GreenEdge Contest:

Energy-Aware Image Classification

I. Overview

In this competition, we are calling on participants to develop a model that excels in both energy efficiency and accuracy when classifying the **TinyImageNet** test set. Specifically, the participants will tackle a multi-class classification problem on this dataset, focusing on **optimizing the energy-accuracy trade-off during the inference phase.** We're eager to see innovative approaches that effectively balance minimizing the average energy consumption with delivering accurate results.

The motivation behind this contest is that oftentimes the long-term costs of repetitive inference jobs outweigh the initial training expenses. Consequently, reducing the resources required for inference without compromising overall performance is paramount for a more efficient and sustainable future of ML applications.

II. Dataset and resources

TinyImageNet is a subset of the ImageNet dataset containing 100K images of 200 classes (500 for each class) downsized to 64 × 64 colored images. In this challenge, participants are encouraged to use the *Deep Lake*[1] database to load the dataset into their code. The queue of test data provided by *Deep Lake* contains $n = 1000$ images. We denote the latter queue by

$$\mathcal{Q} = \left[(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n)\right],$$

[1]Deep Lake, https://github.com/activeloopai/deeplake

where $\mathbf{x}_j$ is the j^{th} test sample in $\mathcal{Q}$, and y_j the corresponding true label, $\forall j \in \{1, 2, \cdots, n\}$. The goal is to classify all samples in $\mathcal{Q}$, regardless of order, while accounting for

1) average classification accuracy A,
2) average energy consumption E,

as defined in Sections III-A and III-B. The candidate's solution will be assessed by the metric defined in Section III-C, which combines the average accuracy and average energy consumption. The contestants are provided with file *greenedge_challenge.ipynb*, containing

- code for loading training and test queues via *Deep Lake*,
- code for estimating the energy consumption,
- a function called *metric* to evaluate their final solution.

The file is available at the following URL: Google Colab.

The next section is dedicated to explaining how these codes should be used to evaluate performance.

III. Evaluation

A. Average Accuracy

Given a model m, the top-1 average accuracy over test queue $\mathcal{Q}$ (of size denoted n) is

$$A(m) \triangleq \frac{1}{n} \sum_{j=1}^{n} \mathbf{1}_{y_j = m(\mathbf{x}_j)},$$

where for any property B, quantity $\mathbf{1}_B$ is equal to 1 if B holds, and 0 otherwise.

B. Average Energy Consumption

In this contest, we consider the number of FLOPs (Floating-Point Operations) as a quantity correlated to the energy consumed on a per-sample basis. To calculate the number of FLOPs in their code, participants can utilize the ***fvcore*** library[2], by following the example given in the provided greenedge_challenge.ipynb. This library allows access to the hardware performance counters. That is, given a classifier m, the code provided approximates energy consumption $E(m)$, in GFLOPs, by

$$\mathrm{E}(m) \triangleq \frac{1}{n}\mathrm{FLOPs}(\mathcal{Q}, m) = \frac{1}{n}\mathrm{fvcore}(\mathcal{Q}, m), \tag{1}$$

where n is the number of test samples.

[2]fvcode, https://github.com/facebookresearch/fvcore/blob/main/docs/flop_count.md

C. Target metric

Each submission should consist of different classifiers, or *variants*, denoted $m_1, ..., m_R$, each having a different energy consumption. The number R of variants is not restricted. For each variant m_i, one can compute $A(m_i)$ as in Section III-A, and $E(m_i)$ as in Section III-B. Then, all resulting values should be given as input to function *metric* in the provided greenedge_challenge.ipynb. The logic behind the latter function is as follows. We require that the amount of energy used by

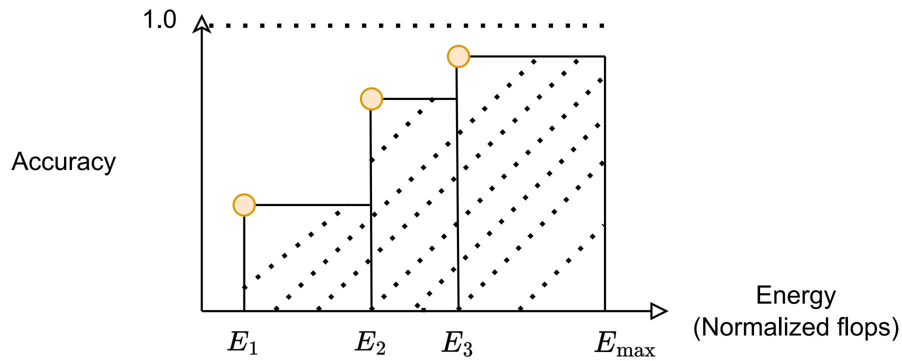

Fig. 1. An example plot for Energy–Accuracy trade-off.

any of the variants not exceed $E_{\max} =$ **39.32 GFLOPs.** Indeed, we consider *EfficientNet B7*[3] as the largest baseline model in terms of energy, which requires the mentioned number of flops for inference on a single image.

We define the *Energy vs. Accuracy curve* (Fig.1) as the zero-th order interpolation of the points $\{(E(m_i), A(m_i))\}_{i=1}^{R}$, as depicted in Fig. 1 (where $R = 3$), in which energies $\{E(m_i)\}_{i=1}^{R}$ have been sorted in increasing order, resulting in a sequence $\{E_i\}_{i=1}^{R}$. **We will rank submissions based on the area under the resulting Energy vs. Accuracy curve between** $E_1 = \min_i E(m_i)$ **and** $E_{\max}$. The larger the area, the better.

As an example, a submission with a single model, e.g., a standard MLP, would result in a single (E_1, A_1), which will have an area under curve computed as: $A_1 \times (E_{\max} - E_1)$. Therefore, to be competitive, a model should result in a sufficiently low $\min_i E(m_i)$. An effective strategy for addressing this is to employ the Early Exiting [1] method. Different variants can be obtained by different choices of thresholds and exit locations. Another strategy is to first cluster the test examples, and then classify each cluster with a different model.

[3] EfficientNet B7, https://github.com/lukemelas/EfficientNet-PyTorch

D. *Recap of requirements*

- Model of size not exceeding 5 GB.
- Participants should submit different models (variants).
- The average energy used on the run of each variant m_i on test queue $\mathcal{Q}$ should not exceed $E_{\max}$.

References

[1] S. Teerapittayanon, B. McDanel, and H. T. Kung, "Branchynet: Fast inference via early exiting from deep neural networks," *CoRR*, vol. abs/1709.01686, 2017. [Online]. Available: http://arxiv.org/abs/1709.01686