Can Deep Learning Revolutionize Mobile Sensing?

Nicholas D. Lane
Microsoft Research

Petko Georgiev
University of Cambridge

ABSTRACT

Sensor-equipped smartphones and wearables are transforming a variety of mobile apps ranging from health monitoring to digital assistants. However, reliably inferring user behavior and context from noisy and complex sensor data collected under mobile device constraints remains an open problem, and a key bottleneck to sensor app development. In recent years, advances in the field of deep learning have resulted in nearly unprecedented gains in related inference tasks such as speech and object recognition. However, although mobile sensing shares many of the same data modeling challenges, we have yet to see deep learning be systematically studied within the sensing domain. If deep learning could lead to significantly more robust and efficient mobile sensor inference it would revolutionize the field by rapidly expanding the number of sensor apps ready for mainstream usage.

In this paper, we provide preliminary answers to this potentially game-changing question by prototyping a low-power Deep Neural Network (DNN) inference engine that exploits both the CPU and DSP of a mobile device SoC. We use this engine to study typical mobile sensing tasks (e.g., activity recognition) using DNNs, and compare results to learning techniques in more common usage. Our early findings provide illustrative examples of DNN usage that do not overburden modern mobile hardware, while also indicating how they can improve inference accuracy. Moreover, we show DNNs can gracefully scale to larger numbers of inference classes and can be flexibly partitioned across mobile and remote resources. Collectively, these results highlight the critical need for further exploration as to how the field of mobile sensing can best make use of advances in deep learning towards robust and efficient sensor inference.

Categories and Subject Descriptors: C.3 [Special-Purpose and Application-Based Systems]: Real-time and embedded systems.

General Terms: Design, Experimentation.

Keywords: Mobile Sensing, Deep Learning, Deep Neural Network, Activity Recognition.

1. INTRODUCTION

By exploiting sensors in wearables and smartphones, apps are exposing users to powerful new mobile experiences that have the potential to change the way users live and interact with each other. Advances in the area of mobile sensing enable users to: quantify their sleep and exercise patterns [6], monitor personal commute behaviors [26], track their emotional state [25], or even measure how long they spend queuing in retail stores [27]. The driving force underpinning these innovations is the use of algorithms to infer behaviors and contexts from sensor data collected by mobile devices.

However, critically today inferring many important behaviors from mobile sensor data under real-world conditions remains brittle and unreliable (e.g., [6]); this in turn is acting as a bottleneck to sensor app development, preventing many apps from being ready for consumers – especially those that require more difficult (but also powerful) forms of behavior modeling. The field of mobile sensing would be transformed overnight if a breakthrough in the level of robustness and efficiency of mobile inference could be achieved – such an advance would revolutionize the sensing app landscape by broadening the number of inference categories accurate enough for mainstream use. But challenges to robust sensor inference are numerous and varied, including for example: coping with uncontrolled device positions [21] (e.g., in a pocket, in a bag); background noise (e.g., outdoors, while driving) when sampling data [23]; and adapting to the differences in data generated by a diverse user population [17] (e.g., lifestyle, demographics). Although the mobile sensing community continues to develop approaches that minimize these effects, more fundamental advances in the machine learning techniques used are also needed to close the gap between the promise and actual reality of sensing apps.

A strong candidate for such fundamental advances in how mobile sensor data is processed is *deep learning*; an emerging area of machine learning that has recently generated significant attention – enabling, for example, large leaps in the accuracy of mature domains like speech recognition, where previously only incremental improvements were seen for many years [2]. In a recent high-profile example [18], deep learning algorithms were also shown to be capable of learning complex concepts – such as the appearance of cats in videos – with incredibly little supervision (i.e., example data manually labeled for each concept of interest). More broadly, deep learning techniques are now key elements in achieving state-of-the-art inference performance in a variety of applications of learning [13] (e.g., computer vision, natural language processing). Promisingly, achieving such levels of robust inference (as seen in speech) often requires overcoming similar data modeling challenges (e.g., noisy data, intra-class diversity) to those found in mobile sensing. In addition, many of the instances where deep learning has been successful are related to inference tasks of importance to mobile sensing (e.g., emotion recognition, speaker identification).

