Neural Decoder for Analog ECC

Xiangwu Zuo and Anxiao (Andrew) Jiang, *Senior Member, IEEE*

***Abstract*—Non-volatile memories (NVMs) have been pivotal in implementing deep neural networks in analog circuits. Analog error-correcting codes (Analog ECCs) have been proposed to make their computation more reliable. Although a number of Analog ECCs have already been designed, how to develop decoders for them remains largely unknown. The only decoder known so far is for the analog $[n, 1]$ repetition code.**

This work explores the designs of neural networks as decoders for Analog ECCs. Principled approaches are used for decoding, including error locating and using regression to find the values of errors. An ensemble method is used to improve the accuracy of error locating. And a transformer-based model is shown to achieve good regression performance (e.g., increasing the signal-to-noise ratio by over 10 or 20 dB).

I. Introduction

Nonvolatile memories (NVMs) have been pivotal in the realization of deep neural networks (DNNs) in analog circuits. Analog chips for DNNs based on phase-change memories (PCMs) and memristors have been developed [3]. The most widely used operations in DNNs, the vector-matrix multiplications, are typically implemented using an NVM crossbar architecture, which computes at a much higher speed and power efficiency than its digital counterpart. A main challenge for the analog computing, however, is the accuracy of computing affected by noise. To make the analog computing of vector-matrix multiplications more reliable, Analog Error-Correcting Codes (Analog ECCs) have been proposed recently [2].

An Analog ECC is defined as follows. Let $G = (g_{i,j})_{k\times n} \in \mathbb{R}^{k\times n}$ be a real-valued $k\times n$ matrix. Then a linear $[n, k]$ Analog ECC $\mathcal{C}$ is $\mathcal{C} = \{\mathbf{u}G|\mathbf{u} \in \mathbb{R}^k\} \subseteq \mathbb{R}^n$. Let $[n\rangle \triangleq \{0, 1, \cdots, n-1\}$ for any $n \in \mathbb{Z}^+$. Given a vector $\mathbf{e} = (e_0, e_1, \cdots, e_{n-1}) \in \mathbb{R}^n$, define its support with respective to a threshold $\nu \geq 0$ as $\text{Supp}_\nu(\mathbf{e}) = \{i \in [n\rangle \ : \ |e_i| > \nu\}$. Then its Hamming weight $\text{w}_H(\mathbf{e})$ is $|\text{Supp}_0(\mathbf{e})|$. Let $\delta, \Delta \in \mathbb{R}^+$ be two thresholds with $\Delta > \delta$. Consider two types of errors: *limited-magnitude errors* (LMEs), and *unlimited-magnitude errors* (UMEs). An error vector $\boldsymbol{\varepsilon} = (\varepsilon_0, \varepsilon_1, \cdots, \varepsilon_{n-1}) \in \mathbb{R}^n$ is called an LME vector if $\varepsilon_i \in [-\delta, \delta]$ for all $i \in [n\rangle$. An error vector $\mathbf{e} = (e_0, e_1, \cdots, e_{n-1}) \in \mathbb{R}^n$ is called an UME vector of Hamming weight w if $\text{w}_H(\mathbf{e}) = w$. Here LMEs model small ubiquitous noise (which DNNs can often tolerate), and UMEs model more significant errors (e.g., stuck memory cells, short cells, etc.).

Analog ECCs focus on correcting UMEs, especially those UMEs whose magnitudes are above Δ. A *decoder* for a linear $[n, k]$ Analog ECC $\mathcal{C}$ is a function $\mathcal{D} : \mathbb{R}^n \to 2^{[n\rangle}$ that returns a set of locations of UMEs. $\mathcal{D}$ is said to "correct t UMEs" if for any noisy codeword $\mathbf{y} = \mathbf{c}+\boldsymbol{\varepsilon}+\mathbf{e}$ (where $\mathbf{c} = (c_0, c_1, \cdots, c_{n-1}) \in \mathcal{C}$ is a codeword, $\boldsymbol{\varepsilon}$ is an LME vector, $\mathbf{e}$ is an UME vector with $\text{w}_H(\mathbf{e}) \leq t$), we have $\text{Supp}_\Delta(\mathbf{e}) \subseteq \mathcal{D}(\mathbf{y}) \subseteq \text{Supp}_0(\mathbf{e})$.

