Proximity – A TinyML Model Transferring Protocol for Dynamic Environments

Dissertation

Visal Gimhan Mediwake Rajapakse

w1742117 / 2018418

Supervisor: Guhanathan Poravi

Date: May 2022

Department: Computer Science

Submitted in partial fulfilment of the requirements for the

BSc (Hons) Computer Science degree at the University of Westminster

Abstract

The swift miniaturization of Machine Learning (ML) for IoT with limited resources has opened gateways to provide intelligence at the very edge (extreme edge), i.e., closer to the sensors and actuators. Coined as Tiny Machine Learning (TinyML), this research paradigm proposes to bring ML and Deep Learning (DL) to extremely energy-efficient Microcontroller Units (MCUs). With concurrent research moving towards updatability/reformability, it is evident, with existing research, that communication will play an integral role in updating procedures, especially in distributed TinyML settings. Although contemporary research in communication-based TinyML updates exists, many solutions disregard the dynamicity in an environment when considering transferring tasks.

"Opportunism" is the exploitation of arising opportunities without considering any principles. Provided the definition, dubbed Proximity, the focus of this project is to research and develop a lightweight yet novel opportunistic protocol and decision support system that will intelligently transfer TinyML models from MCU to other devices. The transfer process considers the dynamicity of the environment an MCU can exist in to reduce failed transfers, as a result, reducing the energy wastage in such resource-constrained devices.

After extensive physical world testing done in two separate environments based on the available metrics and an evaluation process conducted afterwards confirms the lightweight nature and resilience of the produced protocol for achieving its ultimate goal of transferring TinyML models, all whilst being generalizable.

Keywords— TinyML, Microcontrollers, Internet of Things, Updatability

Subject descriptors:

- **Computer systems organization** → Embedded and cyber-physical systems
- **Computing Methodologies** → Artificial Intelligence
- **Network** → Network architectures, Network Algorithms