IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Training of mixed-signal optical convolutional neural networks with reduced quantization levels

Zheyuan Zhu (Member, IEEE), Joseph Ulseth, Guifang Li (Fellow, IEEE), and Shuo Pang
CREOL, The College of Optics and Photonics, University of Central Florida, Orlando, FL, 32816, USA

Corresponding author: Zheyuan Zhu (e-mail: zyzhu@knights.ucf.edu).

This work was supported in part by the Army Research Office (W911NF1910385) and Office of Naval Research (N00014-20-1-2441).

ABSTRACT. Analog computing paradigms are promising solutions to the growing computational demands of machine learning applications. Despite being susceptible to errors, analog and mixed-signal platforms have the potential to achieve higher speed and power efficiency for artificial neural network (ANN) applications than digital computers. Driven by the development of digital fixed-point ANN accelerators, low-precision ANN models have proven to be successful in compressing the size of ANNs and conforming the models to the data format of digital accelerators. While the inputs and weights of these digital, fixed-point ANN models can have low bit widths, the intermediate results (e.g., activations) must be preserved in high precision. As a result, these digital fixed-point models and training algorithms cannot be migrated easily to analog accelerators, because the analog intermediate results typically suffer from reduced precision due to noises and device imperfections. Here, we report on a training method for mixed-signal ANNs that considers two types of analog impairments, namely, random noise and distortion (deterministic in nature). The results show that mixed-signal ANN trained with our method can achieve the same classification accuracy as the digital fixed-point model with noise levels up to 50% of the ideal quantization step size. We demonstrate our training method on a mixed-signal, convolutional neural network based on diffractive optics.

INDEX TERMS neural network, mixed-signal training, analog computation

I. INTRODUCTION

Artificial neural networks (ANN) are growing larger and deeper [1]–[3] to tackle tasks of increasing complexity [4]–[6]. To accommodate the anticipated computational demands of future neural network structures, specialized computing hardware and data formats have been engineered. Various low-precision or even binary neural networks (BNNs), accompanied by specifically designed training algorithms [7]–[11], have proven to be successful in accelerating inferences and reducing memory footprints [12]–[14] via low-bit-width and fixed-point data formats for weights and inputs. When designing and deploying these networks on digital computers [12], the intermediate results (e.g., activations) are typically cached in a format with higher precision than that of the weights and inputs to achieve the expected performance.

Recently, due to their advantages in speed and power efficiency [15], analog computing paradigms have been considered as solutions to the growing demands in neural network computing, with implementations in both electronics [16], [17] and photonics [18], [19]. However, analog computing is susceptible to ambient noise and device imperfections [20]. Ex-situ training has been deployed on a simulated analog unit using a fixed-point data format [21], which is analogous to a low-bit-width neural network using a digital computer. However, a model trained by such a method is likely to have an inferior inference performance [22], as analog intermediate results cannot match the full precision of those obtained with a digital computer. To overcome this performance degradation, fine-tuning of the analog parameters on each computation node [17], [23] can be performed, although doing so requires an exhaustive effort. There is currently no efficient training method that is robust to the errors on analog ANNs.

In this work, we incorporate two types of common impairments in analog processors – random noise and deterministic distortion – into the training process, extending low-precision neural network training to mixed-signal or analog computing platforms. The network trained with our method is robust against analog signal noise levels as high as 50% of the quantization step, indicating that a mixed-signal neural network can operate at a reduced quantization level. We

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3072193, IEEE Access

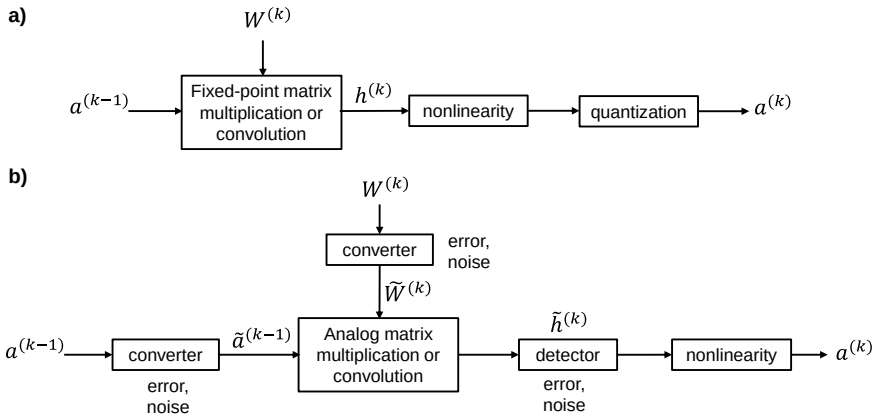


Figure 1. Computational schemes for (a) a digital fixed-point neural network layer and (b) a neural network layer with an analog acceleration unit. Variables with tildes, $\tilde{a}^{(k-1)}$, $\widetilde{W}^{(k)}$, and $\tilde{h}^{(k)}$ are analog signals.

demonstrate a trained model on a programmable optical convolutional neural network.