X. Zuo and A. Jiang are with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX, 77845 USA. E-mail: dkflame@tamu.edu and ajiang@cse.tamu.edu.

The above definition of error correction [2] focuses on finding the locations of UMEs. So we also call it *UME-locating*. In this paper, we further consider how to find the values of UMEs, which we call *error-regression*, defined as follows. An error-regression (ER) decoder is a function $\mathcal{D}_{ER}$ such that given a noisy codeword $\mathbf{y} = \mathbf{c}+\boldsymbol{\varepsilon}+\mathbf{e}$ and its decoding result $\mathcal{D}(\mathbf{y}) = \{i_0, i_1, \cdots, i_{|\mathcal{D}(\mathbf{y})|-1}\} \subseteq [n\rangle$, returns a vector $\mathcal{D}_{ER}(\mathbf{y}, \mathcal{D}(\mathbf{y})) = (\sigma_{i_0}, \sigma_{i_1}, \cdots, \sigma_{i_{|\mathcal{D}(\mathbf{y})|-1}}) \in \mathbb{R}^{|\mathcal{D}(\mathbf{y})|}$. (Here we let $i_0 < i_1 < \cdots < i_{|\mathcal{D}(\mathbf{y})|-1}$.) For $j \in [|\mathcal{D}(\mathbf{y})|\rangle$, σ_{i_j} is an estimation of the value $e_{i_j} + \varepsilon_{i_j}$, which combines the UME and LME at location i_j (as it is difficult to separate the two errors in the same location).

We measure the performance of the two decoders as follows. (This work focuses on the probabilistic decoding performance, instead of worst-case performance.) Let $\mathcal{P}_c$, $\mathcal{P}_\varepsilon$, $\mathcal{P}_e$ and $\mathcal{P}_y$ denote the probability distributions of $\mathbf{c}$, $\boldsymbol{\varepsilon}$, $\mathbf{e}$ and $\mathbf{y} = \mathbf{c}+\boldsymbol{\varepsilon}+\mathbf{e}$, respectively. Let $\tau(\mathbf{y}) = 1$ if $\text{Supp}_\Delta(\mathbf{e}) \subseteq \mathcal{D}(\mathbf{y}) \subseteq \text{Supp}_0(\mathbf{e})$, and let $\tau(\mathbf{y}) = 0$ otherwise. The *accuracy* of the decoder $\mathcal{D}$, $Acc(\mathcal{D})$, is defined as $E_{\mathbf{y}\sim\mathcal{P}_y}(\tau(\mathbf{y}))$. (Here $E(\cdot)$ is the expectation. In experiments, we replace the expectation by its empirical average.) For the ER decoder $\mathcal{D}_{ER}$, we measure its performance by the *Increase in Signal-to-Noise Ratio* SNR_{inc} defined as follows. Let $P \triangleq \text{Supp}_\Delta(\mathbf{e}) \cup \mathcal{D}(\mathbf{y})$ be our "positions of interest", which include the positions where large UMEs exist and need to be corrected (i.e., $\text{Supp}_\Delta(\mathbf{e})$) and positions where we actually correct errors (i.e., $\mathcal{D}(\mathbf{y})$). Define $\sigma_i = 0$ if $i \notin \mathcal{D}(\mathbf{y})$. Then $\text{SNR}_{inc} \triangleq 10\log_{10}\frac{E(\sum_{i\in P} c_i^2)}{E(\sum_{i\in P}(e_i+\varepsilon_i-\sigma_i)^2)} - 10\log_{10}\frac{E(\sum_{i\in P} c_i^2)}{E(\sum_{i\in P}(e_i+\varepsilon_i)^2)} = 10\log_{10}\frac{E(\sum_{i\in P}(e_i+\varepsilon_i)^2)}{E(\sum_{i\in P}(e_i+\varepsilon_i-\sigma_i)^2)}$ dB.

