CS249r Final Project

Important Deadlines

Goals and Scope

Deliverables

- Proposal - 5%
- Check-Ins - 0%
- Final Report and Code - 75%
- Video/Presentation/Poster - 20%

Project Topics

Example Project Ideas

Important Deadlines

- **Proposal**: October 30th
- **Check-Ins**: Around Thanksgiving
- **Final Report and Code**: December 16th
- **Video/Presentation/Poster** (TBD on which): December 16th

Goals and Scope

The CS 249r final project provides an opportunity for you to apply and extend the foundational concepts you have learned in the course. It is intended to encourage you to integrate ideas from different course components and allow you to delve deeper into areas of interest. Pick something that interests you! We know that some students have a stronger embedded systems and/or machine learning background and are fine with projects that lean much heavier into one or the other. However, **all projects must include some aspect of BOTH embedded systems and machine learning as they must be a TinyML project!**

A list of possible course projects has been given below. Their content and scope are meant to be suggestive, not definitive. Final projects that fall outside the specifics but within the spirit of these suggestions are encouraged. It is, however, essential that the project make clear connections with topics covered in the course. If the idea you have for a project varies significantly from the suggestions, or you are unsure of its suitability, please contact the teaching staff enough in advance of the project proposal due date that our discussions with you can help shape your proposal.

There is a limited budget available to procure additional sensors if needed for your final project. Please make sure to document any additional hardware needs in your proposal and remember the budget is limited!

Students are expected to design and carry out final projects working in teams of 2-4 students. Note that we expect all students to demonstrate a roughly equal amount of work, so teams of 4 should be sure to tackle appropriately sized problems.

After the proposal, each final group will be assigned a course mentor who will be the best point of contact throughout the project.

Special Note: If you want to undertake a mostly theoretical project, work alone, or work in a group of larger than 4, please contact the course staff before beginning your proposal.

Deliverables

Proposal - 5%

To ensure that you choose an appropriate project, you are required to turn in a 1–2 page project proposal by the date indicated above. The proposal should begin with a clear, unambiguous statement of your topic, and include all of the following:

1. A brief discussion of the tool/middleware/runtime/application, its constraints, and why TinyML is a good solution/approach/how it benefits the field of TinyML.
2. A high level description of the models you intend to design and train and the data you intend to collect where applicable. Alternative, a discussion of the models/use cases you will use to test/evaluate your tool/middleware/runtime/etc.
3. What sensors you intend to use (and any additional sensors you may need at their cost)
4. A list of papers or other resources you intend to use inform your project effort. This list will form the core of your project report reference list. If your project includes anything usual (such as having significant systems demands), please state this as well.
5. Proposals should identify all members of the team and indicate how you intend to divide work on the project and collaborate given the unique demands placed on all of us due to COVID-19.

We will review your proposal and return it to you with comments and suggestions as soon as we can so that you can get started!

Check-Ins - 0%

To ensure that you are on track with the project and to identify any issues on time, we will be reaching out to check-in with every team around Thanksgiving. While this is not a graded component it is a requirement. Please make sure to have tested any hardware/sensors by this point as it may be impossible to acquire additional hardware/sensors after this meeting.

Final Report and Code - 75%

You must submit a written report on your project, structured as a standard academic conference paper, and the complete, documented source code for it. The report should be formatted using the IEEE LaTeX template - 2 columns - and 5-10 pages in length. It should describe the algorithms you implemented, the data you collected, and the testing and analysis you performed. We recommend GitHub/Overleaf as a method of collaborating and submitting source code. The report must contain all of the following content:

