Freely scalable and reconfigurable optical hardware for deep learning

LIANE BERNSTEIN1,†,5, ALEXANDER SLUDDS1,†,6, RYAN HAMERLY1,2, VIVIENNE SZE1, JOEL EMER3,4, AND DIRK ENGLUND1,7

1Research Laboratory of Electronics, Massachusetts Institute of Technology, 50 Vassar St, Cambridge, MA 02139, USA
2NTT Research Inc. Physics & Informatics Laboratories, 1950 University Ave #600, East Palo Alto, CA 94303, USA
3NVIDIA, 2 Technology Park Drive, Westford, MA 01886, USA
4Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 32 Vassar St, Cambridge, MA 02139, USA
†These authors contributed equally to this work
5e-mail: lbern@mit.edu
6e-mail: asludds@mit.edu
7e-mail: englund@mit.edu

Abstract: As deep neural network (DNN) models grow ever-larger, they can achieve higher accuracy and solve more complex problems. This trend has been enabled by an increase in available compute power; however, efforts to continue to scale electronic processors are impeded by the costs of communication, thermal management, power delivery and clocking. To improve scalability, we propose a digital optical neural network (DONN) with intralayer optical interconnects and reconfigurable input values. The near path-length-independence of optical energy consumption enables information locality between a transmitter and arbitrarily arranged receivers, which allows greater flexibility in architecture design to circumvent scaling limitations. In a proof-of-concept experiment, we demonstrate optical multicast in the classification of 500 MNIST images with a 3-layer, fully-connected network. We also analyze the energy consumption of the DONN and find that optical data transfer is beneficial over electronics when the spacing of computational units is on the order of >10 μm.

Introduction

Machine learning has become ubiquitous in modern data analysis, decision-making, and optimization. A prominent subset of machine learning is the artificial deep neural network (DNN), which has revolutionized many fields, including classification [1], translation and prediction [2,3]. An important step toward unlocking the full potential of DNNs is improving the energy consumption and speed of DNN tasks. To this end, emerging DNN-specific hardware optimizes data access, reuse and communication for mathematical operations: most importantly, general matrix-matrix multiplication (GEMM) and convolution [4]. One approach is to use a specialized memory hierarchy to store and reuse data near an array of computation units, which minimizes reliance on expensive large-scale data distribution networks [5–7]. Another option for GEMM is a large array of electronic multipliers with fewer intermediate memory tiers [8]. The large multiplier array reduces overhead, and if the DNN is sizable enough to keep all the processing elements occupied, this design can be more efficient in energy consumption and throughput, thanks to the ability to perform more parallel operations.

However, despite these advances, a central challenge in the field is scaling hardware to keep up with exponentially-growing DNN models (see Fig. 1 and Ref. [9]). Many popular DNN models comprise matrices exceeding the GEMM capacity of leading DNN processors (e.g., Google’s Tensor Processing Unit (TPU) [8]), and therefore, matrices must be computed in multiple
‘tiles’. Tiling requires many inputs and intermediate values to be stored rather than streamed, which increases data movement. Thus, tiling restricts the use of DNNs in high-throughput applications such as the observation of new phenomena in fundamental physics [10–14], and reduces the throughput of large DNN models such as recommender systems [15], vision [8] and natural language processing [16]. Though the current trend is to scale up conventional electronic hardware, these efforts are impeded by communication [17], clocking [18], thermal management [19] and power delivery [20]. Parallel processing with multiple chips [21] or partitioned chips [22, 23] can ease these constraints and improve performance over a monolithic equivalent through greater mapping flexibility [24], at the cost of increased communication energy.

![Graph showing number of parameters in various neural networks over time.]

In this Article, we introduce an optical DNN accelerator that encodes data into reconfigurable on-off optical pulses for transmission and passive copying (or fan-out) to large-scale electronic multiplier arrays. The near length-independence of optical data routing enables freely scalable systems, where single transmitters are fanned out to many arbitrarily arranged receivers with fast and energy-efficient links. Optics has previously been proposed for analog DNN accelerators, with potential orders-of-magnitude reductions in energy consumption and improved throughput [37–41]. In contrast, we propose an entirely digital system, where we replace electrical on-chip interconnects with optical paths for data transmission, but not computation, thus with the capability to preserve accuracy. This ‘digital optical neural network’ (DONN) performs large-scale data distribution from memory to an arbitrary set of electronic multipliers. We first illustrate the DONN architecture and discuss possible implementations. Then, in a proof-of-concept experiment, we demonstrate that digital optical transmission and fan-out with cylindrical lenses has little effect on the classification accuracy of the MNIST handwritten digit dataset (<0.6%).
Crosstalk is the primary cause of this drop in accuracy, and because it is deterministic, it can be compensated: with a simple crosstalk correction scheme, we reduce our bit error rates by two orders of magnitude. Alternatively, crosstalk can be greatly reduced through optimized optical design. Since shot and thermal noise are negligible (see Discussion), the accuracy of the DONN can therefore be equivalent to an all-electronic DNN accelerator.

We also compare the energy consumption of optical interconnects (including light source energy) against that of electronic interconnects over distances representative of logic, memory, and multi-chiplet interconnects in a 7 nm CMOS node. Our calculations show an advantage in data transmission costs for distances $\geq 5 \mu m$ (roughly the size of the basic computation unit: an 8-bit multiply-and-accumulate (MAC), with length 5-8 $\mu m$). Moreover, the DONN scales favorably with respect to very large DNN accelerators that require partitioning into multi-chiplet: the DONN’s optical communication cost remains nearly constant at $\sim 0.2 fJ/bit$, whereas multi-chiplet systems have much higher electrical interconnect costs ($\sim 90 fJ/bit$). Thus, the efficient optical data distribution provided by the DONN architecture will become critical for continued growth of DNN performance through increased model sizes and greater connectivity.

Results

Problem statement

A DNN consists of a sequence of layers, in which input activations from one layer are connected to the next layer via weighted paths (weights), as shown in Fig. 2a. We focus on inference tasks in this paper (where weights are known from prior training), which, in addition to the energy consumption problem, places stringent requirements on latency and throughput. Modern inference accelerators expend the majority of energy ($> 90\%$) on memory access, data movement, and computation in fully-connected (FC) and convolutional (CONV) layers [5].

Parallelized vector operations, such as matrix-matrix multiplication or successive vector-vector inner products, are the largest energy consumers in CONV and FC layers. In an FC layer, a vector $x$ of input values (‘input activations’, of length $K$) is multiplied by a matrix $W_{K \times N}$ of weights (Fig. 2b). This matrix-vector product yields a vector of output activations ($y$, of length $N$). Most DNN accelerators process vectors in $B$-sized batches, where the inputs are represented by a matrix $X_{B \times K}$. The FC layer then becomes a matrix-matrix multiplication ($X_{B \times K} \cdot W_{K \times N}$). CONV layers can also be processed as matrix multiplications, e.g., with a Toeplitz matrix [4].