II. THEORY

A. Overview of low-precision ANN

Low-precision neural networks perform matrix multiplications and convolutions between fixed-point inputs and weights, which are the required data format for many digital tensor processing units [24]. Fig. 1(a) illustrates the computational scheme for a low-precision neural network layer. A fixed-point processor computes the activations $h^{(k)}$ from the quantized inputs $a^{(k)}$ and weights $W^{(k)}$ with Eq. (1),

$$h^{(k)} = W^{(k)} \cdot a^{(k-1)}, \tag{1}$$

where $\cdot$ denotes matrix multiplication or convolution. The activations $h^{(k)}$ require higher bit widths than the inputs and weights due to the associated accumulation process [12]. A nonlinear function $g(\cdot)$ is then applied to the activations, along with a quantization operation, to match the input data format of the next layer, i.e.:

$$a^{(k)} = quantize\left(g\left(h^{(k)}\right)\right). \tag{2}$$

Here, $g(\cdot)$ can be a neural network layer operation, such as batch normalization, down-sampling, activation, and so on, or a combination of multiple operations. The quantization of the tensor $\boldsymbol{x}$ operates on each tensor element x_i according to Eq. (3),

$$quantize(x_i) = \begin{cases} \left[\frac{x_i}{2^{L-m-1}}\right] 2^{L-m-1}, |x_i| < (2^m - 2^{m+1-L}) \\ sign(x_i)(2^m - 2^{m+1-L}), |x_i| \geq (2^m - 2^{m+1-L}) \end{cases}, \tag{3}$$

where L is the total bit width, m is the shared exponent among all tensor elements, and $[\cdot]$ denotes rounding to the nearest integer. Several nonlinearities, such as clipping and scaling functions, have been purposefully designed for easier integration with the quantization operation [25], [26].

B. Mixed-signal ANN layer with an analog accelerator

A growing number of neural network implementations have replaced traditional digital computers with their analog counterparts for speed and power efficiency [17]–[19]. Fig. 1 (b) illustrates the computational scheme for a mixed-signal neural network layer with an analog acceleration unit. A set of digital inputs $a^{(k-1)}$ and weights $W^{(k)}$ are sent to their corresponding digital-to-analog convertors (DACs) and/or modulators, generating the analog signal representing the inputs $\tilde{a}^{(k-1)}$ and $\widetilde{W}^{(k)}$, respectively. The activations $\tilde{a}^{(k-1)} \cdot \widetilde{W}^{(k)}$ from an analog acceleration unit are collected by a detector, producing signals $\tilde{h}^{(k)}$. The detected signals are then sent to a digital processing unit, which maps the activations to the inputs of the next layer via the activation function $a^{(k)} = g\left(\tilde{h}^{(k)}\right)$. The activation function can include ANN nonlinearities, such as ReLU and sigmoid, as well as quantization operations to match the input data format of the next layer. Computational errors of an analog acceleration unit include deterministic errors and random noises. These two types of impairments can be present in any analog signal, i.e., $\tilde{a}^{(k-1)}$, $\widetilde{W}^{(k)}$, and $\tilde{h}^{(k)}$.

Deterministic errors can originate from the response function $f(\cdot)$ of the modulators, the DACs, or the detectors. For a continuous output, the signal $\tilde{x}$, which represents the distortion of the ideal signal x is given by Eq. (4),

$$\tilde{x} = f(x). \tag{4}$$

Examples of continuous deterministic errors include the gamma curves of the detector, and the sinusoidal relation between the intensity and phase of an interferometry-based intensity modulator [27]. A discrete deterministic error maps tensor x to a set of values, i.e.:

$$\tilde{x} = f(x) = \underset{x' \in \mathbf{X}'}{\operatorname{argmin}} |x' - x|_1, \tag{5}$$

where $\mathbf{X}'$ is the set of discrete values available to the accelerator hardware, and $|\cdot|_1$ represents the L1 norm. Examples of discrete deterministic errors include the effects of DACs and analog-to-digital convertors (ADCs) that can only generate or digitize a fixed number of voltage levels. The quantization in digital low-precision or binary networks can be considered as a special case of Eq. (5), where the set $\mathbf{X}'$ is $\{\pm i \cdot 2^{m+1-L}; i = 0,1, \ldots, 2^{L-1} - 1\}$ for a fixed-point quantization with bit-width L and exponent m.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3072193, IEEE Access

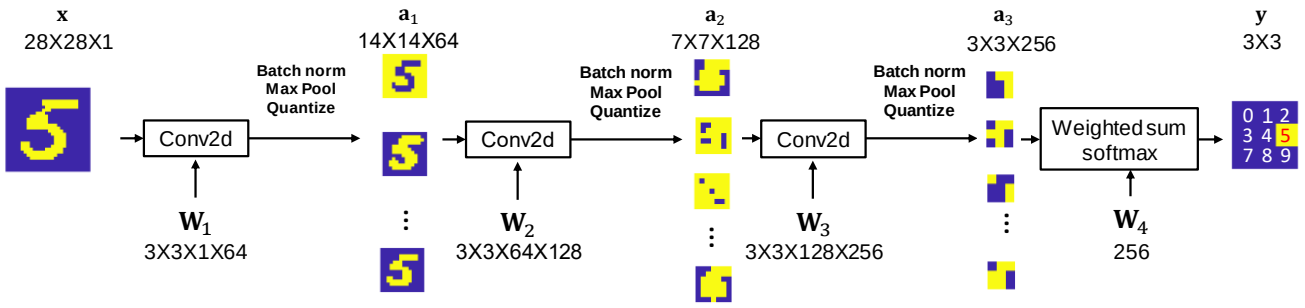


Figure 2. Structure of the convolutional neural network used in the simulation.

