Deep learning

Qq[†]
[†]Northwest University, China, [‡]Lancaster University, UK
[†] jr@stumail.nwu.edu.cn, {gl, hwang}@nwu.edu.cn, [‡]z.wang@lancaster.ac.uk

***Abstract*—**

***Keywords*-workload characterisation**

I. Introduction

II. Motivation

The ImageNet competition has led to significant advancements in exploring various architectural choices in CNNs, such as AlexNet, VGGNet, InceptionNet and ResNets, and the layers of model has increased from 8 to 1000 plus. As the researchers constantly push the boundary of the prediction accuracy, the model size and depth

Behind the excellent performance of these CNNs models is the great computing power

It is expected that the size of the DNNs (i.e., number of weights) and the number of MACs will be larger for the more difficult task than the simpler task and thus require more energy

Behind the excellent performance of the depth model is a powerful computing requirement: while constantly refreshing the limits of the accuracy of the task, its depth and The size has also multiplied. Currently, the most advanced deep learning model usually has millions of parameters that need to be stored, and the memory on the device is limited. In addition, it is not uncommon for even one model to infer calling O(109) memory accesses and arithmetic operations, all of which consume power and dissipate heat, which may consume limited battery capacity and/or thermal limitations of the test equipment. Although these models are often deployed at the back end of the data center, preserving user privacy and reducing user-aware query times requires migrating these deep neural network-provided intelligent services to edge computing devices. However, deploying a large, accurate deep learning model into a resource-constrained computing environment (eg, mobile phones, smart cameras, etc.) to bring about device inference poses some key challenges. Therefore, in order to get the product to land, model compression is an indispensable part. In order to deploy the deep learning model on mobile/embedded devices, it is necessary not only to analyze the performance of the model in a platform environment, but more important work should be devoted to reducing the memory consumption of the model, shortening the inferring time, and reducing power consumption. Therefore, a very natural solution is to compress and accelerate the deep convolutional neural network without affecting the classification accuracy. In summary, the problems faced by the deployment of the deep learning model at the mobile terminal have been urgently solved. In the past two years, related research work has been gradually carried out. On the one hand, the performance of the model itself needs to be analyzed. On the other hand, the compression of the model is also very necessary. Research point. Model compression is mainly about reducing the model size and reasoning time for experiments. By quantifying the shared weights, the size of the model can be compressed very well. By pruning the weights, the reasoning time of the model can be further accelerated. However, these methods mainly stay on the experimental conclusion and are not really on the mobile or embedded system. Deploying the model and putting it into practice, performance testing and compression experiments after actually deploying the model on an embedded platform is a necessary step to further verify the feasibility of the model's deployment on the mobile or embedded system, and is also the focus of this article.

III. Workloads

IV. Overview of our approach

V. System Evaluation

VI. Related Work

Our work lies at the intersection of multiple research areas: web browsing optimization, task scheduling, energy optimization and predictive modeling. There is no existing work that is similar to ours, in respect to optimizing web workloads across multiple optimization objectives on heterogeneous mobile platforms.

VII. Conclusions

This paper has presented an automatic approach to optimize the mobile web