Investigation of Influence Related to Multi-GPUs and Hyperparameters on the ViT Model Based on PyTorch Data Parallelism

Yibo Feng[1] and Haoran Zheng[2,*]

[1] Department of Software Engineering, Central South University, Chang Sha, 410009, China
[2] Department of Computer Science, UCSI University, Kuala Lumpur, 56000, Malaysia
*1002060139@ucsiuniversity.edu.my

Abstract: Neural networks, which are highly effective models for addressing various real-world challenges, face the challenge of excessive computational requirements caused by the substantial size of the models and the extensive datasets employed during training. In this paper, the purpose is to examine the effects of hyperparameters and factors such as the number and model of Graphics Processing Units (GPUs) on training time and accuracy using the Vision Transformer (ViT) model as a case study within parallel algorithms. The experiments conducted in this study employed PyTorch as the framework and CIFAR-10 as the dataset. The learning rate chosen for the experiments was set to 0.001, while two batch sizes, namely 64 and 32, were explored. Additionally, the study investigated different epoch values, including 1, 5, 10, and 20. To handle the computational workload effectively during training, data parallelism was employed as an approach. In the realm of optimizing the performance of parallel algorithms, numerous methods have been proposed to enhance their efficiency. Consequently, this research places particular emphasis on hyperparameter tuning. The primary objectives of this paper are to examine the impact of GPUs on both training time and model accuracy, as well as to analyze the influence of hyperparameter configuration on the accuracy of the model.

Keywords: GPU parallel algorithms; Deep learning; Data parallelism; Vision Transformer; PyTorch.

1 Introduction

Deep learning models, including large-scale ones such as Bidirectional Encoder Representations from Transformers (BERT) and Generative Pre-trained Transformer (GPT) [1,

© The Author(s) 2023
P. Kar et al. (eds.), *Proceedings of the 2023 International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2023)*, Advances in Computer Science Research 108,
https://doi.org/10.2991/978-94-6463-300-9_65

2], have garnered significant attention in the realm of natural language processing due to their remarkable capabilities in addressing intricate problems. These large-scale models, leveraging techniques such as deep learning and self-attention mechanisms [3], have demonstrated excellent performance in various natural language processing tasks. GPT, in particular, is a typical large-scale model based on the Transformer model [4], which has exhibited substantial accomplishments in the field of natural language processing. By incorporating self-attention mechanisms and Transformers [4], GPT overcomes the limitations of traditional convolutional neural networks and recurrent neural networks, further improving the performance of natural language processing models. Inspired by the success of GPT, researchers have begun exploring the application of similar models in the field of computer vision. The Vision Transformer (ViT) model is a transformer-based architecture specifically designed for processing image data [5], and has demonstrated remarkable performance across various computer vision tasks in recent years [6]. However, training such models is a computationally demanding process, and achieving excellent performance in model training frequently relies on a large amount of high-performance computing resources. Therefore, resource allocation is a key factor in determining the computational efficiency of model training.

Vaswani et al. pioneered the Transformer model, which introduced a new computational paradigm and model structure [4]. In many cases, it has replaced convolutional and recurrent neural networks e.g. Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), and has been demonstrated to reduce computation time when training CNNs on parallel CPUs [7]. Furthermore, increasing the number of CPUs leads to higher model accuracy. Dosovitskiy et al. proposed the Vision Transformer (ViT) model, which has had a significant impact on the field of image recognition [5]. By introducing self-attention mechanisms and Transformers, the model overcomes the limitations of traditional CNNs in handling long-range dependencies and capturing global information. These advancements have yielded remarkable outcomes in improving the performance of image models, thereby driving advancements in the field of image recognition.

