T4Train:

Rapid Prototyping of ML-Driven Interactive Applications

Unit 1

Yasha Iravantchi
Ph.D. Candidate, Computer Science & Engineering

Alanson Sample
Associate Professor, Computer Science & Engineering

COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF MICHIGAN

Unit 1: Introductions

- Introduce yourself
  - Background, research interests
- What you hope to get out of this course
- One fun fact about yourself!
- Downloads and Installations (CHI WiFi is SLOW! ;)
  - Arduino
  - T4Train
  - Anaconda3-2022.10

Unit 1: Discussion Agenda

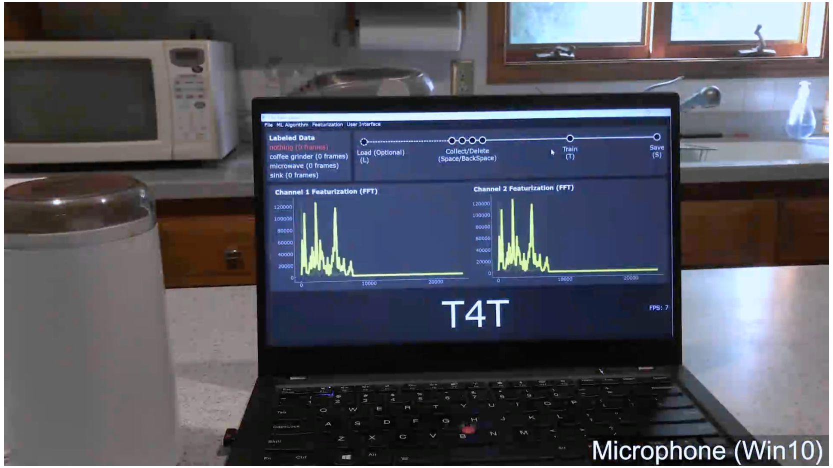
- Overview of Interactive Sensing
  - Types of Sensors
  - Common Sensor Interfaces
  - Signal Processing
    - Filtering, conditioning, and cleanup
  - Featurization
    - Binning, FFT
  - Classical ML Techniques
- Brainstorming an Application
- Installing Software for Workshop

Unit 2: Workshop Agenda

- Connecting a Sensor
- Visualizing the Data
- Training a Model
- Triggering an Action

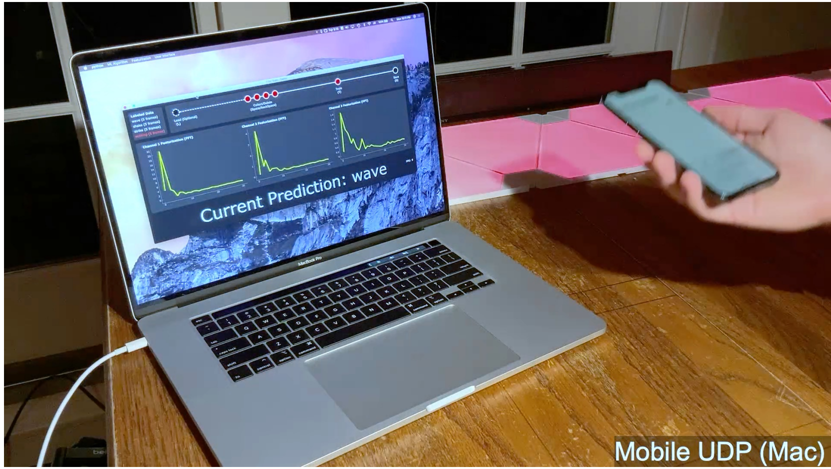
Why do Interactive Sensing?

- Demonstrating interfaces that are responsive in realtime is an important part of technical HCI
- Provides a high-level view of the capabilities of a system



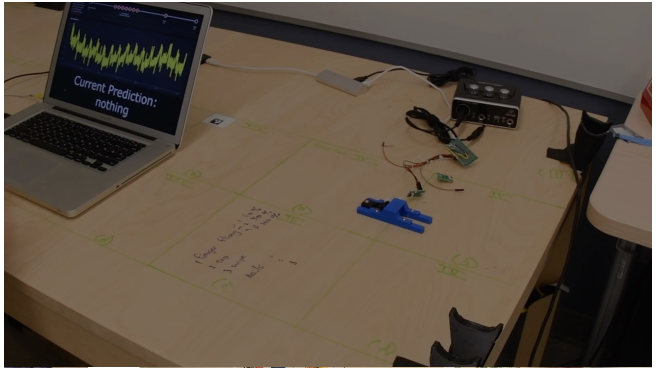
What is Interactive Sensing?

