Title: Optimization and acceleration of convolutional neural networks: A survey

JX-Ma

2024/8/24

This paper describes CNN in detail summarizes architectural evolution of CNN from 1998 to 2019 and introduce three strategies to strengthen speed of CNN in algorithm level and implement level. they are SGD ,fast convolution and parallelism techniques.

INTRODUCE

this section mainly introduce some concept about CNN, next, shows traditional convolution algorithm, finally, Describe improvement of CNN in recent years. for instance, using local receptive field and weight sharing leading to tremendously decrease in the number of parameters that greatly helps in the reduction of training time and inference.

CNN BACKGROUND

This section mainly introduces the background of CNN and its development in recent years. It then introduces the research history of many scholars on fast convolution, including their optimization and improvement on fast convolution, as well as the shortcomings of the improved methods. The main method for accelerating CNN convolution computation is to use parallelism to optimize convolution computation.

- (Zhihao Jia & Sina Lin et al. 2018) Proposed layer-wise parallelism allowing every layer to make utilization of individual parallelization. They allow optimization of each layer using graph search problem.
- (Minji Wang & Chien-Chin Huang et al. 2018) The existing systems like TensorFLow and MXNET focuses only one of the parallelization technique at one time, requiring large sized data set to scale.
- (Zhihao Jia & Matei Zaharia et al. 2019) Proposed extended search space of paralleization techniques for CNN's known as SOAP. SOAP consists of techniques that are responsible for parallelizing CNN in Sample, Operation, Attribute and parameter dimensions.

WHY CNN's

In CNN the weight sharing in CNN while performing convolution operation on input raw images. Which tremendously cut down parameter number in the whole network making network computationally less intensive. The other thing is dimensionality reduction because of introduction of poling layers in the network.

CNN components

This section introduces the components of CNN, including convolutional layers, pooling layers, and nonlinear continuous components. The main components involved in convolutional layers are input tensors and filter tensors Pool layer is responsible for mitigating the problem with output feature map's sensitiveness towards location of particular features present in the input. The Fully-connected layer is used mostly as the last layer of the network, and possess full connectivity with all the activation's of previous layer. Activation functions perform the job of decision making in neu- ral networks and helps in learning complex features from the input image.

Training of convolution neural networks

- Strassen algorithm: Strassen algorithm is used to determine how matrix multiplication is performed efficiently to reduce the number of addition and multiplication operations in comparison to conventional methods.
- Drawbacks: The method is suitable for large matrices but if the matrices are too much large then this method fails and results in degradation of the overall performance of the system.
- Winograd algorithm: Winograd minimal filtering algorithm that was given by Toom (Andrei, 1963) and cook (Cook, 1966) and generalization of this algorithm were done by Y. Xiong et al.
- Strassen-Winograd algorithm: Strassen-Winograd algorithm is the hybrid of a combination of the Strassen and Winograd algo- rithm to improve the computational complexity further.
- Fast Fourier Transform (FFT): In convolutional neural net- works two layers convo and FC layer make contributions to the network bottleneck. The earlier layer is computational intensive whereas the later layer is memory intensive.

SUMMARY

This article provides a detailed introduction to the concept of CNN and its development in recent years. It also discusses the components of CNN and

how parallelization can be used to accelerate the training process. Emphasis was placed on exploring advanced methods for improving computational speed without compromising accuracy in recent times.