The optimization of hyperparameters plays a critical role in model training, with batch size and epochs being two key hyperparameters of interest [8]. Under specific resource allocations, adjusting hyperparameters can control factors such as data size, resulting in changes in model training accuracy. Therefore, this research endeavors to conduct experiments aimed at investigating the influence of epochs and batch size on the training process. Additionally, this study also explored the application of ViT-based image recognition methods in a multi-GPU training environment [5]. Multi-GPU training is a parallel computing approach that significantly accelerates the training speed and improves the performance of deep learning models. The training time of neural networks is influenced by several factors. Firstly, the computational workload of a neural network is primarily dictated by its architectural design. Secondly, the efficiency of the Pytorch framework used to build the neural network also affects training time [9, 10]. For large-scale image recognition tasks, multi-GPU training is an efficient solution. Through this research, an in-depth understanding of the performance of the

ViT model in a multi-GPU training environment can be gained, and its performance can be evaluated and compared. This analysis helps in understanding the advantages and limitations of different models in a multi-GPU environment. [11].

2 Method

2.1 Dataset Preparation

The dataset utilized in this study is CIFAR-10 dataset, which is a widely used computer vision dataset. The dataset consists of a total of 60,000 images, divided equally into 10 categories, with each category containing 6,000 images. Furthermore, each individual image are based on the RGB format with the size of 32×32 pixels.

2.2 ViT Model

The Vision Transformer (ViT) shown in Fig. 1 is an advanced deep learning model that has brought significant breakthroughs to computer vision. Departing from conventional methodologies reliant on convolutional layers, ViT adopts various approaches inspired by natural language processing. It treats images as a sequence of smaller patches and uses a technique called self-attention to capture global relationships between these patches. This paradigmatic shift enables ViT to comprehensively comprehend the entire image holistically, thus exhibiting exceptional performance in tasks such as image classification.

The ViT structure is implemented through a custom model class called VIT. This class inherits from nn.Module and contains an instance of the ViT model from the imported ViT module. The ViT model is initialized with parameters such as image size, patch size, number of classes, dimension, depth, heads, and MLP dimension. The forward() method of the VIT class utilizes the ViT model to perform the forward pass.Therefore, the ViT model in the code follows the principles of treating images as patches and utilizing self-attention to capture global relationships. This enables the model to analyze and classify images effectively.

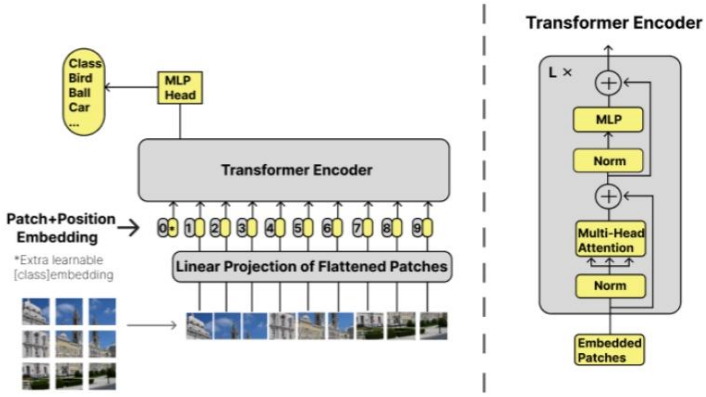

Fig. 1. The architecture of the vision transformer [3].

2.3 Data Parallel

Due to the increasing size of datasets and the complexity of deep learning models, the application of parallelism is the primary approach for addressing the time-consuming nature of training models. Data parallelism can be understood as distributing data across different processors or cores. DataParallel in PyTorch is a data parallelism strategy where the model runs on multiple GPUs, with each GPU receiving its independent batch of data slices during model training and computing gradient updates independently for these data slices.

DataParallel is an algorithm that enables single-threaded data parallel processing using multiple GPUs on a single device, and it supports multi-threaded data parallel training. A notable limitation of DataParallel lies in its relatively low GPU utilization, as it effectively aggregates multiple GPUs into a unified entity for training purposes. Since it uses a single process to compute model weights, it can suffer from communication issues that lead to low GPU utilization.

2.4 Implementation Details

The experiments were conducted using PyTorch framework for implementing the ViT model on the CIFAR-10 dataset. The learning rate used in the experiments was 0.001. The batch size was varied, with values of 64 and 32, and different epochs were explored, including 1, 5, 10, and 20. The loss function employed in these experiments was the CrossEntropyLoss, which is a commonly used loss function for measuring the dissimilarity between the predicted class probabilities and the true class labels. By minimizing this loss, the model is encouraged to make accurate predictions.