Noise is modeled as a random variable on top of an ideal signal x. The signal corrupted by noise ε can be expressed by Eq. (6),

$$\tilde{x} = x + \varepsilon, \tag{6}$$

where ε is assumed to follow an unbiased distribution. If the random noise in an experimental platform introduces a bias to the signal, this bias can be merged into the deterministic error. Notice that random noise and deterministic errors can be combined to model the errors associated with any analog signal in a practical analog acceleration unit.

C. Training of mixed-signal neural networks

Our proposed training method for mixed-signal ANN considers the two types of impairments described above in both the forward pass and the gradient backpropagation during the training process. The gradient flow through Eq. (6) can be computed using the noisy instance of the tensor, $\tilde{x}$, in the forward pass [28], as in Eq. (7),

$$\frac{\partial l}{\partial x} \leftarrow \frac{\partial l}{\partial \tilde{x}}. \tag{7}$$

The gradient flow through the deterministic error process $\tilde{x} = f(x)$ (Eq.(4)) involves the derivate of $f(\cdot)$, i.e.:

$$\frac{\partial l}{\partial x} \leftarrow \frac{\partial l}{\partial \tilde{x}} f'(x). \tag{8}$$

The derivate of a continuous nonlinear response $f(\cdot)$ is readily available. When the output of $f(\cdot)$ is discrete, the gradient is 0 almost everywhere since $f(\cdot)$ is piecewise constant. To preserve the gradient flow, we use a gradient clipping method similar to that for BNN [8] in Eq. (9),

$$\frac{\partial l}{\partial x} \leftarrow \frac{\partial l}{\partial \tilde{x}} \mathbf{1}_{\theta_1 < x < \theta_2}, \tag{9}$$

where $\partial l / \partial \tilde{x}$ is the gradient with respect to the distorted tensor $\tilde{x}$, and $\mathbf{1}_{\theta_1 < x < \theta_2}$ denotes a binary tensor with the same shape as the ideal tensor x, which has value 1 for elements of x within the range (θ_1, θ_2), and 0 for elements of x outside the range (θ_1, θ_2). Here, θ_1 and θ_2 are typically chosen to represent the output range of $f(\cdot)$. For the special case of binarization, θ_1 and θ_2 are -1 and 1 (or 0 and 1 if $\mathbf{X}' = \{0,1\}$ in Eq.(5)), respectively.

III. Mixed-signal convolution neural network simulation

A. Network structure

We have constructed a mixed-signal convolutional neural network (MCNN) that classifies digits with binary inputs and kernels in all layers, shown in Fig. 2. The MCNN consists of convolutional layers only to facilitate its deployment on a mixed-signal diffractive-optics-based system (Sec. IV). The input digit from the MNIST dataset (28×28) is gradually down-sampled to a 3×3 matrix, in which each element representing the probability of a digit being 1 of the 9 labels. The first layer convolves the 28×28 input image with 64 3×3 kernels and outputs a 64-channel activation tensor with size 28×28×64. The 64 channels are then individually batch-normalized and max-pooled with 2×2 down-sampling to a 14×14×64 tensor as the input of layer 2. Layers 2 and 3 use the same convolution and post-processing operations, with the exception that the numbers of kernels used are 128 and 256, respectively. The input of layer 4 is a 3×3×256 tensor that has been down-sampled from the layer 3 activations by extracting the 2nd, 5th, and 7th elements along the horizontal and vertical spatial dimensions. Layer 4 computes a weighted sum of all 256 channels, applies the softmax activation function, and outputs a final 3×3 matrix. Because this MCNN can produce only 9 possible labels, we excluded the digit '6.'

The impairments that we consider in this MCNN simulation are the discrete deterministic errors of the inputs and weights, as well the random noise of the detector. In layer k, the analog inputs $\tilde{a}^{(k)}$ and weights $\widetilde{W}^{(k)}$ produced by the convertors from the ideal, digital values $a^{(k)}$ and $W^{(k)}$ are:

$$\begin{aligned} \tilde{a}^{(k)} &= f_a\left(a^{(k)}\right) = \underset{a' \in \mathbf{A}'}{\operatorname{argmin}} \left|a' - a^{(k)}\right|_1, \\ \widetilde{W}^{(k)} &= f_W\left(W^{(k)}\right) = \underset{W' \in \mathbf{W}'}{\operatorname{argmin}} \left|W' - W^{(k)}\right|_1, \end{aligned} \tag{10}$$

respectively. Here $\mathbf{A}'$ and $\mathbf{W}'$ are the sets of discrete input and weight tensors, respectively. For an input with M×N pixels and Q input channels, the activations in a CNN convolution with 3×3 kernels and C output channels are computed by an analog accelerator as:

$$\tilde{h}_{i,j}^{(k,q,c)} = \sum_{m=i-1}^{i+1} \sum_{n=j-1}^{j+1} \tilde{a}_{m,n}^{(k-1,q)} \widetilde{W}_{m-i+2,n-j+2}^{(k,q,c)} + \varepsilon, \tag{11}$$

