Overview of the Mobile Systems research at NetSys Lab

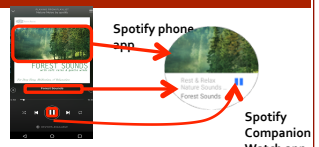
Jian Xu, Qingqing Cao, Aruna Balasubramanian (In collaboration with HexLab, LUNR lab, OSCAR lab, HI Lab, and Nokia)

Stony Brook University

Project #1: Easily extending smartphone apps to wearable devices

Problem: Writing wearable apps is tedious

Opportunity: Most wearable ap simply mirror the smartphone a

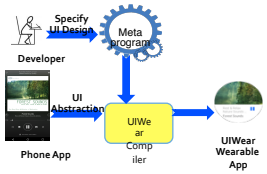


UIWear: Write once, extend to many

Idea: Decouple app design from app management.

Architecture: Developer writes simple metaprogram specifying app design

UIWear creates wearable app



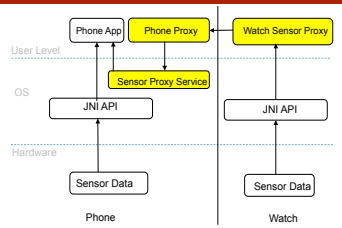
Implementation: On Android OS, AndroidWear, Sony Smartglass

Evaluation: Created 20 wearable apps with fraction of dev effort (some examples Lines-of-Code (LoC))

App	Wearable app LoC	UIWear LoC
4Sound	732	21
Anghami	2,263	79
BandLab	1,802	8
Endomondo	6,522	18

UIWear published at ACM Mobicom 2017

Wearable Sensor Virtualization (Ongoing)



Problem: Wearable sensors are not fully utilized by phone apps.

Idea: Design a sensor virtualization platform for phone apps to seamlessly access wearable sensors

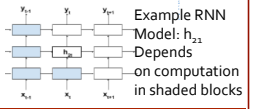
Project Website: sbuwear.cs.stonybrook.edu

Project #2: Enabling Deep Learning on Phones for privacy and performance

Background: Recurrent Neural Networks (RNNs) improve activity recognition, machine translation, and other tasks

Problem: Optimizations to run deep learning on phones focus on Convolutional Neural Networks (CNNs); Don't work for RNNs.

E.g., GPU offloading harder on RNNs due to dependencies

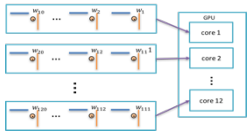


Example RNN Model: h_{21} Depends on computation in shaded blocks

MobiRNN: Offloading RNN models to GPU

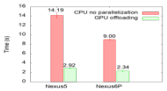
MobiRNN idea: a mobile specific optimization for RNNs that parallelize within a cell rather than across the cells.

MobiRNN performs coarse-grained parallelization where a row of computation is offloaded to a GPU core



Implementation: On TensorFlow, using RenderScript framework

Evaluation: RNN models run 3.9 times faster on GPU, using MobiRNN



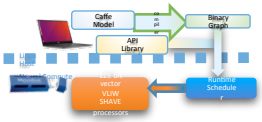
MobiRNN published at EMDL Workshop 2017

Optimizing the deep learning stack for Commodity Accelerators (Ongoing)

Problem: Existing deep learning models not optimized for hardware accelerators

Goal: Optimize deep learning pipeline to be well-suited for the hardware accelerator architecture

Commodity accelerator– Movidius Neural Compute Stick



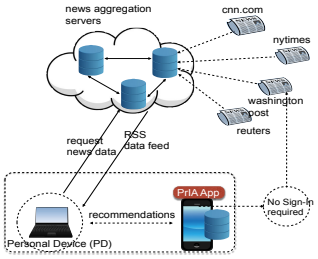
Project #3: Private Intelligent Assistant (PrIA)

Problem: Personalized intelligent assistance costs privacy

Users have no alternatives today

PrIA Idea: Enable intelligent assistance services completely locally from users personal devices

PrIA News Recommendation



Key idea: Decouple news aggregation from personalization

Architecture: PrIA downloads all news articles from an aggregation service, builds a local user profile, and provides recommendations, all on the users personal device

PrIA App

User Study: 6 users for an average of 10 days. Users rate recommendations from PrIA and Google News

PrIA performs poorer than Google News, but not significantly so

Recommendation system	Precision @10
Google	0.45
PrIA	0.38

PrIA published at HotMobile 2017

Ongoing work

- In-depth study of privacy leakage in intelligent home assistants such as Alexa and Google Home
- Designing techniques to reduce these privacy leakages by leveraging ideas from NLP and privacy research.

Project Website: pria.cs.stonybrook.edu

We gratefully acknowledge the National Science Foundation for supporting these projects.