On-Device Deep Learning for IoT-based Wireless Sensing Applications

Manoj Lenka and Ayon Chakraborty

SENSE Lab
IIT Madras
Wireless Sensing 101

- Time Delays: $t_1, t_2, t_3$
- Amplitudes: $a_1, a_2, a_3$
- CIR
- FFT
- CFR
- Frequency Sub-carriers
- CSI Spectrogram
- Multiple Packets
CSI as a Sensing Primitive

Wi-Fi Transmitter

Wi-Fi Receiver

Fall Detection

Gestures

Positioning

Seeing Through Walls
Why **Not** Edge Based Sensing?

- **Low Resources**
- **Add OPEX cost**
- **Increases latency**
- **Requires last mile connectivity**
- **Privacy Issues**

**IoT Devices**
- Sensing Data
- Data
- Reduces QoS of users

**User Devices**
- Sensing Data
- Data

**Edge/Cloud**
- Sensing Data
- Data

Icons taken from https://www.flaticon.com/
Challenges of Inferencing on IoT Devices

More Resources = More Energy

No one-size-fits-all solution
Compressing a Neural Network

Uncompressed

Prune (P)

Cluster (C)

P & C

Quantize (Q)
Related Work and Research Gaps

**Traditional Wi-Fi Sensing**
Mainly focuses on improving performance and finding new and innovative applications. Less interest in actual system implementation.

**System Consideration for IoT**
Some recent work do look into on-device Wi-Fi sensing on microcontrollers (like ESP-32) from a quantization perspective. Works like EfficientFi look into edge-based deployment.

**TinyML Related Work**
Has developed techniques like quantization, pruning, etc. Tools like TensorFlow Lite and Micro. Does not specifically focus on wireless sensing.
Design a framework that provides a best-effort compressed neural network for a Wi-Fi sensing application such that the user can tune the tradeoff between performance and cost.
Wisdom: Inputs

- $w_{acc}$: Accuracy
- $w_{inf}$: Inference Rate
- $w_{eng}$: Energy per Inference
- $w_{flh}$: Flash Consumed
- $w_{ram}$: RAM Required
- $A_{min}$
- $I_{min}$
- $E_{max}$
- $F_{max}$
- $R_{max}$

Weights $w$ -> Filters $c$
WISDOM: Outputs

Architecture Type

\[ T = \{ FCN, CNN, LSTM \} \]

Number of Parameters

\[ N = \{ 250, 1.5K, 3K, 6K, \ldots, 180K \} \]

Compression Techniques

\[ O = \{ \text{none}, \text{prune}, \text{cluster}, \text{qat}, \ldots, \text{pcptq} \} \]

Neural Network Configuration

\[ I = \{ [t, n, o] | t \in T, n \in N, o \in O \} \]

Total of 324 models

Icons taken from https://www.flaticon.com/
**WISDOM: Utility Function**

Utility: $U = P - C$

- **Accuracy:** $w_{acc}A + w_{inf}I$
- **Inference Rate:** $A \geq A_{min}$, $I \geq I_{min}$

Performance:

- **Energy per Inference:** $\epsilon$
- **RAM Required:** $\mathcal{R}$
- **Flash Consumed:** $\mathcal{F}$

Cost:

- **Energy per Inference:** $w_{eng}\epsilon$
- **RAM Required:** $w_{ram}\mathcal{R}$
- **Flash Consumed:** $w_{flh}\mathcal{F}$

where

- $\epsilon \leq E_{max}$
- $\mathcal{R} \leq R_{max}$
- $\mathcal{F} \leq F_{max}$

All metrics are normalized between 0 and 1 for a fair comparison.
Application Use Case: Human Activity Recognition