where i, j denote the pixel indexes of the convolutional; m, n are the pixel indexes of the input image; q is the index of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3072193, IEEE Access

input channels; c is the index of the output channels; and ε denotes the random additive noise, which is modeled by an unbiased Gaussian distribution, i.e., $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. Here, we assume that the inputs are zero-padded. For simplicity, we omit the pixel and channel indexes in the tensor when they are not ambiguous. The activations $\tilde{h}^{(k,q,c)}$ then undergoes digital post-processing, which consists of batch normalization and 2×2 max pooling, as follows:

$$a^{(k)} = g\left(\tilde{h}^{(k,q,c)}\right) = MaxPool\left(BatchNorm\left(S\left(\tilde{h}^{(k,q,c)}\right)\right)\right). \quad (12)$$

Here, S represents the summation over all input channels q of $\tilde{h}^{(k,q,c)}$ for each applied kernel c. Let $\tilde{H}^{(k,c)}$ denote the aggregate results along the dimension, q,

$$\tilde{H}^{(k,c)} = S\left(\tilde{h}^{(k)}\right) = \sum_{q=1}^{Q_k} \tilde{h}^{(k,q,c)}, \quad (13)$$

the operation $BatchNorm(\cdot)$ is performed channel-wise on $\tilde{H}^{(k,c)}$ in Eq. (14),

$$BatchNorm\left(\tilde{H}^{(k,c)}\right) = \gamma^{(k,c)}\left(\frac{\tilde{H}^{(k,c)} - \mu^{(k,c)}}{\sigma^{(k,c)}}\right), \quad (14)$$

where $\mu^{(k,c)}$, $\sigma^{(k,c)}$ are the mean and standard deviation of all the pixels in $\tilde{H}^{(k,c)}$ in channel c, and $\gamma^{(k,c)}$ is a trainable parameter. After $BatchNorm$, the features are binarized to obtain the input of the next layer.

B. Training of MCNN

The random noises and deterministic errors are both quantified by the root mean square error (RMSE), which measures the average deviation per dimension of the tensor, as in Eq. (15),

$$RMSE = \sqrt{\frac{|\tilde{x} - x|_2^2}{\dim(x)}}. \quad (15)$$

Here, $\tilde{x}$ is the tensor x corrupted by errors; $\dim(x)$ is the total number of elements in tensor x, and $|\cdot|_2$ denotes the L2-norm. If $\tilde{x}$ is corrupted by unbiased Gaussian noise ($\varepsilon \sim \mathcal{N}(0, \sigma^2)$), the RMSE reduces to the standard deviation, σ.

The MCNN was trained by considering the binary inputs $\mathbf{A}'$, the kernel sets $\mathbf{W}'$ that can be displayed in the experiment, and a noise level σ=0.5 for ε. These parameters were selected to match the experimental MCNN setup. For comparison, we also trained a reference MCNN model with the same structure using the BNN training method in Ref. [8], but without considering the random noise term ε. The binarization on the kernels in BNN training was replaced with rounding to the nearest experimental kernels, as in Eq. (10). After training, we tested the accuracy of the trained MCNN using the MNIST test digits under various levels of simulated Gaussian noise on the activations. For each Gaussian noise level σ, we ran seven noisy inference instances by randomly sampling ε from $\mathcal{N}(0, \sigma^2)$ to obtain the mean and standard deviation of the inference accuracy.

C. Inference simulation of MCNN

Fig. 3 plots the inference accuracy at various noise levels, as quantified by the RMSE, for the MCNNs with our training method and that of BNN. The MCNN trained with our method maintained the inference accuracy up to σ=0.5. The accuracy was 75.0±3.2% for our method and 47.3±3.1% for BNN training method. These results show that adding random noise during training has improved the performance in a mixed-signal scenario, an effect similar to the regularization of the neural network parameters [29]. Note that we trained the MCNN off-line by modeling the analog computations using random noise and the deterministic errors in the forward and backpropagation processes. An in-situ [16] forward pass through the physical MCNN setup can leverage the full potential of the speed and efficiency provided by the analog accelerator unit.

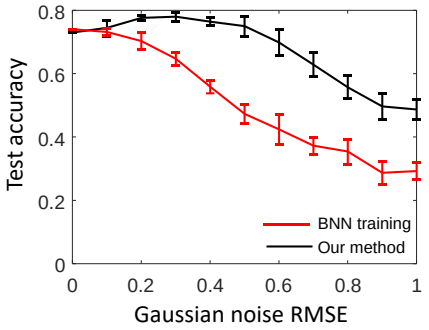


Figure 3. Classification accuracy as a function of the noise RMSE added in the inference simulation.

Fig. 4 exemplifies the layer-by-layer activations of two inference instances in Fig. 3 at σ=0.5 for the input digit '5.' The probabilities of the input digit being classified as '5' were 99.2% and 83.1%, respectively, for the MCNN trained with our method and that of BNN. Although the MCNN trained with the BNN method classifies this digit correctly, the probability of correct identification is reduced and confusions with the digits '0', '3', and '8' can be observed in its output.

