Mesopotamian journal of Big Data
Vol. **(2021)**, 2021, **pp**. 25–28
DOI: https://doi.org/10.58496/MJBD/2021/005 ISSN: 2958-6453
https://mesopotamian.press/journals/index.php/BigData

Research Article

Distributed Reduced Convolution Neural Networks

Mohammad Alajanbi,*[1], Donato Malerba[2], He Liu[3]

[1] *Faculty of Computing, University Malaysia Pahang, Gambang, Malaysia*

[2] *Department of Computer Science, College of Computer Science, Italy*

[3] *Global Energy Interconnection Research Institute, Beijing, China*

ARTICLE INFO

Article History

Received 09 Jan 2021
Accepted 03 Jun 2021
Published 30 July. 2021

Keywords

Distributed learning

Parallel Computing

CNN

Machine Learning

DRCNN

ABSTRACT

A Convolution Neural Network (CNN) is a popular tool in the domains of pattern recognition and machine learning. The performance of KCNN (kernel-based convolutional neural networks) is better than that of regular CNN. Although the KCNN can solve challenging nonlinear problems, doing so when dealing with a large-size kernel matrix is time-consuming and memory-intensive. The computational load and memory usage could be drastically decreased by adopting a reduced kernel strategy. But as the total amount of training data grows at an exponential pace, it becomes hard for a single worker to efficiently store the kernel matrix. Because of this, there can be no effective centralised data mining. In this research, we suggest the use of a distributed reduced kernel, or DRCNN, to train CNN using data that is stored in several locations. The data in the DRCNN will be spread out amongst the nodes at random. Static communication between nodes is defined by the network's architecture rather than the quantity of training data kept on each node. The DRCNN is an alternate direction multiplier (ADMM)-based distributed training technique, in contrast to the standard reduced kernel CNN. Experiments on the huge data set show that the distributed method may yield nearly the same results as the centralized algorithm, and it takes significantly less time. As a result, the amount of time spent computing is drastically reduced.

1. INTRODUCTION

In previous studies, convolution neural networks (CNN)[1, 2] have been discovered to be an innovative machine learning system that is built on the foundation of the feedforward neural network (FNN)[3]. This system is appropriate for both supervised and unsupervised learning situations. The most notable characteristic of this approach is that the weights of the hidden layer nodes are assigned arbitrarily and are not subject to updating; instead, the learning process is solely concerned with calculating the weights of the output nodes. Instead of changing the weights and biases of the hidden nodes in the CNN through lengthy testing, we may instead choose the weights and the biases of the hidden nodes at random for regression and classification purposes. As a result, it requires a less amount of time than comparable algorithms. In addition, algorithms such as pseudo-inverse and regularization are utilized in the final output layer in order to solve the output weights. However, the effectiveness of CNN is impacted by a wide variety of criteria, the most important of which are the selection of activation functions and the number of hidden layers. As a result, the accuracy of every CNN experiment is vary depending on the sizes of the hidden layers that are chosen. In this scenario, a kernel convolution neural network, also known as a KCNN, has been presented in, in which the feature mapping in the hidden layer is determined by the kernel matrix. This study was presented in a recent publication, where it was shown that KCNN will randomly choose a portion of data from the datasets in order to use as the support vectors. This is in contrast to the more conventional support vector machines, which are capable of eliminating repeated learning in order to greatly reduce computing cost while simultaneously improving accuracy. However, the performance of KCNN suffers when dealing with big data sets since the size of the kernel matrix grows at an alarming rate as the size of the data sets increases.

After some time, someone suggested a method that would simplify the kernel matrix. In a similar vein, when the training set is exceptionally huge, the kernel matrix likewise grows to be exceptionally enormous. In this scenario, putting all of your storage eggs in the basket of a single workplace is not the best strategy. In this particular scenario, we suggested a whole new distributed method. In this particular approach, the dataset is randomly segmented into multiple sections, each of which is then run on a separate site appropriately. The communication between sites is determined not by the data set itself but rather by the architecture of the network. Using this strategy, you may avoid the issue in which the size of the kernel function grows at an alarming rate as the data set grows larger.

