Remote Inference for Microcontrollers at the Edge

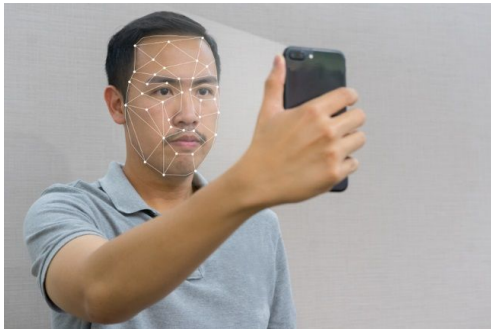
Jingtao Leo Zhang
ECE 202A, Fall 2021

Overview

- Introduction
- Related Work
- Technical Approach
- Evaluation and Results
- Discussion and Conclusions
- Future Directions

Introduction

- ML applications widespread
- Resource constrained microcontrollers
- Offload inference



Related Work

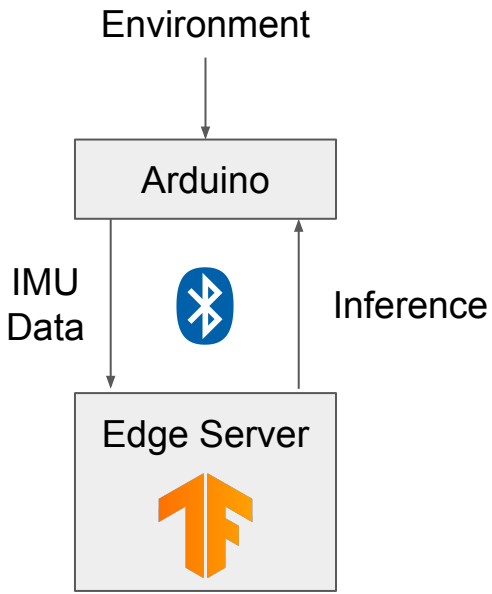
- TinyML
  - Since 2019
  - Models 10s of kB

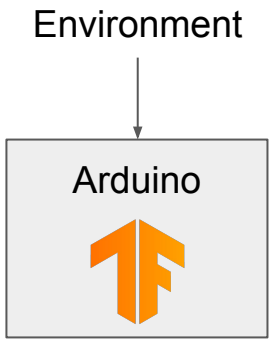
- Remote Inference
  - At the cloud
    - Models 10s of GB
  - At the edge
    - Models 10s of MB

TensorFlow Lite

Project Goal

- Remote vs TinyML
- Metrics:
  - Accuracy
  - Latency
  - Communication

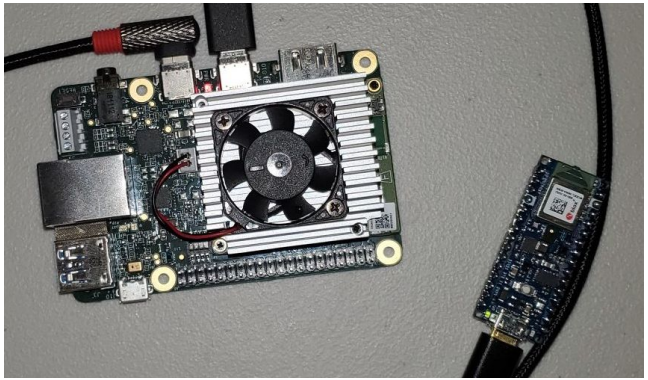




Technical Approach

Setups

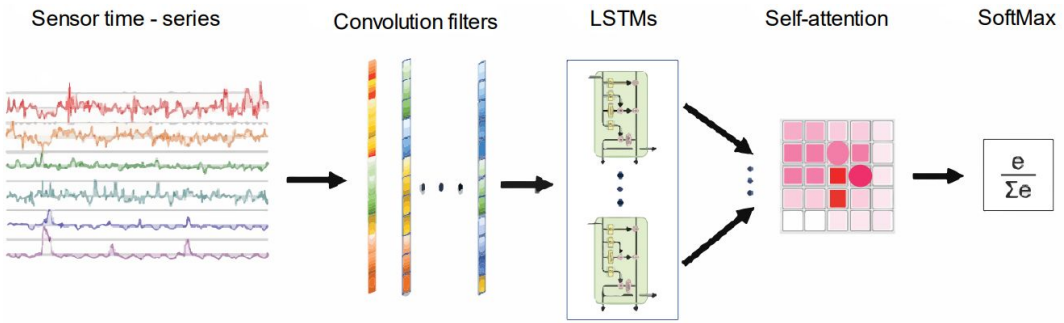
1. Arduino Nano 33 BLE Sense
    - Arm Cortex M4
2. Arduino + Coral Dev Board
    - Edge TPU
3. Arduino + PC
    - GPU