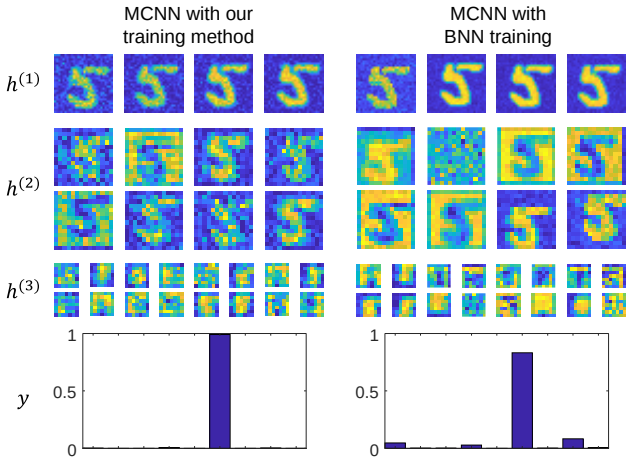


Figure 4. Layer-by-layer activations of the MCNNs trained with BNN and our method for the input digit 5.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3072193, IEEE Access

Our treatment of the analog computation noise is similar to stochastic quantization [12] to a lower precision level on a digital computer. Table 1 shows the inference accuracies of the two MCNNs on a simulated digital low-bit-width accelerator. We kept the same range of the activations, while stochastically quantizing them to 3, 2, and 1 bit(s), corresponding to 8, 4, and 2 quantization levels, respectively. For a binary input convolving with 3×3 binary kernels, the ideal activations range from 0 to 9. The fluctuation due to unbiased Gaussian noise with σ=0.5 can range from -1.0 to 1.0, considering the 95% confidence interval. The MCNN trained with our method maintained its accuracy at 2 bits (4 levels), indicating that the mixed-signal neural network can operate at a reduced quantization level of the intermediate results.

TABLE 1: INFERENCE ACCURACY OF THE MCNN WITH REDUCED ACTIVATION QUANTIZATION BIT WIDTHS

Quantization bit width	MCNN trained with our method	MCNN trained with backpropagation in BNN
3	75.4±2.5%	72.4±3.1%
2	70.4±2.5%	62.1±3.5%
1	37.0±2.1%	26.3±3.9%

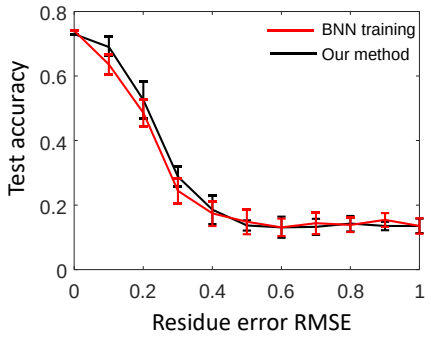


Figure 5. Accuracy vs. RMSE of residue deterministic errors added in the simulated inference process.

Quantization on digital, fixed-point neural networks is often performed stochastically to avoid introducing the quantization bias, which is undesirable in low-precision neural networks. Likewise, if there are some distortions uncorrected for in the training of the MCNN, the residue deterministic error will introduce a cumulative bias, increasing through the layers. Fig. 5 plots the inference accuracy versus the RMSE of the residue error for the two MCNNs trained with our method and the BNN method. The drop in the accuracy as the residue error increases is consistent with the results on digital platforms [8], [12], [30], indicating that our training is still sensitive to the bias from uncorrected deterministic errors.

IV. MCNN experiment

Many of the existing diffractive optics-based neural networks employ pre-recorded diffractive optical elements to represent a trained set of weights [31], [32]; hence, they cannot be re-programmed easily. Here, we constructed a fully programmable optical mixed-signal convolutional network layer based on a 4f system for the deployment of the trained MCNN model. The layer input uses a digital mirror device (DMD, ViALUX, V4100 DLP7000, pixel size 13.7 μm). The analog convolution is performed by a phase-only spatial light modulator (SLM, Meadowlark Optics, P1920-400-800-HDMI, pixel size 9.2 μm) on the Fourier plane, as shown in Fig. 6(a). The DMD is illuminated by a collimated beam from a 12 mW laser source (Coherent, OBIS LX $\lambda = 488nm$). Each element of each input channel, $\tilde{a}_{m,n}^{(k-1,q)}$, is represented by one DMD pixel, in either the on or off state. The light field then passes through a $200mm$ tube lens (Thorlabs TTL-200) L_1 that creates a Fourier transform (FT) of the input onto the SLM. The FT of the kernel $\widetilde{W}^{(k,q,c)}$, approximated as phase only, is loaded onto the SLM. Upon reflection off the SLM, the FT of the input is multiplied by the FT of the kernel, thereby implementing the analog convolution in the frequency domain. A beam splitter directs the reflected beam from the SLM through lens L_2 (identical to L_1), performing the inversion FT to yield the desired convolution between the input and the kernel, which is captured by a camera (JAI Ltd., GO-5000M-USB camera, 5.0 μm pitch). To implement the CNN operation in Eq.(11), the kernels must be flipped horizontally and vertically before use.