It is somewhat surprising that deep learning has yet to have a widespread impact on mobile sensing. Only limited usage exists today coming in the form of largely cloud-based models that provide, for example, speech and object recognition within mobile commercial services [2]. Little exploration has been done into deep learning methods applied to

HotMobile'15, February 12–13, 2015, Santa Fe, NM, USA.
Copyright © 2015 ACM 978-1-4503-3391-7/15/02 ...$15.00.
http://dx.doi.org/10.1145/2699343.2699349.

activity, behavior and context recognition. Deep learning techniques are also absent from the vast majority of mobile sensing prototypes that are deployed and evaluated. Perhaps this is partially due to the computational overhead associated with deep learning, and the fact early mobile sensing efforts were highly computationally constrained. However, mobile architectures have advanced enormously in recent years (an iPhone 6, for instance, is a 10x computational jump over a 5-year old iPhone 3GS). Such advances are radically changing what is possible to locally perform.

What is missing today are systematic studies to understand how advances in deep learning can be applied to inference tasks relevant to mobile sensing; in addition to the development of new mobile runtimes that can perform inference using these models in an energy-efficient low-latency manner. In this paper, we begin to examine this timely issue with an exploratory study into the potential for deep learning to address a range of core challenges to robust and resource efficient mobile sensing. To better understand the interaction between modeling accuracy and system resources, we prototype a mobile Deep Neural Network (DNN) classification engine capable of a variety of sensor inference tasks. The role of the engine is to classify sensor data on the mobile device, assuming deep model training is performed in an offline manner with conventional tools. The design of the engine exploits a broad range of modern mobile hardware and executes most inference operations on the low-power DSPs present in many already available smartphones (e.g., Samsung Galaxy S5, Nexus 6). As a result, this engine achieves resource efficiencies not possible if only using a CPU.

Our study findings show, as would be expected, benefits to inference accuracy and robustness by adopting deep learning techniques. For example, we show our DNN engine can achieve comparable accuracy levels for audio sensing using significantly simpler features (a 71x reduction in features), relative to modeling techniques more typically used. We also discover a number of less expected results related to mobile resource usage. For instance, we find that DNNs can have a resource overhead close to the most simple comparison models, yet simultaneously have accuracy levels equal to any tested alternative. Moreover, our DNN implementation is able to scale gracefully to large numbers of inference categories unlike the models used today. These results indicate forms of deep learning (DNNs in this case) may also provide important improvements in the resource-efficiency of sensing algorithms on mobile devices. We anticipate this preliminary study will provide a foundation for subsequent research that explores the application of deep learning to mobile sensing. More importantly, we believe the findings of this work may even represent the start of transformative changes in how mobile inference algorithms are designed and operate – powered by concepts from deep learning.

2. DEEP LEARNING

Modeling data with neural networks is nothing new, with the underlying technique being in use since the 1940s [22]. Yet this approach, in combination with a series of radical advances (e.g., [16]) in how such networks can be utilized and trained, forms the foundation of *deep learning* [13]; a new area in machine learning that has recently revolutionized many domains of signal and information processing – not only speech and object recognition but also computer vision, natural language processing, and information retrieval.

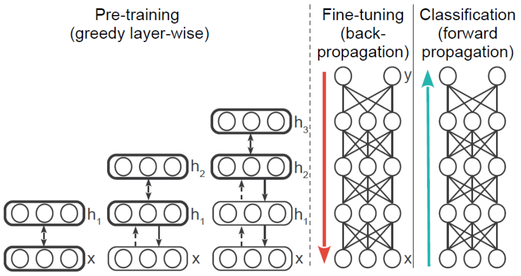


Figure 1: Example phases of building a Deep Neural Network with 3 hidden layers (h_1, h_2, and h_3), input layer x and output layer y. Shown are the pre-training, fine-tuning and classification phases of a DNN variant called a Deep Belief Network.

