations would be misclassified.

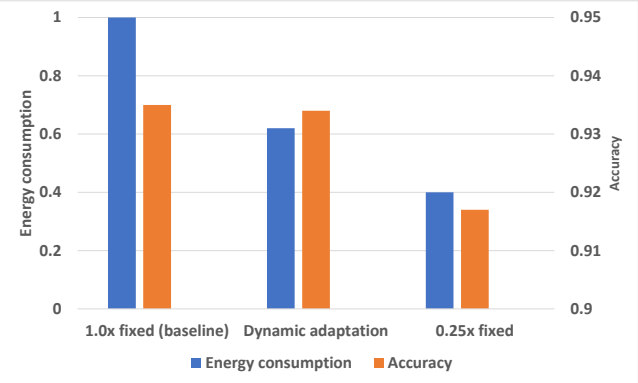

Figure 5: Three-way comparison of the overall accuracy vs. energy consumption between our dynamic SNN adaptive approach and fixed 1.0× and 0.25× networks for the free-living data recorded in [19].

6 CONCLUSIONS & FUTURE WORK

In this paper, fostered by the recent advances in adaptable DL models and harnessing the context-awareness property of ubiquitous systems, we proposed a method of dynamically adjusting the SNN width while preserving the desired inference accuracy with the least amount of computation.

We implemented an SNN HAR classifier based on MobileNet-V1. Our experiments revealed that slimming affects inference accuracy differently in different contexts. Adding to this the observation that the softmax confidence grows with the accuracy as the width of the SNN increases, we used the prediction confidence in an algorithm we developed to guide the dynamic adjustment of the network slimming ratio.

To quantify the resource savings enabled by our approach, we focused on the system-level energy savings (rather than the reduction in the number of FLOPs). We assessed our approach in a real-world scenario, recognising 32 hours worth of elderly activity data, and obtained energy savings of up to 37.8% with just 1% drop in inference accuracy compared to always running the network at full width.

While in this work we adapted the network based on the input, which is causally linked to the user's mobility context, and aimed at maximizing the prediction accuracy, real-world scenarios open the door to network adaptation based on other dimensions. In assisted living applications, such as fall detection, the network can be adapted to surrounding context, i.e. slimmed down to maximize energy savings when a caretaker is present, and upscaled only when the person is unattended, to maximize detection accuracy.

ACKNOWLEDGMENTS

The research presented in this paper was funded by the project "Bringing Resource Efficiency to Smartphones with Approximate Computing" funded by the Slovenian National Research Agency (ARRS), project number: N2-0136. The authors would like to thank Stelios Paraschiakos for sharing with us the HAR dataset collected and described in [19].

REFERENCES

[1] Monsoon Solutions high voltage power monitor. http://msoon.github.io/powermonitor/HVPM.html
[2] D. Anguita, A. Ghio, L. Oneto, X. Parra, and Jorge Luis Reyes-Ortiz. 2013. A Public Domain Dataset for Human Activity Recognition using Smartphones. In *ESANN*.
[3] Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep?. In *Advances in neural information processing systems*. 2654–2662.
[4] Konstantin Berestizshevsky and Guy Even. 2019. Dynamically Sacrificing Accuracy for Reduced Computation: Cascaded Inference Based on Softmax Confidence. In *Int. Conf. on Artificial Neural Networks*. Springer, 306–320.
[5] Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. 2017. Adaptive neural networks for efficient inference. In *Int. Conf. on Machine Learning (ICML)*. Sydney, Australia.
[6] Hadi Esmaeilzadeh, Emily Blem, Renee St Amant, Karthikeyan Sankaralingam, and Doug Burger. 2011. Dark silicon and the end of multicore scaling. In *IEEE/ACM Int. Symp. on Computer Architecture (ISCA)*. San Jose, CA, USA.
[7] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Int. Conf. on Machine Learning (ICML)*. New York City, NY, USA.
[8] Song Han, Huizi Mao, and William J Dally. 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *Int. Conf. on Learning Representations (ICLR)*. San Juan, Pureto Rico.
[9] Seungyeop Han, Haichen Shen, Matthai Philipose, Sharad Agarwal, Alec Wolman, and Arvind Krishnamurthy. 2016. Mcdnn: An approximation-based execution framework for deep stream processing under resource constraints. In *ACM MobiSys*. Singapore, Singapore.
[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*. Montreal, Canada.
[11] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).
[12] Roua Jabla, Félix Buendía, Maha Khemaja, and Sami Faiz. 2019. Balancing Timing and Accuracy Requirements in Human Activity Recognition Mobile Applications. In *Multidisciplinary Digital Publishing Institute Proceedings*, Vol. 31. 15.
[13] Nicholas D Lane, Sourav Bhattacharya, Petko Georgiev, Claudio Forlivesi, Lei Jiao, Lorena Qendro, and Fahim Kawsar. 2016. Deepx: A software accelerator for low-power deep learning inference on mobile devices. In *ACM/IEEE Int. Conf. on Information Processing in Sensor Networks (IPSN)*. Vienna, Austria.
[14] Stefanos Laskaridis, Stylianos I. Venieris, Mario Almeida, Ilias Leontiadis, and Nicholas D. Lane. 2020. *SPINN: Synergistic Progressive Inference of Neural Networks over Device and Cloud.* ACM, New York, NY, USA.
[15] Seulki Lee and Shahriar Nirjon. 2020. Fast and scalable in-memory deep multitask learning via neural weight virtualization. In *ACM MobiSys*. Cyberspace.
[16] Hiroya Maeda, Yoshihide Sekimoto, Toshikazu Seto, Takehiro Kashiyama, and Hiroshi Omata. 2018. Road damage detection and classification using deep neural networks with smartphone images.

