Weight Sparsity Training Performance Enhancement

Samuel Atkins
MASc. University of Toronto ECE
sam.atkins@mail.utoronto.ca

Eugene (Evgeny) Osovetsky
University of Toronto
eugene.osovetsky@mail.utoronto.ca

Abstract

This project will explore several techniques for inducing dynamic weight sparsity (where sparsity changes during training), in an attempt to obtain a net improvement in training time assuming ideal hardware, on a toy BERT-like model.

Introduction

We will explore using weight sparsity to speed up training. GPUs are not capable of exploiting weight sparsity for speed (unless sparsity has special structure). However, emerging hardware architectures may have such capabilities (one of us is employed by an AI hardware manufacturer that may fit this description). The vast majority of current AI research is GPU-based, so we believe there may be many unexplored weight sparsity ideas that may be highly beneficial should the right hardware arise.

We differentiate between “static" sparsity (pruning unimportant weights from a fully-trained model to reduce its size and improve inference time), and “dynamic" sparsity (where weights are dynamically dropped and possibly "re-grown" during training). The former is a fairly established research area. We will instead focus on dynamic sparsity.

We will explore several sparsity ideas on a toy BERT-like model, including implementing at least one established technique and one novel technique, and attempt to get to a net training time speedup (keeping same accuracy or loss). Since we will be performing our experiments on GPUs, we will not be able to observe the speedup directly. Instead, we will use either an estimate of FLOPs or the actual training time adjusted by average sparsity (i.e. assuming ideal hardware)

Related Work

Obtaining an accurate and sparse relationship between the input and output data yields many benefits. A sparse model offers a concise and sometimes interpretable explanation for the relationship between the input and target data. Further, sparse models require far less computational resources to deploy. Much research has been conducted to obtain accurate sparse representations of fully-trained models [3], [4], [5], [6]. Recently, a new pruning technique for obtaining sparse models emerged. Higher accuracy values were observed when network weights were reset to their original values and then retrained. This observation led to the “Lottery Ticket Hypothesis" [2]. This hypothesis states that any sparse neural network can be obtained by training the same network from a set of initial conditions. Clever training and initialization algorithms have since emerged [1].

Method / Algorithm

We will pick a “toy" BERT-like model with a pre-trained embedding (one or more encoder blocks with attention followed by feed-forward FC layers). We will aggressively reduce model size until it is trainable in reasonable time, i.e. significant MLM loss reduction in 20 minutes or less.

We will then apply a standard algorithm (e.g. RigL) at various sparsity levels and take note of the time it takes to train to same loss/accuracy, and compute the net time adjusted for sparsity (e.g. 40min. training time at 50% average sparsity yields net training time of 20min).

We will then apply at least 2-3 ideas to improve net training time - e.g. these can be selected from: varying the layers sparsity applies to, varying the schedule with which sparsity is applied, varying the regularizer function, varying the sparsification algorithm. Overall, we will attempt to reduce net training time to below the training time of the dense model. Instead of net training time, we may use estimated FLOPs if measuring time will turn out to be unreliable. We will make every effort to conduct sufficient literature review to ensure at least one technique we try is novel.

Summary

References

[1] Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. *CoRR*, abs/1911.11134, 2019.

[2] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Training pruned neural networks. *CoRR*, abs/1803.03635, 2018.

[3] Song Han, Huizi Mao, and William Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. 10 2016.

[4] Yann LeCun, J. S. Denker, S. Solla, R. E. Howard, and L. D. Jackel. Optimal brain damage. In David Touretzky, editor, *Advances in Neural Information Processing Systems (NIPS 1989)*, volume 2, Denver, CO, 1990. Morgan Kaufman.

[5] Shaohui Lin, Rongrong Ji, Yuchao Li, Cheng Deng, and Wei Liu. Toward compact convnets via structure-sparsity regularized filter pruning. *IEEE Transactions on Neural Networks and Learning Systems*, PP:1–15, 04 2019.

[6] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.