- Connecting a Sensor
- Visualizing the Data
- Training a Model
- Triggering an Action



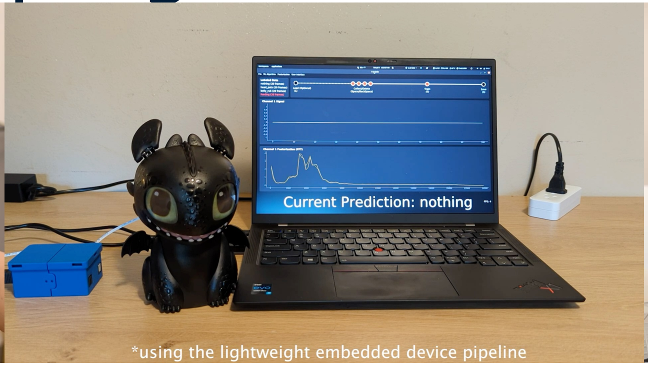
Fast prototyping of Interactive ML

- What could we do if we had a tool that could quickly prototype interactive ML?
- Test different sensors, signal processing approaches, ML models
- Quickly generate useful applications



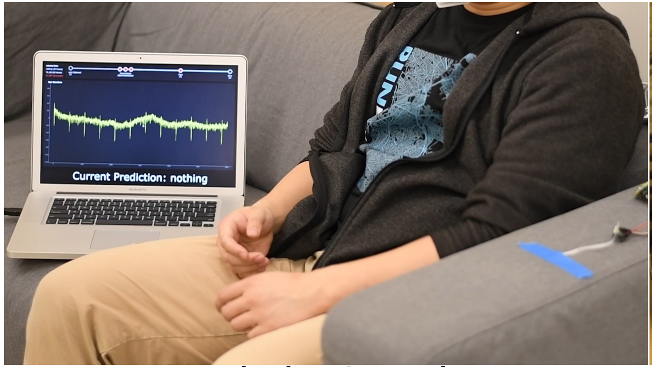
Fast prototyping of Interactive ML

Coffee Machine

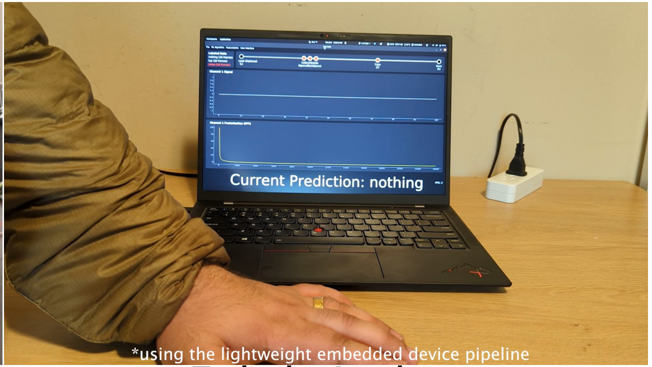


Toy Dragon

Active Speaker



Fabric Couch



Fabric Jacket

Overview of Interactive Sensing

6 Step Process:

1. Identify a potential signal
2. Design/select a sensor
3. Digitize the signal
4. Visualize the signal for human interpretation
5. Use signal processing and ML for computer interpretation
6. Perform an action based on ML predictions