Deep Neural Network Primer. Many forms of deep learning have been developed with example techniques including Boltzmann Machines, Deep Belief Networks, and Deep Autoencoders (each detailed in [13]). Figure 1 illustrates a common example of deep learning; specifically a Deep Neural Network (or DNN). A DNN is a feed-forward neural network (i.e., the network does not form a cycle) that maps provided inputs (e.g., audio or accelerometer data or features derived from them) to required outputs (e.g., categories of behavior or context). The network consists of nodes organized into a series of fully connected layers; in-between the input and output layers the DNN contains additional bridging layers (called "hidden" layers). Each node uses an activation function to transform the data/state in the prior layer that in turn is exposed to the next layer. Commonly used node activation functions are drawn from the sigmoid family. A logistic sigmoid $y = \frac{1}{1+e^{-x}}$, for instance, has the property of returning values in the range $(0, 1)$ making it suitable for representing probabilities. Output nodes are an exception, these typically use a softmax function in which the final inference is determined by the node with the largest value (i.e., the conditional probability). See [13] for more.

A DNN is trained usually in two stages. First, an unsupervised process referred to as "pre-training" is applied to bootstrap hidden node and edge parameters. This stage was a significant breakthrough in deep learning, when it was discovered that this can be effectively done in a greedy layer-wise fashion without labeled data – simplifying the learning when multiple hidden layers are present. Second, a supervised process occurs, referred to as "fine-tuning", that uses backpropagation algorithms to adjust the parameter values initialized in the previous stage. Parameters are adjusted to minimize a loss function that captures the difference between network inferences and ground-truth labeled data.

Of course, many variations on this training process have been proposed; and similarly DNNs themselves can be utilized in various ways to perform inference. Not only are they used simply as classifiers in isolation (as we do in our study) but they are also chained together to interpret data of differing modalities (e.g., [9]) , or combined with other types of models (e.g., HMMs, GMMs) to form hybrids (e.g., [15]) or act as front-end feature selection phase (e.g., [24]). Similarly, beyond a basic DNN is a rich family of approaches and machinery such as (the aforementioned) Deep Belief Networks and Boltzmann Machines or others like Convolutional Neural Networks. However, we limit this work to a relatively simple form of deep learning (single DNNs), leaving the exploration of additional techniques for future study.

Figure 2: Qualcomm Snapdragon 800 MDP/S [1].

stand up chair	sit down chair
get up bed	lie down bed
climb stairs	descend stairs
eat meat	eat soup
drink glass	brush teeth
use phone	walk
comb hair	pour water

Table 1: Activities of daily living (ADL).

Existing Mobile Use of Deep Learning. As previously described, there are some early examples of deep learning being applied in mobile settings. For instance, the speech recognition models used by phones today exploit deep learning techniques (e.g., [2]); but crucially they operate off-device, in the cloud. Some existing application domains of deep learning (such as emotion recognition [15] and others related to audio) are very similar to requirements of mobile sensing and should be able to be adapted for sensor app purposes. Other important sensing tasks like activity recognition are largely unexplored in terms of deep learning, with only isolated examples being available (such as for feature selection [24] or non-mobile activity recognition in controlled or instrumented environments [12, 3]). These inference tasks will require more fundamental study as they lack clear analogs in the deep learning literature. Moreover, significant systems research is required to understand how the full range of deep learning techniques can be used locally on mobile devices while respecting energy and latency constraints. For example, mobile OS resource control algorithms aware of how to regulate the execution of one or more instances of deep learning inference are currently missing; as are new deep learning inference algorithms tailored to mobile SoC components like GPUs and DSPs.

3. PRELIMINARY INVESTIGATION

We now detail our initial study into the suitability and benefits of deep learning when applied to mobile sensing.

Study Aims. Three key issues are investigated:

- *Accuracy:* Are there indications that deep learning can improve inference accuracy and robustness to noisy complex environments? Especially when sensor data is limited, either by features or sampling rates. (See §4).
- *Feasibility:* How practical is it to use deep learning for commonly required sensing tasks on today's mobile devices? Can we push today's hardware to provide acceptable levels of energy efficiency and latency when compared with conventional modeling approaches? (See §5).
- *Scalability:* What are the implications for common scalability challenges to mobile sensing if deep learning is adopted? For example, how well does it perform as the number of monitored categories of activities expands? (A common bottleneck in forms of mobile sensing such as audio [20]). Moreover, how easily can deep learning inference algorithms be partitioned across computational units (i.e., cloud offloading), a frequently needed technique to manage mobile resources [11]. (Also see §5).

