An Analysis of Compression Methods for Deep learning networks

CS 259: Learning Machines
Parth Shettiwar†, Madhav Sankar Krishnakumar†
† University of California, Los Angeles
parthshettiwar@g.ucla.edu,
madhavsankar@g.ucla.edu

***Abstract*—The superior performance of DL models have inspired them to integrate them with IoT devices removing the human intervention altogether in various tasks. The models usually leave huge memory footprint leading to their inability of deploying them on mobile or IoT devices. Various approaches have come up in recent past to compress the Deep learning models without much loss in accuracy. Through this work, we aim to quantify the performance of various network compression algorithms by their application in literature Deep learning models like VGG and ResNet on the basis of metrics like accuracy, storage capacity, compute, inference time and energy consumption.**

I. Approach

Over the years, various compression approaches have come up in the deep learning domain to reduce the model size by a huge merging without losing much on accuracy. Overall the following 4 approaches are used in general to reduce the size of models (Due to limited resources and time, for each section, we plan to implement just the recent state of art approach in that domain and evaluate its performance):

- **Knowledge distillation**: Knowledge distillation is defined as the process of transferring the generalization ability of a teacher model to the student model to improve its performance. The main idea is that student model tries to mimic the teacher model with minimal parameters needed leading to almost same accuracy as teacher model. We plan to implement the [1] as part of this section.
- **Learning-based compression**: We can leverage reinforcement learning to provide the model compression policy instead of conventional rule-based compression policies that require domain experts to explore the trade-offs. We will be using [2] as the reference.
- **Quantisation** : This can be implemented in multiple ways. A few of the options we are looking at are reducing the precision and quantizing weights to one of 5 or 10 possible values. We will be using [3] and [4] as references.
- **Pruning** : Mainly done by removing unnecessary filters, neurons and layers to reduce the size of model. Two types: Structured (Removal of layers and filters in groups ensuring dense matrix operations) and Unstructured (Randomly removing layers and weights resulting in sparse matrix operations, but aggressive pruning can be done). We plan to implement [5], which is a latest CVPR reserach in unstructured pruning section and performs a multi-layer optimized version of magnituide based pruning.

Though in past, there have been survey papers like [4], [6], there has been no work yet to quantify the performance of various compression models uniformly, i.e. across same DL model, same dataset and training on same device. In summary, our key contributions are:

1) We present a small survey like analysis of existing in literature Compression methods for deep learning networks
2) Based on evaluation metrics, we perform an uniform basis comparison of these methods (1 from each section, discussed before) quantitatively to understand their applicability, use cases and various pros and cons depending on situation.
3) An easy to use interface for compression of DL networks, where the user inputs which model he wants to compress, using which algorithm and we return the compressed DL model, which he can use as a module/ block in his main DL network.

II. Evaluation Metrics, Baseline models and Datasets

We will access the compression algorithms based on following metrics:

- **Number of parameters/weights and total model size**
- **Accuracy of model**: We plan to use the classification error and Top 5% accuracy as metric while using models for image dataset classification
- **Inference and training time**: We plan to compute both of these times to understand how easy is to prune the model and once pruned how much time does it take to perform inference in real time on IoT devices.
- **Compute**: To understand the total compute requirements, again on IoT devices, this would play a crucial factor (A low compute requirements is highly desired).

The baseline models we plan to consider are following:

- ResNet
- VGG

We choose these models as these have shown to produce state of art results of classification on imagenet dataset. We plan to run them on **CIFAR-10** and **Imagenet** as part of this

work.

Finally, our eventual goal as part of this work would be to derive a possible approach to combine the pros of the various compression approaches, to get the best of all worlds.

REFERENCES

[1] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *ArXiv*, vol. abs/1503.02531, 2015.

[2] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han, "Amc: Automl for model compression and acceleration on mobile devices," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 815–832.

[3] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://arxiv.org/abs/1510.00149

[4] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," *ArXiv*, vol. abs/2103.13630, 2022.

[5] S. Park*, J. Lee*, S. Mo, and J. Shin, "Lookahead: A far-sighted alternative of magnitude-based pruning," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=ryl3ygHYDB

[6] R. Mishra, H. P. Gupta, and T. Dutta, "A survey on deep neural network compression: Challenges, overview, and solutions," *ArXiv*, vol. abs/2010.03954, 2020.