Error Correction for Hardware-Implemented Deep Neural Networks

Pulakesh Upadhyaya, Xiaojing Yu, Jacob Mink, Jeffrey Cordero, Palash Parmar, Anxiao (Andrew) Jiang
Computer Science and Engineering Department, Texas A&M University

I. Introduction

Deep learning has become a key driving force for artificial intelligence (AI) and has found many applications. It is important to study the implementation of Deep Neural Networks (DNNs) in hardware, because hardware-implemented DNNs are faster and more energy-efficient. In hardware, the real values of weights in the DNN can be stored in non-volatile memory (NVM) cells (e.g. memristors). However, over time, various types of noise will appear in the cells and degrade the performance of DNNs. Hence, it is important to find error correcting schemes for protecting weights of DNNs, and optimize their performance.

In this work, we study how the performance of DNNs degrades when noise is present. We focus on two analog error correcting codes (ECCs) which are suitable for protecting analog weights in DNNs. In the first code, we have designed a *systematic linear analog code*, which allows the weights of the DNN to be stored in their original form. In the second code, which is a *systematic non-linear analog code*, we design a new maximum a posteriori (MAP) decoder for enhanced error correction. We experimentally show that these codes can significantly improve the performance of DNNs. We then extend the study to binarized DNN, and show how noise in different layers affects the DNN performance in different ways. This observation is useful for optimizing the code rates of ECCs protecting different layers of the DNN.

II. How Noise Deteriorates DNN performance

In this section, we study the performance of DNNs when weights are susceptible to noise from the Additive White Gaussian Noise (AWGN) channel. We measure the performance by *accuracy* of DNNs, i.e. their fraction of correct classification outputs. We consider different kinds of DNN models. For example, we use two trained convolutional neural networks(CNNs) of around 1.2 million weights each for the MNIST and CIFAR-10 datasets. MNIST is a well-known dataset of handwritten digits, whereas CIFAR-10 is a dataset used for classification of images into 10 classes (eg. airplanes, cats, dogs etc.). We also use a long short term memory (LSTM) network for the IMDB dataset, which assigns movie rating as "good" or "bad" based on text reviews. Further details on the above models can be found in [2]. The decrease in accuracy caused by decrease in SNR is shown in Fig. 1 a) in solid lines. It is observed that, while the presence of noise decreases the performance of all DNNs, the loss in accuracy can be significant for models like CIFAR-10.

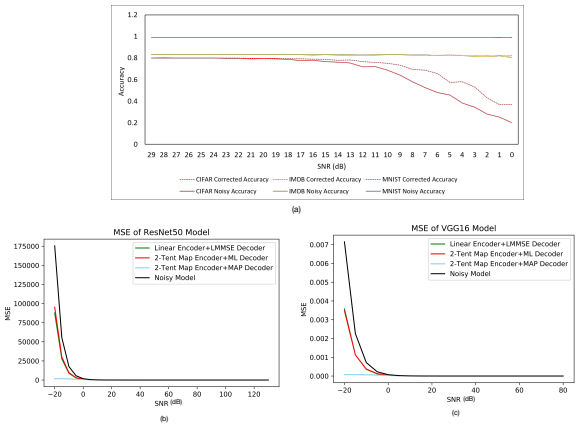

Fig. 1. a) Accuracy vs SNR (high to low) for noisy weights (solid lines) and weights corrected by linear analog codes (dotted lines) for CIFAR-10, IMDB and MNIST datasets. b), c) MSE distortion vs SNR (low to high) for ResNet-50 and VGG-16 respectively.

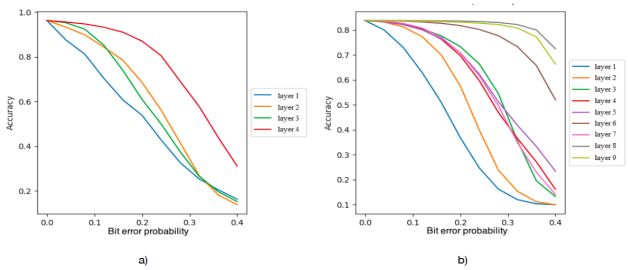

Fig. 2. BNN Accuracy vs bit error probability for a) MNIST, b) CIFAR-10

III. Analog Codes for Noisy DNN

Since typical DNN weights are analog numbers, in this section, we study the use of linear and non-linear analog codes, which convert a vector of analog symbols to an analog codeword for error correction.

