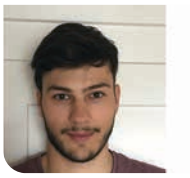
TinyML – Maschinelles Lernen für Ressourcen-beschränkte Eingebettete Systeme

Diplomanden



Jonas Högger

Joël Leirer

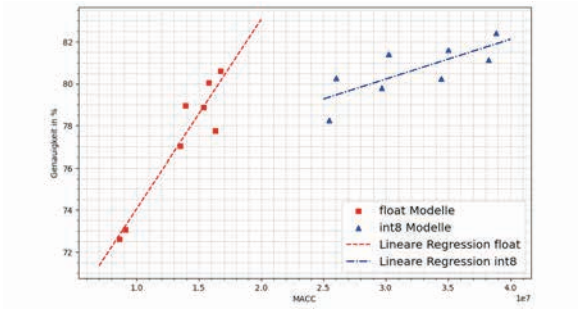
Einleitung: Mit der stetigen Nachfrage nach neuer Technologie gewinnt künstliche Intelligenz (KI) zunehmend an Bedeutung in unserem Alltag. Anwendungen wie Spracherkennung, personalisierte Werbung und medizinische Diagnosen basieren auf umfangreichen Datensätzen und leistungsfähiger Hardware. Ein besonders wachsender Bereich ist der Einsatz von Machine Learning (ML) in eingebetteten Systemen, wie etwa zur Schlüsselworterkennung oder in Hörgeräten. Diese speziellen Anwendungen, auch als TinyML bezeichnet, erfordern angepasste Modelle und Hardwarelösungen, um effizient auf den begrenzten Ressourcen solcher Systeme zu funktionieren. Ziel dieser Arbeit ist es, die Implementierung von Deep Neural Networks (DNN) auf Mikrocontrollern (MCU) der STM32-Plattform zu untersuchen und Benchmarks anhand folgender Metriken zu erstellen: Komplexität (MACC), Berechnungszeit, Speicherbedarf, Genauigkeit und Stromverbrauch.

Vorgehen: Die Arbeit beginnt mit einer umfassenden Literaturrecherche, die die relevanten Grundlagen und Theorien zu ML und DNNs erfasst. Anschliessend werden Python-Skripte entwickelt, um Modelle für die MNIST- und CIFAR10-Datensätze zu trainieren, wobei sowohl Fully Connected Neural Networks (FCNN) als auch Convolutional Neural Networks (CNN) verwendet werden. Diese Modelle werden mithilfe von STM32Cube.AI auf ein STM32 NUCLEO-G474RE-Board geladen. Auf diesem Board werden Benchmarks durchgeführt, u. a. mittels Zusatzhardware wie dem STM32 Power Shield, um die genannten Metriken zu analysieren. Parallel dazu werden Benchmarks für ein ASIC-Projekt am IMES erstellt. Ein Demonstrator bestehend aus LCD-Display und MCU-Kameramodul veranschaulicht die Klassifizierung der handgeschriebenen Ziffern.

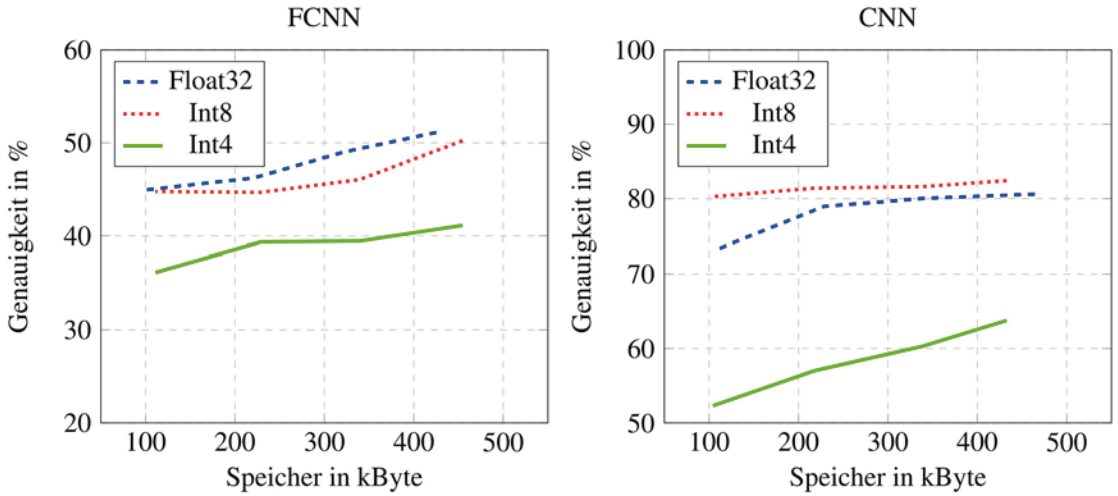
Ergebnis: Die Benchmarks der trainierten Modelle konnten auf der MCU bei unterschiedlicher Quantisierung durchgeführt und der Demonstrator konnte erfolgreich implementiert werden. Wie erwartet erkennt der Demonstrator Ziffern mit einem CNN besser als mit einem FCNN. Die Ergebnisse der Benchmarks zeigten eine positive Korrelation zwischen Speicherbedarf und Genauigkeit. Ein ähnlicher Zusammenhang konnte zwischen MACC und Genauigkeit festgestellt werden.

Der Demonstrator mit implementiertem CNN erkennt die Ziffern des MNIST-Datensatzes.
Eigene Darstellung

Positive Korrelation von Komplexität (MACC) vs. Genauigkeit für je 8 CNN Modelle (Anmerkung: int8 Modelle komplexer)
Eigene Darstellung



Speicher vs. Genauigkeit in selbst trainierten FCNNs und CNNs bei unterschiedlicher Quantisierung, Node- und Layerzahl
Eigene Darstellung



Referent
Prof. Dr. Andreas Breitenmoser

Korreferent
Theo Scheidegger, Swens GmbH, Schänis, SG

Themengebiet
Embedded Systems