EE P 596: TinyML

Spring 2024
Dept. of Electrical and Computer Engineering
University of Washington

Instructor: Dinuka Sahabandu (sdinuka@uw.edu)
Foundations and Applications of TinyML

- **TinyML**: Emerging area where **ultra large powerful ML models are converted into executables for embedded systems** that are battery operated and mostly well beyond the operation capacity of the smart phones (e.g., microcontrollers)

Source: https://towardsdatascience.com/tiny-machine-learning-the-next-ai-revolution-495c26463868
Foundations and Applications of TinyML

TinyML is **real-time processing of time-series data that comes directly from sensors**

*Source: https://www.tinyml.org/about/*
Foundations and Applications of TinyML

- TinyML has **applications in agriculture, health, retail, energy industry**, and more...

  - **Plant disease classification with TensorFlow Lite on Android**
    

  - **Solar Scare Mosquito: A solar-operated device that sits on stagnant water to create air bubbles at regular intervals to avoid the breeding of mosquitoes**
    
    Source: https://theindexproject.org/award/nominees/6558

  - **TinyML for keeping an eye on the inventory of goods on the shelf in retail establishments and sending out warnings when it runs low**
    
    Source: https://www.supermarketnews.com/store-design-construction/amazon-go-goes-smaller
Foundations and Applications of TinyML

- Trend in ML and IoT (TinyML) market

Source: https://www.grandviewresearch.com/industry-analysis/deep-learning-market

A Schematic View of ML and Its Phases

Analysis Phase
- Raw Data
- Features
- Processed Data

Training, Testing, and performance tuning Phase
- Learning Algorithm

Application Phase
- Model $f$
A Schematic View of TinyML and Its Phases

Analysis Phase
- Raw Data
- Features
- Processed Data

Training, Testing, and performance tuning Phase
- Learning Algorithm
- Model $f$

Application Phase
- Model Deployment and Field Testing
- Model Deployment
- Compressed Model $\tilde{f}$

Model Compression Compilation Phase
TensorFlow (TF) and TFLite Workflow for TinyML

Source: https://leonardocavagnis.medium.com/tinyml-machine-learning-for-embedded-system-part-i-92a34529e899
Foundations and Applications of TinyML

- One of the first courses to bring ML, embedded systems, and IoT together
- First two weeks of classes will cover Fundamentals of ML/TinyML
- From week 3 – week 9, we will study one real-word TinyML application per class. Each problem will have a real-world dataset to work on
- From week 3 – week 9, first half of each class will focus on the needed background of the real-world application to be studied that week. The second half is on learning to train ML model, deploy TinyML model, and test the performance
- All labs use Python and C for coding, and we will provide needed modules and also work with the students during the labs
Prerequisites

• **Familiarity with Python programming**: This will be beneficial as Python is a commonly used language in machine learning and deep learning applications.

• **Basic understanding of C or C++**: Familiarity with one of these programming languages is beneficial, especially for deploying compressed machine/deep learning models onto Arduino, which is a key component of the TinyML course.

• **No prior coursework in machine/deep learning is necessary**: We will introduce and cover the essential basics of machine/deep learning in the first two weeks. Additionally, we will explore these concepts in greater depth with respect to various applications studied from week 3 to week 9.
What You Will Learn

- How to **deploy TinyML models on power and performance-constraint devices to solve real-word problems**
- How to **implement machine learning algorithms** such as k-means clustering, regression, classification, and ensemble learning methods
- How to **use Python libraries** - NumPy, Pandas, Seaborn, and Scikit-learn
- Using **TensorFlow for deep learning** and **TensorFlow Lite (TFLite) for TinyML**
- Using **C language for deploying TinyML on Embedded Systems**
- How to **measure the performance** of the deployed TinyML models
- How to efficiently and effectively run TinyML

- **Course Grade will be based upon homework/projects (45%) and a final project (55%)**
Resources

• Textbooks:
  – *TinyML Cookbook* by Gian Marco Lodice

• Free Online Material:
  – TinyML Foundation: https://www.tinymce.org
  – Cainvas Platform: https://cainvas.ai-tech.systems/gallery/

• Software and Hardware:
  – TensorFlow Lite: https://www.tensorflow.org/lite
  – Tiny Machine Learning Kit Arduino (≈$60)
Course Summary: Topics Covered

• Week 1: Introduction to TinyML
  - TinyML Landscape and Related Statistics
  - TinyML Applications
  - TinyML Challenges
  - Introduction to Software and Hardware Used in the Course

• Week 2: Fundamentals of ML and TinyML
  - Background on ML: Curve Fitting, Prediction, Overfitting vs. Underfitting
  - Background on Neural Networks: DNNs, CNNs, Dataset Split (Train, Test, and Validation)
  - TinyML Lifecycle and Workflow
  - Quantization Aware Training (QAT) and Post Training Quantization (PTQ)
  - Tiny Deep Learning
  - TensorFlow Lite (TFLite) for TinyML
Course Summary: Topics Covered

• Week 3: TinyML for Keyword Spotting
  - Background on Keyword Spotting and Streaming Audio
  - Challenges and Constraints in Keyword Spotting
  - Keyword Spotting Architecture and Data Collection
  - Model Training, Evaluation Metrics, and Deployment

• Week 4: TinyML for Visual Wake Words
  - Introduction to Visual Wake Words and Its Challenges
  - Visual Wake Words Dataset
  - MobileNets
  - Transfer Learning for Visual Wake Words
  - Model Training, Evaluation Metrics, and Deployment
Course Summary: Topics Covered

• Week 5: TinyML for Anomaly Detection
  - Background on Anomaly Detection and Signal Processing
  - Real and Synthetic Datasets
  - Unsupervised Learning (K-Means Clustering and Autoencoders)
  - Threshold Choice
  - Model Training, Evaluation Metrics, and Deployment

• Week 6: Robust navigation with TinyCNN
  - Background on robust low power autonomous driving and challenges
  - Closed loop learning system via Imitation Learning
  - TinyML Approach to replace conventional CVA by CNN
  - Data collection and modeling
  - Model Training, Evaluation Metrics, and Deployment
Course Summary: Topics Covered

• Week 7: TinyML for Predictive Maintenance
  - Background on Predictive Maintenance Solutions and Industry Applications
  - Sensors, Sensor Data, and Interface
  - Accelerometer, Gyroscope, Barometer, and Magnetometer
  - TinyML Framework for Predictive Maintenance
  - Model Training, Evaluation Metrics, and Deployment

• Week 8: TinyML for American Sign Language (ASL) Interpretation
  - Background on ASL and ASL Interpretation
  - Gesture Motion Datasets and Features
  - Analyzing Gesture Motion Data using Neural Networks
  - TinyML Framework for ASL Interpretation
  - Model Training, Evaluation Metrics, and Deployment
Course Summary: Topics Covered

- **Week 9: Smart Lock Recognition using TinyML**
  - Audio classification for deploying sensitive smart lock model
  - Data processing on audio data
  - Generate, train, and test a TensorFlow model using the SensiML Python SDK
  - Compile and flash the model to the edge device and display the inferred classes in the SensiML Open Gateway user interface

- **Week 10: Final Project Presentations**
  - Each Group has 12 minutes (Suggested presentation – 9 minutes; Q&A— 3 mins)
  - Signup for the presentation order (Same as the project signup)

- Final report due on June 7th 11:59pm, 2024