The input patterns $\tilde{a}^{(k)}$ that can be displayed are strictly binary due to the use of the DMD as the input device; hence, $\mathbf{A}' \in \{0,1\}$. The use of phase-only masks [33] to approximate a complex Fourier filter gives rise to the distortions in the kernels. For 3×3 binary kernels, there are a total of 511 non-trivial kernels. We pre-calculated the 511 phase masks needed to display all the non-trivial kernels. Due to the experimental artifacts and approximations, the actual 511 kernels, $\mathbf{W}'$, displayed in the experiment are not strictly binary. The distorted kernels were calibrated by imaging a single pixel displayed on the DMD through the optical system for each phase mask.

Due to aberrations and the limited numerical aperture of the 4f system, the full-width-at-half-maximum (FWHM) of its point spread function (PSF) is about four camera pixels. To mitigate crosstalk due to the PSF, we introduced a three-pixel separation between adjacent samples of the input on the DMD and an eight-pixel separation for the kernels on the SLM, accounting for the differences between the pixel sizes of the DMD and the camera. To take advantage of the spatial bandwidth of the 4f system, we tiled multiple input channels in 2×2 and 4×4 formations for layer 2 and layer 3, respectively. Fig. 6 (b) shows the tiled input on the DMD. After the camera acquired the raw image, we performed an 8×8 down-sampling and separated the tiled channels to recover $\tilde{h}$ in its native spatial resolution.

Fig. 6(c) shows the layer-by-layer activations of the MCNN model trained with our method while classifying the input digit '5.' With the calibrated kernel set $\mathbf{W}'$, the ideal convolution between the DMD input and the actual kernels in each layer can be computed. We compared the ideal convolution results with the activations obtained from the raw

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3072193, IEEE Access

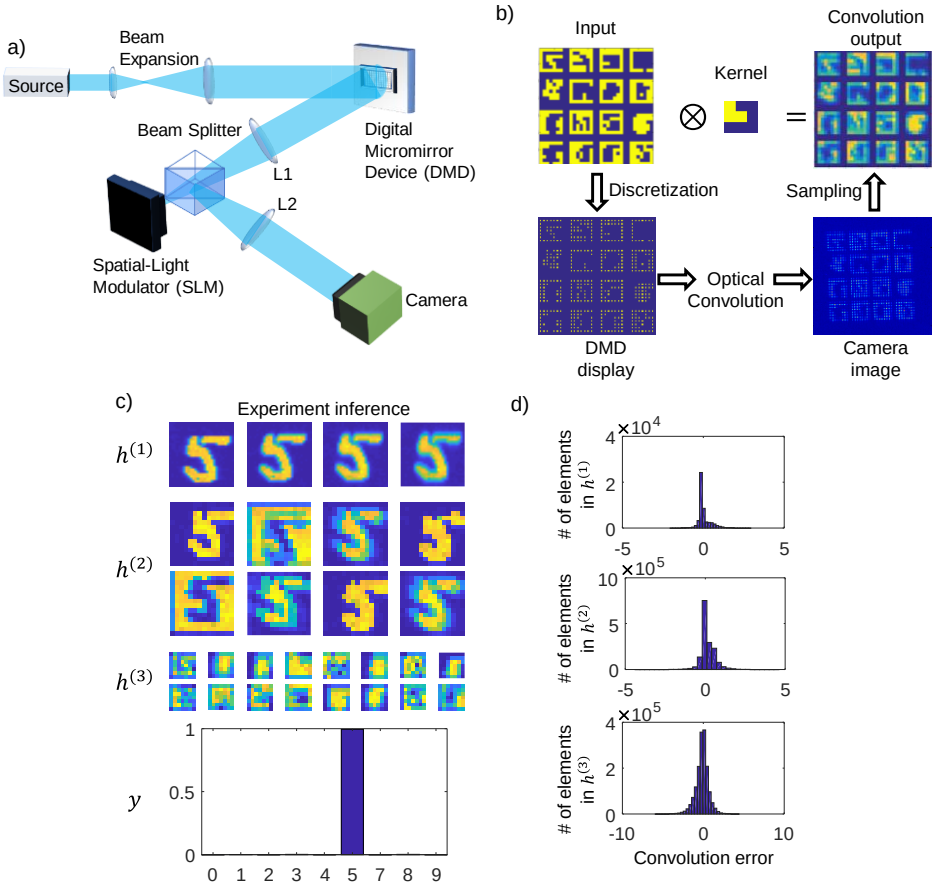


Figure 6. ***(a) Optical setup for implementing each MCNN layer. (b) Parallelized computation of the convolution between multiple input channels in layer 3 and the kernel. (c) Activations and outputs in each MCNN layer for the input digit '5'. (d) Histogram of the errors between ideal (convolution between DMD inputs and calibrated kernels) and experimental activations (down sampled from raw camera images) in each layer.***

camera images. The distributions of the errors for the activation tensor h for all the MCNN layers in the experiment are plotted in Fig. 6(d). These distributions all resemble the Gaussian shape with standard deviations σ=0.37, 0.38, and 0.58 for layers 1 through 3, respectively. The error in each layer is consistent with our choice of the random noise with σ=0.5 in the MCNN simulation. Despite the presence of this activation error, the MCNN trained with our method achieved the correct identification.

V. Summary

We have demonstrated a training method that incorporates analog computational errors in neural network training for deployment on mixed-signal computational platforms. Compared with a neural network trained using conventional backpropagation, mixed-signal neural networks trained with our method are robust against a noise RMSE of 0.5 quantization level in the analog computing process, and thus can tolerate the reduced precision of the activations. Maintaining the inference performance at approximately half the precision levels of the device specification allows us to deploy a trained convolutional neural network on a mixed-signal, diffractive-optics-based convolution system that contains computation errors and kernel distortions.

