CleverName: A Zero-Power All-Optical Convolutional Neural Network

Author Author
Stanford University

Author Author
Stanford University

Author Author
Stanford University

ABSTRACT

Abstract here.

ACM Reference Format:
Author Author, Author Author, and Author Author. 2018. CleverName: A Zero-Power All-Optical Convolutional Neural Network. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Deep neural networks have found success in a wide variety of applications, ranging from computer vision to natural language processing to game playing [7]. Since the explosion of interest following the achievements of convolutional neural networks on ImageNet classification, deep learning has transformed algorithms in both academic and commercial environments. While accuracy has improved to a remarkable level, the number of parameters and connections has grown dramatically, and the power requirements to build and use these networks have increased correspondingly.

While the training phase of learning parameter weights is often considered the slow stage, large models also demand significant energy during inference due to millions of repeated memory references. For example, AlphaGo has a power consumption of approximately 300 W. To this end, there is a large effort to develop new software methods and specialty hardware for improved efficiency. Algorithms for improved efficiency include pruning, quantization, low rank, parallelization, mixed precision, and model distillation. Compressed models have demonstrated preserved accuracy with much fewer parameters [3, 5]. On the hardware front, there are now specialized processing units for deep learning, such as TrueNorth, Movidius's USB-based neural compute stick (NCS), and Google's tensor processing unit (TPU). Despite all these efforts, it remains difficult for embedded systems such as mobile vision, autonomous vehicles/robots, and wireless smart sensors to deploy CNNs due to stringent constraints on power and bandwidth.

Optical computing has been tantalizing for its high bandwidth and inherently parallel processing. If we can come up with scalable optical configurations that together act as a framework for an optical CNN, this would be of interest to computer vision, robotics, machine learning, and optics communities. Optical neural networks (ONNs) could also potentially exploit wave optics for complex-valued neural networks and new types of non-linearities that are currently unavailable to digital computation. Linear transformations can be performed with lens systems or interferometer meshes [16]. Optical nonlinearities include saturable absorption and bistability. Liquid crystal light valves have been used as optical thresholding devices [15]. Convolutions are commonly performed with PSF engineering. We use these to design a model for cascaded parallel convolutions with sandwiched nonlinearity layers.

In this paper, we follow the vein of computational photography to create we zero-power, all-optical convolutional neural network for image classification. We choose a simple classification task, e.g. classify handwritten digits, and build a prototype that performs inference on projected images. We precompute the weights by training with the MNIST dataset on a computer, then fabricate the optical elements accordingly. We compare performance with the same inference performed on the computer. Here we demonstrate proof-of-concept with bulk optics and free-space propagation, but we recommend photonic integrated circuits for scalability. Photonic circuits with up to 4,096 optical devices have been demonstrated [17], and there have also been new three-dimensional photonic integrations that could enable larger networks [13]. Combination of these next-generation large-scale photonic circuits with compressed deep learning models could provide a potential route for high performance ONNs.

To summarize, we make the following contributions:

- We propose an optical toolbox of building blocks for convolutional neural networks.
- We build a zero-power, all-optical two-layer network with precomputed weights for image classification.
- We evaluate against the computer implementation of the same network and show that we achieve similar accuracy.

Overview of limitations. We are limited to non-negative values since we are working with light intensities. This may be avoided with coherent processing. We also hope to proceed without normalization. We focus here on inference and do training on the computer. In our current implementation without active elements, we lose flexibility to update the network. However, in fixed applications, this is not a problem. Photonic circuits remain expensive to fabricate as they are not used in common consumer applications.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

2 RELATED WORKS

CNNs and architecture variations. Artificial neural networks were proposed in X. Early networks were composed of fully connected layers with nonlinear activation functions in between, inspired by the canonical biological neuron and its thresholded activation. Convolutional layers were popularized by LeCun and ... in image classification CITE. Convolutional layers allow for weight sharing... Since then, deeper, more complex, etc. Since an optical implementation of a CNN comes with certain constraints and challenges, we wanted to see what types of CNNs have been explored with non-standard architectures that may align with physical designs. Omission of fully connected layers, i.e. fully convolutional with global average pooling at the top layer has proven to be successful in [5, 8]. Analysis of CNN operations in the Fourier domain, introducing spectral pooling and regularization [14]. Relevant because we can also access optical Fourier plane. We also note the work in the complex-valued deep neural networks [18], as coherent optical signals may be an effective means of propagating complex-valued data.