By examining these important first-order questions regarding deep learning in the context of mobile sensing our study highlights new directions for the community, as well as provides the foundation for follow-up investigations.

Mobile DNN Implementation. In the proceeding two sections, we report experiments performed with a working deep learning implementation developed for an Android smartphone with a Jelly Bean 4.3 OS. The implementation is targeted towards DNN models used in typical continuous sensing tasks such as keyword spotting [5] and activity recognition rather than intermittent workloads, such as speech or image recognition, which require more complex cloud-only models due to their that significant memory and compute requirements. To maximize the mobile resource efficiency, we take advantage of a low power co-processor similarly to [8]: we use the Hexagon DSP of the Qualcomm Snapdragon SoC available in off-the-shelf smartphones and wearables. This Qualcomm SoC is particularly suitable for always-on sensing tasks since the sensors can be continuously monitored at a low cost by the DSP allowing the power-hungry CPU to often remain in low-energy sleep mode. To give a perspective on the possible energy savings, we observe on average an 8× to 14× reduction in the energy consumption when the DNN inference algorithms run on the DSP instead of the CPU. These benefits come at the expense of several DSP limitations including: constraints on the size and complexity of the DNN (due to the small program and memory space of the DSP); as well as only the more simple inference algorithms having acceptable runtime latency (partially due to these algorithms not being fully optimized for the DSP). Naturally, well-known cloud-based models like DeepFace [7] (used by Facebook for face recognition) can not be supported locally with this prototype; rather at this point we can only use carefully constructed simple models.

The co-processor is programmable through the publicly released C/assembly Hexagon SDK but development is enabled only on special boards such as the Snapdragon 800 MDP/S (Figure 2) which we use for the classification engine implementation. We implement the sensing framework and algorithms in C: interfacing between the Android OS and the DSP is achieved through a general computational offloading mechanism (FastRPC) mediated through the Android Native Development Kit (NDK). Our DNN version for the DSP allows several key parameters to be changed, namely the number of hidden layers and their size, the number of features in the input layer, the number of classes in the output layer, as well as the node activation function. In the following sections, we tune these parameters accordingly and report smartphone results. However, the findings can be generalized to other mobile devices since the same Snapdragon architecture, featuring a DSP in addition to the CPU, is present on new consumer wearables like the Android LG G Watch with a Qualcomm Snapdragon 400 SoC.

4. INFERENCE ACCURACY

We begin by investigating the potential for more robust and accurate mobile inference by adopting techniques from deep learning. The two key results from our experiments are:

- Basic DNN techniques do well with noisy accelerometer activities: we observe a 10% accuracy gain over the next best comparison method, even when no deep learning pre-training methods are used to additionally boost the accuracy by initializing the weights of the network;
- For audio sensing (speaker and emotion recognition), a simple DNN model with a 71× reduction in the number of input features provides comparable or superior accuracy against learning techniques in common usage.

Behavioral Context	Dataset Description
Activity Recognition	wrist-worn accelerometer activities [10]
Emotion Recognition	emotional prosody speech [19]
Speaker Identification	10-minute speech from 23 speakers each

Table 2: Sensing datasets overview.

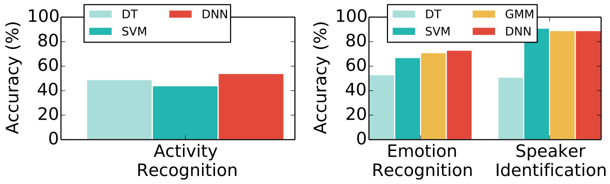
Such preliminary findings are indications of the possible benefits by adopting techniques from deep learning. Here we have only applied some of the most basic DNN-related machinery. Consequently, we believe more comprehensive exploration will lead to even larger performance gains.

