A scalable optical neural network architecture using coherent detection

Sludds, Alexander, Bernstein, Liane, Hamerly, Ryan, Soljacic, Marin, Englund, Dirk

Alexander Sludds, Liane Bernstein, Ryan Hamerly, Marin Soljacic, Dirk Englund, "A scalable optical neural network architecture using coherent detection," Proc. SPIE 11299, AI and Optical Data Sciences, 112990H (24 February 2020); doi: 10.1117/12.2546940

Event: SPIE OPTO, 2020, San Francisco, California, United States

A Scalable Optical Neural Network Architecture Using Coherent Detection

Alexander Sludds[a], Liane Bernstein[a], Ryan Hamerly[a], Marin Soljacic[a], and Dirk Englund[a]

[a]Research Lab for Electronics, MIT, 50 Vassar Street, Cambridge, MA, 02139, USA

ABSTRACT

Keywords: Optical neural network, Optical information processing, Machine learning, Hardware accelerator, Coherent computing, Photonics Accelerators

Storing, processing, and learning from data is a central task in both industrial practice and modern science. Recent advances in modern statistical learning, particularly Deep Neural Networks (DNNs), have given record breaking performance on tasks in game playing,[1,2] natural language processing,[3] computer vision,[4] computational biology,[5,6] and many others. The rapid growth of the field has been driven by an increase in the amount of public datasets,[7] improvements to algorithms,[8] and a substantial growth in computing power.[9] In order to perform well on these tasks networks have had to grow in size, learning more complicated statistical features. The training and deployment of these large neural networks has spurred the creation of many neural network accelerators to aid in the computation of these networks.[10–12]

Existing general purpose computing devices such as CPUs and GPUs are limited both by thermal dissipation per unit area and yield associated with large chips.[13,14] The design of Application Specific Integrated circuits (ASICs) has aided in decreasing the energy consumption per workload substantially by limiting the supported operations on chip. An example of this is the first generation tensor processing unit (TPU)[15] which is able to perform the inference of large convolutional neural networks in datacenter in $< 10ms$ with an idle power of 28W and an workload power of 40W. It may seen counterintuitive then that the limiting factor for the implementation of DNNs is not computation, but rather the energy and bandwidth associated with reading and writing data from memory as well as the energy cost of moving data inside of the ASIC.[15,16] Several emerging technologies, such as in-memory computing,[17] memristive crossbar arrays[18] promise increased performance, but these emerging architectures suffer from calibration issues and limited accuracy.[19]

Photonics as a field has had tremendous success in improving the energy efficiency of data interconnects.[20] This has motivated the creation of optical neural networks (ONNs) based on 3D-printed diffractive elements,[21] spiking neural networks utilizing ring-resonators,[22] reservoir computing[23] and nanophotonic circuits.[24] However, these architectures have several issues. 3D-printed diffractive networks and schemes requiring spatial light modulators are non-programmable, meaning that they are unable to perform the task of training. Nanophotonic circuits allow for an $O(N^2)$ array of interferometers to be programmed, providing passive matrix-vector multiplication. However, the large ($\approx 1mm^2$) size of on chip electro-optic interferometers means that scaling to an array of $100x100$ would require $10,000mm^2$ of silicon, demonstrating the limitations of scaling this architecture. To date no architecture has demonstrated high-speed (GHz) speed computation with more than $N \geq 10,000$ neurons.

Here we present an architecture that is scalable to $N \geq 10^6$ neurons. The key mechanism of this architecture is balanced homodyne detection. By scaling the architecture to such a large size we show that we can decimate energy costs per operation associated with the optical component of this architecture, reaching a bound set by shot noise on the receiving photodetectors which leads to classification error. We call this bound a standard quantum limit (SQL) which reaches 100zJ/MAC on problems such as MNIST. We also analyze the energy consumption using existing technologies and show that sub-fJ/MAC energy consumption should be possible.

Further author information:
Alexander Sludds: E-mail: asludds@mit.edu

AI and Optical Data Sciences, edited by Bahram Jalali, Ken-ichi Kitayama,
Proc. of SPIE Vol. 11299, 112990H · © 2020 SPIE · CCC
code: 0277-786X/20/$21 · doi: 10.1117/12.2546940