Static
- Empty
- Stand

Dynamic
- Sit
- Up/Down
- Walk
- Jump

100 Samples

Dataset is open-sourced at: https://cse.iitm.ac.in/~sense/wisdom/
WISDOM: Optimization Problem

\[ \mathcal{U}_{w,c}(\text{Wisdom}(w, c)) \approx \mathcal{U}_{w,c}(i^{OPT}) \]

\[ i^{OPT} = \arg\max_i \mathcal{U}_{w,c}(i) \quad \forall \; i \in I \]

icons taken from https://www.flaticon.com/
WISDOM: Training

\[ U_{w,c}(\text{Wisdom}(w, c)) \approx U_{w,c}(i^{\text{OPT}}) \]

Generation Script

~27K

Accuracy  Rate  Energy  RAM  Flash

Decision Tree

Measure

\[ i^{\text{OPT}} = \arg\max_{i} U_{w,c}(i) \quad \forall \; i \in I \]

Type  Size  Compression

icons taken from https://www.flaticon.com/
Testbed for Conducting Measurements

- Current Measurement
- Energy per Inference
- Inference Rate

Diagram showing the process of training, compressing, deploying, and inferring with TensorFlow Lite. Icons taken from https://www.flaticon.com/
CNNs/RNNs are more accurate but also consume more resources compared to FCNs.
RNNs are more adversely affected by compression compared to CNNs/FCNs.
Quantization reduces the RAM and energy requirement, while increasing inferencing rate.
Clustering provides significant reduction in flash, but quantization along with clustering is not reasonably effective.

Key Insights (4)
Key Insights (5)

Compressing a model with higher number of parameters yields a more accurate model than an uncompressed model with lesser number of parameters

(while having a similar footprint in terms of energy and memory)
Baseline Models and Scenarios Used for Testing

There are additional 126 different test cases with different combination of weights.

**Scenarios**

- **S1: More weight to accuracy**
  \[ w_{acc} = 0.9, \; w_{inf} = 0.1 \]
  \[ w_{flh} = w_{ram} = w_{eng} = 0.3 \]

- **S2: Equal weight to accuracy and inference rate**
  \[ w_{acc} = w_{inf} = 0.5 \]
  \[ w_{flh} = w_{ram} = w_{eng} = 0.3 \]

- **S3: More weight to flash usage**
  \[ w_{acc} = 0.7, \; w_{inf} = 0.3 \]
  \[ w_{flh} = 0.8, \; w_{ram} = w_{eng} = 0.1 \]

- **S4: More weight to RAM and energy per infer**
  \[ w_{acc} = 0.7, \; w_{inf} = 0.3 \]
  \[ w_{flh} = 0.1, \; w_{ram} = w_{eng} = 0.45 \]

**Type**
- RNN, CNN, FCN

**Size**
- 1.5K, 6K, 24K

**Compression**
- None (NQ), Quantized (Q)

18 baseline models (naive choice)

Icons taken from https://www.flaticon.com/
Results: Models Chosen by WISDOM have Higher Utility

Relative utility of NQ and Q models are lower than WISDOM chosen models. Relative utility is w.r.t to the optimal model i.e., $\frac{U(i)}{U(i^{OPT})}$
Results: Models Chosen by WISDOM have Higher Utility

The CDF of utility difference between Q or NQ model and WISDOM chosen model for all 126 test cases is always positive and starts increasing after 0.5.
Results: Models Chosen by \textit{WISDOM} Uses Less Resources While Maintaining High Accuracy

\textit{WISDOM} chosen models show a percentage decrease similar to Q models for resource consumption, but still maintains higher accuracy of \(\sim 15\%\) compared to Q models. The percentage decrease is w.r.t NQ models.
WISDOM chosen model outperforms the best quantized model 83% of time, and the best non-compressed model 99% of time
Thank You, Questions?

Manoj Lenka, lenka98.github.io and Ayon Chakraborty, cse.iitm.ac.in/~ayon

Contact: cs22s008@cse.iitm.ac.in, ayon@cse.iitm.ac.in

Artifacts available at: https://cse.iitm.ac.in/~sense/wisdom/