*Corresponding author. Email: mohammad.cs88@gmail.com

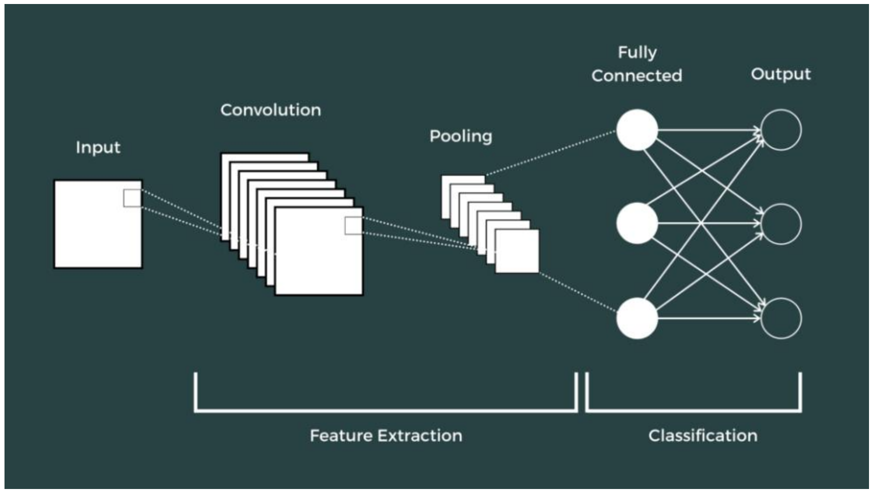

Fig.1. Convolution Neural Networks

2. BACKGROUND

2.1 Literature

Distributed learning is widely regarded as one of the most fruitful avenues of study for large-scale learning, in particular when dealing with data that is already in a distributed format. The most difficult aspect of distributed learning is figuring out how to combine the knowledge from all of the disparate datasets without the time-consuming and labor-intensive process of collecting all of the data. Collecting the outputs of learnt models or the models themselves has always been the method of choice for accomplishing this goal. The first method,[4] which entails the collection of several outputs, is more commonly used in earlier works. For instance, it was suggested to aggregate the outputs of local classifiers using a variety of heuristic decision rules (such as majority vote), and it was also proposed to train a global classifier using the labelled outputs of local classifiers. The technique presented in [5]was changed in order to increase its effectiveness for large-scale dispersed data. Additionally, the solution presented in was expanded in order to include multiple ways of building the global training set. The concept was eventually applied in the construction of a descriptive model based on scattered data. A framework based on distributed voting and pasting was presented in[6] for the purpose of learning sets of classifiers in order to further increase accuracy (ensembles).

2.2 Methods

A. Convolutional Neural Networks (CNN)

CNNs[7] are among the most significant and extensively used deep learning methods. In CNNs, numerous layers are trained using state-of-the-art methodologies, making CNNs one of the most important and frequently used deep learning methods. CNNs are applied rather frequently in the field of picture segmentation because to their high level of effectiveness and efficiency. A CNN used for image classification would often have three distinct layers, which are referred to as the convolution layer, the pooling layer, and the fully connected layer. The deconvolution layer is yet another layer that is crucial to the process of picture segmentation and plays a significant function. Various duties are carried out by the various tiers. The forward phase and the back-propagation phase are the two primary stages that make up the training process for a CNN. After multiplying the inputs and the weights in the first phase of the process—during which the picture is presented to the network—the convolutional operation is carried out on each layer of the network in order to determine the network's final output. The outcome of the output is employed in the calculation of the network error, which is then used to change the parameters of the network. An error function is used to do a comparison between the output and the ground truth in order to detect the network fault. Following this, the back-propagation stage will begin, which will be determined depending on the error rate. During this stage, the gradient of each parameter will be computed, and all parameters will be changed in order to reduce the error rate. This process is carried out over and over again until the conditions necessary for termination are satisfied.

B. Semantic Image Segmentation

Picture segmentation is the process of splitting an image into meaningful parts that are distinct from one another and do not overlap according to human perception. This is now considered to be one of the most fundamental issues in machine vision. Combining the processes of image segmentation with object identification, the semantic segmentation of the picture labels each pixel of the image with the name of an object. This process is known as semantic segmentation. In order to accomplish this goal, a variety of methodologies, including unsupervised, semi-supervised, and fully supervised approaches, have been tried and tested. The dataset that is utilized for this investigation will be discussed in the subsequent parts, as will the methodology that is going to be suggested.