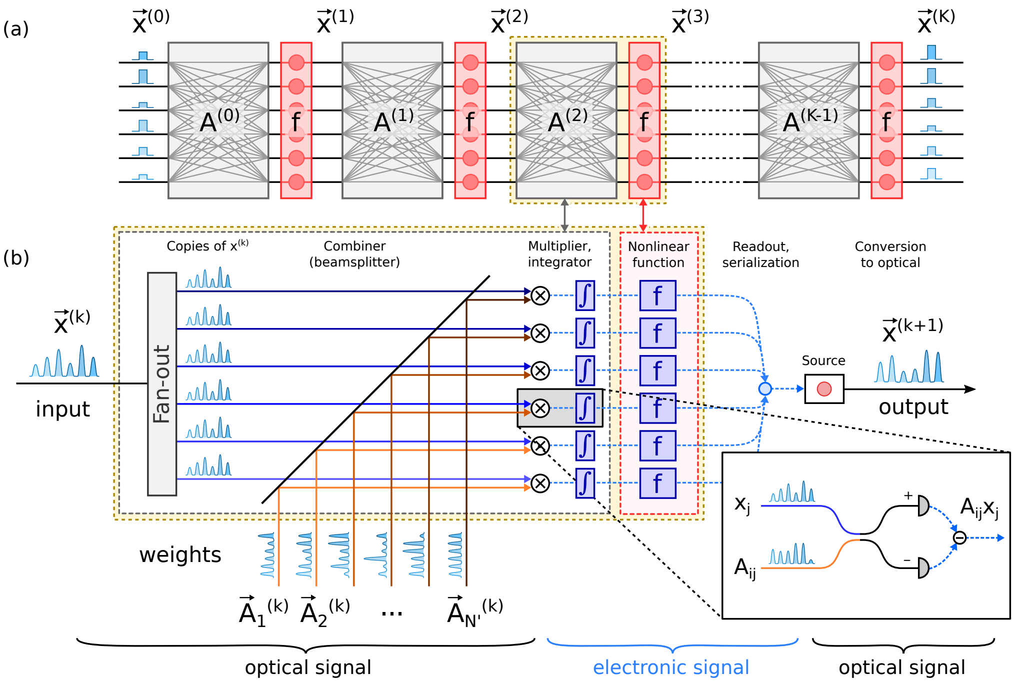


Figure 1. (a) A vector $\vec{x}^{(0)}$ encoding an input to a neural network model passes into the first layer of weighting values $A^{(0)}$ which is a matrix. A non-linear activation function is then applied to the output vector resulting from the matrix-vector product generating the new vector $\vec{x}^{(1)}$. (b) Here how homodyne detection can be utilized for computation is shown. A vector $\vec{x}^{(k)}$ is passively fanned-out. The rows of the associated weight matrix $A^{(k)}$ are transmitted simultaneously in time. These two signals interfere on a balanced homodyne detector which generates charge proportional to the product of the incident electric field strengths. This charge is accumulated and read out to a processing element which can perform the non-linear activation function. Finally, data is serialized for computation in the next layer of the neural network.

This paper is organized as follows: In section 1 we will discuss the function of this architecture as a matrix-matrix processor. In section 2 we will analyze the energy consumption of the architecture. In section 3 we will discuss methods for training and extending the accelerator to a broader scope of problems, namely convolutionally neural networks (CNNs).

Coherent Matrix Multiplication

Here Figure 1 demonstrates the architecture. Deep neural network shown here is a sequence of K layers. Each layer is a matrix-multiplication of a weighting matrix A^k and an input vector $\vec{x}^k$ and, after an element-wise nonlinear function f, returns the input to the next layer $\vec{x}^{k+1}$. Specifically, in terms of the individual vector components we see that

$$x_i^{k+1} = f\left(\sum_i A_{ij}^k x_i^k\right)$$

These nonlinear functions are typically the Rectified Linear Unit (ReLU) function for fully-connected and convolutional neural networks and sigmoid/tanh for RNN problems.[16] The computation of these nonlinear functions, especially for ReLU, is very easy and represents less than a percent of the total computational workload in most CMOS based systems.

For a given layer, let N and N' be the number of input and output neurons. Data is encoded as a sequence of time-multiplexed pulses where each pulse is on it's own seperate channel. Each row of the weighting matrix $\vec{A_i}^k$ is encoded in time on a separate channel. Over N timesteps the data from the weight matrix is transmitted, meaning that on each timestep the j^{th} column of the weight matrix is sent. Data from the inputs is fanned out (passively repeated) onto an array of coherent detectors. Each detector, shown in the Figure 1 inset operate as a quantum photoelectric multiplier. To see why they operate as such, first we calculate the homodyne product of the two signals incident on the detector:

$$Q_i = \frac{2\eta e}{\hbar\omega}\int \mathrm{Re}[\mathrm{E^{(in)}(t) \times E_i^{(wt)}(t)]dt} \propto \sum_{\mathrm{j}} \mathrm{A_{ij}x_j}$$

Here $E_i^{(in)}, E_i^{(wt)}$ are the magnitude of the input and weight electric field respectively for the i^{th} receiver. If these are sequences proportional to x_j and A_{ij} then the accumulated amount of charge on the i^{th} integrator is proportional to the vector-vector inner product. When we consider all N' outputs simultaneously the computed result is the matrix-vector product! The product is now in the electronic domain, stored as charge. The data is then passed to an element-wise nonlinear function before being serialized and passed to another coherent source for further processing (either another layer or the final steps of classification).