Optical computing. High bandwidth, but high cost. Optoelectronics and fully optical. Optical solutions to NP-complete problems that are faster than electronic computation [19]. In the early days of CNNs, there was also momentum for optical implementation, optical neural networks (ONNs). Adaptive optical network using volume holographic interconnects in photorefractive crystals [11]. Hybrid optoelectronic network with feedback loop, computer for subtraction and thresholding operations [9]. Optical thresholding perceptron implemented with liquid crystal light valves (LCLV) [15]. There has also been much development in photonic computing. Recently, two-layer fully connected NN demonstrated with intermediate simulated nonlinearity units on 1D data [16]. However, this required photodetection and reinjection, and it did not involve convolutional layers. Optalysys? We do not work with photonic circuits here, but we think they may be worth exploring for larger networks.

Computational cameras. Computational photography has some intersection with optical computing in that they may perform some operations on the input signal optically, but they are also distinct in that they work with spatially organized inputs that come from physical world (incoherent light). Coded apertures and PSF engineering can perform filtering [CITE]. Optical correlators that essentially perform template matching on images have been explored for optical target detection and tracking [6, 10]. Somewhat similar to our goal is focal plane processing, which refers to the incorporation of image processing on the sensor chip, eliminating or reducing the need to shuttle full image data to a processor. These chips have been designed to detect edges and orientations and to perform wavelet or discrete cosine transforms [2] [RedEye]. Most of these approaches still rely on electronic computation on the image sensor chip, whereas our goal is all-optical implementation with no additional power input. Chen et al. use optically designed angle sensitive pixels, photodiodes with integrated diffraction gratings producing Gabor wavelet impulse responses, to approximate the kernels of the first layer of a typical convolutional neural network [1]. However, this design is limited to a fixed set of convolution kernels, and the output still has to be shuttled to a computer for further processing. Our goal is to build an end-to-end classification system with flexible and rearrangeable optical units that allows for custom optical CNNs.

3 ONN TOOLBOX

Here we describe proposed optical building blocks corresponding to common layers in a CNN. In a standard feed-forward CNN, information is passed in a single direction through a sequence of layers. Cycles, loops, interacting networks, etc. can be incorporated in more complicated architectures that may be interesting to explore in the future. For now, we will focus on the most essential components that define a CNN in the context of an image classification task.

3.1 Convolutional layer

A CNN typically begins with a convolutional layer, which essentially performs pattern matching with a set of learnable visual filters. A standard convolutional layer takes an input volume of depth C_{in}, performs a series of correlations with a set of c_{out} kernels each with depth C_{in}, and outputs a new volume of depth C_{out}. The correlation of the kernel across the width and height of the input volume produces a 2D "activation map", and stacking the C_{out} activation maps for all kernels forms the output volume of depth C_{out}. Hyperparameters include the spatial extent of the kernel F, the stride with which the kernel is applied, and the padding of the input volume. Here we assume a stride of 1, meaning the kernel is shifted by one pixel at a time, and zero-padding such that the output volume has the same height and width as the input.

3.1.1 Tiled PSFs. In optical systems, image formation is often modeled as a spatially invariant convolution of the scene with the point spread function (PSF) of the system:

$$I_{\text{out}} = I_{\text{in}} * \text{PSF} \tag{1}$$

Let us generalize this scene as an input I that can also be a real image relayed by a lens. This simple case can be viewed as a convolutional layer with $C_{\text{in}} = C_{\text{out}} = 1$ and the flipped PSF as the single kernel. We will also refer to the flipped PSF as the kernel since the flipping is trivial. Now suppose we want $C_{\text{out}} = n, n > 1$. By spatially tiling the multiple kernels as the PSF of the system, the output becomes the convolution of the input image with multiple kernels, but now the n outputs are tiled laterally instead of stacked in depth. Consideration can be taken to ensure these outputs are non-overlapping by adjusting the shifts Δx, if desired.

$$\text{PSF} = \sum_{i=1}^{n} W_i * \delta(x - i\Delta x) \tag{2}$$

$$I_{\text{out}} = I_{\text{in}} * PSF = \sum (I_{\text{in}} * W_i) * \delta(x - i\Delta x) \tag{3}$$

The next important extension is to incorporate $C_{\text{in}} = m, m > 1$. If we needed to exactly imitate the digital CNN, we would need m different kernels for each of the m input channels. This could potentially be implemented with many of the single channel modules in parallel, with the addition of a relay that sums m outputs that correspond to the different depth slices of the same kernel, but this type of setup may be prohibitively complicated to build. If we slightly relax our requirements, we could again rely on Fourier optics to perform the summation. Now suppose we tile the input images in addition to the kernels:

$$I = \sum_{j=1}^{m}$$

This combination of tiled images and tiled PSFs results in some cycling of the kernels, but .