Another finding of the mixed-signal neural network is its sensitivity to uncorrected deterministic errors, which introduce cumulative bias throughout the layers. These errors can be reduced with integrated optical components for improved stability. In addition, incorporating weights regularization in the design and training of mixed-signal neural networks can reduce the reliance on kernels that tend to exhibit large errors in experiment, thus improving the tolerance to analog computing errors.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv Prepr. arXiv1409.1556*, 2014.

[3] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[5] R. Collobert and J. Weston, "A unified architecture for natural language processing," in *Proceedings of the 25th international conference on Machine learning - ICML '08*, 2008, pp. 160–167, doi: 10.1145/1390156.1390177.

[6] D. Silver *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science (80-.).*,

vol. 362, no. 6419, pp. 1140–1144, Dec. 2018, doi: 10.1126/science.aar6404.

[7] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks BT - Computer Vision – ECCV 2016," 2016, pp. 525–542.

[8] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *J. Mach. Learn. Res.*, vol. 18, pp. 1–30, 2018.

[9] F. Li, B. Zhang, and B. Liu, "Ternary Weight Networks," *arXiv*, no. Nips, May 2016.

[10] C. Leng, H. Li, S. Zhu, and R. Jin, "Extremely Low Bit Neural Network: Squeeze the Last Bit Out with ADMM," *arXiv Prepr. arXiv1707.09870*, Jul. 2017.

[11] N. Mellempudi, A. Kundu, D. Mudigere, D. Das, B. Kaul, and P. Dubey, "Ternary Neural Networks with Fine-Grained Quantization," *arXiv Prepr. arXiv1705.01462*, May 2017.

[12] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," *32nd Int. Conf. Mach. Learn. ICML 2015*, vol. 3, pp. 1737–1746, 2015.

[13] A. Fan *et al.*, "Training with Quantization Noise for Extreme Model Compression," *arXiv Prepr. arXiv2004.07320*, pp. 1–18, Apr. 2020.

[14] V. Sze, Y. Chen, T. Yang, and J. S. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017, doi: 10.1109/JPROC.2017.2761740.

[15] W. Haensch, T. Gokmen, and R. Puri, "The Next Generation of Deep Learning Hardware: Analog Computing," *Proc. IEEE*, vol. 107, no. 1, pp. 108–122, Jan. 2019, doi: 10.1109/JPROC.2018.2871057.

[16] C. Li *et al.*, "Efficient and self-adaptive in-situ learning in multilayer memristor neural networks," *Nat. Commun.*, vol. 9, no. 1, pp. 7–14, 2018, doi: 10.1038/s41467-018-04484-2.

[17] P. Yao *et al.*, "Fully hardware-implemented memristor convolutional neural network," *Nature*, vol. 577, no. 7792, pp. 641–646, Jan. 2020, doi: 10.1038/s41586-020-1942-4.

[18] R. Hamerly, L. Bernstein, A. Sludds, M. Soljačić, and D. Englund, "Large-Scale Optical Neural Networks Based on Photoelectric Multiplication," *Phys. Rev. X*, vol. 9, no. 2, pp. 1–12, 2019, doi: 10.1103/physrevx.9.021032.

[19] Y. Shen *et al.*, "Deep learning with coherent nanophotonic circuits," *Nat. Photonics*, vol. 11, no. 7, pp. 441–446, 2017, doi: 10.1038/nphoton.2017.93.

[20] H. Tsai, S. Ambrogio, P. Narayanan, R. M. Shelby, and G. W. Burr, "Recent progress in analog memory-based accelerators for deep learning," *J. Phys. D. Appl. Phys.*, vol. 51, no. 28, 2018, doi: 10.1088/1361-6463/aac8a5.

[21] Y. Cai *et al.*, "Training low bitwidth convolutional neural network on RRAM," *Proc. Asia South Pacific Des. Autom. Conf. ASP-DAC*, vol. 2018-Janua, pp. 117–122, 2018, doi: 10.1109/ASPDAC.2018.8297292.

[22] S. Moon, K. Shin, and D. Jeon, "Enhancing Reliability of Analog Neural Network Processors," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 27, no. 6, pp. 1455–1459, 2019, doi: 10.1109/TVLSI.2019.2893256.

[23] M. Cheng *et al.*, "TIME: A training-in-memory architecture for RRAM-based deep neural networks," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 38, no. 5, pp. 834–847, 2019, doi: 10.1109/TCAD.2018.2824304.

[24] N. P. Jouppi *et al.*, "In-Datacenter Performance Analysis of a Tensor Processing Unit," in *Proceedings of the 44th Annual International Symposium on Computer Architecture - ISCA '17*, 2017, pp. 1–12, doi: 10.1145/3079856.3080246.

[25] J. Choi, Z. Wang, S. Venkataramani, P. I.-J. Chuang, V. Srinivasan, and K. Gopalakrishnan, "PACT: Parameterized Clipping Activation for Quantized Neural Networks," pp. 1–15, 2018.