As a device, balanced homodyne detectors offer very promising properties for photoelectric multiplication. First, the upper bound on bandwidth for balanced homodyne detectors is only limited by the bandwidth of beamsplitters and photodetectors, which are both larger than THz. The bandwidth of the integrator can be much slower, since charge only needs to be read out after the completion of each layer. This architecture also avoids major hurdles from all-optical computing associated with the need for low-power nonlinear optical devices. These photodetector based devices are readily scalable in existing CMOS processes to millions of pixels and can be combined in the same process with electronic logic.

Energy Consumption

Standard Quantum Limit

Modern high performance computing is limited by a fundamental energy problem, where general purpose computers operate at a reduced clock so that chips don't exceed a fixed power dissipation budget.[14] Here, we will consider the long-term limits on scalability of the architecture as well as the near-term estimates for the energy consumption per operation.

Fundamental limits to energy consumption stem from quantum limited noise associated with the shot noise of a receiver. Considering the quantization of electrical and optical signals we see that $E_{optical} = \frac{h}{\tau_{photo}}$ and $E_{electrical} = \frac{h}{\tau_{el}}$. τ_{el}, the electrical signal duration, is on the order of 100ps and τ_{photo}, the optical signal duration is $\frac{\lambda}{c}$ which is femtosecond scale. This means that the quantized energy for the optical portion of the system is 10^4 orders of magnitude larger than the electrical portion. As a result, we can view the electrical signal as classical and limited by thermal noise. The optical signal, however, is limited by vacuum fluctuations, which produces a Poisson distributed photocurrent on the photodetector.[25] The photocurrents are substracted in the homodyne detectors, meaning that the standard deviations of the fluctuations adds in quadrature. This leads to the following expression for shot-noise limited performance of an optical neural network (the derivation of this equation can be found in the supplemental section of the original PRX publication:[26]

$$x_i^{(k+1)} = f\left(\sum_j A_{ij}^{(k)} x_j^{(k)} + w_i^{(k)} \frac{|A^{(k)}||x^{(k)}|}{\sqrt{NN'n_{\mathrm{mac}}}}\right)$$

Here $w_i^{(k)}$ is a zero-mean, unity variance Gaussian random variable and the norm taken here is the L^2 norm. n_{mac}. A quick sanity check of this equation says that noise in this equation scales as $n_{mac}^{-1/2}$, leading to a signal to noise ratio scaling as $\mathrm{SNR} \propto \mathrm{n_{mac}}$. This signal to noise ratio places a limit on how low the number of photons per mac can go, limited by how much it changes the accuracy of the model it is running.

A simulation is performed of the effects of shot noise on the accuracy of the MNIST dataset. Each layer is implemented using equation . The results are shown in Figure 2.

A counter-intuitive part of this plot is that the energy consumption per operation is able to fall below the Landaur limit[27, 28] of non-reversible computation. At first glance this may seem counter-intuitive, implying that