The experiments evaluated the impact of GPU models and quantities on training time and accuracy. Specifically, GPU models including the GeForce RT 2080ti, NVIDIA GeForce RTX

3080, and NVIDIA GeForce RTX 3090 were utilized. The experiments compared the performance of single GPU and multi-GPU DataParallel.

3. Experiment Results and Analysis

3.1 The Impact of GPU Types on Model Training Time and Accuracy

The results of experiment are shown in Table 1, indicating that as GPU performance improves, the training time of the model continues to decrease, while the accuracy remains relatively unchanged. An analysis of the mentioned results suggests that the improvement in GPU performance and parallel computing capabilities allows for simultaneous processing of more computational tasks, thereby accelerating the training process. The absence of notable changes in accuracy may be attributed to the fact that enhancements in GPU performance do not directly affect the intrinsic accuracy of the model. Accuracy is primarily influenced by factors such as the choice of model architecture, data quality, hyperparameter tuning, and the training process.

Table 1. Changes in training time and average accuracy with changes in GPU types (epoch=10，lr=0.001，Number of GPU =1，batch-size=64)

GPU type	2080ti	3080	3090
Time(s)	357	269	248
Avg acc	61	61	62

3.2 The Impact of the Number of GPUs on Model Training Time and Accuracy

The results, as shown in Table 2, indicate that as the number of GPUs used for parallel training increases, the training time of the model also increases, while the accuracy remains unchanged. An analysis of these results suggests that certain model architectures may not exhibit good scalability in data parallelism. As the number of GPUs increases, the communication overhead also increases, which can lead to an increase in training time. The Vision Transformer model may face bottlenecks in larger GPU clusters, as its computation and communication patterns may not fully utilize the additional GPU resources. This can result in increased training time without significant improvements in accuracy.

Table 2. Changes in training time and average accuracy with the number of GPU (GPU type=2080ti，lr=0.001，batch-size=64，epoch=10)

GPU	1	2	3	4	5	6	7	8
Time（s）	357	390	507	631	738	854	943	1071
Avg acc	61	60	60	61	61	62	61	61

Table 3. Load on two GPUs

GPU	GPU UTIL
0	59%
1	56%

Table 4. Load on four GPUs

GPU	GPU UTIL
0	39%
1	31%
2	33%
3	34%

Table 5. Load on eight GPUs

GPU	GPU UTIL
0	34%
1	17%
2	17%
3	18%
4	20%
5	20%
6	20%
7	20%

Furthermore, Table 3, Table 4 and Table 5 respectively provide the utilization of GPUs in the scenarios of two, four, and eight GPUs. It can be observed that GPU utilization is insufficient and uneven, with higher utilization on the first GPU. This phenomenon occurs because in data parallel training, data needs to be loaded onto each GPU for parallel computation. If there are bottlenecks or delays in the data loading process, it may take longer for the first GPU to complete data loading, resulting in higher utilization on that GPU. Accuracy is primarily influenced by factors such as the choice of model architecture, data quality, hyperparameter tuning, and the training process.

Therefore, in order to investigate the training time and accuracy of the model under different batch sizes, this study specifically conducted experiments on two GPUs while keeping other parameters unchanged.

3.3 The Impact of Batch-size on Model Training Time and Accuracy

The results, as shown in Table 6, demonstrate that with two GPUs, a smaller batch size leads to higher model accuracy. Notably, a smaller batch size of 32 leads to a superior model accuracy, with a recorded accuracy of 64% and a longer training time of 799 seconds. This result outperforms any other configuration shown in Table 6. The analysis of the phenomenon

mentioned above is that as the batch size increases, each GPU needs to store more input data and gradient information. When the batch size exceeds the GPU memory capacity, it needs to be divided into smaller sub-batches for processing, which introduces additional communication and synchronization overhead. This division and communication process can increase training time and may lead to an increase in gradient estimation noise, thereby affecting accuracy.

Tabel 6. Changes in training time and average accuracy with changes in batch-size (GPU type=2080ti，lr=0.001，Number of GPU =2，epoch=10)

Batch-size	32	64	128	256	512
Time（s）	799	397	231	193	175
Avg acc	64	60	56	49	41

The paper investigates the training time and accuracy of the model under different epochs on two GPUs while keeping other parameters constant.