In matrix multiplication, fan-out, where data is read once from main memory (DRAM) and used multiple times, can greatly reduce data movement and memory access. This amortization of read cost across numerous operations is critical for overall efficiency, since retrieving a single matrix element from DRAM requires two to three orders of magnitude more energy than the MAC [17]. A simple input-weight product illustrates the benefit of fan-out, since activation and weight elements appear repeatedly, as highlighted by the repetition of $X_{11}$ and $W_{11}$:

$$
\begin{bmatrix}
X_{11} & X_{12} \\
X_{21} & X_{22}
\end{bmatrix}
\begin{bmatrix}
W_{11} & W_{12} \\
W_{21} & W_{22}
\end{bmatrix}
= 
\begin{bmatrix}
X_{11}W_{11} + X_{12}W_{21} & X_{11}W_{12} + X_{12}W_{22} \\
X_{21}W_{11} + X_{22}W_{21} & X_{21}W_{12} + X_{22}W_{22}
\end{bmatrix}
$$

(1)

Consequently, DNN hardware design focuses on optimizing data transfer and input and weight matrix element reuse. Accelerators based on conventional electronics use efficient memory hierarchies, a large array of tightly packed processing elements (PEs, i.e., multipliers with or without local storage), or some combination of the these approaches. Memory hierarchies optimize temporal data reuse in memory blocks near the PEs to boost performance under the constraint of chip area [4]. This strategy can enable high throughput in CONV layers [5]. With fewer intermediate memory levels, a larger array of PEs (e.g., TPU v1 [8]) can further increase throughput and lower energy consumption on workloads with a high-utilization mapping due to
Fig. 2. Digital fully-connected neural network (FC-NN) and hardware implementations. (a) FC-NN with input activations (red, vector length $K$) connected to output activations (vector length $N$) via weighted paths, i.e., weights (blue, matrix size $K \times N$). (b) Matrix representation of one layer of an FC-NN with $B$-sized batching. (c) Example bit-serial multiplier array, with output-stationary accumulation across $k$. Fan-out of $X$ across $n \in \{1 \ldots N\}$; fan-out of $W$ across $b \in \{1 \ldots B\}$. Bottom panel: all-electronic version with fan-out by copper wire (for clarity, fan-out of $W$ not illustrated). Top panel: digital optical neural network version, where $X$ and $W$ are fanned out passively using optics, and transmitted to an array of photodetectors. Each pixel contains two photodetectors, where the activations and weights can be separated by, e.g., polarization or wavelength filters. Each photodetector pair is directly connected to a multiplier in close proximity.

potentially reduced overall memory accesses and a greater number of parallel multipliers (spatial reuse). Therefore, for workloads with large-scale matrix multiplication such as those mentioned in the Introduction, if we maximize the number of available PEs, we can improve efficiency.

**Digital optical neural network architecture**

Our DONN architecture replaces electrical interconnects with optical links to relax the design constraints of reducing inter-multiplier spacing or colocating multipliers with memory. Specifically, optical elements transfer and fan out activation and weight bits to electronic multipliers to reduce communication costs in matrix multiplication, where each element $X_{bk}$ is fanned out $N$ times, and $W_{kn}$ is fanned out $B$ times. The DONN scheme shown in Fig. 2c spatially encodes the first column of $X_{B \times K}$ activations into a column of on-off optical pulses. At the first time step, the activation matrix transmitters fan out the first bit of each of the matrix elements $X_{b1}, \forall b \in \{1 \ldots B\}$ to the PEs (here, $k = 1$). Simultaneously, a row of weight matrix light sources transmits the corresponding weight bits $W_{1n}$ to each PE. The photons from these activation and weight bits generate photoelectrons in the detectors, producing the voltages required at the inputs of electronic multipliers (either 0 V for a ‘0’ or 0.8 V for a ‘1’). After 8 time steps, a multiplier has received $2 \times 8$ bits (8 bits for the activation value and 8 bits for the weight value), and the electronic multiplication occurs as it would in an all-electronic system. The activation-weight product is completed, and is added to the locally stored partial sum. The entire matrix-matrix product is therefore computed in $8 \times K$ time steps; this dataflow is commonly called ‘output stationary’. Instead of this bit-serial implementation, bits can be encoded spatially, using a bus of parallel transmitters and receivers. The trade-off between added energy and latency in bit-serial
multiplication versus increased area from photodetectors for a parallel multiplier can be analyzed for specific applications and CMOS nodes.

![Fig. 3. Possible implementations of digital optical neural network. (a) Free-space version. Digital inputs and weights are transmitted electronically to an array of light sources (red and blue, respectively, illustrating different paths). Single-mode light from a source is collimated by a spherical lens (Lens), then focused to a 1D spot array by a diffractive optical element (DOE). A 50:50 beamsplitter brings light from the inputs and weights into close proximity on a custom CMOS receiver. (b) Waveguide or chip-integrated implementation with scatterers above each processing element (PE). (c) Example circuit with 2 photodetectors per PE: 1 for activations; 1 for weights. Received bits proceed to multiplier, then memory or next layer.

We illustrate two exemplary experimental DONN implementations in Fig. 3. In the free-space version (Fig. 3a), each source in a linear array of vertical cavity surface emitting lasers (VCSELs) or µLEDs emits a cone of light into free space, which is collimated by a spherical lens. A diffractive optical element (DOE) focuses the light to a 1D spot array on a 2D receiver, where the activations and weights are brought into close proximity using a beamsplitter. Figure 3b shows a waveguide or chip-integrated alternative, where each light source is coupled into an optical waveguide. The waveguides are low-loss, except at the scattering elements above each detector pixel. These scattering elements are tuned such that, along one row, an equal amount of light enters each photodetector for a ‘1’ (similar concepts have been experimentally demonstrated [42]). In both the free-space and integrated implementations, ‘receiverless’ photodiodes [43] convert the optical signals to the electrical domain (Fig. 3c). An electronic multiplier then multiplies the values. The output is either saved to memory, or routed directly to another DONN that implements the next layer of computation. Note that the data distribution pattern is not confined to regular rows and columns. A spatial light modulator (SLM), an array of micromirrors, scattering waveguides or a DOE can route and fan out bits to arbitrary locations.

There will be some length-dependent optical loss that will vary based on the implementation. Since free-space propagation is lossless and mirrors, SLMs and diffractive elements are highly efficient (> 95%), most length- or receiver-number-dependent losses can be attributed to imperfect focusing, e.g., from optical aberrations far from the optical axis. These effects can be mitigated...
through judicious optical design. There are also waveguide losses in integrated photonics, but these can be very low, e.g., 3 dB/m with mm-scale bend radii in silicon nitride [44]. (The DONN does not require any active components, which makes silicon nitride a good choice here.) Therefore, even if we design a meter-length chip with mm-scale bends in the waveguides, we can compensate optical losses by increasing the number of photons generated at the sources (for example, in silicon nitride, by a factor of 2). We assume for the remainder of our analysis that energy is length-independent.

