Variation-tolerant Architectures for Convolutional Neural Networks in the Near Threshold Voltage Regime

Yingyan Lin, *Student Member, IEEE*, Sai Zhang, *Student Member, IEEE* and Naresh R. Shanbhag, *Fellow, IEEE*
Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, IL, USA
[yingyan, szhang12, shanbhag]@illinois.edu

***Abstract*—Convolutional neural networks (CNNs) have gained considerable interest due to their state-of-the-art performance in many recognition tasks. However, the computational complexity of CNNs hinders their application on power-constrained embedded platforms. In this paper, we propose a variation-tolerant architecture for CNN capable of operating in near threshold voltage (NTV) regime for energy efficiency. A statistical error compensation (SEC) technique referred to as rank decomposed SEC (RD-SEC) is proposed. RD-SEC is applied to the CNN architecture in NTV in order to correct timing errors that can occur due to process variations. Simulation results in $45\,\mathrm{nm}$ CMOS show that the proposed architecture can achieve a median detection accuracy $P_{det} \geq 0.9$ in the presence of gate level delay variation of up to 34%. This represents an $11\times$ improvement in variation tolerance in comparison to a conventional CNN. We further show that RD-SEC-based CNN enables up to $113\times$ reduction in the standard deviation of P_{det} compared with the conventional CNN.**

I. Introduction

Many emerging applications in pattern recognition and data mining require the use of statistical signal processing (SP) and machine learning (ML) algorithms to process massive volumes of data on energy-constrained platforms. Computational complexity of these algorithms makes energy efficiency one of the primary design challenges. A commonly employed kernel in SP and ML algorithms is the matrix-vector multiply or the *dot product ensemble* (DPE), where an input vector $\mathbf{x}$ is projected to a set of weight vectors, i.e.:

$$\mathbf{y} = \mathbf{W}^{\mathrm{T}}\mathbf{x} \tag{1}$$

where $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_M]$ is the $N \times M$ weight matrix, $\mathbf{w}_k$ is the k^{th} $N \times 1$ weight vector, $\mathbf{x}$ is the $N \times 1$ input vector, $\mathbf{y} = [y_1, ..., y_M]^{\mathrm{T}}$ is the $M \times 1$ output vector, and y_k is the k^{th} element of $\mathbf{y}$ which can be expressed as a dot product (DP) $y_k = \mathbf{w}_k^{\mathrm{T}}\mathbf{x}$. The DPE is the most power hungry kernel in a variety of SP and ML algorithms such as convolutional neural networks (CNNs) (see Fig. 1) [1] and accounts for 90% of the computational power in state-of-the-art integrated circuit implementations [2]. As a result, energy-efficient DPE architectures are of great importance.

Techniques such as low power parallel filter design [3] and common subexpression elimination (CSE) [4] can be applied to DPEs to reduce computational complexity. These techniques exploit the redundancy within a multiplier or a DP. Another approach to reduce energy further is to operate the architecture in near threshold voltage (NTV). NTV designs can achieve up to $10\times$ savings in energy, but suffer from a significant increase in variations, which can be as high as $20\times$ [5]. Error-resilient techniques [6–12] have been employed at various levels of design abstraction to compensate for the resultant timing errors caused by NTV operation. At the logic or circuit level, RAZOR [6], error detection sequential (EDS) [7], and Markov Random Field [8] have been proposed. These techniques either compensate for small error rates ($< 2\%$) or have large overhead ($> 5\times$), limiting their ability to enhance energy efficiency. At the system level, conventional fault-tolerance techniques such as N-modular redundancy (NMR) [9] incur $N\times$ complexity and power overhead, restricting their applicability. Statistical error compensation (SEC) [10–12] has been shown to be a promising solution. SEC employs detection and estimation-based techniques for error compensation. Techniques such as algorithmic noise-tolerance (ANT) are able to compensate for error rates of 21% to 89% while achieving 35% to 72% energy savings [11].

In this paper, we propose a new SEC technique referred to as rank decomposed SEC (RD-SEC) that is particularly well-suited for DPEs (see (1)). RD-SEC makes use of the fact that a large fraction of computation inside a DPE can be derived from a small subset, and employs these for low-cost error detection and correction. Simulation results in $45\,\mathrm{nm}$ CMOS for a RD-SEC-based CNN architecture operating in the NTV regime (0.3V-0.7V) show that the proposed architecture can achieve a median detection accuracy $P_{det} \geq 0.9$ in the presence of gate level delay variation of up to 34%. This represents an improvement in variation tolerance of $11\times$ as compared to a conventional CNN. We further show that RD-SEC-based CNN enables up to $113\times$ reduction in the standard deviation of P_{det} compared to the conventional CNN.

The remainder of this paper is organized as follows. Section II provides background on CNNs, low power design techniques, and ANT. Section III presents the proposed DPE-based CNN architecture and RD-SEC technique to enhance robustness. The error model generation and validation are presented in Section IV. Simulation results are shown in Section V. Finally, conclusions and future work are presented in Section VI.