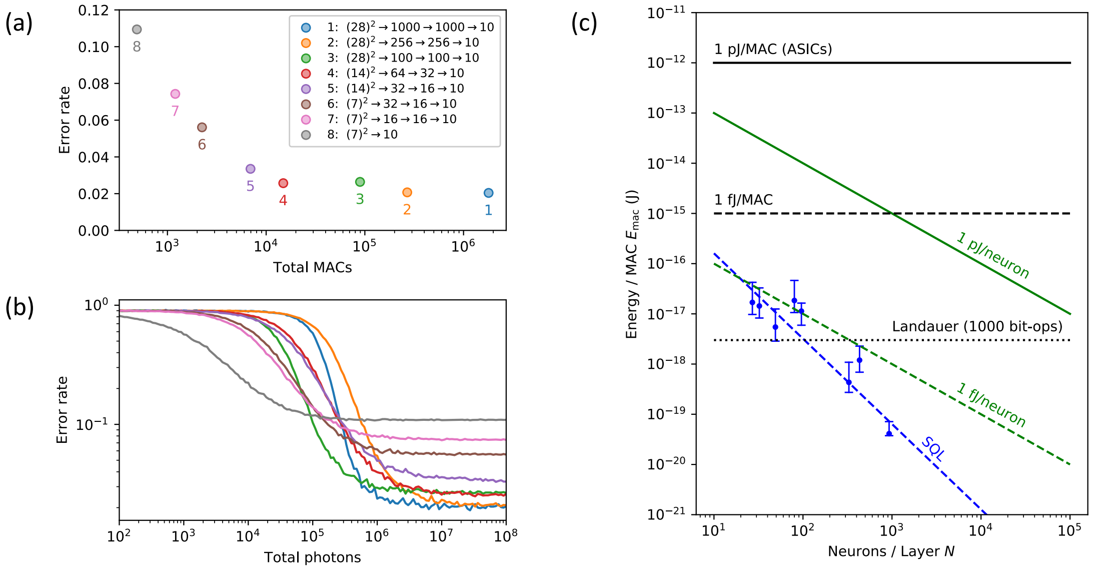


Figure 2. (a) A plot of the original model accuracy of several different layer configurations for the MNIST dataset. As the size of the networks increase here (modeled as the total number of MACs), they learn more complex features and their error drops. (b) The simulation results of the effect of shot-noise on the overall network accuracy for each of these models. (c) The smalling cost of the architecture when compared to several benchmarks. Shown here is the 1pJ/neuron plot, representing an energy consumption feasible in the near term with existing technology. 1fJ/neuron is also shown, represeting future scaling of the technology. The SQL is also plotted, represented here by dots and error bars, where the dots represent the energy per MAC which results from a 1.5 times increase in the model error. The error bars span the range of 1.2 times to 2.0 times increase in model error. The SQL is calculated assuming 1550nm light.

there is some thermodynamic bound not accounted for in the architecture. However, the reason the cost per operation can fall below the Landaur limit is because the process of photoelectric multiplication is a reversible computation (light interferes on a beamsplitter, which is reversible, and then charge accumulates on photodetectors, which is also a reversible process). The non-reversible process comes from the readout of accumulated charge using an ADC and TIA. It is not to say that there does not exist a Landaur limit for this system, but rather than thermodynamic bound for this system is lower than the Landaur limit for conventional computing. In fact, it is a factor of N lower since an ADC read now encompasses N computations, leading to an effective thermodynamic limit of $\frac{kTln(2)}{N}$.

Electronic Energy Consumption

Light by itself cannot best the energy consumption of a custom CMOS ASIC. In particular, many peripherals are required in order to enable computation with light. In particular this system requires $N + N'$ high speed DAC's, to encode data from memory into the amplitude of light. An addition, a configuration with ADCs will be required which enables the readout of the accumulated charge. However, the energy consumption of these ADCs per receiver array readout is independent of the number of ADCs, since we have a fixed number of readouts we wish to do. Since these readouts can be done slowly we leave the design of this ADC readout system as a free parameter, since the energy will be constant anyways, and the tradeoff being made is between adding more area and reducing the readout bandwidth. Each pixel may contain a transimpedance amplifier, leading to $O(N^2)$ transimpedance amplifiers. However, these circuits are very scalable as they are replicated next to the detector in CMOS.

In the near term, the energy consumption of these devices is roughly: ADC's: $1\frac{\text{pJ}}{sample}$.[29]
TIA's: $100\frac{fJ}{sample}$.[30]
DAC's $1\frac{fJ}{sample}$ since many ADCs are implemented as high speed DACs using successive approximation.

Modulators: $1\frac{pJ}{sample}$.[31]
Addition in 45nm node (as a proxy for nonlinearity): $30fJ$ implying 7nm node energy is $\frac{30fJ}{2^5} = 1fJ$[14]

Each of these costs has a different scale associated with it for an $(N, N)X(N, N)$ matrix multiplication. In particular, we need to do N^2 ADC reads, N^2 uses of a TIA, $2N^2$ uses of a DAC, $2N^2$ uses of a modulator, and N^2 uses of simple CMOS logic for a nonlinear function. However, the number of operations performed by all of this hardware is N^3 multiply and accumulate operations. This means that the total cost of all of this hardware can be decimated by increasing the size of the associated hardware. Further advances may leads to decreases in the energy consumption of each of these components.

The benchmark to beat is a modern day custom ASIC (shown in Figure 2 as a solid $1\frac{pJ}{MAC}$ line. It's worth noting however that a significant portion of this cost in modern ASICs comes from the cost of memory + data movement. As noted in[14] the energy consumption of an 8bit MAC operation in a 45nm node is 200fJ. Scaling this to a 7nm node leads to an energy consumption $< 10fJ$ per MAC. Our analysis here shows that this architecture can beat this cost using near-term technology, but it is worth noting here that the cost of a multiply is not the metric to beat, rather the total system energy consumption including memory access and data movement is critical.

Convolutional Neural Networks and Training

The optical computing system in 1 performs matrix-vector products. This was then scaled into a matrix-matrix multiplier by making use of free-space optics to fan out in 3D space 3. One way to view this free space architecture is to view the receiver as a 2D array of processing elements similar to existing accelerator architectures.[11,15] Here, we first discuss a method for performing convolutional problems that maps them directly onto matrix multiplication. Then we discuss a more general framework for modeling convolutional problems on this architecture.

