ML4IoT-HW01

Hojjat Miryavifard s334026
Ali Behlooei Dolatsaraei s334033
Sadjad Zamani s331775

1. Exercise 01: Timeseries Processing for Memory Optimization

How much memory (in GB) is needed to provide this monitoring service for 1000 clients?
To estimate the memory requirements for providing a monitoring service to 1000 clients, we calculate the data points, memory usage per client, and total memory step-by-step.

Data Points Calculation

The monitoring system collects data every two seconds. For raw data retained for 30 days, the total number of data points is:

$$\begin{aligned}\text{Raw Data Points} &= 30 \times 24 \times 60 \times 60 \div 2 \\ &= 1,296,000 \,\text{data points}.\end{aligned} \tag{1}$$

For aggregated data, which includes hourly min, max, and average values retained for one year:

$$\begin{aligned}\text{Aggregated Data Points} &= 365 \times 24 \\ &= 8,760 \,\text{data points}.\end{aligned} \tag{2}$$

For each measurement there is one database for raw data and 3 aggregated database:

$$\begin{aligned}\text{Data Points Per Client} &= 2 \times (1 \times 1,296,000 \\ &+3 \times 8,760) = 2,644,560 \,\text{data points}.\end{aligned} \tag{3}$$

Memory Usage Per Client

Each data point consists of a timestamp (8 bytes) and a value (8 bytes), requiring 16 bytes per data point. Without compression, the memory usage per client is:

$$\begin{aligned}\text{Raw Memory Per Client} &= 2,644,560 \times 16 \\ &= 42,312,960 \,\text{bytes} \approx 40.35 \,\text{MB}.\end{aligned} \tag{4}$$

If we suppose a compression ratio of 90%, and neglect the header size, the memory usage can be approximated by:

$$\text{Compressed Memory} = 40.35 \times 0.1 = \textbf{4.035 MB}. \tag{5}$$

Memory Usage for 1000 Clients

The total memory needed for 1000 clients is:

$$4.035 \,\text{MB} \times 1000 = 4,035 \,\text{MB} \approx \textbf{3.94 GB} \tag{6}$$

Exercise 02: Voice Activity Detection Optimization & Deployment

This exercise focuses on optimizing and deploying a Voice Activity Detection (VAD) system capable of distinguishing between silent and non-silent audio segments. The primary goal is to meet two key constraints: achieving an accuracy greater than 97.6% on the VAD dataset and ensuring a median latency of less than 25 milliseconds when deployed on the Raspberry Pi. Through systematic explo- ration of hyperparameters and careful evaluation of results, the optimal configuration for the VAD system was identified and implemented.

2.1. Report and Discuss the Search Space Used to Discover the VAD Hyperparameters

The search space for optimal VAD hyperparameters focused on ranges tested during the Labs, as we had prior knowledge of their behavior. While we briefly explored other ranges, we primarily stuck to these familiar values for efficient optimization and reliable results. The following ranges were explored:

- **Frame Length in_s**: {0.008, 0.016, 0.032}
- **Frame Step in_s**: Calculated as a percentage of the Frame Length {25%, 50%, 75% }
- **dBFS Threshold**: {10, 12, 15, 20}
- **Duration Threshold**: {0.1, 0.2, 0.3}

The search aimed to find configurations that maximize accuracy while minimizing latency, balancing frame step (calculated using overlap percentages), computational load, and sensitivity (dBFS threshold and duration threshold).

