EE P 596: TinyML

Number of credits: 4 credits

Quarter and year: Spring 2024

Instructor's name: Dinuka Sahabandu

Meeting Time: Wednesdays 6-9:50 p.m.

Classroom: ECE 269

COURSE DESCRIPTION

This hands-on software and hardware course delves into Machine Learning (ML) on intelligent edge devices, particularly in the realm of the Internet of Things (IoT). TinyML, a field that emerged in 2020, is still in its early stages. It offers a rich and wide range of applications, such as wake word detection (Alexa; Siri, Google) and is increasingly relevant given the presence of over 30 billion IoT devices globally. There is a growing demand for integrating ML models that are both compact and high-performing yet require fewer resources to be pluggable into the edge devices below the power of mobile phones. Porting powerful cloud-level ML networks to edge devices has the potential to significantly improve inference, classification, and prediction quality in numerous everyday applications. But such compression also requires clever algorithmic thinking leading to TinyML on edge devices. Each class will address one problem from application description to implementation. Students will be provided with hardware platforms for class engagements.

Each class session of this course is dedicated to exploring one real-world application, drawn from the list of audio/visual wake word detection, wizard magic wand, anomaly detection, sign language interpretation, predictive maintenance, activity detection and classification for smart locks. Each class of the course will focus on developing the full TinyML pipeline starting from the data collection via sensing to the final classification on the edge device. Students will learn to compact these models through pruning, quantization, and knowledge distillation, enabling their deployment on edge-level hardware while maintaining desired performance accuracy. We will also address the often-overlooked vulnerabilities arising from the assumption that edge devices operate in benign environments, with a special focus on Robust TinyML, highlighting the importance of security and resilience in these systems.

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 boards, 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 few weeks. Additionally, we will explore these concepts in greater depth with respect to various applications studied from week 3 to week 9.

LEARNING OBJECTIVES

By the end of this course, students will demonstrate the ability to:

- Deploy TinyML models on power and performance-constrained devices to solve real-word problems.
- Implement machine learning algorithms such as k-means clustering, regression, classification, and ensemble learning methods.
- Utilize Python libraries - NumPy, Pandas, Seaborn, and Scikit-learn.
- Utilize TensorFlow for deep learning and TensorFlow Lite (TFLite) for TinyML.
- Utilize C language for deploying TinyML on Embedded Systems.
- Measure the performance of the deployed TinyML models.
- Efficiently and effectively run TinyML.

LECTURE SCHEDULE

- Week 1: Introduction to TinyML
    - TinyML Landscape, Applications, and Challenges
    - TinyML Lifecycle and Workflow
    - Model Compression Techniques
    - Recap on Necessary ML Background: ML Algorithms, Neural Networks
    - Introduction to Hardware and Software Used in the Course
- Week 2: Fundamentals of ML and TinyML
    - Pruning ML models
    - Quantization Aware Training (QAT) and Post Training Quantization (PTQ)
    - Knowledge Distillation
    - Tiny Deep Learning
    - TensorFlow Lite (TFLite) for TinyML

- 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
- Week 5: TinyML for Anomaly Detection
    - Background on Anomaly Detection and Signal Processing
    - Real and Synthetic Datasets
    - Unsupervised Learning
    - Threshold Choice
    - Model Training, Evaluation Metrics, and Deployment
- Week 6: Wizard Magic Wand
    - Gesture Tracking through Bluetooth
    - CNN for Magic Wand Sketch
    - Data Collection and Labeling
    - Model Training, Evaluation Metrics, and Deployment
- 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

- Week 9: Smart Lock Audio 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: Lecture or possibly additional time for the final project
- Week 11: 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 at 11:59 pm, 2024

COURSE MATERIALS

Textbook

- TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers 1st Edition by Pete Warden and Daniel Situnayake [O'REILLY Publisher]

References

- TinyML Cookbook by Gian Marco Lodice

Additional Resources:

- Free Online Material:
    - TinyML Foundation: https://www.tinyml.org
    - Canvas Platform: https://cainvas.ai-tech.systems/gallery/
- Software:
    - TensorFlow: https://www.tensorflow.org
    - TensorFlow Lite: https://www.tensorflow.org/lite

Materials:

- Tiny Machine Learning Kit Arduino

GRADING CRITERIA

Labs:

- Throughout the course, labs will be integrated into the class and cover each class's second part.