Convolutional Neural Networks

Convolution has become a critical opeartion for accelerators to support in order to perform the computation necessary for machine vision tasks. In Figure 3 a method for converting convolution problems into matrix multiplication is discussed. This method, as well as similar methods such as the use of a Toeplitz matrix, show that the complex problem of convolution can always be converted for processing by vector-vector product accelerators. However, as is discussed in the paper timeloop[32] there are many many efficient ways of mapping convolutional problems onto hardware that do not convert them into matrix-multiplication. As a result, our discussion of CNNs will be more general.

A framework from computer architecture literature that is very useful for considering how to map a given problem onto an accelerator is the general loopnest description of convolutional neural networks. Shown here in Figure 4 we see the loopnest for a convolutional neural network. We can see that matrix-multiplication is a subset of this problem by setting $R = S = P = Q = 1$. From part b of Figure 4 we see a loglog plot of the number of MAC operations that are performed for each layer of these networks compared with the number of distinct values required for those layers (which is the sum of the sizes of the input, weight, and output tensors). From this we see that convolutional problems have a larger amount of data reuse available than matrix-multiplication. That is, the amount of MACs performed per data access is much larger. Problems that maximize this metric are of particular interest, as they represent problems with the greatest opportunity for photonic hardware accelerators to best custom ASICs in CMOS. Because of the limited data access these problems have both the highest chance for CMOS chips to get fundamentally limited by their clock-speed, as well as having the lowest memory hierarchy cost per MAC. A careful analysis of the limits of on-chip memory in CMOS, as well as it's scaling, is required.

Training

The creation of neural network models involves finding weights to the model which minimize a function, known as the loss function L, which represents error in this model (loss functions are typically simple functions of the model error itself such as the mean square error or cross-entropy). Training of a network starts with a forward-pass

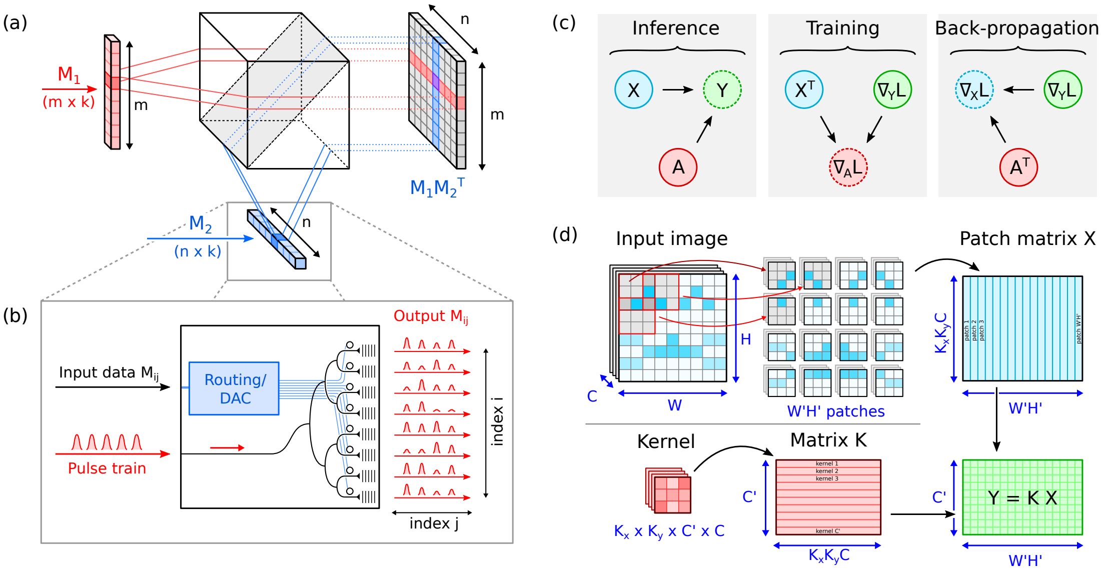
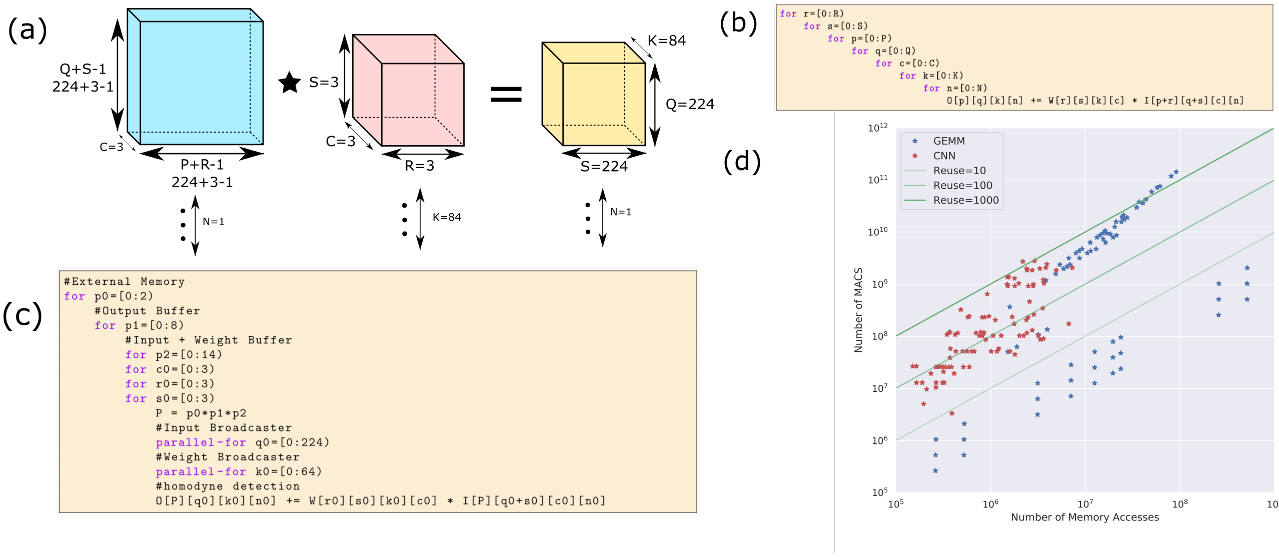


