Power-Efficient Deep Neural Networks with Noisy Memristor Implementation

Elsa Dupraz, Lav R Varshney, François Leduc-Primeau

Power-Efficient Deep Neural Networks with Noisy Memristor Implementation

Elsa Dupraz[†], Lav R. Varshney[‡], and François Leduc-Primeau[*]
[†] IMT Atlantique, Lab-STICC, UMR CNRS 6285, F-29238, France
[‡] Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, USA
[*] Department of Electrical Engineering, École Polytechnique de Montréal, Montreal (QC), Canada

***Abstract*—This paper considers Deep Neural Network (DNN) linear-nonlinear computations implemented on memristor crossbar substrates. To address the case where true memristor conductance values may differ from their target values, it introduces a theoretical framework that characterizes the effect of conductance value variations on the final inference computation. With only second-order moment assumptions, theoretical results on tracking the mean, variance, and covariance of the layer-by-layer noisy computations are given. By allowing the possibility of amplifying certain signals within the DNN, power consumption is characterized and then optimized via KKT conditions. Simulation results verify the accuracy of the proposed analysis and demonstrate the significant power efficiency gains that are possible via optimization for a target mean squared error.**

I. Introduction

When implemented on electronic circuits, Deep Neural Networks (DNNs) demand important data transfers between memories and processors, which severely affects their power consumption and latency. As an alternative emerging paradigm, computation-in-memory consists of performing computation operations directly within the memory [1]. In this work, we consider implementing Deep Neural Networks (DNNs) from memristor crossbars developed for analog in-memory computation [2]–[4], where the synaptic weights of neural networks are encoded in memristance values [5]. When building a computational system from a memristor crossbar, one of the main difficulties is to set up memristance values with arbitrary precision [3]. Hence in this work, we consider the setting where memristance values are noisy, and evaluate the effect of noise on the inference phase of the DNN.

It is shown in [6] that noise can sometimes help in the inference performance of neural networks by getting them unstuck from local minima. Several works in the literature also investigate the robustness of DNNs to undesired noise [7]–[9], though without considering the in-memory computation framework. As a key issue, the DNN robustness is strongly related to the power consumption of its electrical circuit implementation [9]. In this work, we investigate this key issue in the context of in-memory computing, by first aiming to predict the effect of noisy memristor values onto the final DNN computation. The physical substrate of memristors and their computational and noise properties make the mathematical problem quite different from the above works [6]–[9].

This work was supported in part by the "Make our Planet Great Again" Initiative of the Thomas Jefferson Fund, by grant ANR-17-CE40-0020 of the French National Research Agency ANR (project EF-FECtive), and by IVADO grant PRF-2019-4784991664.

We first introduce a theoretical analysis that consists of tracking the evolution of the first and second-order moments over the successive layers of a DNNs, as functions of the noise variance of memristor values. These recursive expressions are obtained from second-order Taylor expansions of the means, variances, and covariances of the layers outputs, taking into account the noisy linear computation realized from memristor crossbars, and the non-linear activation functions at each layer. This extends our previous moment analysis of [10] which considered noisy dot-product computation from memristor crossbars. The proposed analysis is very generic as it only performs a few assumptions on the mean and variance of each memristance value, without any further assumption on their statistical distribution. The accuracy of the proposed analysis is verified using Monte Carlo simulations. Then, as a second important contribution, we propose to optimize the power consumption of a DNN memristor-based implementation. In a first step, we show how to evaluate the expected power consumption of one memristor crossbar, by applying the second-order Taylor expansion approach to electrical formula that provide the power of one specific memristor in a crossbar. In a second step, we formulate a power optimization problem under constraints on the final variance at the network output, and provide analytical solutions to this problem by relying on the Karush-Kuhn-Tucker (KKT) conditions. Our numerical results demonstrate significant power gains compared to the non-optimized case.

The remainder of this paper is organized as follows. Section II presents the noisy memristor crossbar architecture. Section III develops the moment-evolution methodology. Section IV addresses power optimization. Section V provides simulation results. In what follows, $\mathbb{E}[\cdot]$ refers to the expectation, $\mathbb{V}[\cdot]$ to the variance, and $\mathbb{C}[\cdot,\cdot]$ to the covariance. The notation $[\![1,\Psi]\!]$ indicates the set $\{1,2,\ldots,\Psi\}$.