**Bit error rate and inference experiments**

We used a DONN implementation similar to Fig. 3a to test optical digital data transmission and fan-out for DNNs, as described in Methods. In our first experiment, we determined the bit error rate of our system. Figure 4a shows an example of a background-subtracted and normalized image, captured on the camera when the DMDs displayed random vectors of ‘1’s and ‘0’s. The camera’s Bayer filter (described in Methods), as well as optical aberrations and misalignment, caused some crosstalk between pixels (see Fig. 4b). Using a region of $357 \times 477$ superpixels on the camera, we calculated bit error rates (in a single shot) of $1.2 \times 10^{-2}$ and $2.6 \times 10^{-4}$ for the blue and red channels, respectively. When we confined the region of interest to $151 \times 191$ superpixels, the bit error rate (averaged over 100 different trials, i.e., 100 pairs of input vectors) was $4.4 \times 10^{-3}$ and $4.6 \times 10^{-5}$ for the blue and red arms. See Supplementary Note 1 for more details on bit error rate and error maps. Because crosstalk is deterministic, and not a source of random noise, we can compensate for it. We applied a simple crosstalk correction scheme that assumes uniform crosstalk on the detector and subtracts a fixed fraction of an element’s nearest neighbors from the element itself (see Supplementary Note 2). The bit error rates for the blue and red channels then respectively dropped to $2.9 \times 10^{-3}$ and 0 for the $357 \times 477$-pixel, single shot image and $2.6 \times 10^{-5}$ and 0 for the $151 \times 191$-pixel, 100-image average. In other words, after crosstalk correction, there were no errors in the red channel, and the errors in the blue channel dropped significantly.

Next, we experimentally tested the DONN’s effect on the classification accuracy of 500 MNIST images using a three-layer (i.e., two-hidden-layer), fully-connected neural network (FC-NN), with the dataset and training steps described in Supplementary Note 3. We compared our experimental classification results with inference performed entirely on CPU (ground truth) in two ways. The
simplest analysis, reported in Table 1, shows a 0.6% drop in classification accuracy for the DONN versus the ground truth values (or 3 additional incorrectly classified images). Figure 5 illustrates more detailed results, where we analyzed the network output scores. An output score is roughly equivalent to the assigned likelihood that an input image belongs to a given class, and is defined as the normalized (via the softmax function) output vector of a DNN. We found that, along the matrix diagonal, the first and third quartiles in the difference in output scores between the DONN and the ground truth have a magnitude <3%. The absolute difference in average output scores is also <3%. We also performed this experiment with a single hidden layer (‘2-layer’ case), and achieved similar results (a 0.4% drop in accuracy, or 2 misclassified images). No crosstalk error correction was applied to these results.

**Energy analysis: DONN compared with all-electronic hardware**

In this section, we compare the theoretical interconnect energy consumption of the DONN with its all-electronic equivalent, where interconnects are illustrated in green in Fig. 6. The interconnect energy, which must include any source inefficiencies, is the energy required to charge the parasitic wire, detector, and inverter capacitances, where a CMOS inverter is representative of the input to a multiplier. See Methods for full energy calculations. In the electronic case, a long wire transports...
Table 1. MNIST classification accuracy of DONN versus all-electronic hardware with custom fully-connected neural network models

|               | 2 layers | 3 layers |
|---------------|----------|----------|
| Electronic (ground truth) | 95.8%    | 96.4%    |
| DONN          | 95.4%    | 95.8%    |

data to a row of multipliers using low-cost (.06 fJ/bit) repeaters (see Supplementary Note 6). The wire has a large parasitic capacitance, but also produces an effective electrical fan-out. In the DONN, the energetic requirements of the detectors contrast with those of conventional optical receivers, which aim to maximize sensitivity to the optical input field, rather than minimize the energetic cost of the system as a whole. The values for electronic and optical components are summarized in Table 2, where \( h \nu / e \) must be greater than or equal to the bandgap \( E_g \) of the detector material (here, we have chosen silicon as an example, and set \( h \nu / e = E_g \)). \( C_{\text{det}} \) is a theoretical approximation for a (1 x 1 x 1) \( \mu \text{m}^3 \) cubic detector [43] and the optical source power conversion efficiency (wall-plug efficiency, i.e., WPE) is a measured value for VCSELs [45, 46]. \( C_T \) is an approximation for the capacitance of an inverter in a state-of-the-art node [43, 47] and \( L_{\text{wire}} \) is the distance between MAC units in various scenarios.

We find that the optical interconnect energy is independent of length at 0.2 fJ/bit, while the electrical interconnect energy scales from 0.2-0.3 fJ/bit for inter-multiplier communication for abutted MAC units to 90 fJ/bit for inter-chiplet interconnects. The crossover point where the optical interconnect energy drops below the electrical energy occurs when \( L_{\text{wire}} \geq 5 \mu \text{m} \). The DONN therefore provides an improvement in the interconnect energy for data transmission and can scale to greatly decrease the energy consumption of data distribution with regular distribution patterns. Additionally, advanced technologies are emerging which could lower its energy consumption, such as plasmonic photodetectors with ultra-low capacitance [48] and more efficient VCSELs.

![Fig. 6. Fan-out of one bit from memory (Mem) to multiple processing elements (PEs).](image)

(a) Fan-out by electrical wire to a row of PEs in a monolithic chip. (b) DONN equivalent of monolithic chip, where green wire is replaced by optical paths. (c) Fan-out by electrical wire to blocks of PEs divided into chiplets, or separated by memory and logic. (d) DONN equivalent of fan-out to PEs in multiple blocks (energetically equivalent to (b)).

**Discussion**

With minimal impact on accuracy, the DONN yields an energy advantage over all-electronic accelerators with long wire lengths. In our proof-of-concept experiment, we performed inference
Table 2. Interconnect energies over three distances: inter-MAC, inter-SRAM, and inter-chiplet

| Global parameters | $C_{\text{wire}}/\mu\text{m}$ | $\sim 0.2 \text{ fF/} \mu\text{m}$ [43,49,50] |
|-------------------|-----------------------------|---------------------------------|
|                   | $C_T$                       | $\sim 0.1 \text{ fF}$ [43,47]   |
|                   | $C_{\text{det}}$            | $0.1 \text{ fF}$ [43]           |
|                   | $h\nu/e$                    | $1.12 \text{ eV}$               |
|                   | WPE                         | $\sim 0.5$ [45,46]              |

| Inter-MAC (8-bit MAC unit) | Inter-SRAM (7 nm SRAM macro) | Inter-chiplet [22] |
|----------------------------|-----------------------------|-------------------|
| $L_{\text{wire}}$          | 5-8 $\mu$m$^1$              | $L_{\text{wire}}$ | $\sim 2500 \mu$m |
| $V_{DD}$                   | 0.80 V [51]                 | $V_{DD}$          | 0.85 V$^*$        |
| $E_{\text{elec}}$/bit      | 0.2-0.3 fJ/bit              | $E_{\text{elec}}$/bit | 2 fJ/bit          |
| $E_{\text{DONN}}$/bit      | 0.2 fJ/bit                  | $E_{\text{DONN}}$/bit | 0.2 fJ/bit        |

We assume a square multiplier and scale reported 8-bit multiplier areas [53–55] from a 45 nm to a 7 nm node (the current state of the art) with the scaling factors from Ref. [51]. A MAC unit comprises both an 8-bit multiplier and a 32-bit adder, so we are placing a lower bound on the minimum length of $L_{\text{wire}}$. Recent work [56] optimizes MAC units for DNNs, and reports a 337 $\mu$m$^2$ area in a 28 nm node, where the MAC unit comprises an 8-bit multiplier and a 32-bit adder. Extrapolated to a 7 nm node with a fourth-order polynomial fit of the scaling factors from Ref. [51], the MAC unit is of size $(7 \mu m)^2$, which falls within the 5-8 $\mu$m range.

