Factorized Convolution Kernels for Image Processing

Alexander Bergman [awb@stanford.edu]

02/12/2018

1 Motivation & Background

Convolutional neural networks (CNNs) are a class of deep neural networks which have enjoyed success in learning tasks related to image analysis. However, with the increasing resolution of images and the increasing complexity of CNN models, the memory and computational costs of storing and using CNNs quickly can become intractable. The majority of these costs come from the convolutional layers of the CNN, which perform a convolution of an input image with a *n-dimensional* learned kernel.

Applications of CNNs implemented on embedded systems or systems with constrained hardware have limited computing and memory resources. Many computational imaging systems share this requirement, so it is of interest that image processing tasks implemented with CNNs also be computationally and memory efficient without sacrificing accuracy. More specifically, for applications of CNNs in LiDAR systems including object detection [5] and helping perform image processing pipeline steps such as denoising [6], it is necessary to optimize for reduced memory usage and computing time. Since a majority of the costs of using a CNN comes from the convolutional layers, a method of increasing the computing and memory efficiency of the convolution operation and kernel is especially important in these applications.

2 Related Work

One method for speeding up CNN computation has been in decreasing the computational complexity of convolution operations through convolution kernel factorization/decomposition. Factorization can be used to break a higher dimensional convolution into a sequence of lower dimensional convolutions which has a lower computational complexity and approximately the same result. For example, representing a convolutional kernel in a basis of rank-1 kernels as proposed in [3] is able to achieve a 2.5% speedup with no loss in accuracy. A similar low-rank based factorization method involved using the CP-decomposition (which is a generalization of the construction of low-rank approximations of matrices with the SVD to higher dimensional tensors) to decompose $4D$ convolutions into a sequence of four $2D$ convolutions, resulting in a $8.5x$ CPU speedup with only a 1% accuracy drop [4]. These methods both increase the speed of the convolution operation by reducing computational complexity, and reduce the number of parameters needed to be stored to represent convolution kernels, thereby reducing the amount of memory used by the CNN.

In [1] (MobileNets), and [2], a modified CNN architecture is proposed based on replacing all convolution operations in the convolutional layers with depthwise separable convolutions, which are convolution operations that can be broken down into a specific type of sequential lower dimensional convolutions. For example, in [1], depthwise separable convolutions turn N different $3D$ convolutions over M channels into M different $2D$ convolutions over each channel independently, and then N independent $1D$ convolutions over these M outputs. These separated convolutions can then be interpreted as separate, smaller layers in a modified CNN architecture.

This modified CNN architecture contains significantly less parameters and requires less operations in performing convolution, saving both memory and computing time. In [1], this architecture was evaluated quantitatively on a number of tasks in terms of accuracy, Mult-Adds (reflecting number of computations), and parameters (reflecting memory usage), showing improvement in efficiency and little decrease in accuracy.

3 Project Overview

It is of interest to quantify how the proposed different CNN kernel factorizations and modified CNN architectures perform on computational imaging tasks, such as those which use CNNs in the image processing pipeline. In this project, I will examine the performance of different kernel factorizations and corresponding modified CNN architectures on image denoising methods which use CNNs. Performance will be measured in terms of number of computations, memory usage, and accuracy. These metrics will be compared to state of the art image denoising results using CNNs, which show a tradeoff between efficient implementation of CNNs for image denoising and accuracy.

4 Milestones, Timeline, & Goals

The following list contains a timeline/general milestones that I think are important for this course project, and an approximate estimate for how long I expect them to take, although some steps may take longer or shorter than expected, and many can be worked on in parallel. The goal for this project is to have a quantitatively backed idea for the best way to build a memory and computationally efficient high-performance CNN, which can be used in computational imaging tasks such as denoising.

- Background reading: understanding of MobileNets [1] CNN architecture and depthwise separable convolutions, finding, understanding, and deciding upon which several different methods of kernel factorization to evaluate, understanding the use of convolutional neural nets in image denoising applications - 1 week
- Implementation and evaluation results of the MobileNets CNN architecture for both $2D$ and $3D$ convolutional layer operations on the image denoising task - 1.5 weeks
- Implementation and evaluation results of other kernel factorization methods such as rank-1 and CP decomposition (or others methods discovered and decided upon in the first week) on $2D$ and $3D$ convolutional layer operations with the same image denoising task. - 1.5 weeks
- Comparison and interpretation of results and developing report which addresses the ideal way to construct a memory and computationally efficient high-performance CNN and the trade-off involved - 1 week

References

[1] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv:1704.04861, 2017.

[2] M. Wang, B. Liu, and H. Foroosh. Factorized convolutional neural networks. arXiv preprint arXiv:1608.04337, 2016.

[3] Jaderberg, M., Vedaldi, A., Zisserman, A. "Speeding up convolutional neural networks with low rank expansions." arXiv:1405.3866, 2014.

[4] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky. Speeding-up convolutional neural networks using fine-tuned cp-decomposition. arXiv preprint arXiv:1412.6553, 2014.

[5] M. Szarvas, U. Sakai, and J. Ogata, "Real-time pedestrian detection using LIDAR and convolutional neural networks" in Proc. 2006 IEEE Intelligent Vehicles Symp., pp. 213-218

[6] Burger, H.C., Schuler, C.J., Harmeling, S., "Image denoising: Can plain neural networks compete with BM3D?" in CVPR 2012, pp. 2392-2399