II. Noisy architectures for Deep Neural Networks

A. DNN computation

In this work, we consider a feedforward NN with T layers. The network input is a vector $\mathbf{x}$ of length K, and the network output is a vector $\mathbf{y}$ of length N. Layer $t \in [\![1,T]\!]$ of the network is composed by N_t neurons and outputs a vector $\mathbf{x}^{(t)}$

of length N_t. We use $\mathcal{W}^{(t)}$ to denote the weight matrix of layer t, where the matrix $\mathcal{W}^{(t)}$ is of size $N_t \times N_{t-1}$. For a given activation function f, layer t performs the following linear and non-linear operations:

$$\mathbf{z}^{(t)} = \mathcal{W}^{(t)}\mathbf{x}^{(t-1)}, \text{ for all } t \in [\![1, T]\!], \quad (1)$$

$$x_k^{(t)} = f(z_k^{(t)}), \text{ for all } t \in [\![1, T]\!] \text{ and all } k \in [\![1, N_t]\!], \quad (2)$$

where $\mathbf{z}^{(t)}$ is a vector of length N_t. In this work, we consider f to be the sigmoid function $f(z) = \frac{1}{1+e^{-z}}$, which is commonly considered in several NN implementations. The theoretical analysis developed in this paper can be easily extended to other activation functions f, given they are twice differentiable.

B. Memristor-based implementation of DNN

In this work, we consider the setting where the non-linear operation (2) is realized either in the CMOS layer of a CMOL architecture [11], by analog computation [5], or a similarly mathematizable physical substrate. There is no fundamental difference in considering either of these implementations and this implies that in our analysis, the noise is only on the conductance values of the memristors.

Then, the linear operation (1) is a dot-product computation which can be realized by a memristor crossbar. Since memristor conductance values can only be positive, the computation operation (1) is realized from two crossbars: one for the positive part, and one for the negative part of the computation [3]. Therefore, we let $\mathcal{W}^{(t)} = \mathcal{W}^{(t,+)} - \mathcal{W}^{(t,-)}$, where $\mathcal{W}^{(t,+)}$ contains the positive components of $W^{(t)}$, and $\mathcal{W}^{(t,-)}$ contains the absolute values of the negative components of $\mathcal{W}^{(t)}$. The matrices $\mathcal{W}^{(t,+)}$ and $\mathcal{W}^{(t,-)}$ are then converted into matrices $\mathcal{G}^{(t,+)}$ and $\mathcal{G}^{(t,-)}$, respectively, where $\mathcal{G}^{(t,+)}$ and $\mathcal{G}^{(t,-)}$ contain positive conductance values $g_{i,j}^{(t,+)}$ and $g_{i,j}^{(t,-)}$. These conductance values are chosen so that the computation of the $z_j^{(t)}$ in (1) can be equivalently performed as

$$z_j^{(t)} = \sum_{i=1}^{N_{t-1}} \frac{g_{i,j}^{(t,+)} x_i^{(t)}}{g_0 + \sum_{k=1}^{N_{t-1}} g_{k,j}^{(t,+)}} - \sum_{i=1}^{N_{t-1}} \frac{g_{i,j}^{(t,-)} x_i^{(t)}}{g_0 + \sum_{k=1}^{N_{t-1}} g_{k,j}^{(t,-)}} \quad (3)$$

where g_0 is a pull-down conductance. Note that in expression (3), a part of the coefficients $g_{i,j}^{(t,+)}$ and $g_{i,j}^{(t,-)}$ are equal to zero, when the corresponding coefficients in $\mathcal{W}^{(t,+)}$ and $\mathcal{W}^{(t,-)}$ are equal to zero.

C. Noisy conductance values

When considering dot-product computation from memristor crossbars, one key issue is the difficulty in setting up conductance values $g_{i,j}^{(t,+)}$, $g_{i,j}^{(t,-)}$ with arbitrary precision [3]. Therefore, we think of the conductance values as noisy, so that the deterministic quantities $g_{i,j}^{(t,+)}$, $g_{i,j}^{(t,-)}$, are replaced by random variables $G_{i,j}^{(t,+)}$, $G_{i,j}^{(t,-)}$, in (3). We further assume that whenever $g_{i,j}^{(t,+)} \neq 0$, the corresponding random variable $G_{i,j}^{(t,+)}$ has mean $g_{i,j}^{(t,+)}$ and variance σ^2. In the same way, we assume that the random variables $G_{i,j}^{(t,-)}$ such that $g_{i,j}^{(t,-)} \neq 0$ have mean $g_{i,j}^{t,-}$ and the same variance value σ^2. On the contrary, if $g_{i,j}^{(t,+)} = 0$ (respectively $g_{i,j}^{(t,-)} = 0$), we assume that $G_{i,j}^{(t,+)} = 0$ (respectively $G_{i,j}^{(t,-)} = 0$) as well. In order to develop an analysis which applies to a large class of problems, physical substrates, and noise environments, we do not make any further assumption beyond the second-order moments on *e.g.*, the distribution of the random variables $G_{i,j}^{(t,+)}$ and $G_{i,j}^{(t,-)}$. In addition, since the non-linear operations (2) are not realized from memristor crossbars, we assume that no additional noise is introduced during these operations. Finally, we denote the random versions of vectors $\mathbf{x}^{(t)}$ and $\mathbf{z}^{(t)}$ by $\mathbf{X}^{(t)}$ and $\mathbf{Z}^{(t)}$.