Input-output voltage and core logic voltage can differ in CMOS. In optics, however, since the data delivery mechanism does not vary with distance travelled, we assume $V_{DD}$ remains constant at 0.80 V.

We found a <0.6% drop in accuracy and a <3% absolute difference in average output scores with respect to the ground truth implementation on CPU. We attributed these errors to crosstalk due to imperfect alignment and blurring from the camera’s Bayer filter. In fact, a simple crosstalk correction scheme lowered measured bit error rates by two orders of magnitude. We could thus transmit bits with 100% measured fidelity in the activation arm (better aligned than the weight arm), which illustrates that crosstalk can be mitigated and possibly eliminated either through post-processing, charge sharing at the transmitters, greater spacing of receivers, or optimized design of optical elements and receiver pixels. In the hypothetical regime where error due to crosstalk is negligible, the remaining noise sources are shot and thermal noise. Intuitively, shot and thermal noise are also present in an all-electronic system, and the number of photoelectrons at the input to an inverter in the DONN is equal to the number of electrons at the input to an inverter in electronics. Therefore, if these noise sources do not limit accuracy in the all-electronic case, the same can be said for the DONN [43]. For mathematical validation that shot and thermal noise have a trivial impact on bit error rate in the DONN, see Supplementary Note 7. These analyses demonstrate that the fundamental limit to the accuracy of the DONN is no different than the accuracy of electronics, and thus, we do not expect accuracy to hinder DONN scaling in an optimized system.

In our theoretical energy calculations, we compared the nearly length-independent data delivery costs of the DONN with those of an all-electronic system. We found that in the worst case,
when multipliers are abutted in a multiplier array, optical transmitters have a similar interconnect energy cost compared to copper wires in a 7 nm node (−0.2 fJ/bit versus −0.2-0.3 fJ/bit). The regime where the DONN shows important gains over copper interconnects is in architectures with increased spacing between computation units, e.g., with locally-packed memory and logic (−0.2 fJ/bit versus −2 fJ/bit), or with multiple chiplets (−0.2 fJ/bit versus −90 fJ/bit). In the multi-chiplet case, the cost to transmit two 8-bit values in electronics (~1,400 fJ) is therefore significantly larger than that of an 8-bit MAC (25 fJ) [17,51]. On the other hand, in optics, the interconnect cost (~3 fJ for 2×8 bits, including source energy) remains an order of magnitude smaller than the MAC cost. Since multi-chiplet and multi-chip systems offer a promising approach to increasing throughput on large DNN models, optical connectivity can further these scaling efforts by reducing inter-chiplet communication energy by orders of magnitude.

In addition, because length-independent data distribution is a tool currently unavailable to digital system designers, relaxing electronic constraints on locality can open new avenues for DNN accelerator architectures. For example, memory can be devised such that numerous small pieces of memory are located far away from the point of computation and reused many times spatially, with a small fixed cost for doing so. Designers can then lay out smaller memory blocks with higher bandwidth, lower energy consumption, and higher yield. If we keep memory and computation spatially distinct, we have the added benefit of allowing for more compact memories that consume less energy and area, e.g., DRAM, which is fabricated with a different process than typical CMOS to achieve higher density than on-chip memories. Furthermore, due to its massive fan-out potential, the DONN can, firstly, reduce overhead by minimizing a system’s reliance on a memory hierarchy and, secondly, amortize the cost of weight delivery to multiple clients running the same neural network inference on different inputs. Additionally, some newer neural network models require irregular connectivity (e.g., graph neural networks, which show state-of-the-art performance on recommender systems, but are restricted in size due to insufficient compute power [57,58]). These systems have arbitrary connections with potentially long wire lengths between MAC units, representing different edges in the graph. The DONN can implement these links without incurring additional costs in energy from a complex network-on-chip in electronics. Yet another instance of greater distance between multipliers is in higher-bit-precision applications, as in training, which require larger MAC units. Lastly, the DONN could facilitate thermal management in chips, with the option to increase spacing between compute units at no extra cost.

In future work, we plan to assess the performance of the DONN on state-of-the-art DNN workloads, such as the models described in MLPerf [59]. Firstly, we will benchmark the DONN against all-electronic state-of-the-art accelerators by using Timeloop [60]. Through a search for optimal mappings (ways to organize data and computation), this software can simulate the total energy consumption and latency of running various workloads on a given hardware architecture, including computation and memory access. Timeloop therefore enables us to perform an in-depth comparison of all-electronic accelerators against the proposed instances of the DONN, including variable data transmission costs for different electronic wire lengths, and waveguide losses in the chip-integrated DONN. Second, we will design an optical setup and receiver to reduce experimental crosstalk, power consumption and latency. We can then test larger workloads on this optimized hardware. Finally, beyond neural networks, there are many examples of matrix multiplication which a DONN-style architecture can accelerate, such as optimization, Ising machines and statistical analysis, and we plan to investigate these applications as well.

In summary, the DONN implements arbitrary transmission and fan-out of data with an energy cost per bit that is nearly independent of data transmission length and number of receivers. This property is key to scaling deep neural network accelerators, where increasing the number of processing elements for greater throughput in all-electronic hardware typically implies higher data communication costs due to longer electronic path length. Contrary to other proposed
optical neural networks [37–41], the DONN does not require digital-to-analog conversion and is therefore less prone to error propagation. The DONN is also reconfigurable, in that the weights and activations can be easily updated. Our work indicates that the nearly length-independent communication enabled by optics is useful for digital neural network system design, for example to simplify memory access to weight data. We find that optical data transfer begins to save energy when the spacing of MAC computational units is on the order of >10 µm. More broadly, further gains can be expected through the relaxation of electronic system architecture constraints.

**Methods**

**Digital optical neural network implementation for bit error rate and inference experiments**

We performed bit error rate and inference experiments with optical data transfer and fan-out of point sources using cylindrical lenses. Two digital micromirror devices (DMDs, Texas Instruments DLP3000, DLP4500) illuminated by spatially-filtered and collimated LEDs (Thorlabs M625L3, M455L3) acted as stand-ins for the two linear source arrays. For the input activations/weights, each 10.8 µm-long mirror in one DMD column/row either reflected the red/blue light toward the detector (‘1’) or a beam dump (‘0’). Then, for each of the DMDs, an \( f = 100 \) mm spherical lens followed by an \( f = 100 \) mm cylindrical achromatic lens imaged one DMD pixel to an entire row/column of superpixels of a color camera (Thorlabs DCC3240C). Each camera superpixel is made up of four pixels of size (5.3 µm)\(^2\): two green, one red and one blue. The camera acquisition program applies a ‘de-Bayering’ filter to automatically extract color information for each sub-pixel; this filter causes blurring, and therefore it increased crosstalk in our system. In a future version of the DONN, a specialized receiver will reduce this crosstalk and also operate at a higher speed.

