Technische Hochschule
Brandenburg
University of
Applied Sciences
Fachbereich
Informatik und Medien

Benchmarking Post-Training Quantization for Optimizing Machine Learning Inference on compute-limited Edge Devices

Mahmoud Abdelrahman

Bachelorarbeit • Studiengang Informatik • Fachbereich Informatik und Medien • 02.03.2021

Hintergrund und Aufgabenstellung

In den letzten Jahren hat die Edge-KI, d.h. die Übertragung der Intelligenz von der Cloud in Edge-Geräte wie Smartphones und eingebetteten Geräten, an großer Bedeutung gewonnen bzw. gewinnt immer mehr an Bedeutung. Dementsprechend werden optimierte Modelle Maschninellen Lernes (ML) benötigt, welche mit Edge-Computing-Geräten kompatibel sind. Die Quantisierung ist eine der essenziellsten Techniken zur Optimierung von ML-Modellen. Es reduziert die Präzision der Zahlen, die zur Darstellung der Parameter eines Modells verwendet werden.
Das Ziel dieser Arbeit ist es, Post-Training-Quantisierung (Quantisierung nach dem Modelltraining) auf vortrainierte TensorFlow Machine-Learning-Modelle anzuwenden. Die Latenz- und Genauigkeitsergebnisse wurden nach der Ausführung der Inferenz anhand der quantisierten Modelle mit den Ergebnissen der Inferenz des ursprünglichen Modells verglichen.

Konzept und Implementierung

Quantisierung beim Deep Learning bezieht sich auf die Konvertierung der Gewichte und Aktivierungen eines neuronalen Netzwerks vom Fließkommaformat in ein Ganzzahlformat. Dabei wird die Anzahl der Bits verringert, die zur Darstellung der in einem DNN-Modell enthaltenen Informationen benötigt werden.
Das ursprüngliche TensorFlow-Modell wird mit dem TensorFlow-Lite-Konverter in ein TensorFlow-Lite-Modell umgewandelt. Während der Konvertierung kann eine Optimierung durch Quantisierung angewendet werden.

Nachdem das Modell konvertiert ist, interpretiert der TFLite-Interpreter das Modell in ein Format, das für den Prozessortyp (GPU, CPU, usw.) geeignet ist. Während der Arbeit wurde eine Post-Training-Quantisierung auf ein Bildklassifikations- und ein Semantische Segmentierung TensorFlow-Modell angewendet. Diese Modelle wurden auf den MNIST- bzw. Cityscapes-Datensätzen trainiert. Für die Implementierung wurden zwei Python-Skripte geschrieben. Eines davon enthält den Code, der für die Verwendung des TFLite-Konverters erforderlich ist, um die vorab trainierten Modelle in die unquantisierten (float32) und quantisierten TFLite-Modelle zu konvertieren und dann das resultierende Modell auf der Festplatte zu speichern.
Das andere Skript führt die Inferenz auf diesen Modellen aus und speichert die Auswertungsergebnisse und die Bildausgabe auf der Festplatte.

Die Experimente wurden auf einem Laptop mit Intel Prozessor bzw. auf einem Raspberry Pi 4 Modell B mit ARM Prozessor (benutzt im Bauen von IoT-Geräten) ausgeführt.