Figure 3. (a) Here signals from two arrays of sources, M_1 and M_2 are broadcast to the rows and columns respectively of a 2D receiver, generating the outer product between two vectors on each timestep. (b) Incoming data is passed to control circuitry, such as a DAC and router, which distribute RF control signals to modulators. (c) Required matrix operations for inference, training, and backpropagation in a deep neural network. (d) A patching method for performing convolutional matrix multiplication.



```
for r=[0:R)
    for s=[0:S)
        for p=[0:P)
            for q=[0:Q)
                for c=[0:C)
                    for k=[0:K)
                        for n=[0:N)
                            O[p][q][k][n] += W[r][s][k][c] * I[p+r][q+s][c][n]
```

```
#External Memory
for p0=[0:2)
    #Output Buffer
    for p1=[0:8)
        #Input + Weight Buffer
        for p2=[0:14)
        for c0=[0:3)
        for r0=[0:3)
        for s0=[0:3)
            P = p0*p1*p2
            #Input Broadcaster
            parallel-for q0=[0:224)
            #Weight Broadcaster
            parallel-for k0=[0:64)
            #homodyne detection
            O[P][q0][k0][n0] += W[r0][s0][k0][c0] * I[P][q0+s0][c0][n0]
```

Figure 4. (a) The general description of neural network problems described as a loopnest. (b) A graphic depicting a particular problem (the first layer of Alexnet with a batch size of 1). (c) A example of a mapping found using the timeloop[32] software through the network (in the same manner as inference above) which computes $Y = AX$. This generates a vector at the output of the network where the largest value in the vector is the estimate of the class of the input to the network. We can pass this estimation vector, as well as a vector with one-hot encoding into the loss function giving us $\nabla_Y L$. We wish to minimize $\nabla_A L$ for this input, but in order to do this on all layers we must have a way of finding the error in early layers in the network from later layers. Backpropagation[8] is a technique which allows use to compute this by using the chain rule we see that: $\nabla_A L = (\nabla_Y L)X^T, \nabla_X L = A^T(\nabla_Y L)$. One we have propagated a layer output back to it's input the final step is to use chain rule to compute $\nabla_{Y^{k-1}} L = f'(\nabla_{X^k} L)$. This technique can be extended to deal with convolutional neural network. We can take the proposed tiling method from above which casts CNN problems into a matrix multiply, and train on that. Further optimizations are possible by mapping the problem of training onto the hardware.

In the near term, training of a full network from scratch may not be feasible as limitations in DAC/ADC or nonlinearities in modulating devices produce limits on the accuracy possible to train to. Where this technique really has an advantage, however, is in post-training tuning for the hardware. Often in neural network accelerators a process of fine-tuning is employed in order to take a model generated in 32-bit floating point, quantize it to 8-bit integer/fixed-point, and train it for several additional epochs in the presence of quantization noise to regain any lost accuracy from the quantization.[16] In this way, post training fine tuning for photonic hardware could regain lost accuracy from fabrication error, process variation, or other architecture specific phenomena.