3. EXPERIMENTAL RESULTS

In order to evaluate the usefulness of coded computation, we have implemented the suggested algorithms and examined their performance on a cluster hosted by Amazon EC2. In order to determine the frequency with which stragglers arise in our testbed, we begin by obtaining the empirical distribution of task runtime. This is done by monitoring the round-trip timings between the master node and each of 10 worker instances that are hosted on an Amazon EC2 cluster. Each worker does a matrix-vector multiplication and then transmits the output of the calculation to the master node. The master node then measures round trip timings, which include both the amount of time spent computing and the amount of time spent communicating. Following the completion of 500 iterations of this process by each worker, we are able to acquire an empirical distribution of the round trip times for all of the worker nodes.

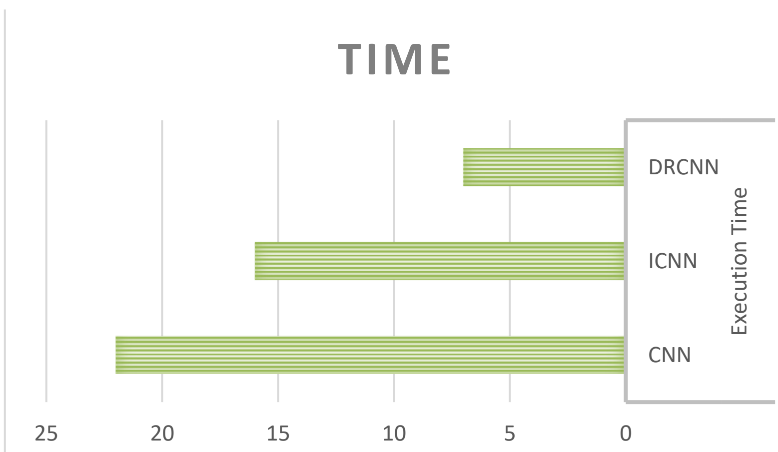

Fig.1. Execution Time of DRCNN

4. CONCLUSION AND FUTURE WORK

An algorithm of DRCNN that is based on the ADMM technique in a distributed environment is proposed in this research. The optimal solution of output weight is found by iterative updates to the algorithm. In recent years, distributed algorithms have been utilized for many common machine learning issues. This is due to the fact that the scale of the data is becoming greater and more complicated. It is believed that the alternating direction approach of the multiplier is particularly suited for various disciplines, including distributed optimization problems, given these circumstances. When compared to the earlier centralized approach, the findings demonstrate that the model performance of this distributed technique is superior and that it has a lower computation time. This results in significant reductions in the amounts of computing space and time consumption.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to express their gratitude to the University Malaysia Pahang, the Informatics Institute for Postgraduate Studies, and the Al Salam University College for their moral support. Please accept my sincere gratitude for the useful recommendations and constructive remarks provided by the anonymous reviewers.

Funding

The authors receive no funding for this work.

References

[1] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.
[2] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET)*, 2017, pp. 1-6: Ieee.
[3] G. Bebis and M. Georgiopoulos, "Feed-forward neural networks," *IEEE Potentials*, vol. 13, no. 4, pp. 27-31, 1994.
[4] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into imaging*, vol. 9, no. 4, pp. 611-629, 2018.
[5] J. Chen, K. Li, K. Bilal, K. Li, and S. Y. Philip, "A bi-layered parallel training architecture for large-scale convolutional neural networks," *IEEE transactions on parallel and distributed systems*, vol. 30, no. 5, pp. 965-976, 2018.
[6] D.-X. Zhou, "Deep distributed convolutional neural networks: Universality," *Analysis and applications*, vol. 16, no. 06, pp. 895-919, 2018.
[7] S. Q. Zhang, J. Lin, and Q. Zhang, "Adaptive distributed convolutional neural network inference at the network edge with ADCNN," in *49th International Conference on Parallel Processing-ICPP*, 2020, pp. 1-11.