II. Background

A. Convolutional Neural Networks (CNNs)

CNNs are a class of multi-layer neural networks [1] [13]. A CNN consists of a cascade of multiple convolutional layers (C-layers) and subsampling layers (S-layers) (feature extractor), and fully-connected layers (F-layers) (classifier). Figure 1 illustrates a state-of-the-art CNN for object recognition [1]. In a C-layer, the DPs between the receptive fields [13] and

weight vectors are computed, to which a bias term is added, and put through a squashing function to generate the output feature maps (FMs) (see Fig. 1):

$$\mathbf{z}_m = f(\mathbf{y}_m + \delta_m),\ (m = 1, \ldots, M) \tag{2}$$

$$\mathbf{y}_m = \sum_{l=1}^{L} \mathbf{w}_{ml}^T \mathbf{X}_l,\ (m = 1, \ldots, M) \tag{3}$$

where L, M, and K are defined in Table I, $\mathbf{w}_{ml}$ denotes the $K^2 \times 1$ weight vector connecting the l^{th} input FM to the m^{th} output FM, $\mathbf{X}_l = [\mathbf{x}_{1l}, \ldots, \mathbf{x}_{Jl}]$ and $\mathbf{x}_{jl}$ denotes the j^{th} $K^2 \times 1$ receptive field in the l^{th} input FM, $\mathbf{y}_m = [y_{1m}, \ldots, y_{Jm}]$ and y_{jm} denotes the j^{th} pixel of the m^{th} convolutional output, $\mathbf{z}_m = [z_{1m}, \ldots, z_{Jm}]$ and z_{jm} denotes the j^{th} pixel of the m^{th} output FM in the C-layer, and δ_m is a trainable bias term corresponding to the m^{th} output FM. The squashing function $f(\cdot)$ usually takes a sigmoid or hyperbolic form. The S-layer reduces the dimension of its input FMs via either an average or a max pooling.

The large amount of computation in a CNN hinders their deployment on embedded platforms [14]. For example, a state-of-the-art CNN AlexNet requires 666 million MACs per 227×227 image ($13k$ MACs/pixel) [2]. Hence, dedicated integrated circuit architectures for energy-efficient CNNs are of great interest.

Table I
SUMMARY OF CNN PARAMETERS FROM [1]

Parameter Definition		CNN Parameter Summary				
Parameter	Description	Layer	L	M	$I_1 \times I_2$	$K \times K$
L/M	# of input/output FMs	C1	1	32	28×28	5×5
$K \times K$	size of kernels	C2	32	64	12×12	5×5
$I_1 \times I_2$	size of input FMs	F1	64	100	4×4	2×1

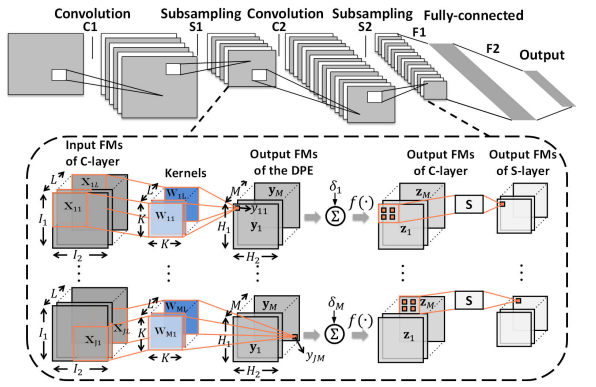
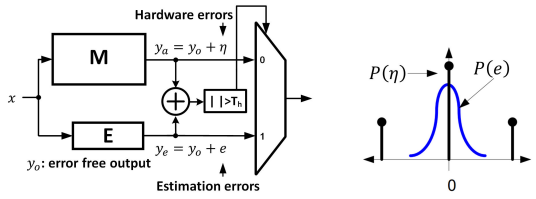


Figure 1. Illustration of a state-of-the-art CNN [1] showing a convolutional layer (C-layer), a subsampling layer (S-layer), feature maps (FMs), and the squashing function $f(\cdot)$.

B. Low Power Design Techniques

Various low power techniques can be used to reduce the energy of DPEs. At the logic level, programmable CSE [4] is a low power technique, where common subexpressions (CSs) in the coefficients are first computed using shift and add, and then summed up to obtain the final product. Programmability is enabled via a look-up table.

In order to further reduce energy, NTV was proposed to operate the devices at or near their threshold voltage (V_{th}), and has shown an energy reduction on the order of $10\times$ [5]. However, the energy efficiency of NTV comes at a cost of exponential increase in the normalized delay variation,



(a) (b)

Figure 2. Algorithmic noise-tolerance (ANT): (a) architecture, and (b) the error statistics in the **M**-block and **E**-block [11].

leading to an increased functional failure. Specifically, circuit simulations in a commercial 45 nm CMOS show that the delay variation of an 8-bit ripple-carry adder (RCA) increases by $8.5\times$ at $V_{dd} = 0.35$V (NTV) compared with that at the nominal $V_{dd} = 1.1$V due to process variations. To address the variation challenge, the traditional approach is to add design margin, which substantially reduces the benefits of NTV [5]. For example, it is estimated that the employing of voltage margining to ensure error-free operation results in $3.1\times$ energy overhead for the 8-bit RCA operating at 0.35V. Techniques such as body biasing [15] or variable pipeline stage latency [16] have been proposed. Although these techniques demonstrated some degree of effectiveness, they can incur significant overheads due to the local nature of variations.