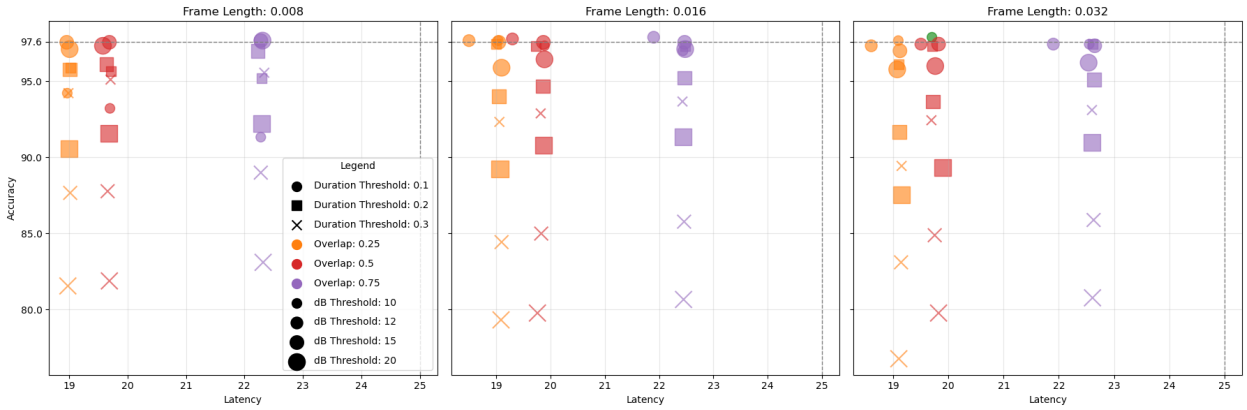

Figure 1. Accuracy vs Latency for different configurations
Optimal configuration is highlighted in green.

2.2. Table Reporting Selected VAD Hyperparameter Values

FL (s)	FS (s)	dBFS	Dur. (s)	Acc. (%)
0.032	0.016	10	0.1	**97.89**
0.032	0.016	15	0.1	97.44
0.032	0.016	10	0.3	92.44
0.032	0.024	10	0.1	97.67
0.032	0.004	10	0.1	97.00
0.016	0.004	12	0.1	**97.89**
0.016	0.004	12	0.3	90.00
0.016	0.012	12	0.1	97.67
0.016	0.008	12	0.1	97.78
0.016	0.012	10	0.1	97.67
0.016	0.004	10	0.1	97.00

Table 1. Selected VAD Hyperparameter Configurations. Configurations are grouped by Frame Length (FL).

2.3. Discussion of How Each Hyperparameter Affects Accuracy and/or Latency

- **Frame Length (‘frame_length_in_s‘)**:
  Longer frame lengths gather more data in each frame, which can improve feature extraction. However, accuracy varies with different frame lengths because performance depends on the specific settings used, showing a balance with other parameters. Also larger frame lengths may slightly increase latency because they require processing more data per frame, though this effect is usually less significant compared to other factors like frame step.
- **Frame Step (‘frame_step_in_s‘)**:
  The choice of frame step (FS) significantly influences latency, with accuracy showing comparatively less variation. As shown in Figure 1, latency varies significantly across configurations, as indicated by the spread of points and different colors. However, accuracy remains relatively stable, despite changes in frame step size and other parameters.
- **dBFS Threshold (‘dBFSthres‘)**:
  Lower thresholds enhance the system’s sensitivity to quieter sounds, resulting in improved detection rates. The best-performing configurations in our tests were observed with smaller dBFS values as it is shown in the table1 highlighting that subtle adjustments to this parameter can lead to higher accuracy while effectively modulating sensitivity. Additionally, dBFS has a negligible impact on latency, as it primarily influences the decision-making process during post-processing without significantly affecting computational load.
- **Duration Threshold (‘duration_thres‘)**:
  Short thresholds ensure that brief voice activity is detected, maximizing accuracy. Increasing this parameter may result in missed detections, lowering overall performance. As it’s evident from figure 1 the larger thresholds shown by x and square have less accuracy than the circles. This parameter has negligible impact on latency.

2.4. Elaborate on Fundamental Choices and Target Constraints

The optimal configuration highlighted by green in the figure 1 achieve (**97.89%**) accuracy with (**19.7**ms) latency. We used a duration threshold of 0.1 seconds, instead of the default 0.4 seconds in *vad_latency.py* script, to better capture sudden changes in the signal over time. Also the overlap which is calculated by the ratio of frame step over frame length in the default is 0.01/0.04 which is 25%, which increases the computational load and thus the latency while our optimal configuration has an overlap of 50%. Additionally, we lowered the db threshold to capture sounds with lower energy, which further improves accuracy.