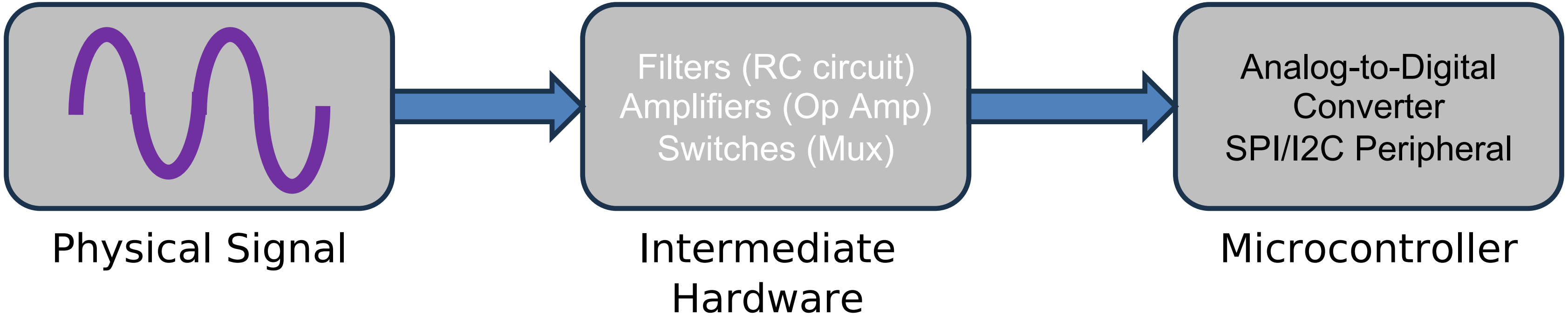
Types of Sensors

- Optical
  - Cameras
  - Depth Sensor
  - LiDAR
- IMU/Inertial
  - Accelerometer
  - Magnetometer
  - Geophone
- Acoustic
  - Microphone
  - Ultrasonic Rangefinder
- What else? Discuss

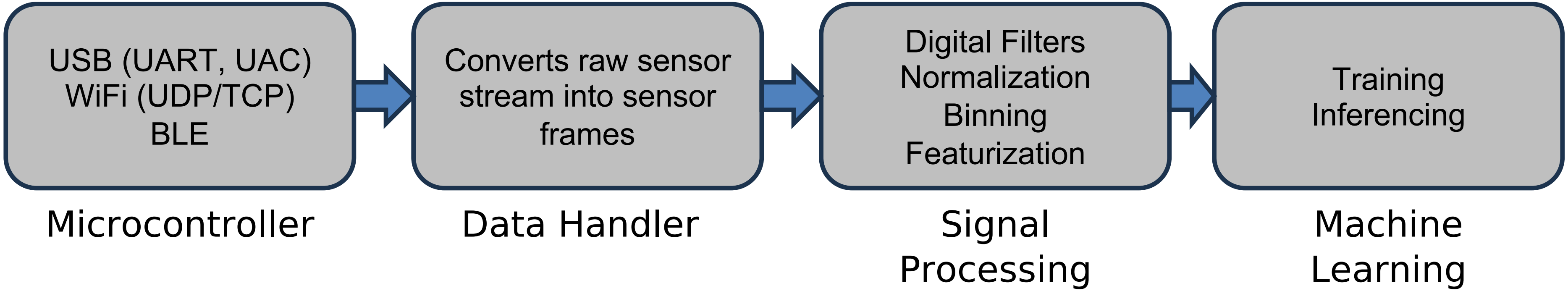
Common Sensor Interfaces

- USB
  - **Serial/UART**
  - **Universal Audio Class**
  - Universal Video Class
- WiFi TCP/UDP
  - Large bandwidth capabilities, comes at the cost of power
- Bluetooth/BLE
  - Constrained bandwidth, great for low power
  - Unintuitive data transfer structure
  - BLE UART is great for sensor data
- Anything else?

Signal Lifecycle (Sensor Side)

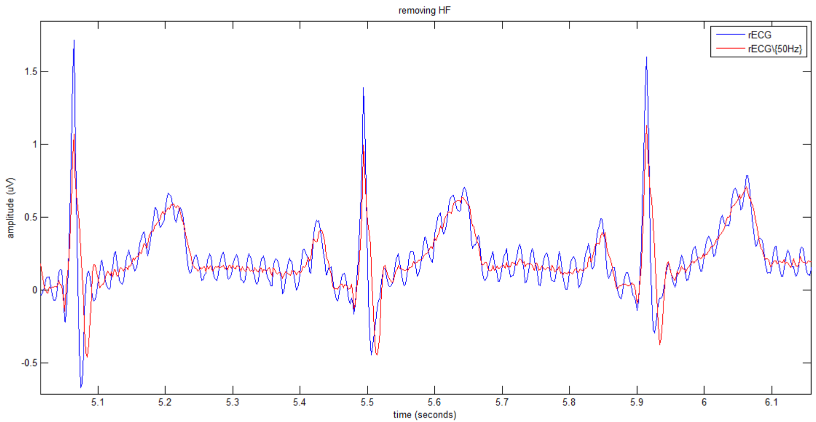


Signal Lifecycle (Computer Side)