Experiment Setup. We examine three inference domains commonly studied within mobile sensing, one based on accelerometer data and the others using the microphone. Specifically these are: activity recognition, emotion recognition, and speaker identification. The particular classes of behavior we study appear in a wide range of proposed and existing sensor-based mobile apps, for example: mHealth [6, 25], digital assistants (e.g., Microsoft's Cortana or Apple's Siri) and life-logging [20, 21]. The complexities of recognizing the categories of behavior evaluated in the wild – using conventional modeling – are well recognized [6, 17].

Datasets. Table 2 details the three datasets we use and specifies the classes of behavior they contain. Two of the datasets are audio-based (for speaker identification and emotion recognition) provided by the authors of [25] and one is accelerometer-based [10] containing a general set of Activities of Daily Life (ADL) shown in Table 1. The ADL dataset is composed of the labeled recordings of 14 simple activities performed by 16 volunteers wearing a single tri-axial accelerometer attached to the right wrist of the volunteer and sampled at a rate of 32Hz. The emotions corpus [19] contains the emotional speech of 7 professional actors delivering a set of 14 distinct emotions grouped by Rachuri et al. [25] into 5 broad categories: happiness, sadness, fear, anger and neutral speech. The speaker data consists of 10-minute voice recordings of 23 speakers reading article excerpts. The microphone sampling rate is set to 8kHz in the datasets.

DNN Design. For the accuracy benchmarks we evaluate a DNN with fairly standard parameters that can be trained fast with a basic backpropagation algorithm. The DNN has 1 hidden layer with nodes equal to $(f + c)/2$ where f is the number of input features and c is the number of output classes. A sigmoid activation function is employed for the hidden layer and a softmax function for the output layer. In the sound processing scenarios, the traditionally adopted Gaussian Mixture Models (GMMs) [4] accept as input features a series of 32 Perceptual Linear Predictive (PLP) coefficients [25] extracted from 30ms audio frames every 10ms for a total of 5 seconds. Consequently, the emotions and speaker inferences are performed on 5-second long utterances. The DNN uses instead summary features (mean, median, std, min, max, 25 percentile, 75 percentile) derived from the original ones to succinctly represent the distribution of each of the 32 PLP coefficients over the window. Thus, the DNN uses 7×32 features in total as opposed to 500×32 which significantly reduces the descriptiveness of the acoustic observations leading to potential accuracy losses.

Benchmark Classifiers. Comparison benchmarks are provided by a set of baseline classifiers commonly adopted in mobile sensing scenarios. Gaussian Mixture Models (GMMs) with diagonal covariance matrices are often used for sound processing [21, 23] which have proven particularly effective for speaker-related inferences [25, 20]. The classifier works with a maximum likelihood principle: each class to be recognized is represented by a single GMM and the classification computes the probability of each class in turn. Other techniques that generally yield good results are Support Vector Machines (SVM) [4] which have successfully been incorporated in emotion recognition systems [14]. Last, Decision Trees (DT) virtually dominate the activity recognition and transportation mode detection landscape [21, 26]. Like the DNN, for audio inference the SVM and DT operate on the summary features instead of those used by the GMM.



(a) Activity (b) Sound

Figure 3: Accuracy results for several popular classifiers applied to typical mobile sensing tasks: (a) activity recognition and (b) audio sensing. For the voice-related inferences, the DNN works with much simpler features (a 71× reduction in the total number of features compared to the GMM case) while still yielding comparable or better accuracy results. This demonstrates the highly discriminative and robust nature of the DNN modeling.

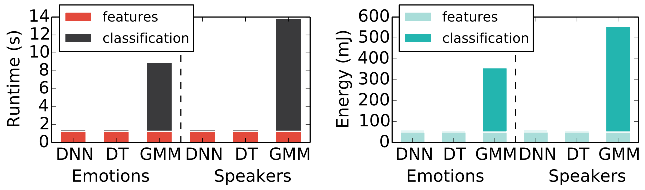
Experiment Results. In Figure 3a we display the various classifiers performance on the ADL dataset. We note that distinguishing between the 14 activities is a challenging task as some of them such as eating meat are fairly complex to be identified with a single accelerometer. The problem difficulty justifies the relatively low ($< 60\%$) accuracy achieved by the classification models with default parameters; yet, the DNN outperforms the DT leader by 10%. In this case, the DNN appears capable of uncovering hidden feature dependencies not easily captured by the DT branching logic.