III. Performance analysis of noisy memristor-based implementation of DNN

With the mathematical model of the noisy memristor-based DNN implementation in place, now we derive our theoretical performance analysis. We aim to express the MSE between the correct network output $\mathbf{x}^{(T)}$ and its noisy version $\mathbf{X}^{(T)}$. For this, we follow the approach of [10], and provide iterative expressions of the first and second-order moments (means, variances, and covariances) of the successive random variables $\mathbf{X}^{(t)}$ and $\mathbf{Z}^{(t)}$ calculated at each layer t of the network. These iterative expressions are obtained from second-order Taylor expansions [12] of the considered moments. Compared to [10] which focused on linear recursive computations, we must now take the non-linear computation (2) into account in the moment expressions. We must also consider the fact that the linear computation (3) is separated into one positive part and one negative part; the effect of this separation was not evaluated in [10]. Finally, in this section, we consider the viewpoint of one layer, and drop the index t in order to simplify the notation.

A. Moments after linear computation

In this part, we evaluate the second-order moments of the random vector $\mathbf{Z}$ after applying the linear operation (3) from memristor crossbars. We first rewrite the components Z_j of $\mathbf{Z}$ as

$$Z_j = Z_j^{(+)} - Z_j^{(-)} = \frac{T_j^{(+)}}{\Delta_j^{(+)}} - \frac{T_j^{(-)}}{\Delta_j^{(-)}}, \quad (4)$$

where $T_j^{(+)} = \sum_{i=1}^{N} G_{ij}^{(+)} X_i$, $\Delta_j^{(+)} = G_0 + \sum_{i=1}^{N} G_{ij}^{(+)}$, and likewise for $T_j^{(-)}$ and $\Delta_j^{(-)}$. We then introduce the notation $\mathbb{E}[X_i] = \nu_i$, $\mathbb{V}ar[X_i] = \gamma_i^2$, and for all $i' \neq i$, $\mathbb{C}[X_i, X_{i'}] = \gamma_{i,i'}$, respectively for the mean, variance, and covariance of the components X_i of $\mathbf{X}$. The next two theorems provide the mean, variance, and covariance of the positive components $Z_j^{(+)}$.

Theorem 1. *For all $j \in [\![1, N]\!]$, the second-order Taylor expansions of the mean $\mu_j^{(+)} = \mathbb{E}[Z_j^{(+)}]$ and variance $\rho_j^{(+)^2} = \mathbb{V}[Z_j^{(+)}]$ of $Z_j^{(+)}$ are given by*

$$\mu_j^{(+)} = \frac{\mathbb{E}[T_j^{(+)}]}{\mathbb{E}[\Delta_j^{(+)}]} - \frac{\mathbb{C}[T_j^{(+)}, \Delta_j^{(+)}]}{\mathbb{E}[\Delta_j^{(+)}]^2} + \frac{\mathbb{V}[\Delta_j^{(+)}]\mathbb{E}[T_j^{(+)}]}{\mathbb{E}[\Delta_j^{(+)}]^3}$$

$$\rho_j^{(+)^2} = \frac{\mathbb{V}[T_j^{(+)}]}{\mathbb{E}[\Delta_j^{(+)}]^2} + \frac{3\mathbb{E}[T_j^{(+)}]^2\mathbb{V}[\Delta_j^{(+)}]}{\mathbb{E}[\Delta_j^{(+)}]^4} - \frac{4\mathbb{E}[T_j^{(+)}]\mathbb{C}[\Delta_j^{(+)}, T_j^{(+)}]}{\mathbb{E}[\Delta_j^{(+)}]^3}$$

where

$$\mathbb{E}[T_j^{(+)}] = \sum_{i=1}^{N} g_{ij}^{(+)} \nu_i \tag{5}$$

$$\mathbb{E}[\Delta_j^{(+)}] = g_0 + \sum_{i=1}^{N} g_{ij}^{(+)} \tag{6}$$

$$\mathbb{V}[\Delta_j^{(+)}] = N_j^{(+)} \sigma^2 \tag{7}$$

$$\mathbb{C}[\Delta_j^{(+)}, T_j^{(+)}] = \sigma^2 \sum_{i=1}^{N} p_{i,j}^{(+)} \nu_i \tag{8}$$

$$\mathbb{V}[T_j^{(+)}] = \sigma^2 \sum_{i=1}^{N} p_{i,j}^{(+)} (\gamma_i^2 + \nu_i^2) + \sum_{i=1}^{N} \sum_{i'=1}^{N} g_{i,j}^{(+)} g_{i',j}^{(+)} \gamma_{i,i'} \tag{9}$$

In the above expressions, $p_{i,j}^{(+)} = 1$ *if* $g_{ij}^{(+)} \neq 0$, $p_{i,j}^{(+)} = 0$ *otherwise, and* $N_j^{(+)} = \sum_{i=1}^{N} p_{i,j}^{(+)}$.