Signal Processing

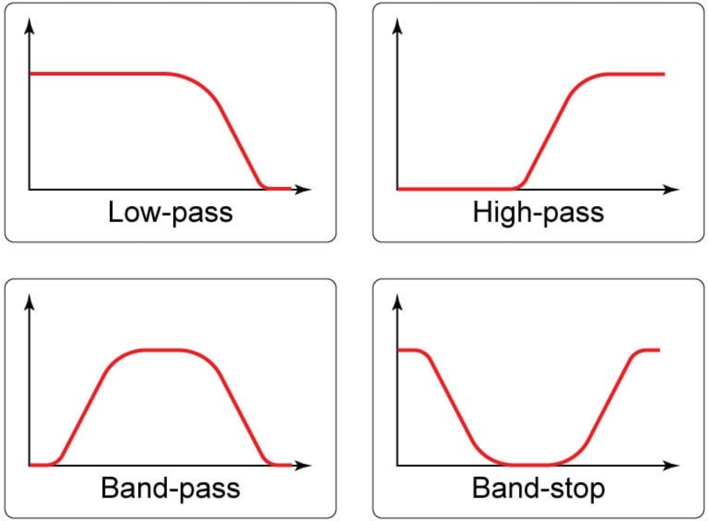
- Why do we need signal processing?
  - Signals are usually imperfect
  - Can have many noise sources
  - Signal we want might be “buried” in sensor data



Sensor data is corrupted by 50Hz line noise, but with filtering and smoothing can be recovered

Noise Rejection Filtering

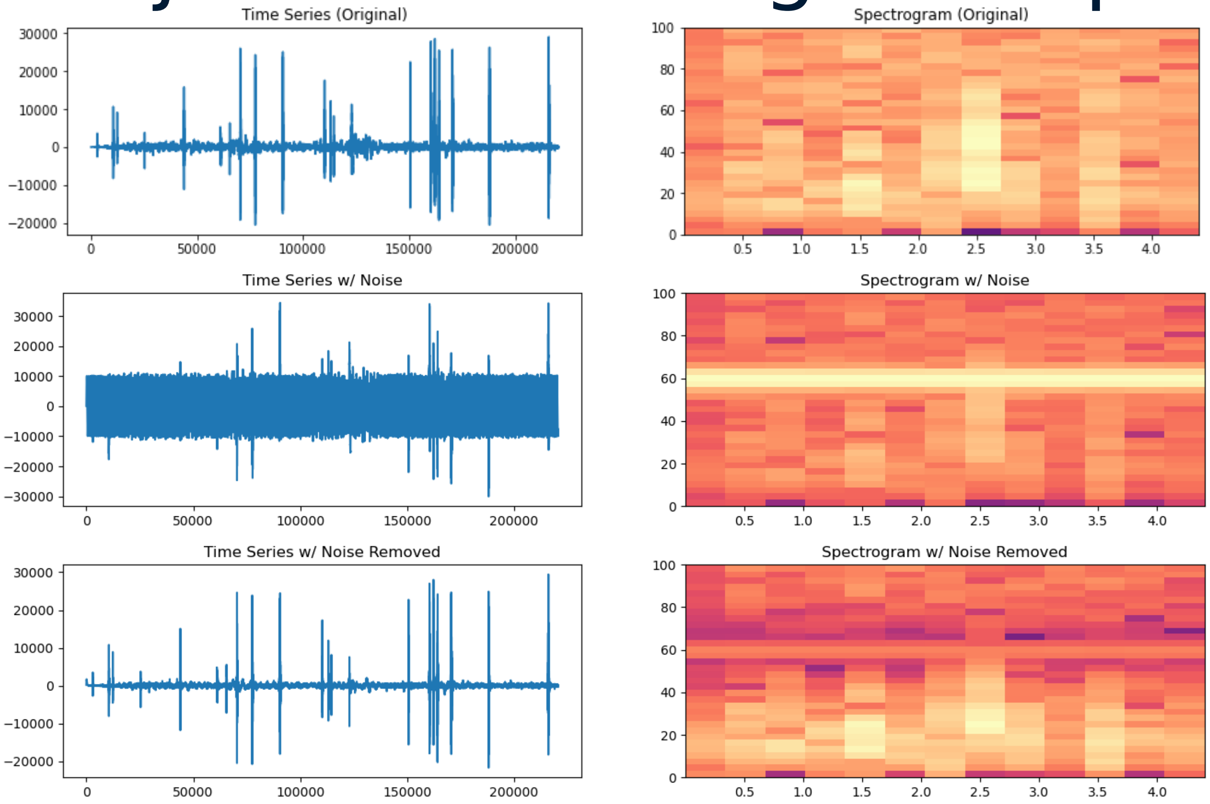
- Applying digital filters is an efficient way to remove unwanted noise in sensor data
- In previous example, a band-stop filter at 50Hz was applied to remove only 50Hz noise
- There are a few different kinds of filters, but generally:
  - Low-pass to reject high frequencies
  - High-pass to reject low frequencies
  - Band-pass to reject some low, some high
  - Band-stop to reject specific frequencies