In Figure 3b we compare the accuracy of the DNN using the weaker summary feature set for the emotion/speech processing. Here, even with the significant loss of feature complexity, *the DNN provides superior accuracy results of 73% for the emotion recognition example and comparable 89% accuracy for the speaker identification.* We highlight that these results are obtained when the DNN is trained without a pre-training step and further accuracy improvements are likely when restricted Boltzmann machines (or similar) are used for the initialization of network weights [13].

5. RESOURCE EFFICIENCY

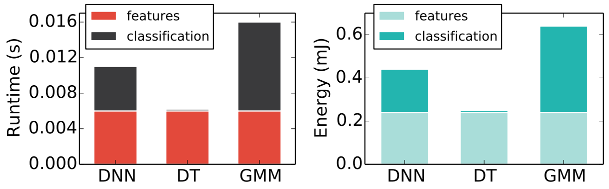
In our next set of results we examine energy and latency properties of DNNs applied to common behavioral inference tasks. The three key results from our experiments are:

- DNN use is feasible on the DSP and has a low energy and runtime overhead allowing complex tasks such as emotion detection or speaker identification to be performed in real time while preserving or improving the accuracy;
- DNN solutions are significantly more scalable as the number of recognized classes increase;
- Splitting models between computational units (e.g., a local device and cloud) is more flexible with a DNN that offers energy/latency trade-offs at a finer granularity.



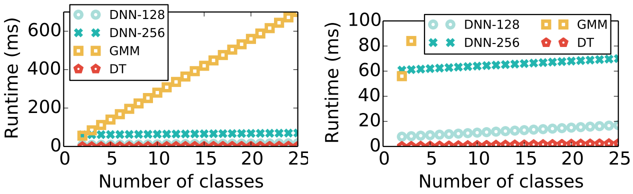
(a) Latency **(b)** Energy

Figure 4: Latency and energy of the emotion recognition and speaker identification when deployed on the DSP. The example DNNs have a low latency/energy overhead similar to a DT.



(a) Activity latency **(b)** Activity energy

Figure 5: Latency and energy of the activity recognition when deployed on the DSP. The accelerometer pipelines are extremely cheap and DNNs still have a lower overhead compared to GMMs.



(a) Full scale (0 to 600 ms) **(b)** Zoom in (0 to 100 ms)

Figure 6: DSP runtime of the inference stage of the various classifiers as a function of the number of classes. The results suggest that DNNs scale extremely well with the increase in the number of classes, in a manner similar to a DT, while often providing superior accuracy.

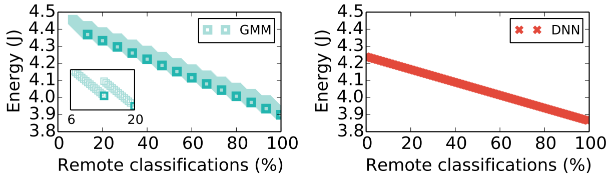
Our early results point to the ability of DNNs to provide energy and latency trade-offs that will be suitable for a wide-range of mobile sensing scenarios; while also having beneficial resource characteristics not found in any other commonly used model.

Experiment Setup. We use the implementation detailed in §3 to evaluate the energy and latency characteristics of the three inference domains from §4. Unless otherwise specified, the adopted default DNN parameters are 3 hidden layers, 128 nodes per layer, and a rectified linear unit (ReLU) [28] activation function. These settings closely match recently applied DNNs to speech and emotion recognition tasks [5, 15]. The DNN model is further used to implement a keyword spotting example [5] illustrating one of the key DNN approaches, namely hybridizing the classification with post-processing. The example brings to light cloud offloading benefits studied in the third of our experiments. The GMMs are set up with 128 mixture components [25]. The classification models (DT, GMM, DNN) and derived sensing applications used in the experiments are all deployed on a smartphone's DSP so that comparisons are put into a low-power context suitable for mobile sensing tasks.