1. A motivation for why your chosen tool/middleware/runtime/application (and its constraints) are both important and need to be solved and ripe for a TinyML solution and/or how they are needed to advance the field of TinyML.
2. A description of related work on which your project and any algorithms and/or data sources you used are based. Include a discussion of whether you adapted a published algorithm/model or devised a new one, the range of problems and issues you addressed, and the relation of these problems and issues to the techniques and ideas covered in the course.
3. A clear specification of the data you collected, algorithm(s) you used, and model(s) you trained. A reader should be able to reconstruct and verify your work from reading your paper.
4. Analysis, evaluation, and critique of your implementation. Include a description of the testing data you used and a discussion of examples that illustrate major features of your system. Testing is a critical part of system construction, and the scope of your testing will be an important component in our evaluation. Discuss what you learned from the implementation. Make sure your analysis includes empirical results ideally presented graphically.
5. In addition, the report should include two appendices:
    a. A clear description of how to use your system and how to generate the output you discussed in the write-up.The teaching staff must be able to run your system.
    b. A list of each project participant and their contributions to the project. If this varies significantly from the project proposal, provide a brief explanation.

Your code should be clearly documented. Submit your code along with the project document to Canvas (or link directly to a public repository in your report). If you have any questions about these specifications, please ask the teaching staff.

Video/Presentation/Poster - 20%

The video/posters/presentations are a chance to explain your problem and approach, and showcase what you've accomplished to the rest of the class. We will release more information about how we are going to do this remotely as the date approaches (and once we have a better understanding of how many groups we have). In general you will be designing PowerPoint slide(s) akin to a short conference poster presentation / paper presentation that quickly summarizes the problem, describes your approach, and presents the most important parts of your analysis. We anticipate that this will occur during our final exam time slot. Please note that the report will also be due on this day so we will be expecting final results.

Project Topics

TinyML Datasets

- Public release of new datasets to TinyML
- Frameworks that automate dataset development
- Survey and analysis of existing tiny datasets that can be used for research

TinyML Applications

- Novel applications across all fields and emerging use cases
- Discussions about real-world use cases
- User behavior and system-user interaction
- Survey on practical experiences

TinyML Algorithms

- Federated learning or stream-based active learning methods
- Deep learning and traditional machine learning algorithms
- Pruning, quantization, optimization methods
- Security and privacy implications

TinyML Systems

- Profiling tools for measuring and characterizing system performance and power
- Solutions that involve hardware and software co-design
- Characterization of tiny real-world embedded systems
- In-sensor processing, design, and implementation

TinyML Software

- Interpreters and code generator frameworks for tiny systems
- Optimizations for efficient execution
- Software memory optimizations
- Neural architecture search methods

TinyML Hardware

- Power management, reliability, security, performance
- Circuit and architecture design
- Ultra-low-power memory system design
- MCU and accelerator architecture design and evaluation

TinyML Evaluation

- Measurement tools and techniques
- Benchmark creation, assessment and validation
- Evaluation and measurement of real production systems

Example Project Ideas

Some more specific thoughts:

Systems oriented:

- Profile TensorFlow Lite Micro for overheads (performance and memory)
- Profile different frameworks and report on your developer experience
- Survey the space of TinyML: what models/ops/features are supported by platforms
- Add a benchmark to tinyMLPerf (https://arxiv.org/pdf/2003.04821.pdf)
- Quantify “AI Tax in TinyML Systems” (pre- and post-processing overheads relative to model execution)
- New (micro) hardware support for TFLite Micro
- Accelerator/MCU system integration for efficient, fast, and powerful TinyML

ML oriented:

- Text to speech training of speech models for TinyML
- Sensor Fusion for better tiny models (e.g. audio and IMU)
- Combining NN with traditional ML for better TinyML
- TinyML computing ON sensors to make them smart (many have small MCUs)
- Novel model architectures (e..g, hybrid methods)
- On device learning
    - Full learning using traditional ML (k-means)
    - Personalization of an NN (pre/post processing)
- Novel compression methods for ML models / algorithms to make them tiny (e.g., how can we go beyond quantization, pruning, and encoding?)

Applications oriented:

- New applications (e.g., soil monitoring, earthquake early warning detection)
- Happy/Frown detector (extension to assignment-2)
- New kernel optimization/additions to TFLite Micro (LSTM supported yet?)
- Adding new hardware/architectural components to TinyML platform (implemented using a simulator like Renode?)