3.1.2 Optimized phase masks. Instead of first optimizing the PSFs and then separately optimizing a phase mask to best produce these PSFs, end-to-end optimization. When optimizing for the phase mask as a whole, we realized we no longer need to think about tiling many small PSFs, but rather just optimize for one large PSF.

3.2 Nonlinear activation layer

Nonlinear activation layers are crucial components in the neural network toolbox that allow for modeling of nonlinear relationships between input and output variables. Most commonly used is a rectified linear unit (ReLU), that simply sets all negative values to 0: $\text{ReLU}(x) = \max\{0, x\}$. In an optical intensity-based system, there are no non-negative values, so the standard ReLU function does not directly apply. However, if we consider the purpose of the ReLU layer to zero out some fraction of the neurons below a threshold response level, then we hypothesize that we can accomplish a similar effect by shifting this threshold to a positive value.

This nonlinear behavior translates to an ideal optical element that is fully opaque when incident light is low intensity and fully transmissive when incident light is above a threshold. A perfectly binary switch is difficult to physically realize, so instead we sought a material that would be less transmissive to lower incident intensities and become more transmissive at higher incident intensities. In fact, this type of nonlinear response is remniscent of the PReLU) [4]. (Swish) [12]

Bacteriorhodopsin

3.3 Fully-connected layer

The fully connected layer is so named because every input neuron is connected to every output neuron. The input is flattened into a single vector and multiplied with a matrix of size $D_{\text{out}} \times D_{\text{in}}$, where $D_{\text{in}} = H \times W \times C_{\text{in}}$.

Spatially-varying convolution with spatial extend equal to the size of the . Maybe not necessary as global average pooling (GAP) has been successful.

3.4 Pooling layer

Pooling layers can be inserted, commonly between convolutional layers, to reduce spatial size and consequently computation. Pooling operations, for example "maximum", operate on each depth slice independently. The same hyperparameters of spatial extent F and stride S also apply, though the most commonly seen pattern is $F = 2, S = 2$.

3.4.1 Average pooling. While it is not obvious how to take the spatial maximum of an optical signal without active sensing, average pooling can be approximated with a reduction in the spatial resolution of the

3.4.2 Spectral pooling. Spectral pooling is another interesting concept that carries over easily to our ONN setup. can be viewed as a generalization of average pooling.

3.5 Other considerations

We have designed these optical building blocks to have input and output in the same format. This allows an arbitrary chaining of blocks to create the desired CNN architecture. Not all of these designs are easily scalable to the same sizes. All weights will be non-negative. It is possible to include negative values when coherent signals are used, but we do not explore that here. We will evaluate the implications of our system constraints in the next section.

4 UNDERSTANDING ONNS

Here we aim to understand the effect of imposing the previously mentioned constraints on the performance of a neural network. We start with a standard network and add in constraints of nonnegative weights, missing or nonnegative bias, limited set of nonlinear activation functions, etc. We set up these models in TensorFlow. Non-negativity constraints included by XXXXX. Biases simply removed from each layer. ReLU replaced by the model of XXXXXX. N iterations of XXXX optimizer (learning rate = X).

Case 1: Fully connected NN for 2D classification. Classification of 2D points.

$$C_{5\times5}^{32} \rightarrow C_{5\times5}^{64} \rightarrow FC \rightarrow \text{Softmax} \tag{4}$$

Case 1: CNN for image classification. To compare performance with modified convolutional layers. This model can be larger. Image classification of MNIST or CIFAR. Network architecture is:

Original error rate is 0.01%.

Constraints	1	2	3	4	5	6	7	8
Non-neg.		✓	✓	✓	✓	✓		
No biases			✓	✓	✓	✓		
Non-ReLU				✓	✓	✓		
Spectral pooling				✓	✓	✓		
Cycled kernels					✓	✓		
...						✓		
1000 epochs	.1%							
10000 epochs	.5%							

Table 1: MNIST classification error rate with various NN constraints.