Computer-Aided Civil and Infrastructure Eng. 33, 12 (2018), 1127–1141.

[17] Wei Niu, Xiaolong Ma, Sheng Lin, Shihao Wang, Xuehai Qian, Xue Lin, Yanzhi Wang, and Bin Ren. 2020. Patdnn: Achieving real-time DNN execution on mobile devices with pattern-based weight pruning. In *ACM ASPLOS*. Cyberspace.

[18] M. Nutter, C. H. Crawford, and J. Ortiz. 2018. Design of Novel Deep Learning Models for Real-time Human Activity Recognition with Mobile Phones. In *2018 Int. Joint Conf. on Neural Networks (IJCNN)*. 1–8.

[19] Stylianos Paraschiakos, Ricardo Cachucho, Matthijs Moed, Diana van Heemst, Simon Mooijaart, Eline P Slagboom, Arno Knobbe, and Marian Beekman. 2020. Activity recognition using wearable sensors for tracking the elderly. *User Modeling and User-Adapted Interaction* 30, 3 (2020), 567–605.

[20] Veljko Pejović. 2019. Towards approximate mobile computing. *GetMobile: Mobile Computing and Communications* 22, 4 (2019), 9–12.

[21] Jorge-L Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita. 2016. Transition-aware human activity recognition using smartphones. *Neurocomputing* 171 (2016), 754–767.

[22] Andreas Schuler and Gabriele Anderst-Kotsis. 2019. Examining the energy impact of sorting algorithms on Android: an empirical study. In *Proceedings of the 16th EAI Int. Conf. on Mobile and Ubiquitous Systems: Computing, Networking and Services* (Houston, Texas) *(MobiQuitous '19)*. ACM, NY, USA, 404–413.

[23] Meiqi Wang, Jianqiao Mo, Jun Lin, Zhongfeng Wang, and Li Du. 2019. DynExit: A Dynamic Early-Exit Strategy for Deep Residual Networks. In *2019 IEEE Int. Workshop on Sign. Proc. Sys. (SiPS)*. IEEE, 178–183.

[24] Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. 2016. Quantized convolutional neural networks for mobile devices. In *IEEE CVPR*. Las Vegas, NV, USA.

[25] Jian Xue, Jinyu Li, and Yifan Gong. 2013. Restructuring of deep neural network acoustic models with singular value decomposition.. In *Interspeech*. Lyon, France.

[26] Kang Yang, Tianzhang Xing, Yang Liu, Zhenjiang Li, Xiaoqing Gong, Xiaojiang Chen, and Dingyi Fang. 2019. cDeepArch: A compact deep neural network architecture for mobile sensing. *IEEE/ACM Transactions on Networking* 27, 5 (2019), 2043–2055.

[27] Shuochao Yao, Yiran Zhao, Huajie Shao, ShengZhong Liu, Dongxin Liu, Lu Su, and Tarek Abdelzaher. 2018. Fastdeepiot: Towards understanding and optimizing neural network execution time on mobile and embedded devices. In *ACM SenSys*. Shenzhen, China.

[28] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. 2019. Slimmable neural networks. In *Int. Conf. on Learning Representations (ICLR)*. New Orleans, LA, USA.

[29] Zhihang Yuan, Xin Liu, Bingzhe Wu, and Guangyu Sun. 2020. ENAS4D: Efficient Multi-stage CNN Architecture Search for Dynamic Inference. *arXiv preprint arXiv:2009.09182* (2020).