Demo: Edge Federated Learning over a LoRa Mesh Network

Nil Llisterri Giménez*, Felix Freitag*, Leandro Navarro*, Mennan Selimi§
* Computer Architecture Department, UPC BarcelonaTech. Spain
§ Max van der Stoel Institute, South East European University. North Macedonia

Abstract—**Training machine learning models in resource-constrained edge devices using federated learning is not yet deeply explored by the research community. There are important technical challenges, such as the limited computing capacity of microcontrollers, and the restricted network connectivity and bandwidth of IoT environments to coordinate and execute the federated learning process. In this paper, we demonstrate a prototype implementation of a federated learning model trained over a LoRa mesh network. We leverage the LoRaMesher library to provide the LoRa mesh network service and integrate it into a keyword spotting federated learning application that runs on a microcontroller-based IoT board. The experimentation with the prototype demonstrates the recent vision of providing edge intelligence with low-capacity networked adaptive microcontroller-based nodes.**

Index Terms—**TinyML, machine learning, IoT.**

I. Introduction

Embedded machine learning is a recent technology that allows machine learning models to run on low-cost, low-power microcontrollers [1]. There are two main approaches to equipping a microcontroller with a trained neural network: off-device training and on-device training of the model. In on-device training, which we focus on in this paper, the machine learning model is trained on the microcontroller. The advantage is the adaptability of the neural network even after being deployed on the microcontroller. Still, but the training process has to be tailored to the resource-constrained environment of the microcontroller [2].

Federated learning has become a popular collaborative training approach for machine learning models. While the research community proposed federated learning for higher-end edge devices, the work on federated learning in networked embedded IoT nodes is still very few [3]. Challenges include the limitations of the IoT network, for instance, when remote nodes to be trained communicate over restricted Low Power, Wide Area Network (LPWAN) radio technology such as LoRa.

LoRa is a popular communication technology for IoT applications. The dominating network architecture used is LoRaWAN, which today has also a limited option for multi-hop [4]. While LoRaWAN is often a suitable option for applications that send sensor values to a nearby gateway node, it is not designed to support the networking needs of federated learning. LoRaWAN lacks node-to-node communication, and the downlink capacity from the gateway to a node, while existing, is not prepared for a symmetric communication of uplink and downlink. LoRa itself, however, can be used for point-to-point data communication. LoRa mesh networks such as proposed in [5] can offer a potential substrate to provide the network for federated learning.

In this demo paper, we combine the capacity of a LoRa mesh network with federated learning to coordinate and interchange machine learning models. We demonstrate a prototype implementation of a federated learning model training over a LoRa mesh network. For this, we leverage the LoRaMesher library, which we have developed, to provide the LoRa mesh network service, and integrate the library into a keyword spotting federated learning application that runs on real microcontroller-based IoT boards.

II. Implementation

For this demo, we use the LoRaMesher library[1] developed in [6]. LoRaMesher enables routed packet delivery to any node in the LoRa mesh network by applying a proactive distance-vector protocol. LoRaMesher is implemented with FreeRTOS and compiles for ESP32-based microcontroller boards with LoRa radio.

Each node containing a networked machine learning federated learning capacity is composed of two microcontroller boards, namely the Arduino Portenta H7 and the TTGO-LoRa32, connected via UART [7].

The Arduino Portenta H7 board is dedicated to tasks related to machine learning. Hence the code flashed to this board integrates the training of the neural network and orchestrates the communications with the other nodes in the LoRa mesh network to perform federated learning.

The TTGO-LoRa32 board is dedicated to the networking task and is in charge of sending the messages received from the application on the Portenta via UART to the other nodes. The LoRa32 uses the LoRaMesher that enables to discover, send, and receive messages from the other nodes. LoRaMesher is integrated as a library into the code for the microcontroller to use its network service as shown in the publicly available git repository[2].

Figure 1 illustrates the LoRa mesh network formed by networking nodes and a few application nodes containing federated learning capacity. When a federated learning round is triggered by the application on the Portenta board, it will query all the nodes of the federated learning network for their metrics.

[1]https://github.com/LoRaMesher/LoRaMesher
[2]https://github.com/NilLlisterri/TTGO-LoRaMesher

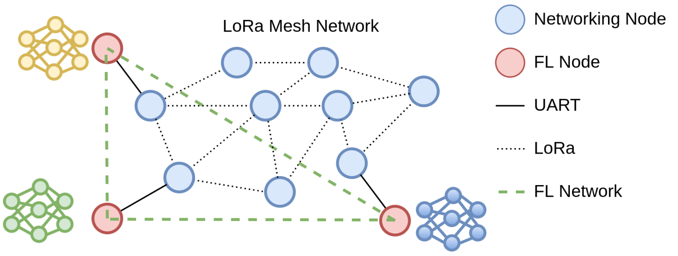

Figure 1: Federated learning nodes interconnected by LoRa mesh network.

Then, the node compares the metrics and selects the best node to perform federated learning with. When the best node is selected, it requests all of the weights of the neural network in batches small enough to be sent via LoRaMesher messages. When each batch is received, the weights are expanded and used to update the local weights via weighted averaging, taking into account the number of local and remote samples each model has been trained on.