Noise Rejection Filtering Example

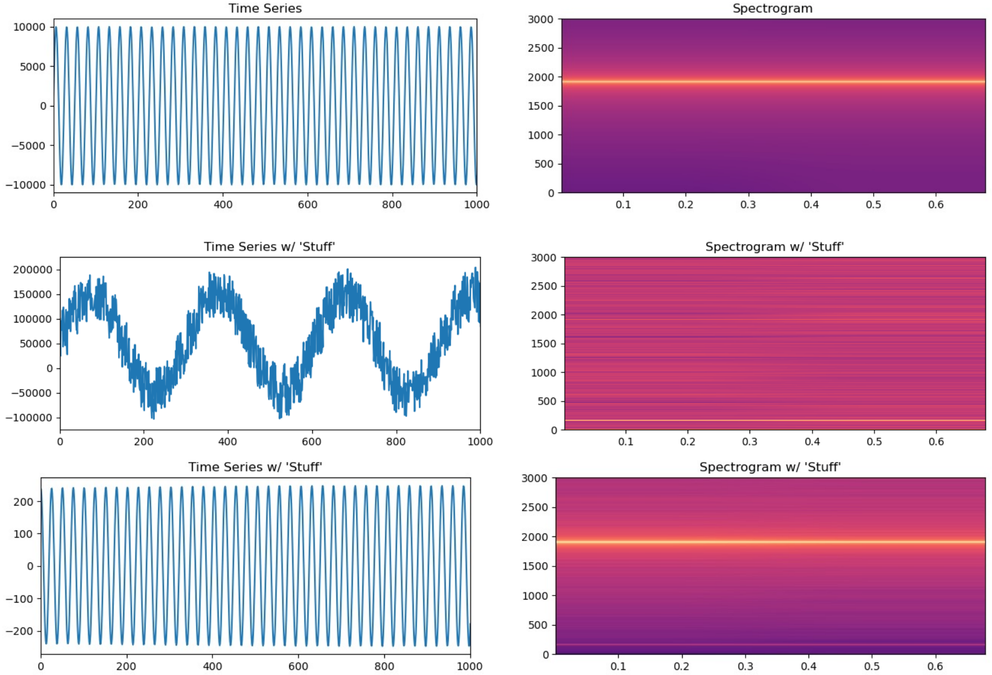
- Applying digital filters is an efficient way to remove unwanted noise in sensor data
- In previous example, a band-stop filter at 50Hz was applied to remove only 50Hz noise
- There are a few different kinds of filters, but generally:
  - Low-pass to reject high frequencies
  - High-pass to reject high frequencies
  - Band-pass to reject some low, some high
  - Band-stop to reject specific frequencies

Noise Rejection Filtering Example



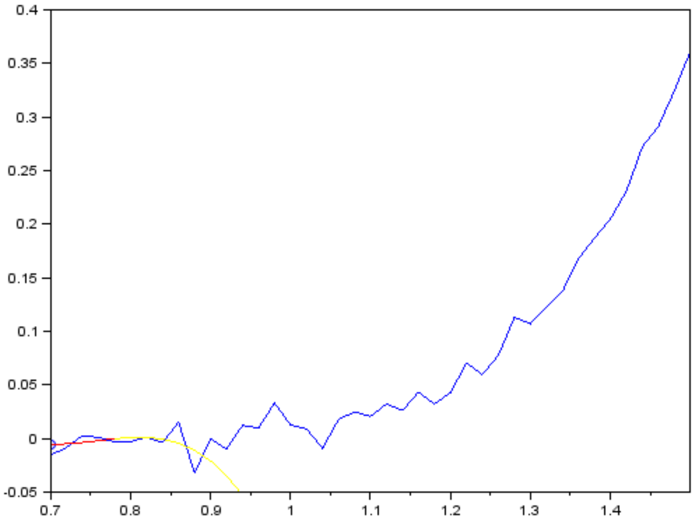
Signal Extraction Filtering

- You can also use digital filters to extract valuable signal buried in the sensor data
- For example, if we know our signal is in a certain frequency range, a band-pass filter can reject all but our desired signal



