CSE 421 – Embedded Machine Learning

Q3: Keyword Spotting from Audio Signals

Can Sarp Aridogan
Ahmet Renda Sendag

1. Introduction

This task implements a simple keyword spotting (KWS) system using audio recordings of spoken digits. The goal is to detect whether an input recording corresponds to the digit "zero" or not, using a model small enough for microcontrollers.

2. Dataset

We use the Free Spoken Digit Dataset (FSDD). Each file name follows the pattern digit_speaker_index.wav. The data is converted into a binary classification problem: label 0 is "zero" and label 1 is "not zero" (all other digits). The speaker "yweweler" is used as the test split, while the remaining speakers are used for training.

3. Feature Extraction (MFCC)

For each audio file, MFCC features are computed with FFTSize = 1024, sampling rate = 8000 Hz, 20 Mel filters, and 13 DCT outputs. To obtain a fixed-length feature vector, the mean and standard deviation of the 13 MFCC coefficients over time are concatenated, producing 26 features per sample.

4. Model

A single-neuron model with sigmoid activation is trained (equivalent to logistic regression). Binary cross-entropy loss and the Adam optimizer are used. Class weighting is applied to emphasize the "zero" class.

5. Results

On the test set (speaker yweweler), the model achieved accuracy = 0.882. The number of false negatives (zero classified as not zero) is 33.

Confusion Matrix (rows = true label, columns = predicted label):
[[17 33]
[26 424]]

6. Embedded Implementation Plan

After training, the model weights and bias are exported into C code. On STM32, the pipeline is: compute MFCC mean and standard deviation features (26 values), compute the dot product plus bias, apply the sigmoid function, and threshold at 0.5 to output the binary decision. UART output can be used to print the probability and predicted label.

7. Conclusion

The results show that MFCC-based features combined with a very small model can successfully perform keyword spotting for "zero". This approach is suitable for embedded deployment due to its small memory footprint and low computational cost.