III. Demonstration

The application we use is a Keyword Spotting Application (KWS), where the nodes are trained to identify four different keywords. A fully-connected neural network with three layers and a hidden-layer size of 20 neurons is integrated into the application. The network is trained by federated learning.

Three nodes are used for this demonstration (Figure 2).

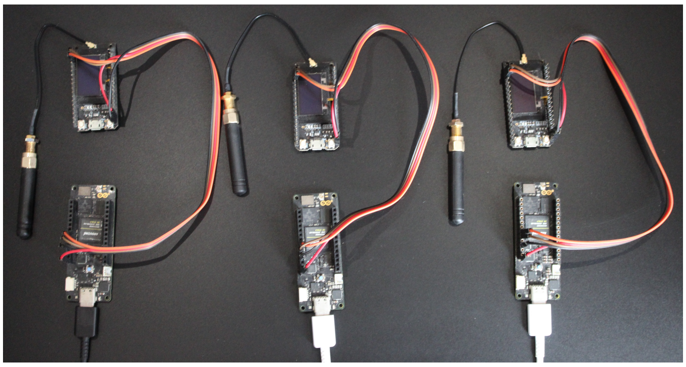

Figure 2: Demonstration with three nodes, where each node is formed by an Arduino Portenta H7 connected to a TTGO-LoRa32.

To be able to control the experiment, all the Arduino Portenta H7 boards are connected to a host computer. Via this serial communication, the demonstration can be orchestrated, sending samples and triggering federated learning rounds as needed.

In the demo, each node is trained with a total of 160 samples, 40 of each keyword. After every 40 samples, a federated learning round is triggered on a node, and the accuracy is tested. Figure 3 shows a snapshot of a terminal output received through the serial port during the federated learning process.

Besides showing the experimentation with the federated learning prototype, the demonstration also provides the opportunity to review with the audience the configuration details

```
[com4] Sending train batch 1...
[com6] Sending train batch 1...
[com11] Sending train batch 1...
[SERVER] Triggering FL round on device com4
[com4] Requesting weights batch 1/6
[com4] Requesting weights batch 2/6
[com4] Requesting weights batch 3/6
      •••
[SERVER] Federated learning round completed
[com4] Sending test samples...
[com6] Sending test samples...
[com11] Sending test samples...
[com4] Sending train batch 2...
[com6] Sending train batch 2...
[com11] Sending train batch 2...
[SERVER] Triggering FL round on device com4
[com4] Requesting weights batch 1/6
[com4] Requesting weights batch 2/6
[com4] Requesting weights batch 3/6
```

Figure 3: Extract of the console output of an experiment with 3 nodes.

of LoRaMesher, and how LoRaMesher was integrated as a network service into the distributed application. Furthermore, concerning edge federated learning, the demonstration offers the opportunity to discuss with the audience ongoing directions such as quantization to reduce the bandwidth needs of federated learning [8].

The code and dataset used in this demo, and LoRaMesher as a tool to enable LoRa node-to-node communication, are available for the community in open git repositories.

Acknowledgment

The work is part of the project PDC2023-145809-I00, funded by MICIU/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR, and the project PID2023-146066OB-I00 funded by MICIU/AEI/10.13039/501100011033/FEDER UE.

References

[1] Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki, and A. S. Hafid, “A comprehensive survey on tinyml,” *IEEE Access*, vol. 11, pp. 96 892–96 922, 2023.

[2] V. Rajapakse, I. Karunanayake, and N. Ahmed, “Intelligence at the extreme edge: A survey on reformable tinyml,” *ACM Comput. Surv.*, vol. 55, no. 13s, jul 2023. [Online]. Available: https://doi.org/10.1145/3583683

[3] H. G. Abreha, M. Hayajneh, and M. A. Serhani, “Federated learning in edge computing: A systematic survey,” *Sensors*, vol. 22, no. 2, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/2/450

[4] A. W.-L. Wong, S. L. Goh, M. K. Hasan, and S. Fattah, “Multi-hop and mesh for lora networks: Recent advancements, issues, and recommended applications,” *ACM Comput. Surv.*, vol. 56, no. 6, jan 2024. [Online]. Available: https://doi.org/10.1145/3638241

[5] R. P. Centelles, R. Meseguer, F. Freitag, R. B. Viñas, and L. Navarro, “A minimalistic distance-vector routing protocol for lora mesh networks,” *IEEE Access*, pp. 1–1, 2024.

[6] J. M. Solé, R. P. Centelles, F. Freitag, and R. Meseguer, “Implementation of a lora mesh library,” *IEEE Access*, vol. 10, pp. 113 158–113 171, 2022.

[7] N. Llisterri Giménez, J. Miquel Solé, and F. Freitag, “Embedded federated learning over a lora mesh network,” *Pervasive and Mobile Computing*, vol. 93, p. 101819, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574119223000779

[8] N. L. Giménez, J. Lee, F. Freitag, and H. Vandierendonck, “The effects of weight quantization on online federated learning for the iot: A case study,” *IEEE Access*, vol. 12, pp. 5490–5502, 2024.