Report on Paper “Adaptive Deep Reuse: Accelerating CNN Training on the Fly” [2]

Hao Lin[1]

[1]Department of Computer Science and Technology, Nanjing University
njulh2017@outlook.com

1 Summary of the paper

The paper is intended to propose a new approach to accelerate the original CNN training. Due to the wide application of CNN networks and its compute-expensive property, many researchers have devoted to study how to accelerate CNN training. However, most of them only focus on reducing weight parameters instead of the inputs of convolutional layers. Xipeng Shen and his team seize this opportunity to conduct in-depth research in neuron vector similarities. Finally, they propose adaptive deep reuse as a method to accelerate the original CNN training. This method uses LSH to identify similarities among neuron vectors in the forward propagation and reuse them in the backward propagation to enable deep computation reuses. Meanwhile, it introduces adaptive deep reuse strategy for training to achieve further computation savings.

For actual benefits it can generate, this method requires holding two conditions, similarities among neuron vectors and less detecting and exploiting time than that it saves. Experiments prove the efficiency of the adaptive strategies from two aspects. Firstly, LSH is effective in identifying the similarities among neuron vectors. What’s more, automatically adjusts the parameter {L, H} for different stages is more effective than the others. In general, their method can save up to 69% training time with no accuracy loss.

2 Advantages of the method compared to other alternative methods

There are multiple alternative methods and I’d like to divide them into three parts.

First of all, adaptive deep reuse has different foothold from previous methods. It focuses on the properties in convolutional layers’ inputs, while lots of researches focus mainly on identifying the weight redundancy and reducing the number of computations of the convolutional layer. This is a completely new aspect of CNN training accelerating. People will conduct more in-depth research on this aspect based on the points raised in this article, making it a groundbreaking point of view.

Secondly, as the team mentioned in Section 7, LSH is firstly used in figuring out similarities among neuron vectors(It is used for different purpose previously). It performs quite well in recovering the original inference accuracy with a very small r_c. There are also randomized and approximate algorithms that can complete similar task before. For those randomized methods, solving the same instance of the problem sought with the same randomization algorithm twice may result in completely different answers. And not only the results, but even the time may vary greatly. In this case, non-randomized algorithm leads to a determined result, which makes it more realistic in the industry. For approximate methods, they can potentially be combined with adaptive deep reuse. This means that adaptive deep reuse is highly adaptable to different approximate algorithms. Even at some time later, this algorithm is not efficient when used alone, but it can still be used with other more efficient approximate algorithms to increase efficiency, which makes it “live longer”.

And thirdly, as we can see in the evaluation part of the article, three different strategies can also be used in saving computation time. The second strategy, which automatically adjusts the parameter set {L, H} for different training stages, performs the best with saving up to 69% training time. It gives us a larger speedup and a better balance between computation savings and training errors. We can achieve higher efficiency with less loss than other methods using this method.

Finally, the characteristics and fundamentals of this technology's accelerated low-level computing may be successfully applied to different platforms, or even mobile computing resources or embedded computing devices with limited computing resources [1]. While other types of technologies have not yet explored issues related to cross-platform applications, and probably even cannot be applied to different platforms.

In general, groundbreaking, adaptiveness, efficiency, accuracy and portability should be the five main advantages of the method.

3 The limitations of the method

Adaptive deep reuse also has its limitations. I reckon that there are four main limitations. Firstly, this article only focuses on how to accelerate Convolutional Neural Network training based on convolutional layers’ inputs. It lacks comprehensive consideration of reducing the number of computations of the convolutional layer while exploiting the similarities and the reuse. Both of these can be counted as the cornerstones of accelerating CNN training process, but it may be misleading when considered separately.

Secondly, I think that the article's exhaustive classification method doesn’ t cover the complete set of all methods. In the process of proposing the LSH method, the author's wording is ”After a thorough exploration of several different methods, we identified LSH as the clustering method for the adaptive deep reuse”. And the remainder of the paper does not further prove that the LSH method is indeed the best method of all clustering methods. Although LSH is indeed widely used for solving the Neural Network problem in high dimension space, there still exist doubts that whether LSH is surely the best answer for the issue we discuss here.

Thirdly, this paper lacks rigorous mathematical proof. Many policies are short of the basis of mathematical theory. Instead, they are based primarily on observations. This is very dangerous. Because the correctness of the final conclusion depends on strict proof rather than observing the degree of conformity between the fact and the conclusion. If there is some kind of observation that points out the flaws with the existing conclusions, then the conclusions put forward by the article naturally cannot be recognized.

Fourthly, the team only evaluated adaptive deep reuse by using it during the training of three popular convolutional neural network (CNN) architectures for image classification. Whether other types of convolutional neural networks will achieve the same degree of significant acceleration is still unknown. In other words, the generalization ability of the model still has doubts.

4 Possible improvements

First of all, there may be many input layer clusters that do not change between forward iterations of different rounds. Probably detecting and identifying them and ignoring unnecessary re-clustering can speed up the training of the model, too.

Secondly, the article lacks rigorous arguments, and many conclusions or methods are derived from observations and experiments. So, authors may add some specific arguments for lemmas, theorems, etc. to make the foundation of the article stronger.

Thirdly, authors can think about the efficiency issues of applying adaptive deep reuse to other types of CNN networks. Meanwhile, they can do further research in the possibility of applying this technology to neural networks of other architectures, such as LSTM.

Fourthly, there will be various computing platform with different computing power. Authors can do further research in demonstrating the ability of adaptive deep reuse to apply to different computing resources. There exists an opportunity making it not only adaptable to those various datasets or different iterations, but also the resources provided by the hardware [1].

Last but not least, this method focuses on computation reuse based on properties in convolutional layers’ inputs, while prior work has its focus on reducing weights, reducing precision and leverage sparsity. Could these improvements from different aspects be combined into a faster neural network training method? Whether this combination will result in greater overhead or higher efficiency? In this case, can the iteration of the neural network converge? Multiple questions can be raised and they are also waiting for us to solve.

References

[1] Anthony Alford. Xipeng shen on a new technique to reduce deep-learning training time. https://www.infoq.com/news/2019/05/shen-reduce-dnn-training/.

[2] Xipeng Shen Lin Ning, Hui Guan. Adaptive deep reuse: Accelerating cnn training on the fly. (ICDE2019). https://en.wikipedia.org/wiki/Benford's_law.