Discussion

This paper presents an optoelectronic accelerator which is scalable to $N \geq 10^6$ neurons in the near-term. The primary benefit in this architecture is that multiply and accumulate operations can be performed passively via optical inference. As a result the architecture does not have a fixed cost per operation, meaning that the number of data reads and writes required for a matrix-multiplication operation scales as $O(3N^2)$ while the number of operations performed scales as $O(N^3)$. This means that for large problems and scaled hardware the IO cost per operation can be decreased by a factor $\frac{1}{N}$. To an extent CMOS accelerators can also do this, but as large portions of memory cost are from data movement scaling a CMOS architecture leads to both an increase in total IO energy. Even if IO energy does not scale in CMOS they fundamentally run into the bound of the cost per operation on the order of $10fJ$.

The exponential growth in computing performance associated with Moore's law is slowing. As statistical models become more complicated the need for larger and larger layers which can learn more complicated features increases.[15,16] Photonics, in the future, may become necessary to assist in computation as the cost of both logic and interconnects[13] stops scaling. The architecture presented in this paper promises significant short-term performance gains over state-of-the-art electronics, with long term performance bounded by a standard quantum limit, of many orders of magnitude improvement.

ACKNOWLEDGMENTS

A.S. is supported by an NSF GRFP. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1122374. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of the National Science Foundation. L.B. is supported by a Doctoral Postgraduate Scholarship from the Natural Sciences and Engineering Research Council of Canada (NSERC). D.E. and M.S. acknowledge support from the U.S. ARO through the ISN at MIT (no. W911NF-18-2-0048).

The author would like to acknowledge Vivienne Sze (MIT), Joel Emer (Nvidia/MIT), and Angshuman Parashar (Nvidia) for informative discussions on modeling this architecture.

REFERENCES

[1] Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky,

Y., Molloy, J., Paine, T. L., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Kavukcuoglu, K., Hassabis, D., Apps, C., and Silver, D., "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature* **575**, 350–354 (11 2019).

[2] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., Van Den Driessche, G., Graepel, T., and Hassabis, D., "Mastering the game of Go without human knowledge," *Nature* **550**, 354–359 (10 2017).

[3] Young, T., Hazarika, D., Poria, S., and Cambria, E., "Recent Trends in Deep Learning Based Natural Language Processing," (8 2017).

[4] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "ImageNet Classification with Deep Convolutional Neural Networks," tech. rep.

[5] Schreiber, J., Durham, T., Bilmes, J., and Noble, W. S., "Multi-scale deep tensor factorization learns a latent representation of the human epigenome," (2018).

[6] Zhou, J. and Troyanskaya, O. G., "Predicting effects of noncoding variants with deep learning–based sequence model," *Nature Methods* **12**, 931–934 (10 2015).

[7] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L., "ImageNet Large Scale Visual Recognition Challenge," (9 2014).

[8] "lecun-88,"

[9] Moore, G. E., "Cramming More Components onto Integrated Circuits," tech. rep.

[10] Chen, Y., Luo, T., Liu, S., Zhang, S., He, L., Wang, J., Li, L., Chen, T., Xu, Z., Sun, N., and Temam, O., "DaDianNao: A Machine-Learning Supercomputer," tech. rep.

[11] Chen, Y. H., Krishna, T., Emer, J. S., and Sze, V., "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," *IEEE Journal of Solid-State Circuits* **52**, 127–138 (1 2017).

[12] Yin, S., Ouyang, P., Tang, S., Tu, F., Li, X., Zheng, S., Lu, T., Gu, J., Liu, L., and Wei, S., "I AM LEAD ON THIS!!! - A High Energy Efficient Reconfigurable Hybrid Neural Network Processor for Deep Learning Applications," *IEEE Journal of Solid-State Circuits* **53**(4), 968–982 (2018).

[13] Miller, D. A., "Attojoule Optoelectronics for Low-Energy Information Processing and Communications," *Journal of Lightwave Technology* **35**, 346–396 (2 2017).

[14] Horowitz, "Computing Energy Problem,"

[15] Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., Boyle, R., Cantin, P.-L., Chao, C., Clark, C., Coriell, J., Daley, M., Dau, M., Dean, J., Gelb, B., Ghaemmaghami, T. V., Gottipati, R., Gulland, W., Hagmann, R., Ho, C. R., Hogberg, D., Hu, J., Hundt, R., Hurt, D., Ibarz, J., Jaffey, A., Jaworski, A., Kaplan, A., Khaitan, H., Killebrew, D., Koch, A., Kumar, N., Lacy, S., Laudon, J., Law, J., Le, D., Leary, C., Liu, Z., Lucke, K., Lundin, A., Mackean, G., Maggiore, A., Mahony, M., Miller, K., Nagarajan, R., Narayanaswami, R., Ni, R., Nix, K., Norrie, T., Omernick, M., Penukonda, N., Phelps, A., Ross, J., Ross, M., Salek, A., Samadiani, E., Severn, C., Sizikov, G., Snelham, M., Souter, J., Steinberg, D., Swing, A., Tan, M., Thorson, G., Tian, B., Toma, H., Tuttle, E., Vasudevan, V., Walter, R., Wang, W., Wilcox, E., and Yoon, D. H., "In-Datacenter Performance Analysis of a Tensor Processing Unit TM," tech. rep. (2017).