Feasibility Results. In this first experiment we provide insights with respect to the DSP runtime and energy footprint of DNNs compared against other techniques (DT, GMM) widely used in the mobile sensing literature. In Figure 4 we plot the latency and energy profiles of the sound-related apps detailed in §4. The emotion recognition task with GMMs, for example, runs for approximately 9 seconds and requires 350mJ on the DSP to process 5 seconds of audio data. A most notable observation is that *the DNN classification overhead is extremely low compared to a GMM-based inference and matches the overhead of a simple Decision Tree*. We recall that both the emotion recognition and speaker identification operate on acoustic features extracted from 5 seconds of audio samples which means that *the DNN versions of the applications, unlike the GMM-based implementations, can perform complex sound-related inferences in real time with comparable or superior accuracy*. The prohibitively high GMM overhead stems from both the large amounts of features (500×32) serving as acoustic observations and the additive nature of the classification where one full GMM is required per class. In the activity recognition scenario examined in Figure 5, the results are similar: the DNN has a lower overhead compared to GMMs and inferences can be performed in real time. The runtime values for all models are reported for processing 4 seconds of accelerometer data and 24 features so that the low runtimes of barely 16ms indicate how cheap accelerometer-based sensor apps are.

Scalability Results. In this part of the analysis we shed light on how the DNN scales with the increase in the number of inferred classes. Mobile context inference tasks often require a larger number of behaviors or activities being recognized such as multiple activity categories [21] (e.g. still, running, walking with phone in pocket, backpack, or belt etc.), multiple words, emotional states or speakers [25]. In Figure 6 we plot the runtime of the classification stage of the three models (DT, GMM, DNN) as a function of the number of recognized contextual categories. Again, the DNN behaves in manner similar to a simple Decision Tree where the larger number of supported classes does not significantly affect the overall inference performance. The runtime of the feed-forward stage of a deep neural network is dominated by the propagation from the input and multiple hidden layers which are invariant to the number of classes in the output layer. The GMM-based classification computes probability scores for each class represented by an entire GMM so that an inference with 25 added categories/classes is 25× more expensive than one with a single class. This justifies the more than 11× slower inference compared to a 256-node DNN [15] for 25 recognized categories and an identical number ($750 = 25 \times 30$) of input features for all models.

Cloud Partitioning Results. In this experiment we investigate the benefits of DNN-based inference usage with respect to cloud offloading. To set up the experiment we consider a speech recognition scenario where a set of keywords need to be detected from voice on the mobile device. A common DNN approach adopted in speech processing [5, 15] is repeatedly invoking the DNN feed-forward stage on short segments, such as once every 10ms in a keyword spotting application [5], and then performing post-processing on the sequence of extracted DNN scores for obtaining the final inference, such as the probability of encountering a keyword. In Figure 7b we demonstrate that the high frequency of DNN propagations facilitates *cloud offloading decisions to be per-*



(a) GMM cloud offloading **(b)** DNN cloud offloading

Figure 7: Energy footprint of a speech recognition inference model based on GMMs or DNNs when a proportion of the classifications are performed in the cloud. For the GMM case a zoom-in for the 6% to 20% partition range is also provided. Experiment duration is 15 seconds with a WiFi connection assumed (5Mbps uplink). DNN usage allows for a graceful reduction in the energy consumption unlike the choppy GMM offloading.

formed at a fine level of granularity with a graceful reduction in the total energy consumption when a larger proportion of the DNN inferences are performed in the cloud.

In contrast, a GMM-based approach would usually increase the total amount of time acoustic observations (features) are accumulated before resorting to an inference. This together with the overhead of evaluating the probability of multiple GMMs (e.g. one per keyword) for a single inference, lead to the much choppier falls in the energy consumption for this model when a percentage of the GMM computations are offloaded to the cloud, as illustrated in Figure 7a. This phenomenon is portrayed in Figure 7a with the saw-like shape of the energy curve. We highlight that *such a curve is harder to control to a specific energy budget.* Situations where a certain number of the per-class GMM inferences need to be performed remotely may often be encountered because of latency/resource constraints, for instance, which introduces the above mentioned local-remote split inefficiencies. The DNN energy curve with a smoother gradient is therefore largely preferable.

