Masters Project on tinyML Non-Linear Quantization

Emil Njor

July 8, 2022

Supervisor: Xenofon Fafoutis <xefa@dtu.dk>

Co-Supervisor: Emil Njor <emjn@dtu.dk>

Background: tinyML is a promising research area concerned with running machine learning models on ultra-low power devices, typically in the range of milliwatts or below. The state of the art of tinyML involves training machine learning models on larger computers (e.g., a laptop computer), and subsequently applying optimizations and deploying the models on ultra-low power devices. tinyML has interesting use cases in rural areas where a working network connection cannot always be guaranteed or where a network is undesirable due to latency or privacy concerns.

Arguably the most important optimization done to run neural networks in ultra-low power devices is quantization. In typical neural networks we store parameters as 32-bit floating point values. Research has shown that these can be reduced to at least 8-bit integers without incurring a significant accuracy loss.

Project Description: Current state of the art tinyML libraries for neural networks computes quantization from floating point values to integer values using the following equation [1]:

$$x_{int} = \text{clamp}(\lfloor \frac{x}{s} \rceil + z; 0; 2^b - 1) \tag{1}$$

Where x_int is the result of the quantization. x is the floating point input, s is a scaling factor and z is the value of the zero point in the quantized representation. $\lfloor * \rceil$ rounds to the nearest integer and the definition of clamp is:

$$\text{clamp}(x; a, c) = \begin{cases} a, & x < a, \\ x, & a \leq x \leq c, \\ c, & x > c. \end{cases} \tag{2}$$

This method of quantization maps an equal amount of possible floating point values onto every integer value. However, neural networks center most actual

floating point values around the zero point. This means that many actual floating point values will be mapped onto few integer values. The idea in this project is to propose a quantization algorithm that splits actual floating point values equally between the quantized integer values.

Recommended Background Knowledge: Embedded Systems, Machine Learning, C++ & Python.

References

[1] Markus Nagel et al. "A white paper on neural network quantization". In: *arXiv preprint arXiv:2106.08295* (2021).