A number of Analog ECCs have been designed [1], [2], which focus on encoding (i.e., generator matrices) and their theoretical error-correction capabilities. However, how to design decoders for Analog ECCs remains largely unknown. The only known decoder is for the $[n, 1]$ repetition code for UME-locating, along with some discussions on bounding the sizes of UMEs [2]. In this work, we explore the designs of decoders for Analog ECCs based on neural networks for both UME-locating and error-regression, and demonstrate the performance of the decoders through extensive experiments.

II. Main Results

A. Neural Decoders for UME-Locating

For $i \in [n\rangle$, let q_i be 0 if $e_i = 0$, be 1 if $|e_i| \in (0, \Delta]$, and be 2 if $|e_i| > \Delta$. Let $Q(\mathbf{e}) = (q_0, q_1, \cdots, q_{n-1}) \in \{0, 1, 2\}^n$ be the *quantized-UME pattern*. Given that $\text{w}_H(\mathbf{e}) \leq t$, there are totally $T \triangleq \sum_{i=0}^{t} 2^i \binom{n}{i}$ such patterns. Let $\mathcal{N}_\alpha$ be a T-class classification neural network that, given a noisy codeword $\mathbf{y} = \mathbf{c}+\boldsymbol{\varepsilon}+\mathbf{e}$ as input, classifies its UME vector

$\mathbf{e}$ into one of those T patterns. Let $\mathcal{N}_\alpha(\mathbf{y})$ denote the pattern that $\mathcal{N}_\alpha$ predicts to have the maximum likelihood. Then the learning objective for $\mathcal{N}_\alpha$ is to make $\mathcal{N}_\alpha(\mathbf{y}) = Q(\mathbf{e})$. Let $\mathcal{N}_\alpha(\mathbf{y}) = (\hat{q}_0, \hat{q}_1, \cdots, \hat{q}_{n-1}) \in \{0,1,2\}^n$. Let $\mathcal{D}_\alpha(\mathbf{y}) = \{i \in [n\rangle | \hat{q}_i = 1 \text{ or } 2\}$. Such a decoder $\mathcal{D}_\alpha$ tends to have much fewer false negatives (i.e., excluding i from $\mathcal{D}_\alpha(\mathbf{y})$ when $|e_i| > \Delta$) than false positives (i.e., including i in $\mathcal{D}_\alpha(\mathbf{y})$ when $e_i = 0$). Such a property can be used to build a stronger decoder $\mathcal{D}_\beta$.

Let $m \in \mathbb{Z}^+$. Let $\mathcal{N}_\alpha^{[0]}$, $\mathcal{N}_\alpha^{[1]}$, $\cdots$, $\mathcal{N}_\alpha^{[m-1]}$ be m independently trained neural decoders as $\mathcal{N}_\alpha$ above. For $i \in [m\rangle$, let $\mathcal{D}_\alpha^{[i]}(\mathbf{y})$ be the output of $\mathcal{N}_\alpha^{[i]}$. Let $\mathcal{D}_\beta(\mathbf{y}) = \cap_{i \in [m\rangle} \mathcal{D}_\alpha^{[i]}(\mathbf{y})$. Then $\mathcal{D}_\beta$ is a neural decoder based on ensemble learning, whose accuracy $Acc(\mathcal{D}_\beta)$ can be optimized by tuning m.

Due to page limitation, we demonstrate the performance of the neural decoders $\mathcal{D}_\alpha$ and $\mathcal{D}_\beta$ via two examples of Analog ECCs. The first code $\mathcal{C}_1$ is a linear $[12, 8]$ code from [2]. It is proved that when $\Delta/\delta \geq 12$, theoretically there exists a decoder that can achieve perfect accuracy for UME-locating when $t = 1$. The second code $\mathcal{C}_2$ is a linear $[10, 4]$ code from [1]. It is proved that when $\Delta/\delta \geq 16.45$, theoretically there exists a decoder that can achieve perfect accuracy for UME-locating when $t = 2$. The generator matrices G of the two codes are shown in an extended version of this paper [4].