C. Algorithmic Noise-Tolerance (ANT)

ANT is an algorithmic technique that employs error statistics to perform error compensation, and has been shown to be effective for SP and ML kernels [10] [11]. Specifically, ANT incorporates a main block (**M**-block) and an estimator (**E**-block) which is an approximate version of the **M**-block (see Fig. 2(a)). The **M**-block is subject to large magnitude errors η (e.g., timing errors which typically occur in the MSBs) while the **E**-block is subject to small magnitude errors e (see Fig. 2(b), e.g., due to quantization noise in the LSBs), i.e.:

$$y_a = y_o + \eta \tag{4}$$

$$y_e = y_o + e \tag{5}$$

where y_o, y_a, and y_e are the error-free, the **M,** and **E**-block outputs, respectively. ANT exploits the difference in the error statistics of η and e to detect and compensate for errors to obtain the final corrected output $\hat{y}$ as follows:

$$\hat{y} = \begin{cases} y_a & \text{if } |y_a - y_e| \leq T_h \\ y_e & \text{otherwise} \end{cases} \tag{6}$$

where T_h is an application dependent threshold parameter chosen to maximize the performance of ANT.

III. THE PROPOSED RD-SEC TECHNIQUE

This section describes the proposed error compensation technique RD-SEC to enable robust CNN design in the NTV regime. First, we reformulate the C-layer computation in terms of the DPE.

A. DPE-based CNNs

From (2) and (3), the computation of one pixel in the output FM of the C-layer can be described as follows:

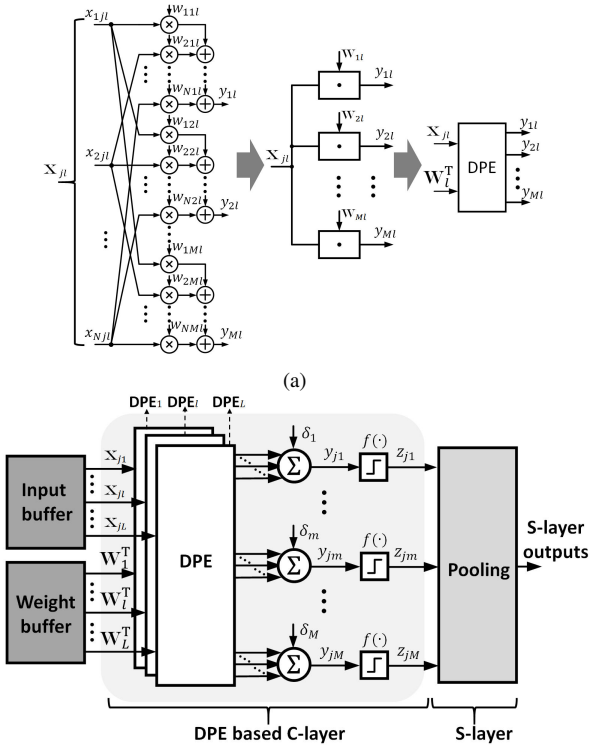
$$z_{jm} = f(y_{jm} + \delta_m),\ (m = 1, \ldots, M) \tag{7}$$

$$y_{jm} = \sum_{l=1}^{L} \mathbf{w}_{ml}^T \mathbf{x}_{jl},\ (m = 1, \ldots, M) \tag{8}$$

Equation (8) shows that the j^{th} pixel y_{jm} of the m^{th} convolutional output FM $\mathbf{y}_m$ is obtained by first performing DPs between the L input vectors $\mathbf{x}_{jl}$ and weight vectors $\mathbf{w}_{ml}$, and summing up the results. Equation (8) can be rewritten in a vector form as follows:

$$\begin{bmatrix} y_{j1} \\ \vdots \\ y_{jm} \\ \vdots \\ y_{jM} \end{bmatrix} = \begin{bmatrix} \sum_{l=1}^{L} \mathbf{w}_{1l}^T \mathbf{x}_{jl} \\ \vdots \\ \sum_{l=1}^{L} \mathbf{w}_{ml}^T \mathbf{x}_{jl} \\ \vdots \\ \sum_{l=1}^{L} \mathbf{w}_{Ml}^T \mathbf{x}_{jl} \end{bmatrix} = \sum_{l=1}^{L} \mathbf{W}_l^T \mathbf{x}_{jl} \qquad (9)$$

where $\mathbf{W}_l = [\mathbf{w}_{1l}, \ldots, \mathbf{w}_{Ml}]$. It can be seen that (9) is the sum of L DPEs, where the l^{th} DPE is given by $\mathbf{W}_l^T \mathbf{x}_{jl}$ (see Fig. 3(a)). A single stage of a DPE-based CNN in Fig. 3(b) consists of input and weight buffers, a DPE-based C-layer, and an S-layer. Specifically, the input images and kernel weights are streamed from the input and weight buffers, respectively. The DPE-based C-layer accepts the input vectors and weight matrices, and obtains the M outputs according to (7) and (8). In the S-layer, the spatial resolution of the C-layer output FMs is reduced by either averaging or max pooling.



