Optical 4F Correlator for Acceleration of Convolutional Neural Networks

E.W.R. Schultz, J.V. de Nijs, B. Shi, R. Stabile

Institute for Photonic Integration, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands

Abstract *- Convolutional neural networks (CNNs) represent one of the most effective methods for image classification. The de-facto approach for performing the required 2D convolutions is to run an iterative algorithm consisting of point wise multiplication and kernel shifting on a graphical processing unit (GPU) or tensor processing unit (TPU). However, the computational complexity of this algorithm is $O(n^2k^2)$ for convolution of an $(n \times n)$ image and a $(k \times k)$ kernel, suggesting that 2D convolutions scale poorly for large matrices, leading to high power consumption and long execution times. A possible solution is a 4F optical correlator, which can, using Fourier optics, perform the convolutions in parallel and is not bound by conventional electronic limitations. In this paper we implement a 4F optical correlator using off-the-shelf components (Fig. 1) including spatial light modulators (SLMs) and a camera, while a PC is used to interact with the computing system. We experimentally demonstrate that a CNN utilizing such optical correlator has a best-case classification accuracy of 91% for the MNIST handwritten digit dataset and we show that the processing speed of the optical correlator can be in the same order of magnitude as a conventional GPU if maximum parallelism is exploited.*

Introduction

In the last decade, convolutional neural networks (CNN) have proven to be one of the most effective deep learning methods for image classification [1]. CNNs are deep neural networks that rely on at least one convolutional layer for the computations. When the dataset is a large 2D matrix, such as an image, these convolution computations are done in parallel on graphical processing units (GPUs) and tensor processing units (TPUs), which consist of hundreds to thousands of compute cores. However, the convolutional layers in CNN architectures are computationally complex: The computational complexity of the most used iterative convolution algorithm is $O(n^2k^2)$ for a convolution of an $(n \times n)$ image and a $(k \times k)$ kernel [2], which shows that the convolution algorithm scales poorly with higher input dimensions in terms of power usage and computation time.

A potential solution to improve computation speed and decrease power usage is to use a 4F optical correlator to perform the convolution operations by exploiting Fourier transforming properties of lenses, reducing the computational complexity of any matrix convolution from $O(n^2k^2)$ to $O(1)$ [3]. This means that, in theory, a convolution operation will always consume a constant amount of time and power, regardless of the sizes of the input and kernel matrix. The convolution theorem describes that convolution between an input image $f(x, y)$ and a kernel $k(x, y)$ in the spatial domain is equivalent to their product in the spectral domain:

$$h(x, y) = f(x, y) * k(x, y) = \mathcal{F}^{-1}\left(\mathcal{F}\big(f(x, y)\big) \cdot \mathcal{F}\big(k(x, y)\big)\right) \quad (11)$$

where $\mathcal{F}$ refers to the Fourier transform and $\mathcal{F}^{-1}$ refers to the inverse Fourier transform, thus the 4F correlator can perform 2D spatial convolutions optically. In this paper we

implement the 4F optical correlator to perform optical convolution for the MNIST dataset recognition problem.

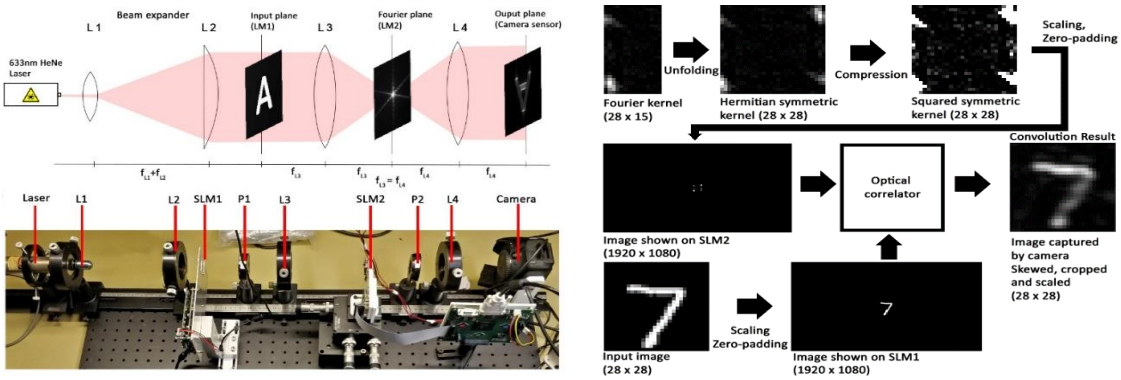

Fig. 1 – Left: The optical correlator. (Top) Schematic view of the 4F optical correlator. The character "A" is displayed as an example input image. (Bottom) Photo of the realized 4F optical correlator using off-the-shelf components. Polarizers P1 and P2 are added since liquid crystal on silicon (LCOS) SLMs are used as light modulators. Right: Procedure to perform inference on the optical correlator. All images shown in this figure are from the actual inference procedure on the optical correlator.