Let $\mathbf{u} = (u_0, u_1, \cdots, u_k)$, where each u_i is uniformly distributed in $[-1, 1]$. Let codeword $\mathbf{c} = \mathbf{u}G$. Let each LME ε_i be uniformly distributed in $[-\delta, \delta]$. For $i \in [t+1\rangle$, let θ_i denote the probability that $\mathrm{w}_H(\mathbf{e}) = i$, with $\sum_{i=0}^{t} \theta_i = 1$; and each UME follows a normal distribution with mean 0 and standard deviation η. Let $\mathcal{N}_\alpha$ be a dense classification neural network with three hidden layers of 64 neurons each, whose input size is n (for the n numbers in a noisy codeword $\mathbf{y}$) and output size is T (as a probability distribution over T classes).

The experimental performance is as follows. For code $\mathcal{C}_1$, let $\delta = 0.05$, $\Delta = 0.6$ (so that $\Delta/\delta = 12$), $\theta_0 = 0.2$, $\theta_1 = 0.8$, $\eta = 0.6$. We get $Acc(\mathcal{D}_\alpha) = 95.8\%$. For code $\mathcal{C}_2$, let $\delta = 0.05$, $\Delta = 0.85$ (so that $\Delta/\delta = 17 > 16.45$), $\theta_0 = 0.04$, $\theta_1 = 0.32$, $\theta_2 = 0.64$, $\eta = 0.85$. We get $Acc(\mathcal{D}_\alpha) = 94.6\%$. When Δ/δ increases up to 30 (by increasing Δ and let $\eta = \Delta$), $Acc(\mathcal{D}_\alpha)$ further increases in general, as shown in Fig. 1 (a), where the y-axis is $1 - Acc(\mathcal{D}_\alpha)$.

Now consider the ensemble-based decoder $\mathcal{D}_\beta$. For code $\mathcal{C}_1$, let $\Delta/\delta = 12$. When m increases from 1 to 10, $Acc(\mathcal{D}_\beta)$ increases from 95.8% to 98.9%. When m increases up to 300, the values of $1 - Acc(\mathcal{D}_\beta)$ are shown as the red curve in Fig. 1 (b). $\mathcal{D}_\beta$ reaches the highest accuracy 99.6% when $m = 228$. For code $\mathcal{C}_2$, let $\Delta/\delta = 17$. When m increases up to 200, the values of $1 - Acc(\mathcal{D}_\beta)$ are shown as the green curve in Fig. 1 (b). $\mathcal{D}_\beta$ reaches the highest accuracy 99.8% when $m = 12$.

B. Neural Decoder for Error-Regression

For $i \in [n\rangle$, let $\tilde{q}_i = -2, -1, 0, 1, 2$ when $e_i < -\Delta$, $e_i \in [-\Delta, 0)$, $e_i = 0$, $e_i \in (0, \Delta]$, $e_i > \Delta$, respectively. Let $\tilde{Q}(\mathbf{e}) = (\tilde{q}_0, \tilde{q}_1, \cdots, \tilde{q}_{n-1}) \in \{-2, -1, 0, 1, 2\}^n$, which is a more refined error pattern compared to $Q(\mathbf{e})$. Given that $\mathrm{w}_H(\mathbf{e}) \leq t$, there are totally $\tilde{T} \triangleq \sum_{i=0}^{t} 4^i \binom{n}{i}$ such patterns. Let $\mathcal{N}_\gamma$ be a $\tilde{T}$-class classification neural network, which is similar to $\mathcal{N}_\alpha$ except that it has $\tilde{T}$ outputs instead of T outputs,