Theorem 2. *For all* $(j,j)' \in [\![1,N]\!]^2$ *such that* $j' \neq j$, *the covariance* $\rho_{j,j'}^{(+)} = \mathbb{C}(Z_j^{(+)}, Z_{j'}^{(+)})$ *can be expressed as*

$$\rho_{j,j'}^{(+)} = \sum_{i=1}^{N} \sum_{i'=1}^{N} \mathbb{E}\left[\frac{G_{i,j}^{(+)}}{\Delta_j^{(+)}}\right] \mathbb{E}\left[\frac{G_{i',j'}^{(+)}}{\Delta_{j'}^{(+)}}\right] \gamma_{i,i'} \tag{10}$$

and the second-order Taylor expansion of $\mathbb{E}\left[\frac{G_{i,j}^{(+)}}{\Delta_j^{(+)}}\right]$ *is*

$$\mathbb{E}\left[\frac{G_{i,j}^{(+)}}{\Delta_j^{(+)}}\right] = \frac{\mathbb{E}[G_{i,j}^{(+)}]}{\mathbb{E}[\Delta_j^{(+)}]} - \frac{\mathbb{C}[G_{i,j}^{(+)}, \Delta_j^{(+)}]}{\mathbb{E}[\Delta_j^{(+)}]^2} + \frac{\mathbb{V}(\Delta_j^{(+)})\mathbb{E}[G_{i,j}^{(+)}]}{\mathbb{E}[\Delta_j^{(+)}]^3}$$

where $\mathbb{C}[G_{i,j}^{(+)}, \Delta_j^{(+)}] = \sigma^2$.

In order to obtain the mean, variance, and covariance of $Z_j^{(-)}$, it suffices to replace "+" by "−" in Theorems 1 and 2. The next theorem then provides the mean, variance, and covariance of the components Z_j of $\mathbf{Z}$.

Theorem 3. *For all* $(j,j)' \in [\![1,N]\!]^2$, *and* $j' \neq j$, *the mean* $\mu_j = \mathbb{E}[Z_j]$, *variance* ${\rho_j}^2 = \mathbb{V}[Z_j]$, *and covariance* $\rho_{j,j'} = \mathbb{C}(Z_j, Z_{j'})$ *are given by*

$$\mu_j = \mu_j^{(+)} - \mu_j^{(-)}$$

$$\rho_j^2 = {\rho_j^{(+)}}^2 + {\rho_j^{(-)}}^2 - 2\mathbb{C}[Z_j^{(+)}, Z_j^{(-)}]$$

$$\rho_{j,j'} = \rho_{j,j'}^{(+)} + \rho_{j,j'}^{(-)} - \mathbb{C}[Z_j^{(+)}, Z_{j'}^{(-)}] - \mathbb{C}[Z_j^{(-)}, Z_{j'}^{(+)}]$$

where

$$\mathbb{C}[Z_j^{(+)}, Z_j^{(-)}] = \sum_{i=1}^{N} \sum_{i'=1}^{N} \mathbb{E}\left[\frac{G_{i,j}^{(+)}}{\Delta_j^{(+)}}\right] \mathbb{E}\left[\frac{G_{i',j}^{(-)}}{\Delta_j^{(-)}}\right] \gamma_{i,i'}.$$

B. Moments after non-linear computation

We now evaluate the means, variances, and covariances, of random variables $X_i = f(Z_i)$, after the sigmoid activation function f.

Theorem 4. *For all* $(i,i)' \in [\![1,N]\!]^2$, *and* $i' \neq i$, *the second-order Taylor expansions of the mean* $\nu_i = \mathbb{E}[X_i]$ *and variance* $\gamma_i^2 = \mathbb{V}[X_i]$ *of the random variable* X_i, *are given by*

$$\nu_i = f(\mu_i) + \frac{1}{2} f''(\mu_i) \rho_i^2$$

$$\gamma_i^2 = \frac{1}{2} g''(\mu_i) \rho_i^2 - f(\mu_i) f''(\mu_i) \rho_i^2$$

$$\gamma_{i,i'} = f'(\mu_i) f'(\mu_{i'}) \rho_{i,i'}$$

where $f(u) = \frac{1}{1+\exp(-x)}$, *and* $g = f^2$.