Arduino and Coral Dev Board

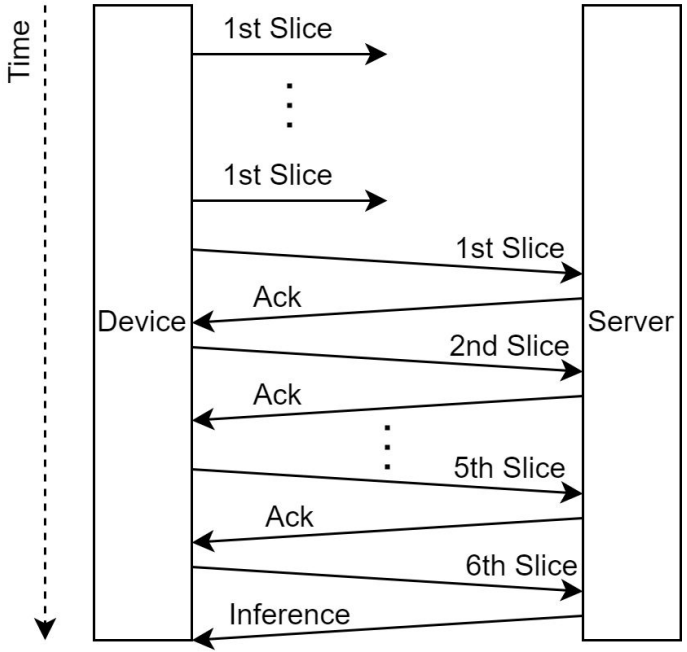
Technical Approach

- Benchmark Application
  - Human Activity Recognition
- Data
  - UCI Smartphone-Based Recognition of Human Activities and Postural Transitions
  - Gyroscope and Accelerometer at 50Hz
- Models
  - Arduino
    - CNN, 364 kB
  - Arduino + Coral
    - CNN, 1364 kB
  - Arduino + PC
    - DeepConvLSTM, 1785 kB

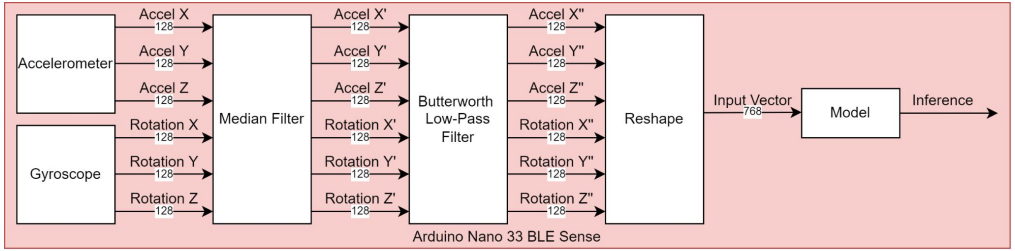


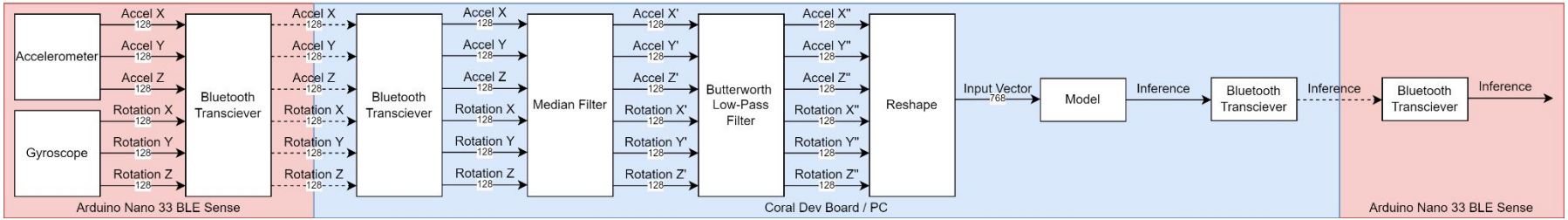
Technical Approach

- Communication
- 768 float input vector = 3072 Bytes
- BLE 4.2
  - 512 Byte/packet
- Requires 6 transmissions
  - 2 characteristics: TX and RX
  - Send and acknowledge



Technical Approach





Evaluation and Results

Setup	Model Size	Accuracy	Processing	Communication	Inference	Total
Arduino	364 kB	74.7%	0.05090	N/A	1.20305	3.80777
Arduino + Coral	1364 kB	92.1%	0.01258 (4.05x)	2.92701*	0.01843 (66.76x)	5.51802* (-1.45x)
Arduino + PC	1785 kB	98.7%	0.00312 (16.3x)	2.92701	0.01428 (84.25x)	5.50441 (-1.45x)

* I damaged the bluetooth chip on the Coral board, assuming PC measurements in its place

- Constant 2.56s sampling time
- Averaged over 100 trials
- BLE throughput: 3072 B / 2.92701s = 1.05kB/s
- Large communication penalty

Discussion and Conclusions

Setup	Sampling	Processing	Communication	Inference
Arduino	67.2%	1.3%	N/A	31.6%
Arduino + Coral	46.4%	0.228%	53%	0.334%
Arduino + PC	46%	0.057%	53.2%	0.259%

- 3s from end of data collection to classification for remote v.s. 1.2s for local
  - Could be OK depending on application
- Improved accuracy better be worth it

Future Directions

How to reduce the communication penalty

- Compressed Sensing
- Model Partitioning
- Aggressive Quantization
- Communication Protocol Refinement