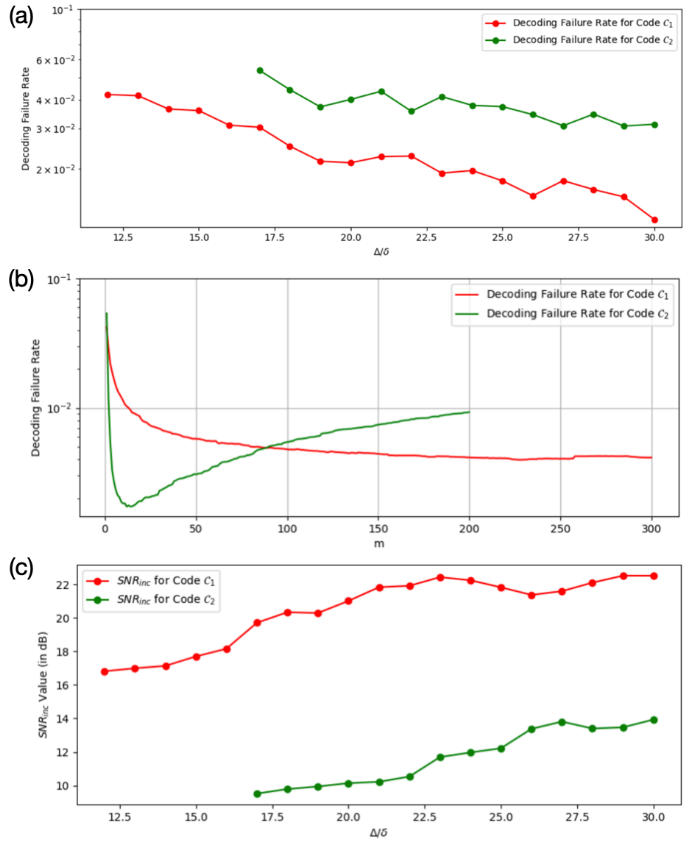

Fig. 1. Experimental performance of neural decoders. (a) Decoding failure rate (i.e., $1 - Acc(\mathcal{D}_\alpha)$) for different Δ/δ. (b) Decoding failure rate (i.e., $1 - Acc(\mathcal{D}_\beta)$) for different values of m. (c) SNR$_{inc}$ for different Δ/δ.

and let $\mathcal{N}_\gamma(\mathbf{y})$ denote its predicted pattern given input $\mathbf{y}$. (Its learning objective is to make $\mathcal{N}_\gamma(\mathbf{y}) = \tilde{Q}(\mathbf{e})$.) Let $\mathcal{N}_\lambda$ be a Transformer neural network model (based on self-attention), which takes $\mathbf{y}$ and $\mathcal{N}_\gamma(\mathbf{y})$ as input and outputs t numbers, the first $\mu \triangleq \mathrm{w}_H(\mathcal{N}_\gamma(\mathbf{y}))$ of which are the predicted values for the UME+LME values at the μ positions indicated by the non-zeros in $\mathcal{N}_\gamma(\mathbf{y})$. (Here $\mathcal{N}_\gamma(\mathbf{y})$ serves the role of $\mathcal{D}(\mathbf{y})$, and $\mathcal{N}_\lambda$ serves the role of the error-regression decoder $\mathcal{D}_{ER}$.)

The neural decoder $\mathcal{N}_\lambda$ consists of a concatenation layer (which combines $\mathbf{y}$ and $\mathcal{N}_\gamma(\mathbf{y})$ into a sequence of shape $(n, 2)$), a positional-embedding layer, a transformer block, a global average pooling layer and 3 dense layers.

The experimental performance is as follows. For code $\mathcal{C}_1$, let δ, Δ, θ_0, θ_1 and η be as before (with $\Delta/\delta = 12$). We get SNR$_{inc}$ = 16.82 dB. For code $\mathcal{C}_2$, let δ, Δ, θ_0, θ_1, θ_2, η be as before (with $\Delta/\delta = 17$). We get SNR$_{inc}$ = 9.52 dB. Fig. 1 (c) shows the SNR$_{inc}$ values for $\mathcal{C}_1$ and $\mathcal{C}_2$ for different Δ/δ.

References

[1] A. Jiang and X. Zuo, "Analysis and Designs of Analog ECC," submitted to Non-Volatile Memories Workshop (NVMW) 2024.

[2] R. M. Roth, "Analog Error-Correcting Codes," in *IEEE Transactions on Information Theory*, vol. 66, no. 7, pp. 4075-4088, July 2020.

[3] W. Zhang *et al.*, "Edge Learning Using a Fully Integrated Neuro-inspired Memristor Chip," in *Science*, vol. 381, no. 6663, pp. 1205-1211, 2023.

[4] X. Zuo and A. Jiang, "Neural Decoder for Analog ECC," December 2023. Available at $https://drive.google.com/file/d/1DaneaeYIfuQoHT6ZxuE4OHJd68EErXCy$.