(a)

(b)

Figure 3. Architecture of: (a) a (N, M) dot product ensemble (DPE), where $\mathbf{w}_{ml} = [w_{1ml}, \cdots, w_{Nml}]$ and $\mathbf{W}_l = [\mathbf{w}_{1l}, \cdots, \mathbf{w}_{Ml}]$, and (b) one stage DPE-based CNN consisting of a C-layer and an S-layer.

B. Rank Decomposed SEC (RD-SEC): Principle and Architecture

The formulation of a DPE-based CNN in Section III-A enables us to exploit redundancy within a DPE for statistical error compensation. The proposed approach RD-SEC employs low-cost estimators from a set of basis vectors in the $N \times M$ weight matrix $\mathbf{W}$ (see (1)). To do so, we make use of the rank

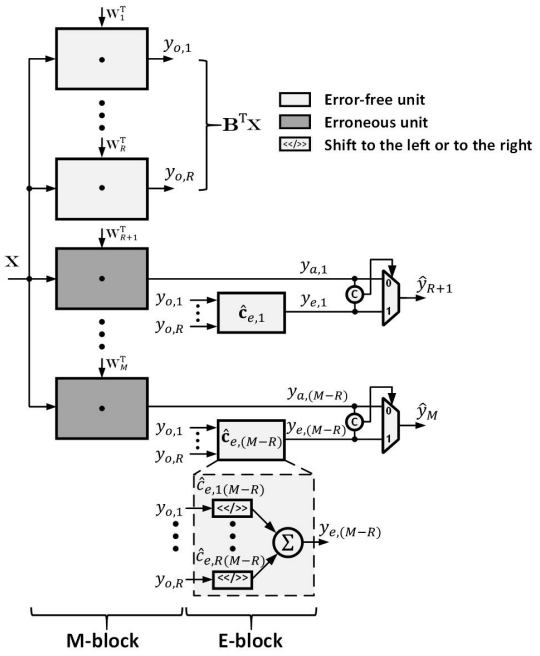


Figure 4. RD-SEC applied to a DPE.

decomposition of $\mathbf{W}$ which exists for every finite-dimensional matrix [17]:

$$\mathbf{W} = \mathbf{BC} \qquad (10)$$

where $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_R]$ is a $N \times R$ basis matrix with $R = rank(\mathbf{W})$ (assume $M > N$, then $R \leq N$), $\mathbf{b}_r$ $(r = 1, \ldots, R)$ is the r^{th} $N \times 1$ basis vector, $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_M]$ is a $R \times M$ coefficient matrix, and $\mathbf{c}_m$ $(m = 1, \ldots, M)$ is the m^{th} $R \times 1$ coefficient vector. For example, in Fig. 1 and Table I, $N = 25$, $R = 24$ or 25, $M = 32$ for the C1 and C2 layers, and $N = 2$, $R = 2$, $M = 100$ for the F1 layer of the CNN in [1]. We choose $\mathbf{b}_i = \mathbf{w}_i$ $(i = 1, \ldots, R)$ so that,

$$\mathbf{W} = \mathbf{BC} = \mathbf{B}[\ \mathbf{I_R} \quad \mathbf{C}_e\ ] \qquad (11)$$

where $\mathbf{I_R}$ is a $R \times R$ identity matrix, and $\mathbf{C}_e = [\mathbf{c}_{e,1}, \ldots, \mathbf{c}_{e,M-R}]$ is a $R \times (M-R)$ matrix. Substituting (11) into (1), we have:

$$\begin{aligned} \mathbf{y} &= [\ \mathbf{I_R} \quad \mathbf{C}_e\ ]^{\mathrm{T}}(\mathbf{B}^{\mathrm{T}}\mathbf{x}) \\ &= [\ \mathbf{I_R} \quad \mathbf{C}_e\ ]^{\mathrm{T}}\mathbf{y}_o \\ &= \begin{bmatrix} \mathbf{y}_o \\ \mathbf{y}_a \end{bmatrix} \end{aligned} \qquad (12)$$

where $\mathbf{y}_o = \mathbf{B}^{\mathrm{T}}\mathbf{x} = [y_{o,1}, \ldots, y_{o,R}]^{\mathrm{T}}$ is the error-free $R \times 1$ vector, and $\mathbf{y}_a = \mathbf{C}_e^{\mathrm{T}}\mathbf{y}_o = [y_{a,1}, \ldots, y_{a,(M-R)}]^{\mathrm{T}}$ is a $(M - R) \times 1$ output vector from the $\mathbf{M}$-block subject to errors. In RD-SEC, we derive a low-cost estimator of $\mathbf{y}_a$ using the error-free output $\mathbf{y}_o$ and a rounded coefficient matrix $\hat{\mathbf{C}}_e^{\mathrm{T}}$, i.e.:

$$\mathbf{y}_e = \hat{\mathbf{C}}_e^{\mathrm{T}}\mathbf{y}_o = [\hat{\mathbf{c}}_{e,1}, \ldots, \hat{\mathbf{c}}_{e,(M-R)}]^{\mathrm{T}}\mathbf{y}_o \qquad (13)$$

where $\mathbf{y}_e = [y_{e,1}, \ldots, y_{e,(M-R)}]^{\mathrm{T}}$ is a $(M - R) \times 1$ estimation vector, $\hat{\mathbf{C}}_e^{\mathrm{T}} = round(\mathbf{C}_e^{\mathrm{T}})$ where the $round(\cdot)$ operator rounds an element to the nearest power of 2, and $\hat{\mathbf{c}}_{e,m} = [\hat{c}_{e,1m}, \ldots, \hat{c}_{e,Rm}]^{\mathrm{T}}$ is the m^{th} $R \times 1$ coefficient vector corresponding to $y_{e,m}$. Equation (13) indicates that $y_{e,m}$ can be implemented using only shifts and adds. Finally, the m^{th} error compensated output $\hat{y}_m$ is obtained as follows:

$$\hat{y}_m = \begin{cases} y_{o,m} & \text{if } m \leq R \\ y_{a,(m-R)} & \text{if } m > R \ \& \ \left|y_{a,(m-R)} - y_{e,(m-R)}\right| \leq T_h \\ y_{e,(m-R)} & \text{otherwise} \end{cases} \tag{14}$$

where the threshold T_h is an application dependent parameter chosen to maximize system performance [10]. The RD-SEC architecture is shown in Fig. 4.

C. RD-SEC Overhead

The overhead of a RD-SEC-based CNN can be approximated relative to the **M**-block in a DPE. The computational overhead γ of RD-SEC relative to the **M**-block is defined as:

$$\gamma = \frac{N_{\rm P} - N_{\rm conv}}{N_{\rm conv}} = \frac{(M-R)}{M}\alpha \tag{15}$$

where $N_{\rm P}$ and $N_{\rm conv}$ denote the complexities of the RD-SEC-based DPE and the conventional DPE in terms of the number of full adders (FAs), respectively, α quantifies the ratio of the complexities of one **E**-block and **M**-block (only $M-R$ out of M channels have **E**-blocks (see Fig. 4)). The detailed expression for α is provided in the Appendix.

The $\gamma_{\rm C}$ and $\gamma_{\rm F}$ in Fig. 5 correspond to the computational overhead of the C1/C2 layers and the F1 layer of the CNN in [1], respectively. Figure 5 shows that $\gamma_{\rm C}$ increases with N for $N \leq 5$, and then decreases with N. This is because α increases with N due to the increased number of adders in (13), while at the same time, the number of **E**-blocks $M-R$ reduces since $R = N$. Similar results were obtained for $\gamma_{\rm F}$. This indicates that RD-SEC overhead reduces with N for large vector length (i.e., $N \geq 10$). Specifically, $\gamma_{\rm C} \approx 5\%$ and $\gamma_{\rm F} \approx 15\%$ when RD-SEC is applied to the CNN in [1].

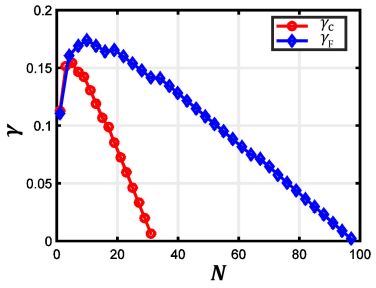


Figure 5. Overhead of the RD-SEC-based DPE: computational overhead γ vs. N, where the corresponding parameters are summarized in Table II.

Table II
PARAMETERS FOR THE $\gamma_{\rm C}$ AND $\gamma_{\rm F}$ IN FIG. 5

	Parameters
$\gamma_{\rm C}$	$B_{in}=7$, $B_w=8$, $R=N$, $M=32$
$\gamma_{\rm F}$	$B_{in}=7$, $B_w=8$, $R=N$, $M=100$

IV. Error Model Generation and Validation

This section presents the timing error model generation methodology [18] and the validation of this timing error model in a commercial 45 nm CMOS. A complete HDL simulation for the entire CNN is infeasible due to the large amount of the DPEs, thus we validate the model for a single DPE employing the circuit-level signal-to-noise ratio (SNR) of the main block (see Fig. 2 and (4)) as follows:

$$SNR = 10\log_{10}\left(\frac{\sigma^2_{y_o}}{\sigma^2_{\eta}}\right) \tag{16}$$

where $\sigma^2_{y_o}$ and σ^2_{η} are the variances of the error-free output y_o and the timing error η, respectively. The error model generation and validation methodology is shown in Fig. 6(a), and described below:

1) Characterize the gate delay distribution vs. operating voltage V_{dd} of basic gates such as AND and XOR using HSPICE in the NTV range 0.3V-0.7V.

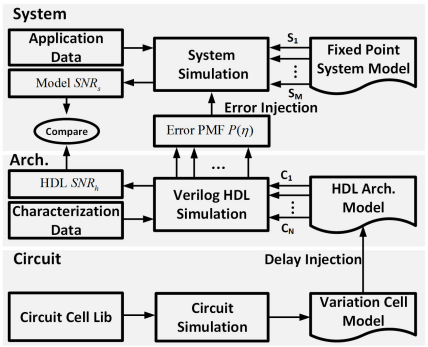
2) Implement the DPE architecture shown in Fig. 3(a) using structural Verilog HDL using the basic gates characterized in Step 1.

