Research Project Proposal: Input-Aware Dynamic Quantization in Deep Neural Networks

Mehmet Emre Akbulut, mehmetemre.akbulut@mail.polimi.it

1. Introduction to the problem

Tiny Machine Learning (TinyML) is a subset of Machine Learning that serves as a link between the ML domain and the embedded system ecosystem. It enables us to employ advanced machine learning capabilities on edge devices with limited memory, computation power, and energy consumption, while also offering reduced latency and better privacy through on-device data processing.

Besides all the advantages mentioned above, many challenges arise when deploying machine learning models into resource-constrained devices. Most of the obstacles originate from heterogeneity and resource constraints of the edge devices with lower memory capacity, compute power, and energy consumption [1],[12]. In the field of TinyML, a variety of approaches and methods are used to overcome these issues. One of the main techniques used to compress and deploy these models on devices with limited resources is low-precision quantization. Quantization algorithms are mainly grounded on the idea of reducing the precision of weights, biases, and activations. In other words, a 32-bit full precision model is compressed to a low-bit representation by employing bit widths from 8-bit to 1-bit.

Quantization of machine learning models leads to reduced memory consumption and computation without changing the current architecture. Prior works show that different quantization algorithms enable us to deploy models that have significantly lower memory consumption and fewer arithmetic operations with little loss in task performance. Quantization also increases the inference speed since it enables faster execution through simpler operations instead of full-precision floating-point operations.

Various quantization techniques are available in the literature so far. Earlier works generally focused on the weight quantization methods [5], [17], [7]. Later activation bit precisions are taken into consideration [11],[16], [4]. One of the important techniques is the **mixed-precision quantization of neural networks**. Deep neural networks consist of different layers such as convolutional layers and fully connected layers which also vary among themselves. Mixed-precision quantization offers flexibility to select different precisions for different layers with the support of specific hardware [10], [14], [3], [13]. However, in the mixed-precision approaches, current solutions in the literature use pre-defined, fixed bit widths for each layer, that can not be modified without retraining the model. **The thesis aims to explore the solutions to reduce memory and computation overhead at run-time by employing a dynamic quantization approach while focusing on choosing efficient bit precisions for different layers considering the given input**. Eventually, an instance-aware dynamic quantization framework that reduces memory consumption and computation overhead while ensuring minimal accuracy loss. In other words, from a set of candidates varying from 1 to k bit width ($b^1, b^2, ..., b^k$), for each layer L_n, efficient bit precision is chosen depending on the given input to the model with n layers.

In most existing quantization methods, the bit width for each layer stays the same for all samples in a dataset. Even though a mixed precision approach is employed, bit precisions of the layers stay the same for all the input at runtime when doing inference. But natural images vary a lot in their content, so using the same quantization settings for every sample can be sub-optimal in many scenarios [9]. Especially for tasks such as image classification, this approach leads to better performance and efficient use of resources because more easily classifiable images with relatively low details need lower computational effort. Naturally, the model can use lower bit precision for these types of images, while using higher bit precision for complex images that need more computation effort to be classified.

2. Main related works

Even though lots of work is done in quantization, it is more limited for the research in dynamic quantization based on the input data.

In the dynamic network quantization literature, one of the very first works was published in 2018 [15]. In this paper, a Dynamic Network Quantization framework utilizes bit widths of layers via bidirectional LSTM which exploits the reinforcement learning. The framework was based on weight quantization and uses RL which leads to high computation overhead. Also, layer bit widths are fixed during inference.

Furthermore, AdaBits were proposed to allow dynamically adjust bit precision of the model during inference [6]. The main drawback of this paper is that all layers share the same bit precision so the framework cannot exploit the advantages of mixed precision.

In [2], a bit predictor, Bit-Mixer, which focuses on choosing bit precisions for each layer on inference time is proposed. Bit-Mixer optimizes the bit precisions considering the resource availability and performance instead of focusing on the input.

One of the earliest research focused on the input is D^2NN [8] which executes a subset of the neurons depending on the given input.

An Instance-Aware DQNet framework , which consists of a predictor bit controller network trained together with the model itself, is proposed in [9]. This framework is the most similar work to our thesis so we will focus on the possible directions considering the results of this paper.