Quizzes/Homework / Project:

- Each class will recap the materials learned through team-based quizzes. There will be three homework the whole course. Throughout the quarter, you will also form groups with others to work on a course project we suggest or proposed by the students.

Grading:

- In-class labs/quizzes: 10%
- Homework: 45%
- Project: 45%
    - Proposal: 5%
    - Final presentation: 20%
    - Final report: 20%

COURSE POLICIES

- Please complete the homework by yourself and do not copy code from others or the internet. Any answer from Chat-GPT needs to be identified as such. Suppose your homework is identical to others or any sample code snippets online. In that case, you will receive zero scores, and UW mandates us to report it to the College of Engineering.
- You are encouraged to discuss lab and homework content with your classmates offline and on the discussion board. However, please limit your discussion to ideas and not discuss the code. Specifically, do not copy-paste any homework answers into the discussion board.
- Students must submit homework and project materials online by the posted due date. Throughout the quarter, we will provide you with six late-day credits for the three homework assignments, which you can use to submit one or more homework without a penalty. However, if you have used all six days, each additional late day will result in a 20% penalty in the corresponding homework.
- Students need to submit project reports and presentations on time. The project does not have any credit days. Any overdue project materials will receive a zero score.

RELIGIOUS ACCOMMODATIONS

Washington state law requires that UW develop a policy for accommodation of student absences or significant hardship due to reasons of faith or conscience, or for organized religious activities. The UW's policy, including more information about how to request an accommodation, is available at Religious Accommodations Policy (https://registrar.washington.edu/staffandfaculty/religious-accommodations-policy/). Accommodations must be requested within the first two weeks of this course using the

Religious Accommodations Request form (https://registrar.washington.edu/students/religious-accommodations-request/).

ACCOMMODATIONS AND ACCESS

If you have already established accommodations with Disability Resources for Students (DRS), please communicate your approved accommodations to me at your earliest convenience so we can discuss your needs in this course. If you have not yet established services through DRS, but have a temporary health condition or permanent disability that requires accommodations (conditions include but not limited to; mental health, attention-related, learning, vision, hearing, physical or health impacts), you are welcome to contact DRS at 206-543-8924 or uwdrs@uw.edu or disability.uw.edu. DRS offers resources and coordinates reasonable accommodations for students with disabilities and/or temporary health conditions. Reasonable accommodations are established through an interactive process between the student, instructor, and DRS. It is the policy and practice of the University of Washington to create inclusive and accessible learning environments consistent with federal and state law.

ACADEMIC INTEGRITY

Engineering is a profession demanding a high level of personal honesty, integrity and responsibility. Therefore, it is essential that engineering students, in fulfillment of their academic requirements and in preparation to enter the engineering profession, adhere to the College of Engineering Statement of Principles. Any student in this course suspected of academic misconduct (e.g., cheating, plagiarism, or falsification) will be reported to the College of Engineering Dean's Office and the University's Office of Community Standards and Student Conduct to initiate the student conduct process.

TITLE IX

"UW, through numerous policies, prohibits sex- and gender-based violence and harassment, and we expect students, faculty, and staff to act professionally and respectfully in all work, learning, and research environments. For support, resources, and reporting options related to sex- and gender-based violence or harassment, visit UW Title IX's webpage (https://www.washington.edu/titleix/), specifically the Know Your Rights & Resources guide (https://www.washington.edu/titleix/files/2020/08/KYRR-guide-8-10-2020-LINKED.pdf).

If you choose to disclose information to me about sex- or gender-based violence or harassment, I will connect you (or the person who experienced the conduct) with resources and individuals who can best provide support and options. You can also access those resources directly:

- Confidential: Confidential advocates (https://www.washington.edu/sexualassault/support/advocacy/) will not share information with others unless given express permission by the person who has experienced the harm or when required by law.
- Private and/or anonymous: SafeCampus (https://www.washington.edu/safecampus/) provides consultation and support and can connect you with additional resources if you want them. You can contact SafeCampus anonymously or share limited information when you call

Please note that some senior leaders and other specified employees have been identified as "Officials Required to Report." (https://www.washington.edu/titleix/title-ix-officials-required-to-report/) If an Official Required to Report learns of possible sex- or gender-based violence or harassment, they are required to call SafeCampus and report all the details they have in order to ensure that the person who experienced harm is offered support and reporting options (https://www.washington.edu/titleix/resources/)."