ECGR 4127/5090 Project 2: Keyword Spotting

v1.0

Due May 5, 2023

In this project you will train and deploy a keyword spotting algorithm, using the Arduino Nano 33 BLE Sense (or Pico4ML board). You will present your work during the final examination period on May 5.

Specifications

- The project is to be performed individually or in groups of two. Between groups, you may discuss the concepts, but you should not share code.
- The model should recognize at least two words, at least one of which should be a custom word that you choose (e.g. your name, your favorite color, the name of your pet). It should **not** be one of the "Google 30" words in the Speech Commands dataset. Every team must choose a unique word. Your second word can either be one of the words from the Speech Commands dataset, or another custom word of your choosing.
- To train your model, you will collect several utterances of your target word. You **are** encouraged to cooperate on collecting training data, so for example, you can record another team saying your target word, and in turn they can record you saying theirs. You can use either the embeddd board itself as a recording device, or other recording hardware, e.g. phone, laptop, etc.
- The deployed model should work in a streaming scenario. In other words, it should detect the target words in a continuos stream of speech from the microphone, rather than simply classify a discrete frame of audio.

Table 1: Grading

Criteria	**Weight**	**Notes**
Meeting Specifications	35%	Did you meet the minimum specifications outlined here?
Performance	20%	Your approach to maximizing performance.
Exploration	10%	To what extent did you attempt new methods beyond what is contained in the baseline example?
Report Clarity	20%	Explanations, clear and legible figures, grammar, formatting, etc.
Phase 1 Submission	10%	Clarity and completeness of your phase 1 turn-in.

Report Contents and Format

Document the design and testing of your amplifier in a report. You may optionally format the report in the IEEE journal style. Figures should be included in the body of the document, not printed separately and stapled to the back. Your report should thoroughly explain the process of training and deploying your model, as well as your experimental results. The items below represent key parts of the report, but ***this does not constitute an exhaustive list of what your report should explain***. The report should include the following sections:

Introduction Describe the purpose of the project, what you intended to accomplish, and inform the reader about the upcoming sections. Two to three paragraphs should be sufficient.

Model Architecture

- How did you choose the topology of your neural network? Was the size of the network constrained by number of parameters, runtime, or both?
- What input features did you choose? How did you choose them?
- What experiments did you perform to explore other architectures?
- Include any plots or explanation that might help explain the design of your model.

Data and Training

- How did you collect your data? Did you pool resources with others? What hardware did you use? Did you utilize any other sources of data?
- What measures did you take to ensure that your training data covers the natural variation in environment, voice, accent, etc?
- What augmentations did you perform, if any?
- Did you do any experiments to estimate how much data you need?
- Did you employ any specific training techniques such as distillation, transfer learning, or synthetic data generation?
- Any other plots or information you feel would be useful in explaining your data acquisition and training pipeline.

Results

- Report the accuracy of your model *in software*, i.e. as measured in TensorFlow on the training, validation, and test sets.
- Describe the performance of the model in its deployed streaming context. Does it respond to the target words. Does it exhibit false positives on other sounds? Do any specific sounds or words induce false positives more strongly than others (e.g. near-homophones). Does it fail more often on any specific group of speakers or on any particularly style of speech (e.g. a model might fail when the word is said slowly or for speakers of one gender, etc.)
- How long does the inference take to run? Relatedly, what sampling rate (in frames per second) do you use for the deployed model?
- Play a 5-10 minutes recording of a noise source at a few different volume levels and record the number of false alarms. Translate into a rate of false alarms per hour. Measure the sound level at the microphone using a phone and a dB-meter application.

Summary Table Consolidate the following key results in a summary table.

- Accuracy in TensorFlow on training, validation, and test data.
- False Rejection Rate (FRR) in TensorFlow for the word from the Speech Commands dataset.
- FRR in TensorFlow for your custom word.
- Number of Parameters, and MACs.
- Input tensor shape.
- Sampling rate (aka frame rate).
- False Alarm Rate (FA/hour).

Discussion Overall, how would you assess the performance of your deployed model? Does it seem like a usable voice control system? What improvements would help it become a more usable voice interface? (E.g. “It needs to be more responsive to quiet voices.” or “It needs fewer false alarms in response to the TV.”) If there are any odd behaviors in your model or your results, describe them and attempt to explain them. What difficulties were encountered? What would you have done differently?

Conclusion Briefly summarize your results. What did you learn?

References If you used information from any papers, applications notes, web sites, or textbooks other than the course text, cite those works here.