[26] S. Jung *et al.*, "Learning to quantize deep networks by optimizing quantization intervals with task loss," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 4345–4354, 2019, doi: 10.1109/CVPR.2019.00448.

[27] B. E. A. Saleh and M. C. Teich, *Fundamentals of photonics*. john Wiley & sons, 2019.

[28] Y. Bengio, N. Léonard, and A. Courville, "Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation," pp. 1–12, Aug. 2013.

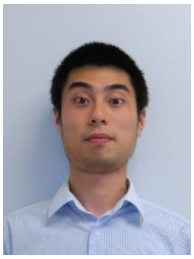
[29] C. M. Bishop, "Training with Noise is Equivalent to Tikhonov Regularization," *Neural Comput.*, vol. 7, no. 1, pp. 108–116, 1995, doi: 10.1162/neco.1995.7.1.108.

[30] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "DoReFa-Net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients," *arXiv Prepr. arXiv1606.06160*, vol. 1, no. 1, pp. 1–13, Jun. 2016.

[31] J. Chang, V. Sitzmann, X. Dun, W. Heidrich, and G. Wetzstein, "Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification," *Sci. Rep.*, vol. 8, no. 1, p. 12324, Dec. 2018, doi: 10.1038/s41598-018-30619-y.

[32] T. Yan *et al.*, "Fourier-space Diffractive Deep Neural Network," *Phys. Rev. Lett.*, vol. 123, no. 2, p. 023901, Jul. 2019, doi: 10.1103/PhysRevLett.123.023901.

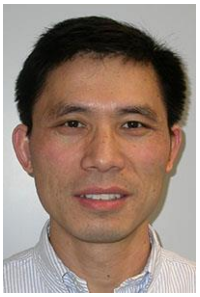
[33] D. Mendlovic, G. Shabtay, U. Levi, Z. Zalevsky, and E. Marom, "Encoding technique for design of zero-order (on-axis) Fraunhofer computer-generated holograms," *Appl. Opt.*, vol. 36, no. 32, p. 8427, Nov. 1997, doi: 10.1364/AO.36.008427.



Dr. Zheyuan Zhu (M'18) received the B.S. degree in physics from Nanjing University in Nanjing, China, and MS and PhD in optics and photonics from CREOL, The College of Optics and Photonics in University of Central Florida (UCF).

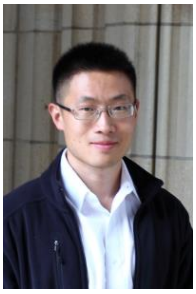
Dr. Zhu's research focuses on designing, modeling and prototyping novel computational imaging platforms using minimal system resources in both visible and x-ray regimes, with a specialty in x-ray transmission / diffraction tomography. He is a recipient of CREOL Student of the Year Finalist award, UCF Graduate Research Support award, as well as various other travel grants. He also serves as the vice president of IEEE Photonics Society student chapter at CREOL and the president of CREOL Association of Optics Students.

Mr. Joseph Ulseth received the bachelor's degrees in astronomy, physics, and mathematics from The University of Florida in Gainesville, FL in 2015. He completed an MS degree in optics and photonics at CREOL, The College of Optics and Photonics in University of Central Florida (UCF). Mr. Ulseth's research has focused on modeling and designing computational imaging systems including x-ray tracing for x-ray diffraction tomography simulations, and EO platforms for neural network inference. He is a student member of OSA, SPIE, and IEEE.



Dr. Guifang Li (F'13) received the Ph.D. degree in electrical engineering from the University of Wisconsin at Madison, Madison, WI, USA. He is currently a Professor of optics, and electrical & computer engineering at the University of Central Florida, Orlando, FL, USA. His research interests include optical communications and networking, RF photonics, all-optical signal processing, free-space optics, and optical imaging. He is the recipient of the NSF CAREER award, the Office of Naval Research Young Investigator award, the UCF Research Incentive Award in 2006, and the UCF Innovator Award in 2012. He is a Fellow of SPIE, the Optical Society of America, and the National Academy of Inventors. He served as a Deputy Editor for Optics Express and an

Associate Editor for Optical Networks, Chinese Optics Letters, and the IEEE Photonics Technology Letters. He currently serves as the Editor-in-Chief OSA's *Advances in Optics & Photonics*.



Dr. Shuo Pang received the B.S. degree in optical engineering from Tsinghua University in Beijing, China. He received the M.S. degree in biomedical engineering from the Texas A&M University in College Station, Texas. He received the M.S. degree and Ph.D. degree in electrical engineering from California Institute of Technology in Pasadena, California. He was a postdoctoral associate at the Department of Electrical and Computer Engineering in Duke University, 2013-2014. He is currently an assistant professor at CREOL, the College of Optics and Photonics of the University of Central Florida.

Dr. Pang's current research focuses on developing computational imaging systems, image processing in both visible and x-ray regimes, and machine-learning approach in optics. He is a recipient of Ralph E. Powe Junior Faculty Award in 2016 and SPIE Defense and Commercial Sensing (DCS) Rising Researcher Award in 2017. He was the Chair of Optical Microscopy Group of Optical Society of America. He serves on the organizing committee of Anomaly Detection and Imaging with X-ray Conference of SPIE DCS, and a co-editor of the CREOL Institutional Focus Issue in Applied Optics, 2019.