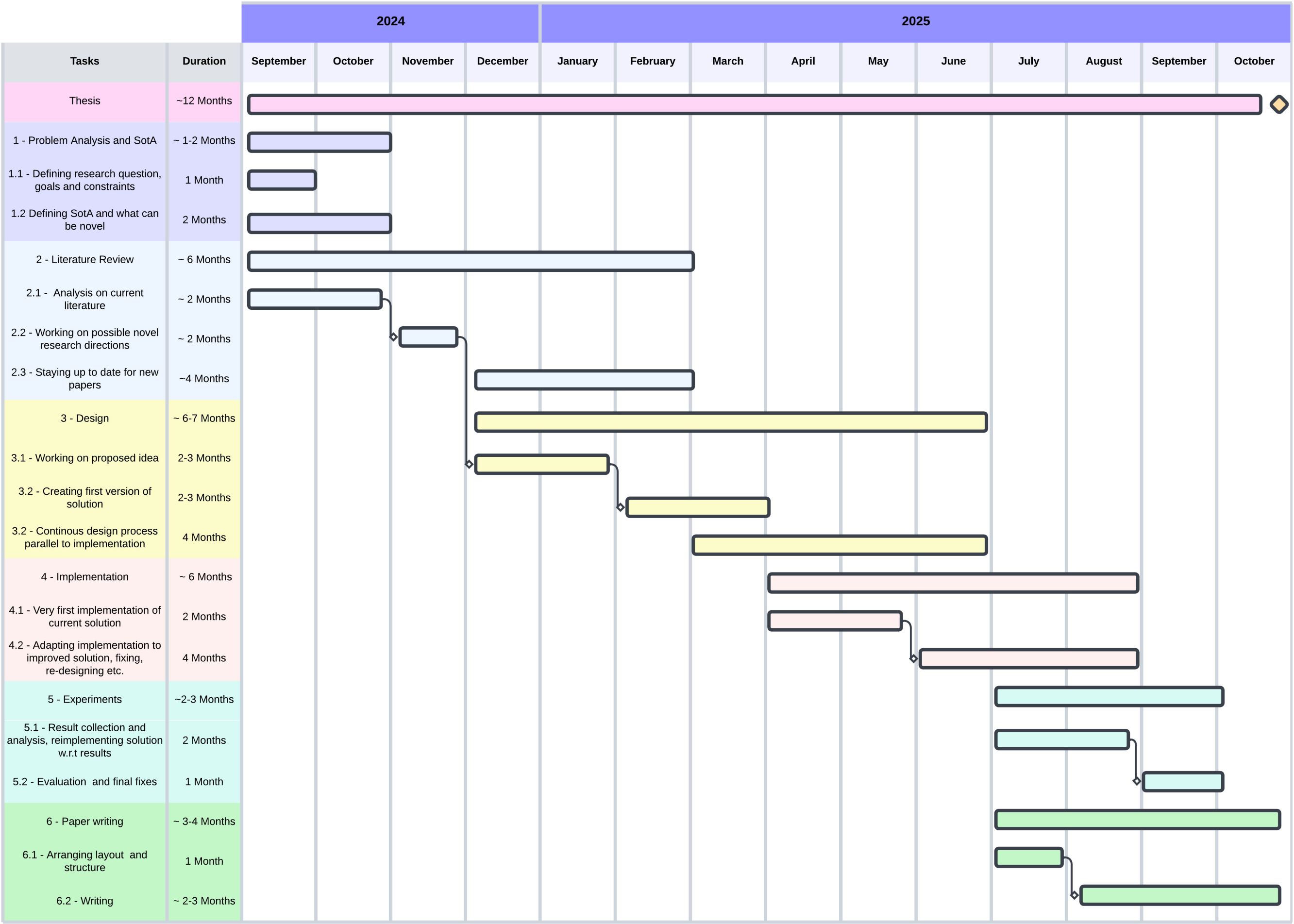
3. Research plan

The primary goal of the research is to develop and evaluate a novel dynamic quantization framework for resource-constrained TinyML applications. Mainly, we aim to answer the research question *"How can we design and implement an instance-aware dynamic quantization framework that adapts bit precision considering given input for devices with limited memory and computation power while maintaining model accuracy?"*. In short, the goal of the research is to design, implement, and deploy a framework that enables dynamic quantization concerning given input in edge devices, by exploring the current solutions in literature and improving them.

The nature of this research mainly lies between theory and application. To create a novel approach we need to take current literature further and explore possible ways to apply instance aware bit precision selection process in dynamic settings. To achieve this many architectures and algorithms will be discussed in detail. Furthermore, the research focuses on deploying this framework on a real edge device with limited resource availability, so at this step, the applicability of the developed research will be considered.

Currently, we are at the problem analysis and formulation phase of the research plan. Defining the problem to be worked on and grounding the research plan are the two possible outcomes of this step. Secondly, the following task is to analyze the current state-of-the-art solutions in detail for the instance-aware dynamic quantization domain. Our main goal after a detailed literature review is to explore possible improvements on top of the research available. After all these steps together, a more solid research direction will be determined.

After these two steps, the implementation phase will be started by choosing the proper neural networks that will be worked on such as ResNet, MobileNet, etc. Afterward, our theoretical assumptions and conclusions around input-based dynamic quantization will be shaped. A set of different solutions and versions are aimed to start experiments after the implementation and development of these candidate solutions. At this step, setting the experiment setup is so crucial to avoid some biases at the evaluation phase. Tasks such as training the neural network, implementing quantization schemes, and collecting metrics/results need to be done in an attentive and precise way. Parallel to the results and evaluation, meanwhile thesis writing should be started to create a detailed and meticulous final submission document and consider possible applications to journals and conferences. The evaluation methods are explained below in the GANTT chart.