[16] Sze, V., Chen, Y. H., Yang, T. J., and Emer, J. S., "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," (12 2017).

[17] Kim, K. H., Gaba, S., Wheeler, D., Cruz-Albrecht, J. M., Hussain, T., Srinivasa, N., and Lu, W., "A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications," *Nano Letters* **12**, 389–395 (1 2012).

[18] Li, C., Belkin, D., Li, Y., Yan, P., Hu, M., Ge, N., Jiang, H., Montgomery, E., Lin, P., Wang, Z., Song, W., Strachan, J. P., Barnell, M., Wu, Q., Williams, R. S., Yang, J. J., and Xia, Q., "Efficient and self-adaptive in-situ learning in multilayer memristor neural networks," *Nature Communications* **9** (12 2018).

[19] Feinberg, B., Wang, S., and Ipek, E., "Making Memristive Neural Network Accelerators Reliable," tech. rep.

[20] Miller, S. E., Marcatili, E. A. J., and Li, T., "Research Toward Optical-Fiber Transmission Systems Part I: The Transmission Medium Part It: Devices and Systems Considerations," Tech. Rep. 12 (1973).

[21] Lin, X., Rivenson, Y., Yardimci, N. T., Veli, M., Luo, Y., Jarrahi, M., and Ozcan, A., "All-optical machine learning using diffractive deep neural networks," tech. rep.

[22] Tait, A. N., Nahmias, M. A., Shastri, B. J., and Prucnal, P. R., "Broadcast and weight: An integrated network for scalable photonic spike processing," *Journal of Lightwave Technology* **32**, 3427–3439 (11 2014).

[23] Duport, F., Schneider, B., Smerieri, A., Haelterman, M., and Massar, S., "Backpropagation-decorrelation: online recurrent learning with O(N) complexity," Tech. Rep. 5667 (2004).

[24] Shen, Y., Harris, N. C., Skirlo, S., Prabhu, M., Baehr-Jones, T., Hochberg, M., Sun, X., Zhao, S., Larochelle, H., Englund, D., and Soljacic, M., "Deep learning with coherent nanophotonic circuits," *Nature Photonics* **11**, 441–446 (6 2017).

[25] Hayat, M. M., Saleh, E. A., and Gubner, J. A., "Shot-Noise-Limited Perfor of Optical Neural Netw," Tech. Rep. 3 (1996).

[26] Hamerly, R., Bernstein, L., Sludds, A., Soljačić, M., and Englund, D., "Large-Scale Optical Neural Networks Based on Photoelectric Multiplication," *Physical Review X* **9** (5 2019).

[27] Landauer, R., "Irreversibility and Heat Generation in the Computing Process," tech. rep.

[28] Lloyd, S., "Physical Limits to Computation," in [*Quantum Communication, Computing, and Measurement 3*], 189–198, Kluwer Academic Publishers (2 2006).

[29] Jonsson, B. E., "Testing and Data Converter Analysis and Design and IEEE 2011 ADC Forum," tech. rep.

[30] Saeedi, S., Menezo, S., Pares, G., and Emami, A., "A 25 Gb/s 3D-Integrated CMOS/Silicon-Photonic Receiver for Low-Power High-Sensitivity Optical Communication," *Journal of Lightwave Technology* **34**, 2924–2933 (6 2016).

[31] Sun, C., Wade, M. T., Lee, Y., Orcutt, J. S., Alloatti, L., Georgas, M. S., Waterman, A. S., Shainline, J. M., Avizienis, R. R., Lin, S., Moss, B. R., Kumar, R., Pavanello, F., Atabaki, A. H., Cook, H. M., Ou, A. J., Leu, J. C., Chen, Y. H., Asanović, K., Ram, R. J., Popović, M. A., and Stojanović, V. M., "Single-chip microprocessor that communicates directly using light," *Nature* **528**, 534–538 (12 2015).

[32] Parashar, A., Raina, P., Yakun, Shao, S., Chen, Y.-H., Ying, V. A., Mukkara, A., Venkatesan, R., Khailany, B., Keckler, S. W., Emer, J., and Nvidia, ., "Timeloop: A Systematic Approach to DNN Accelerator Evaluation," tech. rep.