3) Emulate process variations at NTV by generating multiple (30) architectural instances and assigning random gate delays obtained via sampling the gate delay distributions obtained in Step 1.

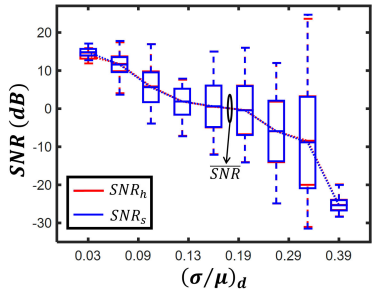
4) Run HDL (bit and clock accurate) simulations of each instance to obtain error samples and circuit-level signal-to-noise ratio SNR_h.

5) Generate the error PMF $P(\eta)$ employing the procedure in [18].

6) Run fixed-point MATLAB simulations using the PMF to inject errors for the DPEs in CNNs to obtain circuit-level signal-to-noise ratio SNR_s. Compare SNR_s with SNR_h.



(a)



(b)

Figure 6. Error model generation and validation methodology: (a) model generation methodology, and (b) validation by comparing SNR from HDL simulations and the NTV methodology based on 30 DPE instances with 10^5 random input samples for each instance operating at gate level delay variation of 3%-39%.

Figure 6(b) plots SNR_h obtained in Step 4 (HDL simulations using gate delay distributions) and SNR_s obtained in Step 6 (MATLAB simulations using error PMF) as a function of the gate level delay variation $(\sigma/\mu)_d$. It is found that the difference between the median SNR_h ($\overline{SNR_h}$) and SNR_s ($\overline{SNR_s}$) is no more than 5% when $(\sigma/\mu)_d$ increases from 3% to 39%. Figure 6(b) shows that the variation of SNR increases

for $3\% \leq (\sigma/\mu)_d \leq 34\%$, and then decreases because all the instances are subject to large timing errors. Figure 6(b) further shows that the maximum and minimum values of SNR_h and SNR_s differ by no more than 6% and 4%, respectively. These results indicate that the timing error is well-modeled by its PMF.

V. Simulation Results

In this section, we evaluate the performance of RD-SEC-based CNNs employing the MNIST database [1] and the error PMFs from Section IV.

A. System Set-up

The parameters of the CNN being studied are summarized in Table I [1]. The bias term δ_m in (2) and kernel $\mathbf{w}_{ml}$ in (3) are trained using the back propagation algorithm [13]. The following two architectures are considered: 1) a slow CNN architecture with RD-SEC applied to the C-layers and F1 layer (denoted as RD-SEC CNN), where the multipliers and adders are implemented using Baugh-Wooley (BW) multiplier and RCA, respectively; 2) an uncompensated fast CNN architecture (denoted as Conv CNN), where the multipliers and adders are implemented using the programmable CSE technique in [4] and Kogge-Stone adder, respectively. The fast architecture is chosen for comparison because it will result in the largest energy savings in the error-free case when voltage scaling is employed. For both CNNs, B_{in} and B_w are set to 7 bits and 8 bits, respectively, ensuring the error-free fixed-point detection accuracy to be within 0.2% of the floating-point detection accuracy of 0.98.

B. Characterization

First, the extent of process variation in NTV is characterized in terms of $(\sigma/\mu)_d$. Figure 7(a) shows that $(\sigma/\mu)_d$ increases by $13\times$ from 3% to 39% as the supply voltage V_{dd} decreases from $0.7\,\text{V}$ to $0.3\,\text{V}$. Note that process variation makes the detection accuracy $p_{det} = P\{\hat{T} = t\}$ ($\hat{T}$ and t are the classifier decision and the true label, respectively) a random variable, which is denoted as P_{det}. Figure 7(b) shows that the median error rate $\bar{p}_\eta$ (where the error rate is defined as $p_\eta = P\{\eta \neq 0\}$) increases by $70\times$ from 1.4×10^{-2} to 0.99 as V_{dd} decreases from $0.7\,\text{V}$ to $0.3\,\text{V}$. At a $(\sigma/\mu)_d = 34\%$, the median error rate $\bar{p}_\eta = 0.57$.

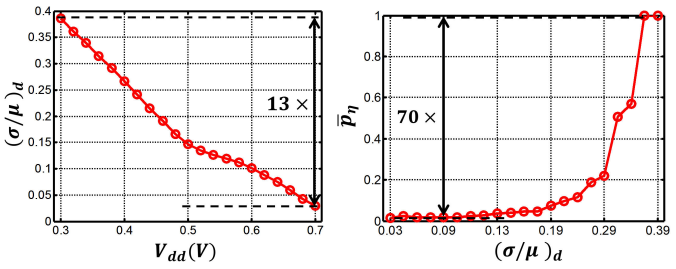
Next, we employ the error PMFs obtained from Step 5 of the NTV error modeling methodology (see Section IV) to inject errors in fixed-point MATLAB simulations of CNN architectures to evaluate their robustness to timing errors in NTV. We compare the two architectures in terms of median ($\bar{p}_{det}$) and standard deviation ($\sigma_{p_{det}}$) of the detection accuracy P_{det}. This is because p_η and p_{det} are spatially distributed random variables in the presence of process variations.