This theorem can be adapted to any other twice-differentiable activation functions (tanh, softplus, etc.), by replacing the functions f, g, and their derivatives, according to the newly considered activation function.

To summarize, the four theorems given in this section provide recursive expressions of the first and second-order moments of random vectors $\mathbf{X}^{(t)}$ and $\mathbf{Z}^{(t)}$ at successive layers t. In addition, the MSE at each layer can be estimated as the average of the variance terms ρ_j^2 and γ_i^2. In the simulation section, we show from numerical simulations that these estimates provide good approximations of the successive MSE. At the end, the analysis we have developed allows us to investigate the effect of noisy conductance values onto the final NN output $\mathbf{Z}^{(T)}$, depending on various parameters such as the crossbar size, the conductance noise variance, etc..

IV. Power optimization of memristor-based DNNs

Having characterized the final MSE as a function of system parameters and chosen operating points, in this section we address the trade-off between the power consumption and the robustness to noise of memristor crossbars. We first evaluate the power consumption of a memristor crossbar, and then optimize this power consumption for a given target MSE at the network output. Here, we only evaluate the memory power consumption, and therefore do not take into account the non-linear activation parts, which may be realized in CMOS or similar technologies. As in Section III, we consider the viewpoint of one layer, and drop the index t in the notation.

A. Power consumption of a memristor crossbar

We assume that the variance σ^2 of conductance values $G_{ij}^{(+)}$ and $G_{ij}^{(-)}$ is fixed by internal properties of the memristors [13], [14]. Therefore, in order to make the computation more robust to noise, we consider coefficients c_j such that target conductance values are now given by $\tilde{g}_{i,j}^{(+)} = c_j g_{i,j}^{(+)}$, $\tilde{g}_{i,j}^{(-)} = c_j g_{i,j}^{(-)}$, and $\tilde{g_0} = c_j g_0$. This does not change the computation operation (3), since the coefficient c_j appears as a factor in both parts of the ratio for Z_j. The corresponding random variables $\tilde{G}_{i,j}^{(+)}$ and $\tilde{G}_{i,j}^{(-)}$ then have means $c_j g_{i,j}^{(+)}$ and $c_j g_{i,j}^{(-)}$, respectively, and both have variance σ^2. Note that it is physically possible for each component Z_j to have its own parameter c_j, which in turn leads to many degrees of freedom in the optimization.

Then, for the positive part of the computation, the power $P_{i,j}^{(+)}$ consumed by the memristor at position (i,j) is a random variable which can be expressed as $P_{i,j}^{(+)} = \tilde{G}_{i,j}^{(+)} U_{i,j}^2$, where $U_{i,j}$ is the voltage across memristor at position (i,j). We can note that

$$U_{i,j} = (X_i - Z_j^{(+)}) = \left(X_i - \frac{\sum_{i'=1}^{N} \tilde{G}_{i',j}^{(+)} X_{i'}}{\tilde{G}_0 + \sum_{i'=1}^{N} \tilde{G}_{i',j}^{(+)}} \right). \tag{11}$$

Since $P_{i,j}$ is a random variable, we are interested in its expectation $\mathbb{E}[P_{i,j}^{(+)}]$, which can be rewriten as

$$\mathbb{E}[P_{i,j}^{(+)}] = \mathbb{E}\left[\tilde{G}_{i,j}^{(+)} \frac{\tilde{V}_{i,j}^{(+)}}{\tilde{\Delta}_j^{(+)}}\right] \tag{12}$$

where $\tilde{V}_{i,j}^{(+)} = \sum_{i'=1}^{N} \tilde{G}_{i',j}^{(+)}(X_i - X_{i'})$, and $\tilde{\Delta}_j^{(+)} = \tilde{G}_0 + \sum_{i'=1}^{N} \tilde{G}_{i',j}^{(+)}$. The following theorem gives the second-order Taylor expansion of $\mathbb{E}[P_{i,j}^{(+)}]$.

Theorem 5. *For all $(i,j) \in [\![1,N]\!]^2$, the second-order Taylor expansion of $\mathbb{E}[P_{i,j}^{(+)}]$ is given by*

$$\mathbb{E}[P_{i,j}^{(+)}] = \frac{\mathbb{E}[\tilde{G}_{ij}^{(+)}]\mathbb{E}[{V_{i,j}^{(+)}}^2]}{\mathbb{E}[\Delta_j^{(+)}]^2} - \frac{4\mathbb{E}[V_{i,j}^{(+)}]\mathbb{E}[\tilde{G}_{i,j}^{(+)}]\mathbb{C}[V_{i,j}^{(+)}, \Delta_j^{(+)}]}{\mathbb{E}[\Delta_j^{(+)}]^3}$$
$$- \frac{2\mathbb{E}[V_{i,j}^{(+)}]^2\mathbb{C}[\tilde{G}_{i,j}^{(+)}, \Delta_j^{(+)}]}{\mathbb{E}[\Delta_j^{(+)}]^3} + \frac{3\mathbb{E}[\tilde{G}_{ij}^{(+)}]\mathbb{E}[V_{i,j}^{(+)}]^2\mathbb{V}[\Delta_j^{(+)}]}{\mathbb{E}[\Delta_j^{(+)}]^4}$$