3.4 The Impact of Epochs on Model Training Time and Accuracy

Based on the results, as shown in Table 7, it can be observed that the model's accuracy indeed increases with the increase in epoch. However, the impact of increasing epoch on training time is much greater than the improvement in model accuracy, and the model accuracy reaches a plateau after a certain number of epochs. As more training epochs are completed, the model has a better opportunity to learn the features and patterns of the dataset, thereby improving its predictive ability. The reason for the accuracy to stabilize is that as the training progresses, the model gradually learns the features and patterns present in the dataset. Initially, the model may experience rapid improvements in accuracy, but over time, the incremental benefit of new training data to improve the model's performance may diminish.

Table 7. Changes in training time and average accuracy with changes in epoch (GPU type=2080ti，lr=0.001，batch-size=64，Number of GPU =2)

Epoch	1	5	10	20
Time（s）	42	200	390	796
Avg acc	31	52	60	60

From the previous experiments, it can be concluded that model training is influenced by batch size, epoch, the number and performance of GPUs. Evaluating the impact of a single parameter in isolation is not feasible. Based on the experimental data, different parameters have varying degrees of influence on the training time and accuracy of the model. Therefore, it is possible to hypothesize an optimal parameter configuration based on the existing results.

4. Conclusion

With the rapid growth of big data, data volumes are increasing exponentially, and data

parallelism has become a primary approach to address this challenge. The PyTorch framework is a prominent system for data parallel machine learning. In this paper, experiments were conducted to investigate the impact of data parallel algorithms in PyTorch on model training performance. In this study, the collective influence of batch size, epoch, and the performance and quantity of GPUs on model training was analyzed. The experimental results provide insights for setting hyperparameters to minimize training time while improving model accuracy. Limitations of this work include: 1) No control experiments were conducted with other models. The impact of communication overhead with increasing GPU quantities may vary for different models. 2) Other forms of data parallelism, such as distributed data parallelism, were not explored. Distributed data parallelism could potentially reduce training time. 3) There are additional parameters that could be analyzed when assessing the impact of hyperparameters on parallel training. These issues will be solved in the future study.

Acknowledgment

All the authors contributed equally and their names were listed in alphabetical order.

References

1. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805 [cs.CL] (2018).
2. Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I.: Improving Language Understanding by Generative Pre-Training (2018).
3. Arnold, L., Rebecchi, S., Chevallier, S., Paugam-Moisy, H.: An Introduction to Deep Learning. Tao, INRIA-Saclay, LRI, UMR8623, Université Paris-Sud 11, F-91405 Orsay, France; LIMSI, UMR3251, F-91403 Orsay, France; Université Lyon 2, LIRIS, UMR5205, F-69676 Bron, France (2016).
4. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I.: Attention Is All You Need. In Advances in Neural Information Processing Systems (pp. 5998-6008) (2017).
5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3156-3164) (2020).
6. Dong, Y.-n., & Liang, G.-s.: Research and Discussion on Image Recognition and Classification Algorithm Based on Deep Learning. In 2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI) (pp. 1-6). Taiyuan, China: IEEE. doi:10.1109/MLBDBI48998.2019.00061 (2019).
7. Potluri, S., Fasih, A., Vutukuru, L. K., Al Machot, F., & Kyamakya, K.: CNN based high performance computing for real time image processing on GPU. IEEE (2011).

8. Smith, L. N.: A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay (Version 2). arXiv preprint arXiv:1803.09820v2 [cs.LG] (2018).
9. Abadi, M.: TensorFlow: learning functions at scale. In ICFP 2016: Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming (pp. 1) (2016).
10. Paszke, A., Gross, S., Massa, F., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32 (2019).
11. Miao, X., Wang, Y., Jiang, Y., Shi, C., Nie, X., Zhang, H., & Cui, B.: Galvatron: Efficient Transformer Training over Multiple GPUs Using Automatic Parallelism. arXiv preprint arXiv:2211.13878v1 [cs.LG] (2022).

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.