C. Comparison of $\bar{p}_{det}$ and $\sigma_{p_{det}}$

Figure 8(a) shows that RD-SEC CNN is able to maintain $\bar{p}_{det} \geq 0.9$ for $(\sigma/\mu)_d \leq 34\%$, whereas Conv CNN can only maintain the same performance for $(\sigma/\mu)_d \leq 3\%$. Thus, RD-SEC CNN is able to deliver a high detection accuracy in the presence of high error rate of $\bar{p}_\eta \leq 0.57$ (see Fig. (7(b))). This indicates an $11\times$ improvement compared with the Conv CNN. Figure 8(b) shows that the RD-SEC CNN can achieve $113\times$ reduction in $\sigma_{p_{det}}$ as compared to the Conv CNN at $(\sigma/\mu)_d = 11\%$. Figure 8(b) also shows that $\sigma_{p_{det}}$ of the RD-SEC CNN is no more than 4.8×10^{-2}, whereas the maximum $\sigma_{p_{det}}$ of the Conv CNN is 0.32 for $3\% \leq (\sigma/\mu)_d \leq 39\%$.

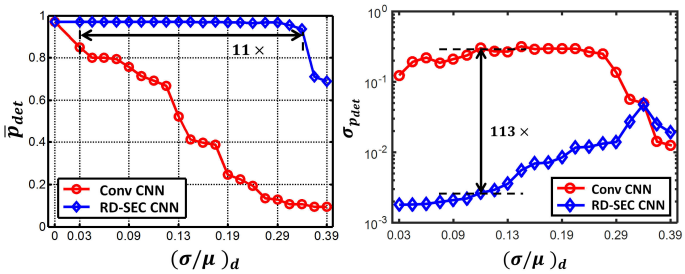
Furthermore, Fig. 8(b) demonstrates that $\sigma_{p_{det}}$ of the RD-SEC CNN increases from 1.8×10^{-3} to 4.8×10^{-2} when $(\sigma/\mu)_d$ increases from 3% to 34%, and then decreases. When $(\sigma/\mu)_d > 34\%$, $\sigma_{p_{det}}$ of the RD-SEC CNN is larger than that of the Conv CNN because all the instances of the Conv CNN achieve a low $P_{det} \approx 0.1$, whereas some instances of the RD-SEC CNN can still achieve a $P_{det} \geq 0.9$, leading to a larger $\sigma_{p_{det}}$.

To understand the robustness improvement achieved by RD-SEC, the input, C1 FMs (12 out of 32), the output vector and the final decision $\hat{T}$ are analyzed (see Fig. 9). Note that $\hat{T}$ is chosen as the index of the maximum element in the output vector. Figure 9(a) shows that the timing errors contaminate the extracted features in the Conv CNN, leading to classification failure. Specifically, the output vector has two peaks (at positions "3" and "5") due to the contaminated features, resulting in a wrong decision "3" instead of the correct one "5". On the other hand, RD-SEC is able to



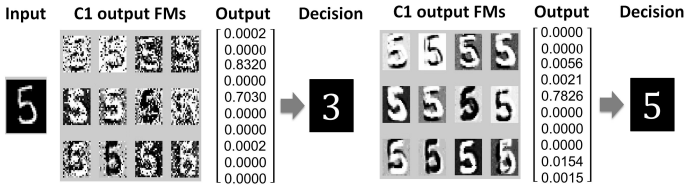
(a) (b)

Figure 7. Characterization of: (a) process variations in terms of $(\sigma/\mu)_d$ vs. V_{dd}, and (b) impact of process variations on DPE error rate $\bar{p}_\eta$ based on 30 DPE instances.



(a) (b)

Figure 8. Simulation results comparing: (a) $\bar{p}_{det}$ vs. $(\sigma/\mu)_d$, and (b) $\sigma_{p_{det}}$ vs. $(\sigma/\mu)_d$, based on 30 CNN instances in the presence of process variations.



(a) (b)

Figure 9. An example of the C1 FMs and the output vector from: (a) the Conv CNN, and (b) the RD-SEC CNN, when the input digit is "5" and $(\sigma/\mu)_d = 27\%$.

compensate for timing errors, and thus enables the RD-SEC CNN to extract correct features for correct classification even in the presence of a large number of timing errors.

VI. Conclusions

In this paper, we propose a new algorithmic error compensation technique RD-SEC, which exploits the inherent redundancy within a DPE for low-cost error correction. RD-SEC makes use of the fact that there are redundant and non-redundant computations within vector-matrix multiplication operations, which is an essential and pervasive operation in many ML and SP algorithms. This work opens up the possibility to exploit inherent redundancy within parallel data-paths for error detection and correction with low overhead. Future work includes imposing constraints that favor the reduction of estimation error into ML training algorithms and exploring low-cost estimators through clustering techniques such as k-means.