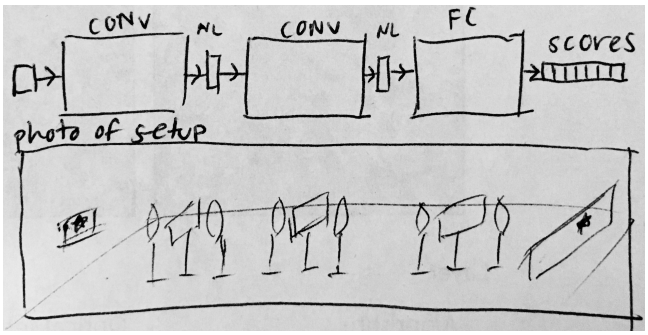

Figure 1: setup

5 IMPLEMENTATION

5.1 Network architecture

For our prototype, we limit to two convolutional layers and one fully connected layer. We train a small n-layer network to perform classification.

5.2 Optical protoype

We fabricate phase masks / print amplitude masks. Build optical setup on benchtop. Project images for inference.

6 RESULTS

These are the results captured with the prototype.

7 DISCUSSION

- brief summary

- extended discussion of coherent vs incoherent light, how to train optically, and other aspects that may be interesting (this discussion could also include the limitations)

- future work

8 CONCLUSION

We hope this will inspire more research in the area.

REFERENCES

[1] Huaijin G Chen, Suren Jayasuriya, Jiyue Yang, Judy Stephen, Sriram Sivaramakrishnan, Ashok Veeraraghavan, and Alyosha Molnar. 2016. ASP vision: Optically computing the first layer of convolutional neural networks using angle sensitive pixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 903–912.

[2] Viktor Gruev and Ralph Etienne-Cummings. 2002. Implementation of steerable spatiotemporal image filters on the focal plane. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 49, 4 (2002), 233–244.

[3] Song Han, Huizi Mao, and William J Dally. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149* (2015).

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*. 1026–1034.

[5] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size. *arXiv preprint arXiv:1602.07360* (2016).

[6] Bahram Javidi, Jian Li, and Qing Tang. 1995. Optical implementation of neural networks for face recognition by the use of nonlinear joint transform correlators. *Applied optics* 34, 20 (1995), 3950–3962.

[7] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[8] Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network. *arXiv preprint arXiv:1312.4400* (2013).

[9] Taiwei Lu, Shudong Wu, Xin Xu, and TS Francis. 1989. Two-dimensional programmable optical neural network. *Applied optics* 28, 22 (1989), 4908–4913.

[10] Tariq Manzur, John Zeller, and Steve Serati. 2012. Optical correlator based target detection, recognition, classification, and tracking. *Applied optics* 51, 21 (2012), 4976–4983.

[11] Demetri Psaltis, David Brady, and Kelvin Wagner. 1988. Adaptive optical networks using photorefractive crystals. *Applied Optics* 27, 9 (1988), 1752–1759.

[12] Prajit Ramachandran, Barret Zoph, and Quoc Le. 2017. Searching for activation functions. (2017).

[13] Mikael C Rechtsman, Julia M Zeuner, Yonatan Plotnik, Yaakov Lumer, M Segev, and A Szameit. 2013. Photonic Floquet topological insulators. In *Lasers and Electro-Optics (CLEO), 2013 Conference on*. IEEE, 1–2.

[14] Oren Rippel, Jasper Snoek, and Ryan P Adams. 2015. Spectral representations for convolutional neural networks. In *Advances in Neural Information Processing Systems*. 2449–2457.

[15] Indu Saxena and Emile Fiesler. 1995. Adaptive multilayer optical neural network with optical thresholding. *Optical Engineering* 34, 8 (1995), 2435–2440.

[16] Yichen Shen, Nicholas C Harris, Scott Skirlo, Dirk Englund, and Marin Soljačić. 2017. Deep learning with coherent nanophotonic circuits. In *Photonics Society Summer Topical Meeting Series (SUM), 2017 IEEE*. IEEE, 189–190.

[17] Jie Sun, Erman Timurdogan, Ami Yaacobi, Ehsan Shah Hosseini, and Michael R Watts. 2013. Large-scale nanophotonic phased array. *Nature* 493, 7431 (2013), 195–199.

[18] Chiheb Trabelsi, Olexa Bilaniuk, Dmitriy Serdyuk, Sandeep Subramanian, João Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J Pal. 2017. Deep Complex Networks. *arXiv preprint arXiv:1705.09792* (2017).

[19] Kan Wu, Javier García De Abajo, Cesare Soci, Perry Ping Shum, and Nikolay I Zheludev. 2014. An optical fiber network oracle for NP-complete problems. *Light: Science & Applications* 3, 2 (2014), e147.