Tasks	Duration
Thesis	~12 Months
1 - Problem Analysis and SotA	~ 1-2 Months
1.1 - Defining research question, goals and constraints	1 Month
1.2 Defining SotA and what can be novel	2 Months
2 - Literature Review	~ 6 Months
2.1 - Analysis on current literature	~ 2 Months
2.2 - Working on possible novel research directions	~ 2 Months
2.3 - Staying up to date for new papers	~4 Months
3 - Design	~ 6-7 Months
3.1 - Working on proposed idea	2-3 Months
3.2 - Creating first version of solution	2-3 Months
3.2 - Continous design process parallel to implementation	4 Months
4 - Implementation	~ 6 Months
4.1 - Very first implementation of current solution	2 Months
4.2 - Adapting implementation to improved solution, fixing, re-designing etc.	4 Months
5 - Experiments	~2-3 Months
5.1 - Result collection and analysis, reimplementing solution w.r.t results	2 Months
5.2 - Evaluation and final fixes	1 Month
6 - Paper writing	~ 3-4 Months
6.1 - Arranging layout and structure	1 Month
6.2 - Writing	~ 2-3 Months

Figure 1: Gantt Chart

In the field of TinyML, the real concern is maintaining model success (accuracy, precision, AUC, etc) while reducing resource usage such as memory consumption and computation. Apart from this, several parameters, FLOPs, and MAC (multiply-accumulate) operations are some of the pivotal metrics used when calculating efficiency. At this step, bit precisions of MAC operations are taken into account because doing an operation with 32-bit floating point representation consumes much more resources than doing it with lower bit precision such as an 8-bit integer or less. Eventually, the model accuracy will be evaluated together with memory consumption and computation overhead. Apart from this, a real deployment on a resource-constrained edge device will strengthen our assessment through testing the developed solution in real-world scenarios, which differs from previous works that generally introduce the results concerning metrics such as several FLOPs, MACs, etc. Briefly, the successful deployment of the proposed framework in real-world settings is considered the final step of the research.

References

[1] Banbury, C. R., Reddi, V. J., Lam, M., Fu, W., Fazel, A., Holleman, J., Huang, X., Hurtado, R., Kanter, D., Lokhmotov, A., Patterson, D. A., Pau, D., sun Seo, J., Sieracki, J., Thakker, U., Verhelst, M., and Yadav, P. Benchmarking tinyml systems: Challenges and direction. *ArXiv abs/2003.04821* (2020).

[2] Bulat, A., and Tzimiropoulos, G. Bit-mixer: Mixed-precision networks with runtime bit-width selection. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), 5168–5177.

[3] Cai, Z., and Vasconcelos, N. Rethinking differentiable search for mixed-precision neural networks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), 2346–2355.

[4] Choi, J., Wang, Z., Venkataramani, S., Chuang, P. I.-J., Srinivasan, V., and Gopalakrishnan, K. Pact: Parameterized clipping activation for quantized neural networks. *ArXiv abs/1805.06085* (2018).

[5] Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *arXiv: Computer Vision and Pattern Recognition* (2015).

[6] Jin, Q., Yang, L., and Liao, Z. A. Adabits: Neural network quantization with adaptive bit-widths. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 2143–2153.

[7] Leng, C., Li, H., Zhu, S., and Jin, R. Extremely low bit neural network: Squeeze the last bit out with admm. In *AAAI Conference on Artificial Intelligence* (2017).

[8] Liu, L., and Deng, J. Dynamic deep neural networks: Optimizing accuracy-efficiency trade-offs by selective execution. *ArXiv abs/1701.00299* (2017).

[9] Liu, Z., Wang, Y., Han, K., Ma, S., and Gao, W. Instance-aware dynamic neural network quantization. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), 12424–12433.

[10] Micikevicius, P., Narang, S., Alben, J., Diamos, G. F., Elsen, E., García, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., and Wu, H. Mixed precision training. *ArXiv abs/1710.03740* (2017).

[11] Rastegari, M., Ordonez, V., Redmon, J., and Farhadi, A. Xnor-net: Imagenet classification using binary convolutional neural networks. *ArXiv abs/1603.05279* (2016).

[12] Sanchez-Iborra, R., and Skarmeta, A. F. Tinyml-enabled frugal smart objects: Challenges and opportunities. *IEEE Circuits and Systems Magazine 20* (2020), 4–18.

[13] Uhlich, S., Mauch, L., Cardinaux, F., Yoshiyama, K., García, J. A., Tiedemann, S., Kemp, T., and Nakamura, A. Mixed precision dnns: All you need is a good parametrization. In *International Conference on Learning Representations* (2019).

[14] Wu, B., Wang, Y., Zhang, P., Tian, Y., Vajda, P., and Keutzer, K. Mixed precision quantization of convnets via differentiable neural architecture search. *ArXiv abs/1812.00090* (2018).

[15] Xu, Y., Zhang, S., Qi, Y., Guo, J., Lin, W., and Xiong, H. Dnq: Dynamic network quantization. *2019 Data Compression Conference (DCC)* (2018), 610–610.

[16] Zhou, S., Ni, Z., Zhou, X., Wen, H., Wu, Y., and Zou, Y. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *ArXiv abs/1606.06160* (2016).

[17] Zhu, C., Han, S., Mao, H., and Dally, W. J. Trained ternary quantization. *ArXiv abs/1612.01064* (2016).