where $\mathbb{E}[\tilde{G}_{ij}^{(+)}] = c_j g_{i,j}^{(+)}$, $\mathbb{C}(\tilde{G}_{i,j}^{(+)}, \Delta_j^{(+)}) = \sigma^2$, $\mathbb{V}(\Delta_j^{(+)}) = N_j^{(+)}\sigma^2$, and

$$\mathbb{E}[V_{i,j}^{(+)}] = c_j \sum_{i'=1}^{N} g_{i',j}^{(+)}(\nu_i - \nu_{i'})$$

$$\mathbb{E}[\Delta_j^{(+)}] = c_j \left(g_0 + \sum_{i'=1}^{N} g_{i',j}^{(+)}\right)$$

$$\mathbb{C}[V_{i,j}^{(+)}, \Delta_j^{(+)}] = \sigma^2 \sum_{i'=1}^{N} p_{i,j}^{(+)}(\nu_i - \nu_{i'})$$

$$\mathbb{V}[V_{i,j}^{(+)}] = \sum_{(i',k)\ i'\neq i,k\neq i,k\neq i'_i} c_j^2 g_{i',j}^{(+)} g_{k,j}^{(+)}(\gamma_i^2 - \gamma_{i,i'} - \gamma_{i,k} + \gamma_{i',k})$$
$$+ \sum_{i'=1}^{N} p_{i,j}^{(+)} \left((\sigma^2 + c_j^2 {g_{i',j}^{(+)}}^2)(\gamma_i^2 + \gamma_{i'}^2 - 2\gamma_{i,i'}) + 2\sigma^2(\nu_i - \nu_{i'})^2\right).$$

Finally, for one crossbar, we express the expected total power as $\mathbb{E}[P] = \sum_{i,j}\left(\mathbb{E}[P_{i,j}^{(+)}] + \mathbb{E}[P_{i,j}^{(-)}]\right)$, where $\mathbb{E}[P_{i,j}^{(-)}]$ is also obtained from Theorem 5 by replacing "+" by "−" in the various expressions. The average total power can then be evaluated for the full DNN architecture, by summing over all T crossbars.

B. Power optimization

In this subsection, we use $\tilde{\rho}_j$ to denote the variance of component Z_j, calculated given that the conductance values were multiplied by c_j. In addition, we can show that $\tilde{\rho}_j = \rho_j^2/c_j^2$, where ρ_j^2 was given in Theorem 3. Then, at each layer of the network, both the average power $\mathbb{E}[P]$ and the variances $\tilde{\rho}_j^2$ depend on the parameters c_j. In this section, we propose to optimize these parameters so as to minimize $\mathbb{E}[P]$ under a certain performance constraint expressed onto the targeted variance of each component Z_j of $\mathbf{Z}^{(t)}$. We treat layers one after the other, assuming that previous layers from 1 to $(t-1)$ were already optimized when considering layer t. Therefore, we define the following optimization problem for layer t:

$$\min_{c_j} \sum_{i,j}\left(\mathbb{E}[P_{i,j}^{(+)}] + \mathbb{E}[P_{i,j}^{(-)}]\right) \text{ s.t. } \forall j,\ \tilde{\rho}_j^2 \leq \overline{\rho},\ c_j \geq 0 \tag{13}$$

where $\overline{\rho}$ is a constraint on the maximum variance on each component Z_j at the output of the linear part. We further constrain that $c_j \geq 0$ in order to get positive conductance values. Then, the Lagrangian of the constrained optimization problem (13) is given by

$$\mathcal{L} = \sum_{i,j}\left(\mathbb{E}[P_{i,j}^{(+)}] + \mathbb{E}[P_{i,j}^{(-)}]\right) + \sum_j \alpha_j(\tilde{\rho}_j^2 - \overline{\rho}) - \beta_j c_j \tag{14}$$

and can be rewritten as

$$\mathcal{L} = \sum_{j=1}^{N}\left(F_{1,j}c_j + \frac{F_{2,j}}{c_j} + \alpha_j\left(\frac{\rho_j^2}{c_j^2} - \overline{\rho}\right) - \beta_j c_j\right) \tag{15}$$

where the terms $F_{1,j}$ and $F_{2,j}$ can be identified from the expressions of $\mathbb{E}[P_{i,j}]$ given in Theorem 5, and we show that $F_{1,j} \geq 0$ and $F_{2,j} \leq 0$. The Lagrangian is separable with respect to j, and its derivative with respect to c_j is

$$\frac{\partial \mathcal{L}}{\partial c_j} = F_{1,j} - \frac{F_{2,j}}{c_j^2} - 2\alpha_j\frac{\rho_j^2}{c_j^3} - \beta_j. \tag{16}$$