To process the image received on the camera, we subtracted the background, normalized, then thresholded. (We acquired normalization and background curves with all DMD pixels in the ‘on’ and ‘off’ states, respectively. This background subtraction and normalization could be implemented on-chip by precharacterizing the system, and biasing each receiver pixel by some fixed voltage.) If the detected intensity was above the threshold value, it was labeled a ‘1’; below threshold, a ‘0’. For the bit error rate experiments, we compared the parsed values from the camera with the known values transmitted by the DMDs, and defined the bit error rate as the number of incorrectly received bits divided by the total number of bits. In the inference experiments, the DMDs displayed the activations and pre-trained weights, which propagated through the optical system to the camera. After background subtraction and normalization, the CPU multiplied each activation with each weight, and applied the nonlinear function (ReLU after the hidden layers and softmax at the output). We did not correct for crosstalk here, to illustrate the worst-case scenario of impact on accuracy. The CPU then fed the outputs back to the input activation DMD for the next layer of computation. We used a DNN model with two hidden layers with 100 activations each and a 10-activation output layer. We also tested a model with a single hidden layer with 100 activations.

**MNIST preprocessing**

For the inputs to the network, a bilinear interpolation algorithm transformed the 28 × 28-pixel images into 7 × 7-pixel images, which were then flattened into a 1D 49-element vector. The following standard mapping quantized both input and weight matrices into 8-bit integer representations:

\[
\text{Quantized} = \text{QuantizedMin} + \frac{\text{Input} - \text{FloatingMin}}{\text{Scale}}
\]

where Quantized is the returned value, QuantizedMin is the minimum value expressible in the quantized datatype (here, always 0), Input is the input data to be quantized, FloatingMin is
the minimum value in Input, and Scale is the scaling factor to map between the two datatype ranges \( \left( \frac{\text{FloatingMax} - \text{FloatingMin}}{\text{QuantizedMax} - \text{QuantizedMin}} \right) \). See gemmlowp documentation [61] for more information on implementations of this quantization. (In practice, 8-bit representations are widely used in DNNs, since 8-bit MACs are generally sufficient to maintain accuracy in inference [8, 62, 63]).

**Electronic and optical interconnect energy calculations**

When an electronic wire transports data over a distance \( L_{\text{wire}} \) to the gate of a CMOS inverter (representative of a full-adder’s input, which are the building blocks of multipliers), the energy consumption per bit is:

\[
E_{\text{elec}} / \text{bit} = \frac{1}{4} \left( \frac{C_{\text{wire}}}{\mu m} \cdot L_{\text{wire}} + C_T \right) \cdot V_{DD}^2
\]

where \( V_{DD} \) is the supply voltage, \( L_{\text{wire}} \) is the wire length between two multipliers and \( C_T \) is the inverter capacitance. Interconnects consume energy predominantly when a load capacitance, such as a wire, is charged from a low (0 V) to a high (~1 V) voltage, i.e., in a 0 \( \rightarrow \) 1 transition. If we assume a low leakage current, maintaining a value of ‘1’ (i.e., 1 \( \rightarrow \) 1) consumes little additional energy. To switch a wire from a ‘1’ to a ‘0’, the wire is discharged to the ground for free (Supplementary Note 4). Lastly, maintaining a value of ‘0’ simply keeps the voltage at 0 V, at no cost. Assuming a random distribution of ‘0’ and ‘1’ bits, we therefore include a factor of 1/4 in equation (3) to account for this dependence on switching activity.

In the DONN, a light source replaces the wire for fan-out. The low capacitances of the receiverless detectors in the DONN allow for the removal of receiving amplifiers [43]. Thus, the DONN’s minimum energy consumption corresponds to the optical energy required to generate a voltage swing of 0.8 V on the load capacitance (i.e., the photodetector \( C_{\text{det}} \) and an inverter \( C_T \)), all divided by the source’s power conversion efficiency (called wall-plug efficiency, WPE). Subsequent transistors in the multiplier are powered by the off-chip voltage supply, as in the all-electronic architecture. Assuming a detector responsivity of \( \sim 1 \) [64], the DONN interconnect energy cost is:

\[
E_{\text{DONN}} / \text{bit} = \frac{1}{2} \cdot WPE \cdot h \cdot n_p
\]

where \( h \) is the photon energy and the number of photons per bit, \( n_p \), is determined by:

\[
n_p = \frac{(C_{\text{det}} + C_T) \cdot V_{DD}}{e}
\]

As in the all-electronic case, we assume low leakage on the receiverless photodetector. Photons are received for every ‘1’ and therefore, to avoid charge buildup, charge on the output capacitor must be reset after every clock cycle. In Supplementary Note 5, we propose a CMOS discharge circuit that actively resets the receiver. (Another possible method is a dual-rail encoding scheme [43].) Thus, the switching activity factor is 1/2 instead of 1/4: as for the all-electronic case, we assume a random distribution of bits, but here, both 1 \( \rightarrow \) 1 and 0 \( \rightarrow \) 1 have a nonzero cost.

**References**

1. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25.1097–1105 (2012).
2. Esteva, A. et al. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **542**, 115–118 (2017).
3. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
4. Sze, V., Chen, Y., Yang, T. & Emer, J. S. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE* **105**, 2295–2329 (2017).
5. Chen, Y., Krishna, T., Emer, J. S. & Sze, V. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE Journal of Solid-State Circuits* **52**, 127–138 (2017).
6. Chen, Y.-H., Yang, T.-J., Emer, J. & Sze, V. Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* **9**, 292–308 (2019).

7. Yin, S. et al. A high energy efficient reconfigurable hybrid neural network processor for deep learning applications. *IEEE Journal of Solid-State Circuits* **53**, 968–982 (2018).

8. Jouppi, N. P. et al. In-datacenter performance analysis of a tensor processing unit. In *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, 1–12 (2017).

9. Xu, X. et al. Scaling for edge inference of deep neural networks. *Nature Electronics* **1**, 216–222 (2018).

10. Duarte, J. et al. Fast inference of deep neural networks in FPGAs for particle physics. *Journal of Instrumentation* **13**, P07027–P07027 (2018).

11. Petillo, C. E. et al. LinkKS: discovering galaxy-scale strong lenses in the kilo-degree survey using convolutional neural networks. *Monthly Notices of the Royal Astronomical Society* **484**, 3879–3896 (2019).

12. Ciprianović, A., Snyder, G., Nord, B. & Peak, J. Deepemerge: Classifying high-redshift merging galaxies with deep neural networks. *Astronomy and Computing* **100390** (2020).

13. Jwa, Y., Di Guglielmo, G., Carloni, L. P. & Karagiorgi, G. Accelerating deep neural networks for real-time data selection for high-resolution imaging particle detectors. In *2019 New York Scientific Data Summit (NYSDS)*, 1–10 (2019).

14. Huerta, E. A. et al. Enabling real-time multi-messenger astrophysics discoveries with deep learning. *Nature Reviews Physics* **1**, 600–608 (2019).

15. Gupta, U. et al. The architectural implications of Facebook’s dnn-based personalized recommendation. In *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 488–501 (2020).

16. Lan, Z. et al. ALBERT: A lite BERT for self-supervised learning of language representations. Preprint at arXiv:1909.11942 (2019).

17. Horowitz, M. Computing’s energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 10–14 (2014).

18. Poulton, J. W. et al. A 1.17-pJ/b, 25-gb/s/pin ground-referenced single-ended serial link for off- and on-package communication using a process- and temperature-adaptive voltage regulator. *IEEE Journal of Solid-State Circuits* **54**, 43–54 (2019).

19. Srivastava, M. et al. Physical insight toward heat transport and an improved electrothermal modeling framework for FinFET architectures. *IEEE Transactions on Electron Devices* **59**, 1353–1363 (2012).