Acknowledgment

This work was supported in part by Systems on Nanoscale Information fabriCs (SONIC), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA.

References

[1] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Computer Vision, IEEE 12th International Conference on*, 2009, pp. 2146–2153.

[2] Y. H. Chen, T. Krishna, J. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2016.

[3] J.-G. Chung and K. K. Parhi, "Frequency spectrum based low-area low-power parallel FIR filter design," *EURASIP J. Appl. Signal Process.*, vol. 2002, pp. 944–953, 2002.

[4] R. Mahesh and A. Vinod, "New reconfigurable architectures for implementing FIR filters with low complexity," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 29, no. 2, 2010.

[5] R. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-threshold computing: reclaiming Moore's law through energy efficient integrated circuits," *Proceedings of the IEEE*, vol. 98, no. 2, 2010.

[6] S. Das, D. Blaauw, D. Bull, K. Flautner, and R. Aitken, "Addressing design margins through error-tolerant circuits," in *Design Automation Conference (DAC), 46th ACM/IEEE*, 2009, pp. 11–12.

[7] J. Tschanz, K. Bowman, C. Wilkerson, S.-L. Lu, and T. Karnik, "Resilient circuits: enabling energy-efficient performance and reliability," in *Computer-Aided Design (ICCAD), IEEE/ACM International Conference on*, 2009.

[8] R. Bahar, J. Mundy, and J. Chen, "A probabilistic-based design methodology for nanoscale computation," in *Computer Aided Design (ICCAD), IEEE/ACM International Conference on*, 2003, pp. 480–486.

[9] N. Vaidya and D. Pradhan, "Fault-tolerant design strategies for high reliability and safety," *Computers, IEEE Transactions on*, vol. 42, no. 10, pp. 1195–1206, 1993.

[10] B. Shim, S. Sridhara, and N. Shanbhag, "Reliable low-power digital signal processing via reduced precision redundancy," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 12, no. 5, pp. 497–510, 2004.

[11] J. Choi, E. P. Kim, R. A. Rutenbar, and N. R. Shanbhag, "Error resilient MRF message passing architecture for stereo matching," in *Signal Processing Systems (SiPS), IEEE Workshop on*, 2013, pp. 348–353.

[12] R. A. Abdallah and N. R. Shanbhag, "Error-resilient systems via statistical signal processing," in *Signal Processing Systems (SiPS), IEEE Workshop on*, 2013.

[13] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, 1998.

[14] P. H. Hung, C. H. Lee, S. W. Yang, V. S. Somayazulu, Y. K. Chen, and S. Y. Chien, "Bridge deep learning to the physical world: An efficient method to quantize network," in *Signal Processing Systems (SiPS), IEEE Workshop on*, 2015, pp. 1–6.

[15] R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas, "Mitigating parameter variation with dynamic fine-grain body biasing," in *Microarchitecture (MICRO), 40th Annual IEEE/ACM International Symposium on*, 2007.

[16] X. Liang, G.-Y. Wei, and D. Brooks, "Revival: a variation-tolerant architecture using voltage interpolation and variable latency," *Micro, IEEE*, vol. 29, 2009.

[17] G. Strang, *Introduction to Linear Algebra*, 3rd ed. Wesley-Cambridge Press, 2003.

[18] S. Zhang and N. Shanbhag, "Probabilistic error models for machine learning kernels implemented on stochastic nanoscale fabrics," in *Design, Automation Test in Europe (DATE)*, 2016.

[19] J. M. Rabaey, A. Chandrakasan, and B. NikoliÄĞ, *Digital integrated circuits: a design perspective*. Upper Saddle River (N.J.): Prentice-Hall, Inc., 2003.

Appendix

In this Appendix, we provide a detailed expression for α in (15). The complexity is calculated in terms of the number of FAs. From (15), α is given by:

$$\alpha = \frac{N_E}{N_M} = \frac{N_{add-R} + N_{MUX}}{N_{DP}} \tag{17}$$

where N_E and N_M denote the complexities of one **E**-block and **M**-block, respectively, N_{add-R} denotes the complexity of the summer in (13), N_{MUX} denotes the complexity of MUX-based shifter in (13), N_{DP} denotes the complexity of one DP implemented using a Baugh-Wooley (BW) multiplier and ripple-carry adder (RCA). Specifically,

$$N_{add-R} = (R-1)(B_{out} + \lceil \log_2(R) \rceil - 1) \tag{18}$$

$$N_{MUX} = B_{out}(\lceil \log_2(B_{out}+1) \rceil \, r_{_{M2F}})R \tag{19}$$

$$N_{DP} = NB_wB_{in} + (N-1)(B_{in} + B_w + \lceil \log_2(N) \rceil - 1) \tag{20}$$

where $r_{_{M2F}}$ denotes the normalized complexity of a $2:1$ MUX over a FA and we use $r_{_{M2F}} = 3.5/9$ [19], the $\lceil a \rceil$ is the ceiling operation, and B_{in}, B_{out} and B_w denote the precision for the input/output and weights, respectively.