Methods

The 4F optical correlator is implemented as shown in Fig. 1, left side. A 633nm He-Ne laser beam is expanded by a Keplerian beam expander (lenses L1 and L2) to cover the active area of the first spatial light modulator (SLM1) where an image is encoded. The light then enters L3, after which the spatial Fourier transform appears at one focal distance behind the lens. At the Fourier plane, SLM2 with polarizer P2 is placed, acting as a programmable filter where during inference the Fourier transform of different kernels are displayed. The light then passes through L4 after which the inverse Fourier transform appears at the focal plane of L4, where a camera is placed to transform the image from the optical back to the electronic domain. To compare both accuracy and speed of a GPU-based CNN to the optical CNN, a simple CNN architecture (including Convolution, Batch Normalization, Max Pooling, Flatten, FC layer and Relu) was implemented using the open-source machine learning library PyTorch [4]. To train the CNN, a custom (electronic) Fourier convolution function was programmed which ensures that the kernels are initialized and trained in the Fourier domain [5], which saves computation time since, during inference, there is no more need for Fourier transformation of the kernel. Testing showed that this custom function had similar accuracy results in a CNN as PyTorch's built-in 2D convolution function. Furthermore, we train only the positive half of the kernel. Since the kernel must be Hermitian symmetric, we can unfold the kernel to get a square matrix which can be displayed on SLM2 (see Fig. 1, right side). To boost the amount of light passing through SLM2, an undesirable but necessary compression function is applied. Finally, both the input image and the squared symmetric kernel are scaled and zero-padded so that the images are displayed on the centers of the light modulators.

To evaluate the system accuracy, the MNIST dataset of handwritten digits [6] was used to train the CNN electronically using the custom Fourier convolution function. Then, with a subset of 300 images, inference was performed using the 4F correlator. Each input image was sequentially convolved with the 16 different Fourier kernels. The 16 resulting output images were skewed, cropped and scaled accordingly and then fed through the

fully connected (FC) layers of the CNN. The predicted labels were then compared to the correct labels of the input images to get the prediction accuracy.

Results

Fig. 2, left side, shows a comparison between the result of electronic convolution (using the custom Fourier convolution function) and convolution using the optical correlator. Both convolution methods highlight the same areas of the picture, which indicates that the optical correlator is working properly.

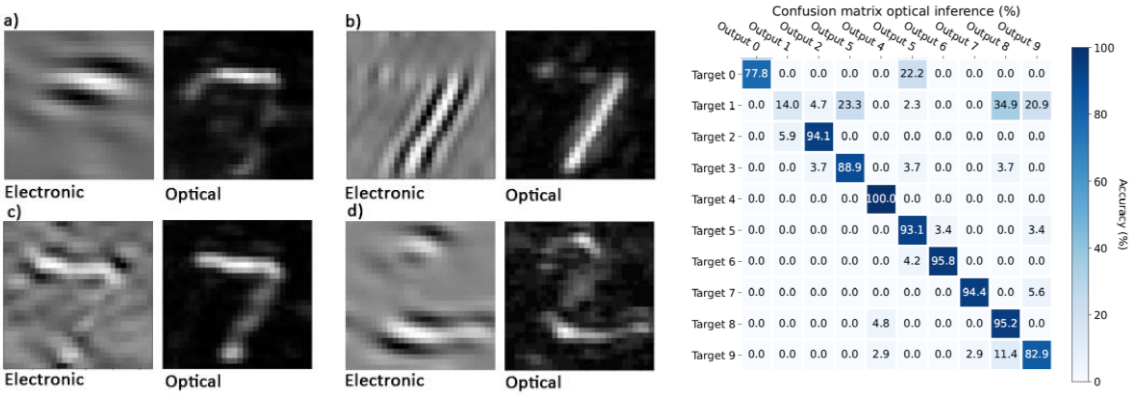

Fig. 2 – Left: The result of electronic Fourier-based convolution and the result of the convolution on the optical correlator. a), b), c) show the number 7 when different filters are applied. d) shows the number 2 with a filter applied. The overall tone of the convolution results of the 4F correlator is darker, most likely because the camera captures the irradiance rather than the amplitude of the light. Right: Confusion matrix of the inference on the optical correlator, showing accuracy. To limit the run time, a random subset of roughly 300 images was selected to be validated on the optical correlator. The overall accuracy is 81.01%.