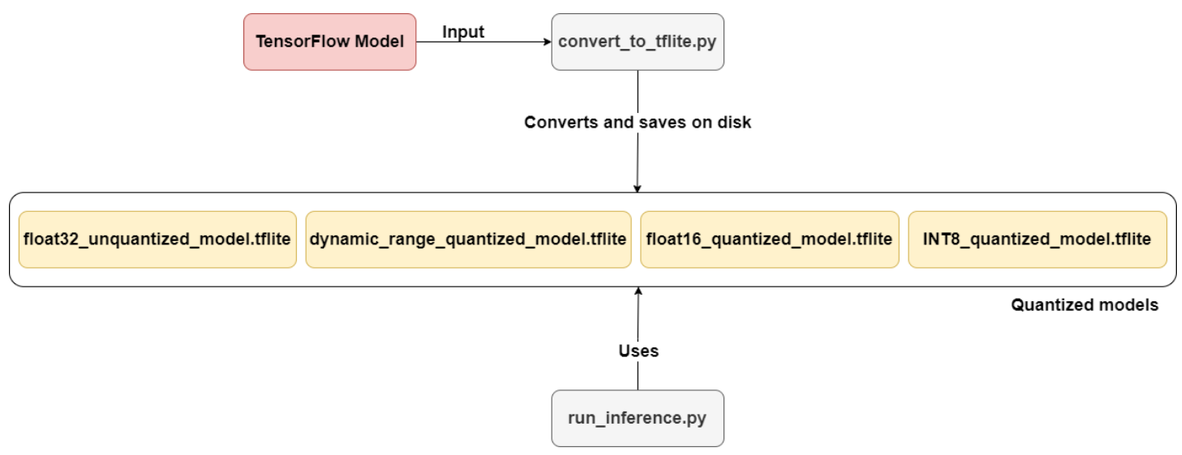

Abb. 1: Systementwurf für Modellkonvertierung und Inferenz Ausführung

Ergebnisse

Sowohl für Bildklassifizierungs- als auch für semantische Segmentierungsmodelle zeigten die Ergebnisse eine erwartete Verringerung der Modellgröße, wenn verschiedene Quantisierungstechniken angewendet wurden. Genauigkeit hat sich im Bezug auf Originalmodell in beiden Fällen im Wesentlichem kaum verändert. In einigen Fällen führte die Anwendung der Quantisierung sogar zu einer Verbesserung der Genauigkeit. Dabei hat sich die Inferenzgeschwindigkeit bezüglich des Bildklassifizierungsmodells adäquat verbessert. In manchen Fällen schien dies aber bezüglich des semantischen Segmentierungsmodelles doch nicht der Fall zu sein. In einigen Fällen erhöhte sich die Inferenzgeschwindigkeit auf dem Raspberry Pi sogar um den Faktor 10.

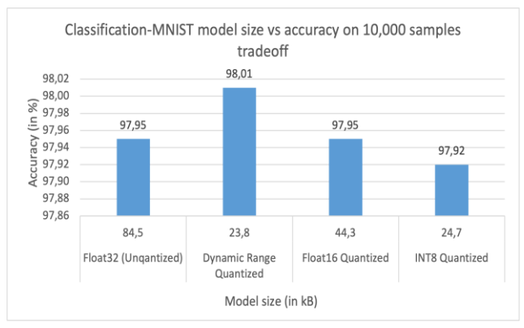

Abb. 2: INT8 MNIST-Klassifikation quantisiertes Modell mit ungefähr ¼ der originalen Modelgröße verwirklicht eine Genauigkeit, die nur 0,09 % vom Originalmodell abweicht.

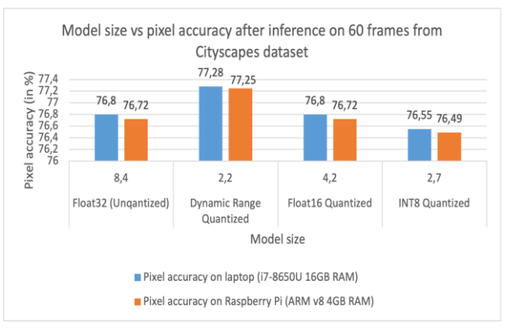

Abb. 3: INT8 Cityscapes-semantische Segmentierung quantisiertes Modell mit ungefähr 30% der originalen Modelgröße verwirklicht eine Genauigkeit, die 0,25 % vom Originalmodell abweicht.

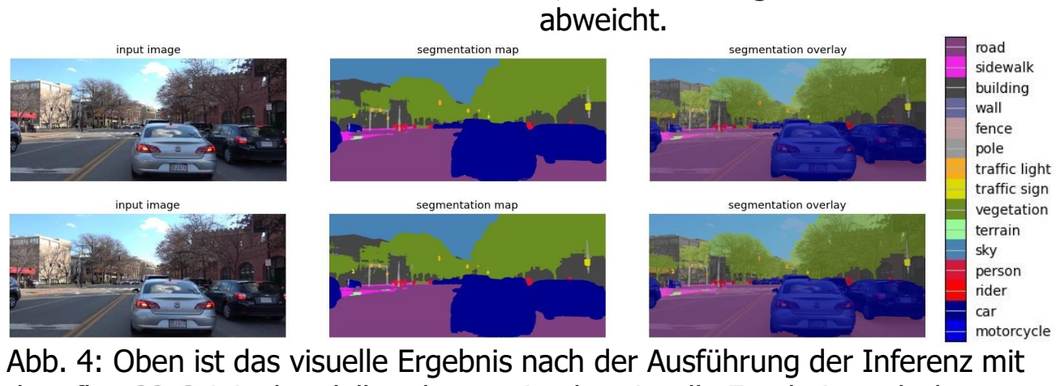

Abb. 4: Oben ist das visuelle Ergebnis nach der Ausführung der Inferenz mit dem float32-Originalmodell und unten ist das visuelle Ergebnis nach der Ausführung der Inferenz mit dem INT8-quantisierten Modell, das eine um 0,25% geringere Pixelgenauigkeit als das Originalmodell aufweist. Diese Abbildungen zeigen, dass dieser winzige Unterschied in der Genauigkeit die visuellen Ergebnisse nicht wirklich beeinflusst.

Quellen

F. Chollet. Deep Learning with Python. 1st. USA: Manning Publications Co., 2017.isbn:1617294438
Jacob *et al*. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. 2017.

Betreuung Prof. Jochen Heinsohn und Dipl-Inform. Ingo Boersch • Technische Hochschule Brandenburg • Abhishek Saurabh (MSc) • Volkswagen Car.Software Organization