20. Gupta, M. S., Oatley, J. L., Joseph, R., Wei, G. & Brooks, D. M. Understanding voltage variations in chip multiprocessors using a distributed power-delivery network. In *2007 Design, Automation and Test in Europe Conference and Exhibition*, 1–6 (2007).

21. Fowers, J. et al. A configurable cloud-scale dnn processor for real-time AI. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, 1–14 (2018).

22. Shao, Y. S. et al. Simba: Scaling deep-learning inference with multi-chip-module-based architecture. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture - MICRO ’19*, 252–257 (2019).

23. Yin, J. et al. Modular routing design for chiplet-based systems. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, 726–738 (2018).

24. Samajdar, A. et al. A systematic methodology for characterizing scalability of DNN accelerators using SCALE-Sim. In *2019 IEEE International Symposium on Performance Analysis of Systems and Software*, 304–315 (IEEE, 2020).

25. Simonyan, K. & Zisserman, A. In *International Conference on Learning Representations* (2015).

26. Szegedy, C. et al. Going deeper with convolutions. Preprint at arXiv:1409.4842 (2014).

27. Mnih, V. et al. Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).

28. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818–2826 (2016).

29. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778 (2016).

30. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1800–1807 (2017).

31. Vaswani, A. et al. Attention is all you need. In *Advances in Neural Information Processing Systems* 30.5998–6008 (2017).

32. Zoph, B., Vasudevan, V., Shlens, J. & Le, Q. V. Learning transferable architectures for scalable image recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8697–8710 (2018).

33. Hu, J., Shen, L. & Sun, G. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7132–7141 (2018).

34. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. Preprint at arXiv:1810.04805 (2018).

35. Radford, A. et al. Language models are unsupervised multitask learners. *OpenAI Blog* **1** (2019). https://openai.com/blog/better-language-models.

36. Dai, Z. et al. Transformer-XL: attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2978–2988 (Association for Computational Linguistics, Florence, Italy, 2019).

37. Hamerly, R., Bernstein, L., Sludds, A., Soljačić, M. & Englund, D. Large-scale optical neural networks based on ...
photoelectric multiplication. *Physical Review X* **9**, 021032 (2019).

38. Tait, A. N. et al. Neuromorphic photonic networks using silicon photonic weight banks. *Scientific Reports* **7**, 1–10 (2017).

39. Lin, X. et al. All-optical machine learning using diffractive deep neural networks. *Science* **361**, 1004–1008 (2018).

40. Shen, Y. et al. Deep learning with coherent nanophotonic circuits. *Nature Photonics* **11**, 441–446 (2017).

41. Feldmann, J. et al. Parallel convolution processing using an integrated photonic tensor core. Preprint at arXiv:2002.00281 (2020).

42. Sun, J., Timurdogan, E., Yaacobi, A., Hosseini, E. S. & Watts, M. R. Large-scale nanophotonic phased array. *Nature* **493**, 195 (2013).

43. Miller, D. A. B. Attojoule optoelectronics for low-energy information processing and communications. *Journal of Lightwave Technology* **35**, 346–396 (2017).

44. Bauters, J. F. et al. Ultra-low loss silica-based waveguides with millimeter bend radius. In *36th European Conference and Exhibition on Optical Communication*, 1–3 (2010).

45. Iga, K. Vertical-cavity surface-emitting laser: its conception and evolution. *Japanese Journal of Applied Physics* **47**, 1 (2008).

46. Jäger, R. et al. 57% wallplug efficiency oxide-confined 850 nm wavelength GaAs VCSELs. *Electronics Letters* **33**, 330–331 (1997).

47. Zheng, P., Connelly, D., Ding, F. & Liu, T.-J. K. FinFET evolution toward stacked-nanowire FET for CMOS technology scaling. *IEEE Transactions on Electron Devices* **62**, 3945–3950 (2015).

48. Miller, D. A. B. Attojoule optoelectronics for low-energy information processing and communications. *Journal of Lightwave Technology* **35**, 346–396 (2017).

49. Keckler, S. W., Dally, W. J., Khailany, B., Garland, M. & Gasco, D. GPUs and the future of parallel computing. *IEEE Micro* **31**, 7–17 (2011).

50. Dally, W. J. et al. Hardware-enabled artificial intelligence. In *2018 IEEE Symposium on VLSI Circuits* (2018).

51. Stillmaker, A. & Baas, B. Scaling equations for the accurate prediction of CMOS device performance from 180 nm to 7 nm. *Integration* **58**, 74–81 (2017).

52. Chang, J. et al. A 7nm 256Mb SRAM in high-k metal-gate FinFET technology with write-assist circuitry for low-VMIN applications. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, 206–207 (IEEE, 2017).

53. Saadat, H., Bokhari, H. & Parameswaran, S. Minimally biased multipliers for approximate integer and floating-point multiplication. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **37**, 2623–2635 (2018).

54. Shoba, M. & Nakkeeran, R. Energy and area efficient hierarchy multiplier architecture based on Vedic mathematics and GDI logic. *Engineering Science and Technology, an International Journal* **20**, 321–331 (2017).

55. Ravi, S., Patel, A., Shabaz, M., Chaniyara, P. M. & Kittur, H. M. Design of low-power multiplier using UCSLA technique. In *Artificial Intelligence and Evolutionary Algorithms in Engineering Systems*, 119–126 (2015).

56. Johnson, J. Rethinking floating point for deep learning. Preprint at arXiv:1811.01721 (2018).

57. A. Z. et al. A comprehensive survey on graph neural networks. *IEEE Transactions on Knowledge and Data Engineering Systems* 1–21 (2020).

58. Zhang, Z., Cui, F. & Zhu, W. Deep learning on graphs: a survey. *IEEE Transactions on Knowledge and Data Engineering* **1–12** (2020).

59. Matsoukas, P. et al. MLPerf: An industry standard benchmark suite for machine learning performance. *IEEE Micro* **40**, 8–16 (2020).

60. Parashar, A. et al. Timeloop: A systematic approach to DNN accelerator evaluation. In *2019 IEEE International Symposium on Performance Analysis of Systems and Software*, 304–315 (IEEE, 2019).

61. Jacob, B. & Warden, P. et al. gemmlowp: a small self-contained low-precision GEMM library. https://github.com/google/gemmlowp (2015, accessed 2020).

62. Judd, P., Albericio, J., Hetherington, T., Aamodt, T. M. & Moshovos, A. Stripes: Bit-serial deep neural network computing. In *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 1–12 (2016).

63. Albericio, J. et al. Bit-pragmatic deep neural network computing. In *2017 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 382–394 (2017).

64. Coimbatore Balram, K., Audet, R. & Miller, D. Nanoscale resonant-cavity-enhanced germanium photodetectors with lithographically defined spectral response for improved performance at telecommunications wavelengths. *Optics express* **21**, 10228–33 (2013).