We now apply the KKT conditions in order to solve the optimization problem. The constraint $c_j \geq 0$ is necessarily inactive ($c_j > 0$, $\beta_j = 0$) since $c_j = 0$ does not allow us to satisfy $\tilde{\rho}_j^2 \leq \overline{\rho}$. In addition, considering an inactive constraint $\tilde{\rho}_j < \overline{\rho}$ with $\alpha_j = 0$ does not lead to a solution for c_j. Therefore, the constraint $\tilde{\rho}_j \leq \overline{\rho}$ is active, which leads to

$$c_j = \sqrt{\rho_j^2/\overline{\rho}}. \tag{17}$$

Finally, by calculating α_j from the condition $\frac{\partial \mathcal{L}}{\partial c_j} = 0$, the second-order derivative of the Lagrangian is given by

$$\frac{\partial^2 \mathcal{L}}{\partial c_j^2} = \frac{1}{c_j^2}\left(3F_{1,j}c_j - \frac{F_{2,j}}{c_j}\right) > 0, \tag{18}$$

which allows us to conclude that (17) provides a strict global minimum. Therefore, the values of c_j in (17) allow us to minimize the power consumption of a crossbar, while satisfying the variance conditions $\tilde{\rho}_j^2 \leq \overline{\rho}$.

V. Simulation results

In this part, we first evaluate the accuracy of the proposed theoretical analysis. We consider a Neural Network structure with 7 layers, with the following number of neurons per layer: $(100, 100, 200, 150, 120, 80, 10)$, and with sigmoid activation functions. We generate the weight matrices $\mathcal{W}^{(t)}$ at random and uniformly between 0 and 10, and then convert them into conductance values $g_{i,j}^{(t,+)}$ and $g_{i,j}^{(t,-)}$, with $g_0 = 10$. We also generate input vectors $\mathbf{x}$ at random, with components distributed uniformly between -5 and 5. In Figure 1, we compare the variance measured from Monte-Carlo simulations and evaluated with the theoretical analysis of Section III, with respect to the noise variance σ^2, after one layer and after the seven layers. After one layer, we observe that the theoretical variance value accurately predicts the MSE measured from Monte Carlo simulations: the two curves are visually indistinguishable. In addition, as expected, the MSE increases with σ^2. Then, after seven layers, we also observe that the

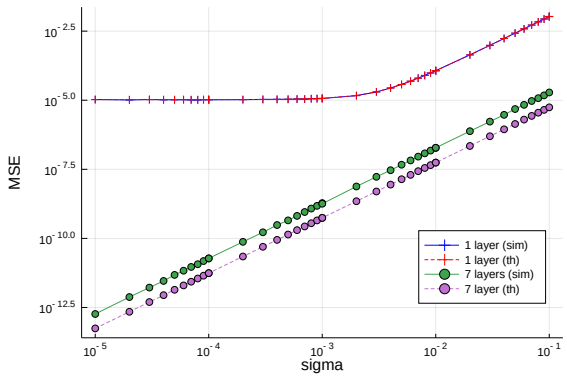

Fig. 1. Comparison between MSE measured from Monte Carlo simulations and evaluated with the theoretical analysis developed therein. The red and blue curve (one layer) are superimposed.

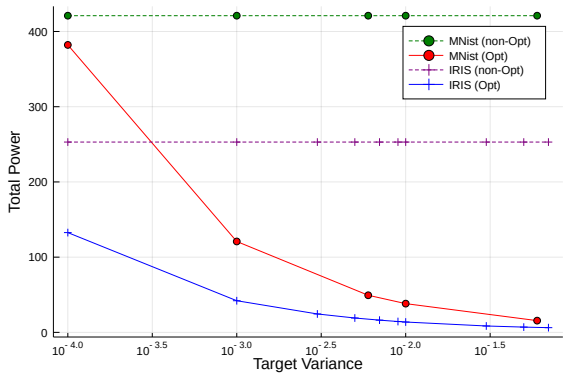

Fig. 2. Optimized and non-optimized total power calculated over the T layers, for two NNs. The non-optimized case does not necessarily achieve the final target variance.

theoretical variance values are close to the MSE measured from simulations, although there is a small gap which is probably due to error accumulation over the successive layers. Finally, we note that after one layer, the MSE is higher or very close to the corresponding noise variance σ^2. On the opposite, after seven layers, the MSE is actually smaller than the noise variance value σ^2 on memristors. This shows that memristor crossbar computation can be inherently robust to noise on conductance values.

