

A THEORETICAL ANALYSIS FOR OUR LOSS FUNCTION

In manuscript section 4.2.1, we have proposed a hash center loss in Eq.(9) as follows:

$$L_{hc} = -\frac{1}{N} \sum_i^N \log \frac{\exp(\phi(h_i, c'_i)/\tau)}{\sum_{l=1}^m \exp(\phi(h_i, c_l)/\tau)}. \quad (1)$$

This section aims to demonstrate that this optimization objective L_{hc} can also promote a maximum distance between the dynamic hash centers. Without loss of generality, let us employ the variable j to denote an arbitrary dimension in h_i and c'_i . We can then reformulate the loss function, Eq. 1, as follows:

$$\begin{aligned} L_{hc}^j &= -\frac{1}{N} \sum_i^N \log \frac{\exp(h_{ij}c'_{ij}/b\tau)}{\sum_{l=1}^m \exp(h_{ij}c_{lj}/b\tau)} \\ &= -\frac{1}{N} \sum_{i=1}^N (h_{ij}c'_{ij}/b\tau - \log \sum_{l=1}^m \exp(h_{ij}c'_{ij}/b\tau)). \end{aligned} \quad (2)$$

Next, using AM-GM inequality, we can get the following:

$$\begin{aligned} L_{hc}^j &\geq -\frac{1}{N} \sum_{i=1}^N \left(\frac{h_{ij}c'_{ij}}{b\tau} - \frac{1}{m} \sum_{l=1}^m \frac{h_{ij}c'_{lj}}{b\tau} - \log m \right) \\ &= -\frac{1}{b\tau Nm} \sum_{i=1}^N \sum_{l=1}^m \frac{h_{ij}}{c'_{ij}} (c'_{ij}{}^2 - c_{lj}c'_{ij}) + \log m. \end{aligned} \quad (3)$$

Note that if the loss function L_{hc} can optimize h_i to be sufficiently close to c'_i , it implies the existence of a $\delta > 0$ such that $h_{ij}/c'_{ij} \leq 1 + \delta$. Then we can get:

$$\begin{aligned} L_{hc}^j &\geq -\frac{1}{b\tau Nm} \sum_{i=1}^N \sum_{l=1}^m (1 + \delta)(1 - c_{lj}c'_{ij}) + \log m \\ &= \sum_{l \neq k} \frac{\mathcal{N}(l, k)}{b\tau Nm} c_{lj}c_{kj} + C. \end{aligned} \quad (4)$$

where $C = \log m - \frac{(1+\delta)(m-1)}{b\tau m}$ is a constant and $\mathcal{N}(l, k)$ depends on the number of samples per category in the dataset. Thus, minimizing the loss function L_{hc} is equivalent to maximizing the distance between hash centers $c_l, c_k, l \neq k$ with the weight $\frac{\mathcal{N}(l, k)}{b\tau Nm}$.

B EFFICIENCY ANALYSIS

In this section, we analyzed the reduction in memory consumption and acceleration in computation when employing our ODH model in a CPU-only environment. Following the calculation method in [1], the memory usage was computed by summing the product of 32-bit values with the number of real-valued parameters and the product of 1-bit values with the number of binary parameters in the network. For computation, we utilized FLOPs to measure the total real-valued multiplication computation. As the bitwise XNOR operation and POPCOUNT can be executed in parallel by the current generation of CPUs, the FLOPs is calculated as the sum of real-valued floating point multiplication plus 1/64 of the sum of 1-bit multiplication [2]. Please note that ODH is a hashing model that can apply any BNN backbone, and it does not introduce any additional parameters beyond the BNN itself during practical deployment. Therefore, the efficiency improvements of ODH primarily depend on the selected BNN backbone.

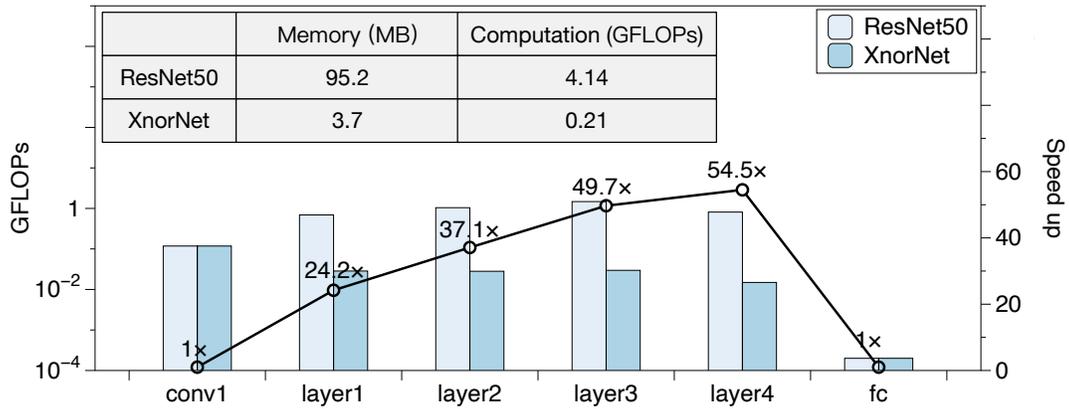


Figure 1: The efficiency analysis on ODH.

For example, Figure 1 depicts the reduction in memory usage and the acceleration in computational speed achieved when employing XnorNet as the backbone, in comparison to the commonly used ResNet50 in DH [3]. We can find adopting XnorNet yields a 25.7× improvement in storage overhead and a 19.7× improvement in computation. Note that our results differ from the work [2] due to the inclusion of the cost associated with BatchNorm. In practical scenarios, one can select an appropriate BNN as the backbone for ODH based on the specific requirements of the given task.

REFERENCES

- [1] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. 2018. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *Proceedings of the European conference on computer vision (ECCV)*. 722–737.
- [2] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. 2016. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*. Springer, 525–542.
- [3] Li Yuan, Tao Wang, Xiaopeng Zhang, Francis EH Tay, Zequn Jie, Wei Liu, and Jiashi Feng. 2020. Central similarity quantization for efficient image and video retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3083–3092.