Acknowledgements

Thanks to Christopher Panuski for helpful discussions about µLEDs and Angshuman Parashar and Yannan (Nellie) Wu for insights into all-electronic DNN accelerators. We also would like to thank Mohamed Ibrahim for useful discussions on receiver discharging circuits. Anthony Pennes helped with several machining tasks. Thanks to Ronald Davis III and Zhen Guo for manuscript revisions. We also thank the NVIDIA Corporation for the donation of the Tesla K40 GPU used for training the fully-connected networks. Equipment was purchased thanks to the U.S. Army
Research Office through the Institute for Soldier Nanotechnologies (ISN) at MIT under grant no. W911NF-18-2-0048. L.B. is supported by a Postgraduate Scholarship from the Natural Sciences and Engineering Research Council of Canada, National Science Foundation (NSF) E2CDA grant no. 1640012 and the afore-mentioned ISN grant. A.S. is supported by an NSF Graduate Research Fellowship Program under grant no. 1122374, NTT Research Inc., NSF EAGER program grant no. 1946967, and the NSF/SRC E2CDA and ISN grants mentioned above. R.H. was supported by an Intelligence Community Postdoctoral Research Fellowship at MIT, administered by ORISE through the U.S. DoE / ODNI.

Author contributions
D.E. and R.H. developed the original concept. L.B. designed and performed the hardware experiments with the support of A.S. and D.E. A.S. developed the data acquisition, training, and confusion matrix analysis software. L.B. developed the output image processing software and performed the bit error rate calculations. L.B. and A.S. performed the energy calculations, with critical insights from R.H. J.E. and V.S. provided critical insights into all-electronic hardware comparisons. L.B. and A.S. wrote the manuscript with input from all authors. R.H., J.E., V.S. and D.E. supervised the project.

Competing interests
The authors declare no competing interests.

Additional information

Correspondence
Correspondence and requests for materials should be addressed to L.B., A.S. or D.E.

Supplementary information
Supplementary information is available for this paper.
Freely scalable and reconfigurable optical hardware for deep learning: supplementary material

LIANE BERNSTEIN1,†,5, ALEXANDER SLUDDS1,†,6, RYAN HAMERLY1,2, VIVIENNE SZE1, JOEL EMER3,4, AND DIRK ENGLUND1,7

1Research Laboratory of Electronics, Massachusetts Institute of Technology, 50 Vassar St, Cambridge, MA 02139, USA
2NTT Research, 1950 University Ave #600, East Palo Alto, CA 94303, USA
3NVIDIA, 2 Technology Park Drive, Westford, MA 01886, USA
4Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 32 Vassar St, Cambridge, MA 02139, USA
†These authors contributed equally to this work
5e-mail: lbern@mit.edu
6e-mail: asludds@mit.edu
7e-mail: englund@mit.edu

Compiled June 25, 2020

SUPPLEMENTARY NOTE 1: BIT ERROR RATE DUE TO CROSSTALK

Here, we show experimental bit error rate maps for both the blue and red channels. Each DMD pixel is fanned out to a row (column) of superpixels on the camera for the input activations (weights). The Bayer filter allows the discrimination of the input activations from the weights into red and blue channels, respectively. Since the camera has four sub-pixels per superpixel, we bin the sub-pixels into $2 \times 2$ blocks. As described and shown in Fig. 4 of the main text, random vectors of ‘1’s and ‘0’s were displayed on the DMDs to assess bit error rates in data transmission from two 1D source arrays to the camera. In Fig. S1, we show bit error rate maps. Images S1(a) and (c) show the error from a single shot (one random vector pair displayed on the DMDs). Images S1 (b) and (d) show the error averaged over 100 frames (100 different random vector pairs displayed on the DMDs) in the low-error region of interest used for the proof-of-concept experiment. The error is larger on the edges of each image due to optical aberrations, and is larger in the blue channel than the red channel due to misalignment.

SUPPLEMENTARY NOTE 2: CROSSTALK CORRECTION

The bit error rate described in the previous section is mainly attributable to optical crosstalk at the detector, due to imperfect lenses and alignment. Since this error is deterministic (as opposed to random fluctuations), it can be compensated by post-processing. To illustrate this principle, we performed simple crosstalk correction: we multiplied each line of an image detected on the camera by a tridiagonal crosstalk reduction matrix, per equation (S1) (where $I_n$ is the corrected line of the camera image).

To maximize energy efficiency and throughput, the final version of this system (with a custom CMOS chip that integrates detection with digital MAC computation) will not perform any post-processing. Instead, we might use a charge-sharing scheme at the transmitters to implement a version of equation (S1). Alternatively, we could simply reduce crosstalk by changing the system design; for example, we could choose to space the PEs further apart or shrink the active region of the detectors to improve the ratio of signal at the current pixel to noise from neighboring pixels.

SUPPLEMENTARY NOTE 3: TRAINING AND TEST SETS

In our proof-of-concept experiment, we performed inference on 500 images using a two-hidden-layer, fully-connected neural network, where each hidden layer had 100 activations. We used TensorFlow’s built-in dataset importer to download the first 500 images in the test set of the MNIST handwritten digit dataset [1], as downloaded from the TensorFlow 2 Keras database. Relevent code can be found in the GitHub repository for user Alexander Sludds (alexsludds):

https://github.com/alexsludds/Digital-Optical-Neural-Network-Code

The model’s weights were pre-trained on an NVIDIA K40 GPU using the entire MNIST training set. Categorical cross-entropy was used as a loss function. Dropout regularized the
SUPPLEMENTARY NOTE 4: ELECTRONIC INTERCONNECT SWITCHING ENERGY IN 0 TO 1 TRANSITIONS

The dynamic switching energy of CMOS devices is the amount of energy required to charge the output capacitance of a CMOS gate. Energy is only consumed in CMOS inverters for low-to-high transitions on the outputs of these gates. Consider the toy circuit model shown in Fig. S3. On the left is a CMOS inverter, and on the right are a low-to-high and high-to-low transitions, respectively. In the low-to-high transition, the PMOS has to switch closed, shorting the output to the supply rail by charging the load capacitance. In the high-to-low transition, the NMOS already has a sufficient drain-to-source voltage from the load capacitance charge, so it can discharge the output without...
Fig. S3. A demonstration of where dynamic energy consumption goes during switching of a CMOS inverter. The circuit, shown left, consists of a stacked NMOS and PMOS device. During an output low to high transition, shown center, charge is deposited on the lumped output capacitance. During an output low to high transition, shown right, that charge from the lumped output capacitance is discharged through the NMOS into ground.

Fig. S4. A proposed circuit for resetting the receiver lumped capacitor model.

There are several circuit methods by which the accumulated charge on the input capacitor can be reset. In the method shown in Fig. S4, we place the NMOS device $\text{NMOS}_{\text{Discharge}}$ between the photodetector and ground and drive the gate with an external reference voltage $V_{\text{ref}}$. The benefit of this solution is that it consumes no dynamic energy when there is no optical input power. However, it has the tradeoff that it requires additional area on chip and, because it is ratioed logic, requires careful design to ensure functionality. The width of $\text{NMOS}_{\text{Discharge}}$ is set such that the accumulated charge on the capacitor generates a voltage high enough to overcome the input threshold of the load (modeled here as a CMOS inverter), but not so small that it cannot dissipate the charge quickly in a single clock cycle. One problem that arises from receiverless photodetection is that a constant steam of '1’s coming into a photodetector without a strong enough $\text{NMOS}_{\text{Discharge}}$ fill causes additional charge to slowly build on the load capacitance. To compensate, we propose a P-N junction diode (Clamp Diode).