6. CONCLUSION

In this paper, we have investigated the potential for techniques from deep learning to address a number of critical barriers to mobile sensing surrounding inference accuracy, robustness and resource efficiency. Significantly, we performed this study by implementing a DNN inference engine by broadly using the capabilities of modern mobile SoCs, and heavily use the DSP in addition to the CPU. Our findings show likely increases to inference robustness, and acceptable levels of resource usage, when DNNs are applied to a variety of mobile sensing tasks such as activity, emotion and speaker recognition. Furthermore, we highlight beneficial resource characteristics (e.g., class scaling, cloud offloading) missing from models in common use today (e.g., GMMs).

We believe this first step in understanding how deep learning can be used in mobile contexts provides a foundation for more complete studies, and will lead to the development of important innovative classifier designs for sensing apps. Our study only scratches the surface of potentially a revolution in the widespread adoption of consumer-ready sensing apps powered by deep learning.

7. REFERENCES

[1] Qualcomm Snapdragon 800 MDP. http://goo.gl/ySfCFl.

[2] L. Deng, et al. Recent Advances in Deep Learning for Speech Research at Microsoft. In *ICASSP '13*.

[3] S. Ji, et al. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Trans. Pattern Anal. Mach. Intel*, 35(1):221–231, Jan 2013.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

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

[6] S. Consolvo, et al. Activity Sensing in the Wild: A Field Trial of UbiFit Garden. In *CHI '08*.

[7] Y. Taigman, et al. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *CVPR '14*.

[8] P. Georgiev, et al. DSP. Ear: Leveraging Co-Processor Support for Continuous Audio Sensing on Smartphones. In *SenSys '14*.

[9] S. E. Kahou. Combining Modality Specific Deep Neural Networks for Emotion Recognition in Video. In *ICMI '13*.

[10] B. Bruno, et al. Analysis of human behavior recognition algorithms based on acceleration data. In *ICRA '13*.

[11] E. Cuervo, et al. MAUI: Making Smartphones Last Longer with Code Offload. In *MobiSys '10*.

[12] M. Zeng, et al. Convolutional Neural Networks for Human Activity Recognition using Mobile Sensors. In *MobiCASE '14*.

[13] L. Deng and D. Yu. Deep Learning: Methods and Applications. Now Publishers Inc. Jan. 2014.

[14] F. Eyben, M. Wöllmer, and B. Schuller. OpenEar – Introducing the Munich Open-source Emotion and Affect Recognition Toolkit. In *In ACII*.

[15] K. Han, D. Yu, and I. Tashev. Speech Emotion Recognition using Deep Neural Network and Extreme Learning Machine. In *Interspeech '14*.

[16] G. E. Hinton, S. Osindero, and Y.-W. Teh. A Fast Learning Algorithm for Deep Delief Nets. *Neural Comput.*, 18(7):1527–1554, July 2006.

[17] N. Lane, et al. Enabling large-scale human activity inference on smartphones using community similarity networks (CSN). In *UbiComp '11*.

[18] Q. V. Le, et al. Building high-level features using large scale unsupervised learning. In *ICML '12*.

[19] M. Liberman, et al. Emotional prosody speech and transcripts. 2002.

[20] H. Lu, et al. Speakersense: Energy efficient unobtrusive speaker identification on mobile phones. In *Pervasive '11*.

[21] H. Lu, et al. The jigsaw continuous sensing engine for mobile phone applications. In *SenSys '10*.

[22] W. S. McCulloch and W. Pitts. A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biology*, 5(4):115–133, Dec 1943.

[23] E. Miluzzo, et al. Darwin phones: The evolution of sensing and inference on mobile phones. In *MobiSys '10*.

[24] T. Plötz, et al. Feature learning for activity recognition in ubiquitous computing. In *IJCAI '11*.

[25] K. K. Rachuri, et al. Emotionsense: A mobile phones based adaptive platform for experimental social psychology research. In *UbiComp '10*.

[26] S. Reddy, et al. Using mobile phones to determine transportation modes. *ACM Trans. Sen. Netw.*, 6(2):13:1–13:27, Mar. 2010.

[27] Y. Wang, et al. Tracking human queues using single-point signal monitoring. In *MobiSys '14*.

[28] M. D. Zeiler, et al. On rectified linear units for speech processing. In *ICASSP '13*.