We then evaluate the power optimization method proposed in Section IV. To do so, we consider two standard datasets that are IRIS [15] and MNIST [16]. For IRIS (input length 4), we consider a NN with 2 layers and respectively 50 and 10 neurons on each layer, and for MNIST (input length 784), we consider a NN with 3 layers, and respectively 200, 50, 10, neurons on each layer. Both networks integrate sigmoid activation functions, and were trained over $2/3$ of the dataset, with 100 epochs and learning rate $\eta = 0.1$. For both networks, we set $\sigma = 10^{-2}$, and we optimize the parameters c_j with the method of Section IV, for various target variance values $\overline{\rho}$. The results are shown in Figure 2, where we also display the total power in the non-optimized case (this non-optimized total power is for comparison and does not necessarily allow to achieve the corresponding value of $\overline{\rho}$). We observe a clear power gain after optimization, which is due to the fact that the optimized values c_j allow to exactly reach the target variance $\overline{\rho}$. In addition, as expected, the total power decreases as $\overline{\rho}$ increases, and the NN for MNIST requires more power than the NN for IRIS.

VI. Conclusion

In this paper, we considered a DNN implementation from noisy memristor crossbars, and we developed a theoretical analysis to predict the effect of noise onto the final inference computation performance. We further proposed a method to optimize the memristor crossbars power consumption, and demonstrated significant power gains compared to the non-optimized case. Future works will include the study of other neural network architectures such as convolutional neural networks and recurrent neural networks.

References

[1] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *nature*, vol. 453, no. 7191, p. 80, 2008.

[2] C. Yakopcic, M. Z. Alom, and T. M. Taha, "Memristor crossbar deep network implementation based on a convolutional neural network," in *IJCNN*, Jul. 2016, pp. 963–970.

[3] S. Liu, Y. Wang, M. Fardad, and P. K. Varshney, "A memristor-based optimization framework for artificial intelligence applications," *IEEE Circuits and Systems Magazine*, vol. 18, no. 1, pp. 29–44, 2018.

[4] C. Li, Z. Wang, M. Rao, D. Belkin, W. Song, H. Jiang, P. Yan, Y. Li, P. Lin, M. Hu, N. Ge, J. P. Strachan, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang, and Q. Xia, "Long short-term memory networks in memristor crossbar arrays," *Nature Machine Intelligence*, vol. 1, pp. 49–57, 2019.

[5] J. Kendall, R. Pantone, K. Manickavasagam, Y. Bengio, and B. Scellier, "Training end-to-end analog neural networks with equilibrium propagation," arXiv:2006.01981 [cs.NE]., Jun. 2020.

[6] A. Karbasi, A. H. Salavati, A. Shokrollahi, and L. R. Varshney, "Noise facilitation in associative memories of exponential capacity," *Neural Computation*, vol. 26, no. 11, pp. 2493–2526, Nov. 2014.

[7] S. Kim, P. Howe, T. Moreau, A. Alaghi, L. Ceze, and V. S. Sathe, "Energy-efficient neural network acceleration in the presence of bit-level memory errors," *IEEE Trans. on Circuits and Systems I: Regular Papers*, pp. 1–14, 2018.

[8] G. B. Hacene, F. Leduc-Primeau, A. B. Soussia, V. Gripon, and F. Gagnon, "Training modern deep neural networks for memory-fault robustness," in *ISCAS*, 2019.

[9] A. Chatterjee and L. R. Varshney, "Energy-reliability limits in nanoscale feedforward neural networks and formulas," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 250–266, May 2020.

[10] E. Dupraz and L. R. Varshney, "Noisy in-memory recursive computation with memristor crossbars," in *Proceedings of the 2020 IEEE International Symposium on Information Theory (ISIT)*, Jun. 2020, pp. 804–809.

[11] K. K. Likharev and D. B. Strukov, "CMOL: Devices, circuits, and architectures," in *Introducing Molecular Electronics*, ser. Lecture Notes in Physics, G. Cuniberti, K. Richter, and G. Fagas, Eds. Berlin: Springer, 2006, vol. 680, pp. 447–477.

[12] H. Seltman, "Approximations for mean and variance of a ratio," *unpublished note*, 2012.

[13] J. A. Starzyk and Basawaraj, "Memristor crossbar architecture for synchronous neural networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 8, pp. 2390–2401, Aug. 2014.

[14] A. James and L. Chua, "Analog neural computing with super-resolution memristor crossbars," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2021, to appear.

[15] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[16] Y. LeCun, C. Cortes, and C. J. Burges, "The MNIST database of handwritten digits," 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/