SUPPLEMENTARY NOTE 5: RESETTING A ‘RECEIVERLESS’ CIRCUIT

There are several circuit methods by which the accumulated charge on the input capacitor can be reset. In the method shown in Fig. S4, we place the NMOS device $\text{NMOS}_{\text{Discharge}}$ between the photodetector and ground and drive the gate with an external reference voltage $V_{\text{ref}}$. The benefit of this solution is that it consumes no dynamic energy when there is no optical input power. However, it has the tradeoff that it requires additional area on chip and, because it is ratioed logic, requires careful design to ensure functionality. The width of $\text{NMOS}_{\text{Discharge}}$ is set such that the accumulated charge on the capacitor generates a voltage high enough to overcome the input threshold of the load (modeled here as a CMOS inverter), but not so small that it cannot dissipate the charge quickly in a single clock cycle. One problem that arises from receiverless photodetection is that a constant steam of '1’s coming into a photodetector without a strong enough $\text{NMOS}_{\text{Discharge}}$ fill causes additional charge to slowly build on the load capacitance. To compensate, we propose a P-N junction diode (Clamp Diode).

SUPPLEMENTARY NOTE 6: ELECTRONIC REPEATERS

A naive implementation of a repeater is a double inverter. The energy required is $C_1V^2_{\text{DD}}$, since in any transition, one inverter must be making a low-to-high transition and the other a high-to-low transition. As a result, in any ‘flip’ of a repeater, one inverter does not consume energy. Using the values in Table 2 of the main text, the cost of a repeater is 0.06 fJ/bit for an output low-to-high transition. Therefore, even in the worst-case scenario where we place a repeater between every multiplier in an array of abutted 8-bit MAC units, the inter-multiplier interconnect energy cost is larger than that of the repeater.

SUPPLEMENTARY NOTE 7: SHOT AND THERMAL NOISE

In a hypothetical crosstalk-free DONN, the remaining noise sources are thermal (Johnson) and shot noise. To gain insight into whether they would affect classification accuracy, we estimate the ensuing bit error rates (BERs). The detector registers a ‘1’ when $q \geq q_0$ photoelectrons are generated, and a ‘0’ when $q < q_0$, where we assume the threshold charge is set by $q_0 = n_p/2$ electrons. Fig. S5 illustrates the probability distributions of the number of photoelectrons, as well as the probabilities that a ‘0’ is received when a ‘1’ is sent (BER$_1$), and vice-versa (BER$_0$).

Fig. S5. Schematic representation of probability density function of received charge (curves) and bit error rate (shaded region) - not to scale.

In a receiverless photodetector scheme, thermal noise can be approximated as ‘KT/C’ noise [2], with:

$$\sigma_V = \sqrt{\frac{k_B T}{(C_{\text{det}} + C_T)}}$$  \hspace{1cm} (S2)

where $\sigma_V$ is the standard deviation of voltage, $k_B$ is the Boltzmann constant, $T$ is the temperature in Kelvin, $C_{\text{det}}$ is the capacitance of the photodetector, and $C_T$ is the capacitance of the inverter. The temperature depends on quality of heat sinking and proximity to hot spots; from Ref. [3], we assume it is in the range $T \in [300 - 500]$. Using the values from Table 2 of the main text, we find $\sigma_V \approx 5 - 6 \text{ mV} \ll V_{\text{DD}}$. We can further verify whether thermal noise is likely to cause bit errors by approximating the probability distribution due to thermal noise, $p_1(q)$, by a Gaussian:

$$p_1(q) = \frac{1}{\sqrt{2\pi} \sigma_1} e^{-\frac{q^2}{2\sigma_1^2}}$$  \hspace{1cm} (S3)
with \( \sigma_J = \sqrt{k_B T (C_{det} + C_T) / e} \approx 6 - 7 \) electrons.

To first order, shot noise will not affect the transmission of '0's (BER\(_0\)) since the number of transmitted photons is \( n_p = 0 \).

Thus:

\[
\text{BER}_0 = \sum_{q=0}^{\infty} p_0(q) = \sum_{q=0}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{q^2}{2\sigma_J^2}} \approx \frac{1}{2} \text{erfc} \left( \frac{q_0}{\sqrt{2\sigma_J}} \right) \quad (S4)
\]

BER\(_0\) for different \( n_p = 2q_0 \) are reported in Table S1.

We assume shot noise follows a Poissonian probability distribution:

\[
p_{\text{shot}}(q) = \frac{e^{-n_p}(n_p)^q}{q!} \quad (S6)
\]

where \( n_p \) is the number of photons per detector per clock cycle. For ease of computation with large \( n_p \), we take the natural logarithm:

\[
\ln(p_{\text{shot}}(q)) = \ln(e^{-n_p}(n_p)^q) = -n_p + q \ln(n_p) - \ln(q!) \quad (S7)
\]

\[
\uparrow
\]

\[
p_{\text{shot}}(q) = \exp(-n_p + q \ln(n_p) - \sum_{m=1}^{q} \ln(m)) \quad (S11)
\]

BER\(_1\) due to shot noise is therefore:

\[
\text{BER}^{\text{shot}}_1 = \sum_{q=1}^{q_0-1} p_{\text{shot}}(q) \quad (S12)
\]

Results of this computation for various \( n_p \) are shown in Table S1.

**Table S1.** Expected values for BER\(_1\) due to shot noise for different numbers of transmitted photons/bit

| \( n_p \) | \( \text{BER}_0^* \) | \( \text{BER}^{\text{shot}}_1 \) | \( \text{BER}^{\text{total}}_1 \) |
|---|---|---|---|
| 10 | \( 10^{-1} \) | \( 10^{-2} \) | \( 10^{-1} \) |
| 100 | \( 10^{-18} - 10^{-12} \) | \( 10^{-6} \) | \( 10^{-6} - 10^{-5} \) |
| 1000 | small\(^*\) | \( 10^{-69} \) | \( 10^{-65} - 10^{-63} \) |

\(*\)BER\(_0^*\) = BER\(_0^{\text{thermal}}\)

\(^*\)Too small for MATLAB to compute.

Note: We report a range since thermal noise, and therefore BER, depends on quality of heat sinking.

Thermal noise will also contribute to BER\(_1\); we convolve the probability distributions to find the total bit error rate:

\[
\text{BER}^{\text{total}}_1 = \sum_{q=1}^{q_0-1} p_t(q) = \sum_{q=1}^{q_0-1} p_{\text{shot}}(q) \oplus p_t(q) \quad (S13)
\]

From equation (5) in the main text, we find \( n_p \approx 1000 \) photons/bit to generate a voltage swing of 0.8 V on the load capacitance; therefore, the expected BER is negligible, per Table S1.

**REFERENCES**

1. LeCun, Y., Cortes, C. & Burges, C. J. C. MNIST handwritten digit database. [http://yann.lecun.com/exdb/mnist/](http://yann.lecun.com/exdb/mnist/) (1998).

2. Miller, D. A. B. Attojoule optoelectronics for low-energy information processing and communications. *Journal of Lightwave Technology* 35, 346–396 (2017).

3. Shrivastava, M. *et al.* Physical insight toward heat transport and an improved electrothermal modeling framework for finfet architectures. *IEEE Transactions on Electron Devices* 59, 1353–1363 (2012).