Fig. 2, right side, shows the confusion matrix of the CNN inference with a subset of roughly 300 images of the MNIST dataset. The overall accuracy is 81.01% which is lower than the accuracy of an electronic CNN (98.2%). Subsequent tests using the same MNIST subset resulted in similar overall accuracies (±1%). The matrix shows that most digits were correctly identified with high accuracy. A clear exception is the digit 1, which only has an accuracy of 14.0%: This may be caused by misalignment of the camera which causes normally inactive neurons in the FC layers to be activated for this few-feature digit. Also the chance of FC layer overfitting is high, since they are only trained with ‘perfect’ electronic convolution. Furthermore, the lower overall accuracy might be caused by flickering, which is caused by the asynchronous refresh rate of the camera and SLMs. If we disregard the entries of the 1-row, assuming that these are inaccurate due to misalignment or flickering, the accuracy is 91.57%. This is similar to what other papers have presented [7]. Due to the lack of light intensity and to minimize flickering, the exposure time of the camera, and thus the convolution time of the correlator was fixed to a rather high 0.3s, but this time could be reduced significantly by using faster, more sensitive cameras. For reference, a single convolution operation on a Quadro M1200 GPU took $2.911 \cdot 10^{-4}$s, much faster than the correlator. Massive parallelism could be exploited by displaying a grid of images on SLM1. In total up to 1444 input images could be displayed on SLM1 and convolved with the same kernel simultaneously without increasing convolution time, greatly increasing throughput. Performing 1444 convolutions on a Quadro M1200 GPU takes 0.1328s, which is in the same order of magnitude as the 0.3s convolution time of the setup. The correlator could be orders of magnitude faster if a faster camera is used.

Discussion

Only a small subset of roughly 300 images was used for inference due to the low throughput of the setup, limited mostly by the exposure time of the camera. To increase throughput the camera should be replaced with a proper high-speed camera and the light modulators could be replaced with high-speed DMDs which have a much higher refresh rate. Furthermore, it would be interesting to see what accuracy could be achieved while exploiting maximum parallelism by displaying a grid of input images on SLM1. Additionally, the accuracy that was achieved in this paper is likely lower than the electronic accuracy because of misalignment, flickering and optical aberration. In future work, after electronic training, the FC layers could be trained with the outputs of the optical correlator so that the network can adjust to slight misalignments in the optical correlator. Flickering can be countered in future work by utilizing a camera and SLMs which support generator locking (Genlock). Finally, future work should take into account the diffraction orders of the used light modulators when choosing their lenses, since the diffraction orders can result in unwanted interference for large input matrices.

Conclusions

This paper experimentally demonstrated the implementation of a 4F amplitude only optical correlator which can perform convolutions optically. A CNN, that uses the optical correlator for its convolutional layer, has classified a subset of the MNIST dataset with an accuracy of 81.01% (or 91.57% excluding the '1'-row). The main sources of error are likely misalignment, flickering and optical aberration. The speed of the optical correlator for single MNIST convolutions is much lower than the M1200 GPU, but if maximum parallelism were to be exploited the speed of the optical correlator is in the same order of magnitude as the speed of the Quadro M1200 GPU. Considering that there are much, much faster cameras and light modulators available on the market today, it is very likely that an optical correlator such as the one presented in this paper can greatly surpass the processing speed of a conventional GPU.

References

[1] A. Bhandare, M. Bhide, P. Gokhale, and R. Chandavarkar, "Applications of Convolutional Neural Networks," International Journal of Computer Science and Information Technologies, vol. 7, no. 5, pp. 2206–2215, 2016.

[2] M. Miscuglio, Z. Hu, S. Li, J. George, R. Capanna, P. M. Bardet, P. Gupta, and V. J. Sorger, "Massively parallel amplitude-only fourier neural network," arXiv, no. December, 2020.

[3] E. Cottle, F. Michel, J. Wilson, N. New, and I. Kundu, "Optical Convolutional Neural Networks – Combining Silicon Photonics and Fourier Optics for Computer Vision," 2020.

[4] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. K¨opf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in Advances in Neural Information Processing Systems, vol. 32. Neural information processing systems foundation, 12 2019.

[5] H. Pratt, B. Williams, F. Coenen, and Y. Zheng, "FCNN: Fourier Convolutional Neural Networks," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10534 LNAI, 2017, pp. 786–798.

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2323, 1998.

[7] M. Miscuglio, Z. Hu, S. Li, J. George, R. Capanna, P. M. Bardet, P. Gupta, and V. J. Sorger, "Massively parallel amplitude-only fourier neural network," arXiv, no. December, 2020.