УДК 681.518.3

FORWARD AND INVERSE TASKS FOR INFERENCE RATE ESTIMATION IN BINARY NEURAL NETWORKS

Shakkouf A. (ITMO University)
Scientific research supervisor – C.T.S. Gromov V. S.
(ITMO University)

Abstract. This work leverages the " XNOROP" metric to define and solve two critical tasks in BNNs. The first task, referred to as the "Forward Task", involves estimating the inference rate of a given model M when deployed on a specific target device T. The second task, known as the "Inverse Task", outlines a systematic procedure to identify a set of target devices T capable of achieving a required inference rate R when deploying the model M. Additionally, we extend the foundational formula of "XNOROP" and introduce “LXNOROP” which incorporates considerations for memory access time, enhancing its applicability for real-world deployment scenarios.

Introduction. The AI community is rapidly evolving, with state-of-the-art advancements emerging in fields such as natural language processing, computer vision, and reinforcement learning. Metrics play a fundamental role in benchmarking performance, comparing efficiency, and evaluating AI models. For instance, task-specific metrics like mAP, accuracy, precision, recall, F1-score, RMSE, and AUC-ROC ensure consistent evaluations, while efficiency metrics—such as FLOPs, MACs, energy consumption, memory usage, and latency—help estimate the training and inference costs. FLOPs measure the total number of floating-point operations, whereas MACs count the multiply-accumulate operations, with one MAC roughly equivalent to two FLOPs. Although these metrics provide useful approximations rather than exact runtime performance, they offer valuable insights, especially for edge computing. In 2015, compact Binary Neural Networks (BNNs) emerged, constraining weights and activations to binary values and primarily using XNOR binary operations during inference, which renders traditional metrics like MACs and FLOPs inadequate. Since no metric can fully capture the computational cost of BNNs without considering the target device’s bus size (i.e., the size of its core registers), the XNOROPs metric was introduced to quantify BNN computation on CPUs and MCUs, while Binary Operations Per Second (BOPS) is less suitable for estimating BNN performance.

Main part. Given a BNN model M and a target T on which we need to deploy our model M, what is the inference rate R_{fps} that we may get after the deployment of model M on the target (platform) T.

Mathematically we can express problem statement as follows: given a model

$$M = \left(Memory_M, Structure_M \right); \begin{cases} Memory_M = Weights_M + Activations_M + Other_M \\ Structure_M = \left(\left\{ L_i \right\}_{i=1}^{L}, Connections, Operations, Hyper \right) \end{cases}$$

, and a target $T = \left(Bus_T, Memory_T, CPU_T \right); \begin{cases} Bus_T \in \left\{ 8,16,32,64,128,256 \right\} \\ Memory_T = \left(Size_T, Speed_T \right) \\ CPU_T = \left(f_T, C_T \right) \end{cases}$, What is the inference rate R_{fps} obtained by deploying M over T ?

where $Memory_M$ is the memory the model occupy which comes mainly from model weights $Weights_M$, model activations $Activations_M$, and other parameters of the model $Other_M$, $Structure_M$ is the configuration of the model that shows what are the layers $\left\{ L_i \right\}_{i=1}^{L}$ in the model and how many LXNOROP and FLOP operations are embedded in the model, $Connections$ is the arrangement of layers, including sequential, residual, or parallel structures, $Operations$ the types of operations (e.g., convolution, pooling, attention), and some hyper parameters $Hyper$ (kernel size, stride, or number of filters). Bus_T is bus size of the target T, $Memory_T$ is the memory that target T is occupied with. It has a size of $Size_T$ and a speed of $Speed_T$ which represents how many CPU clocks are needed to access one element from the memory $Memory_T$. CPU_T is the processing unit that target T is occupied with. CPU_T ticks at a frequency of f_T and has C_T cores embedded in it. How to calculate $Weights_M$? $Weights_M$ occupy most of memory place that is needed to store the model M . We iterate over all blocks of each layer $\{L_i\}_{i=1}^{L}$ and sum up to find $Weights_M$.

The computation power of target T is calculated as $CP = \sum_{i=1}^{cores_T} f_{i_T}$ where $cores_T$ is the number of cores integrated into the target and f_{i_T} is the frequency of core i . The term computation power CP in our case refers to the maximum quantity of CPU clocks that our target can offer. R_{fps} is calculated as:

$$R_{fps} = CP / \left(N_{LXNOROPs} . Clocks_{LXNOROP} \right)$$

$N_{LXNOROPs}$ is the quantity of LXNOROPs and $Clocks_{LXNOROP}$ is the cost of one LXNOROP in terms of clock cycles.

Conclusion. The modification proposed to XNOROP metric (LXNOROP) enhances its applicability and accuracy in evaluating computational cost within BNN models. By integrating memory access time, the improved metric provides a more comprehensive and realistic measure of performance, aligning better with deployment constraints and real-world scenarios. The forward and inverse tasks of LXNOROP provides a systematic approach to solve real-life tasks with BNNs deployment.

Shakkouf A. (author)	Signature
Gromov V. S. (Scientific research supervisor)	Signature