**Titre :** On Device Learning for AIoT

**Supervisors** :

Rémi Nahon (remi.nahon@telecom-paris.fr)

Van-Tam Nguyen (van-tam.nguyen@telecom-paris.fr)

**Summary:**

On-device learning allows the AI model to adapt to new data collected from IoT devices by fine-tuning a pre-trained model. The main objective of this master II internship is to optimize an algorithm-system co-design framework to improve on-device training with limited memory and computing capabilities.

**Description:**

On-device learning adapts pre-trained AI model to newly collected data after deployment. By training and adapting locally at the edge, the AI model can learn to improve its predictions and achieve continuous learning and user personalization. For example, adjusting a language model enables continuous learning from users' typing and writing; adapting a vision model enables recognition of new objects from a mobile camera. By bringing the training closer to the sensors, it allows, among other things, to protect the privacy of users when processing sensitive data (*e.g.,* healthcare).

However, on-device learning on tiny IoT devices is extremely difficult and fundamentally different from training in the cloud. Small IoT devices (*e.g.,* microcontrollers) typically have a limited SRAM size, on the order of 256 KB. Such a small memory budget is barely sufficient for deep learning model inference, let alone training, which requires additional computation for backward and additional memory for intermediate activation. On the other hand, modern deep learning frameworks (*e.g.,* PyTorch, TensorFlow) are typically designed for cloud servers and require a large memory footprint (>300 MB) even for small model training.

This huge gap (>1000×) makes it impossible to run on tiny IoT devices with current frameworks and algorithms. Current deep learning systems like PyTorch, TensorFlow, etc. do not take into account the limited resources of edge devices. Edge deep learning inference frameworks such as TVM, TF-Lite, NCNN, etc. offer reduced execution time, but do not support backpropagation. Although there are efficient and low-cost transfer learning algorithms, such as train to classifier final layer, bias update only, etc., the accuracy drop is significant and existing training systems cannot turn theoretical savings into measured savings. Furthermore, devices such as microcontrollers lack the operational system and runtime support needed by existing training frameworks. Therefore, we need to co-design the algorithm and system to enable tiny learning on the device.

This internship aims to bridge this gap and enable tiny on-device learning through algorithm-system co-design. The main objective is to optimize an algorithm-system co-design framework recently proposed at MIT to improve on-device learning with with limited memory and computing capabilities.