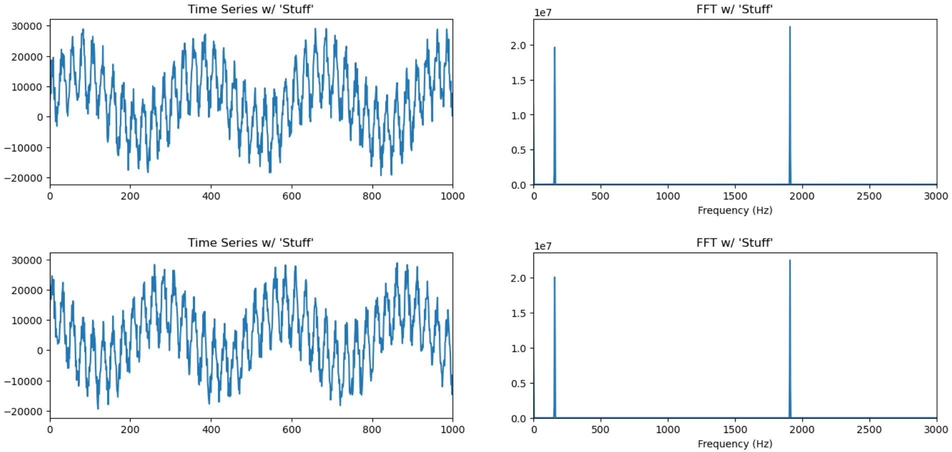
Signal Conditioning and Smoothing

- Filters can be used to smooth
  - SavGol Filter
  - Kalman Filter
- Sometimes it may be easier to smooth with sliding window methods
- Few different ways:
  - Median Filter
  - Moving Average
  - Moving Maximum
  - Moving Minimum



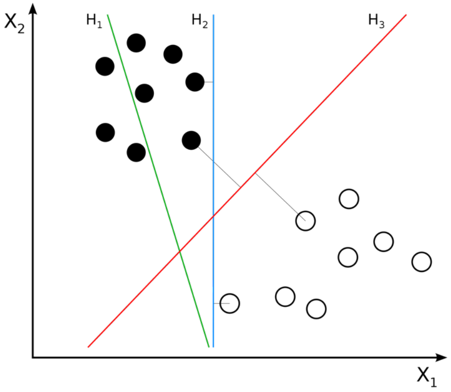
Featurization

- Why do we need featurization?
- For many time-series sensor data, it is difficult to represent it consistently for classical ML



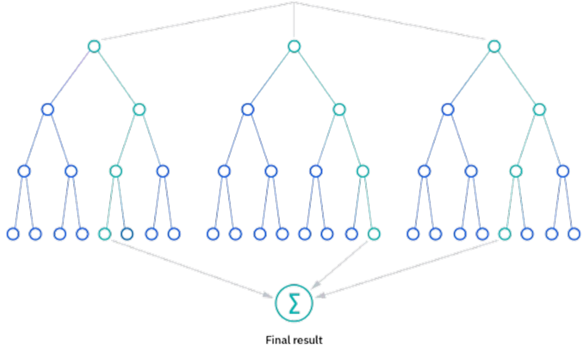
Classical ML Techniques

- T4Train utilizes classical ML approaches:
  - Can generally train “on the fly”
  - Fast for realtime inferencing
  - Does not require GPU



Support Vector Machines

Random Forests



Jupyter Notebook Example

- Sound Classification

T4Train Hello World

- Microphone Demo
- Nano 33 Sense Accelerometer Demo

Brainstorming an Application

- Pair in groups of 3-4

T4Train:

Rapid Prototyping of ML-Driven Interactive Applications

Unit 2

Yasha Iravantchi
Ph.D. Candidate, Computer Science & Engineering

Alanson Sample
Associate Professor, Computer Science & Engineering

COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF MICHIGAN

Unit 2: Workshop Agenda

- Connecting a Sensor
- Visualizing the Data
- Training a Model
- Triggering an Action