A. Linear Analog Codes

In this section, we design systematic linear analog codes, which is an extension of the codes in non systematic form in [4]. A linear analog code converts a real vector $\mathbf{u} \in \mathcal{R}^{K\times 1}$ to codeword vector $\mathbf{v} \in \mathcal{R}^{N\times 1}$, by $\mathbf{v} = \mathbf{G}^T\mathbf{u}$. Each coordinate u_i has mean 0 and variance D_u. $\mathbf{G}$ is called a $K \times N$ generator matrix. The codeword $\mathbf{v}$ passes through a channel, characterized by the noise vector $\mathbf{n} \in \mathcal{R}^{N\times 1}$, where each

i.i.d. coordinate $n_i \sim N(0, \sigma_n^2)$. The received signal is given by $\mathbf{r} = \mathbf{v} + \mathbf{n}$. The decoder takes the noisy message $\mathbf{r}$ to produce $\hat{\mathbf{u}}$, which is an estimate of $\mathbf{u}$ by $\hat{\mathbf{u}} = \mathbf{A}\mathbf{r}$, where $\mathbf{A}$ is the decoding matrix. $\mathbf{G}$ is designed to ensure that the energy per information symbol of the signal $\mathbf{v}$ is E_b. Also, for an maximum likelihood (ML) decoder $\mathbf{A}$, the optimal $\mathbf{G}$ to reduce the mean squared error is given by $\mathbf{G}\mathbf{G}^T = diag\{\frac{E_b}{D_u}, \frac{E_b}{D_u}, ..., \frac{E_b}{D_u}\}$. $\mathbf{G}$ can be formed by deleting $(N-K)$ rows of an *orthogonal matrix*, and scaling it appropriately by a factor $\sqrt{\frac{E_b}{D_u}}$.

In our case, let us assume we have a vector $\mathbf{w}$ of K weights in a DNN, where weights w_i come from a distribution with mean μ and variance D_u. We first convert the vector $\mathbf{w}$ to the message vector $\mathbf{u}$ where $u_i = w_i - \mu$. We would also want the analog codes to be *systematic*, so that the actual weights can be stored in hardware. In other words, we want the $\mathbf{G}$ to be of the form $(\mathbf{I}_K \mid \mathbf{P})$ where $\mathbf{P}$ is a $K \times N-K$ matrix and $\mathbf{I}_K$ is a $K \times K$ identity matrix. The matrix $\mathbf{P}$ can be constructed as follows. $\mathbf{P}$ can be constructed by by deleting $(N-2K)$ rows of an $N-K \times N-K$ *orthogonal matrix*, and scaling it appropriately by a factor $\sqrt{\frac{E_b}{D_u} - 1}$, for $K \leq N-K$ and $E_b \geq D_u$. The following theorem can be proved :

Theorem 1. $\mathbf{G} = (\mathbf{I}_K \mid \mathbf{P})$ *is a systematic matrix which satisfies the condition* $\mathbf{G}\mathbf{G}^T = diag\{\frac{E_b}{D_u}, \frac{E_b}{D_u}, ..., \frac{E_b}{D_u}\}$

B. Non Linear Analog Codes

Now, we discuss error correction of weights of a deep neural network using systematic *non-linear* analog codes based on two dimensional tent maps. The noise in this case corresponds to a Gaussian distribution $n \sim N(0, \sigma_n^2)$. The non-linear code is an extension of previous work [5], [1], which uses a two dimensional tent map encoder. In this work, we introduce a MAP decoder as compared to the ML decoder, because the neural network weights are far away from a uniform distribution.

This encoder takes an input message sequence $\mathbf{u} = (u_0, u_1)$ of length $K = 2$, where each message has a range $[-1, 1]$. The 2-dimensional tent map function can be written as $(x_i, y_i) = F(x_{i-1}, y_{i-1}) = (1-|x_{i-1}+y_{i-1}|, 1-|x_{i-1}-y_{i-1}|)$ where $x_0 = u_0$, $y_0 = u_1$, x_i, y_i are elements of codeword. We set our code rate here to be 1/2. Then the codeword for $\mathbf{u} = (u_0, u_1)$ is the vector (x_0, y_0, x_1, y_1). The codeword (x_0, y_0, x_1, y_1) passes through the AWGN channel to give the received codeword $(R_{x_0}, R_{y_0}, R_{x_1}, R_{y_1})$. In this case, the two-dimensional tent code is decoded by the maximum a posteriori probability (MAP) rule. The estimates $(\hat{u}_0, \hat{u}_1)$ are given by

$$\arg\max \; P(R_{x_0}, R_{y_0}, R_{x_1}, R_{y_1} | x_0, y_0, x_1, y_1) P(x_0, y_0)$$

We assume all weights of the deep neural network are subject to Gaussian distribution $u \sim N(\mu, \sigma_w^2)$, and derive the following formula for $\hat{u}_0$ and $\hat{u}_1$:

$$\hat{u}_0 = \frac{R_{x_0} + \mu\frac{\sigma_n^2}{\sigma_w^2} + s_1(1 - R_{x_1}) + s_2(1 - R_{y_1})}{3 + \frac{\sigma_n^2}{\sigma_w^2}}$$

$$\hat{u}_1 = \frac{R_{y_0} + \mu\frac{\sigma_n^2}{\sigma_w^2} + s_1(1 - R_{x_1}) + s_2(R_{y_1} - 1)}{3 + \frac{\sigma_n^2}{\sigma_w^2}}$$

Here, s_1 and s_2 are signs of the received information. s_1 is 1, when $R_{x_0} + R_{y_0} \geq 0$ and -1 otherwise. s_2 is 1, when $R_{x_0} - R_{y_0} \geq 0$ and -1 otherwise.

IV. Results and Extended work

A. Performance of Analog Codes

We now present some results of the accuracy of different networks, when the weights are protected by linear analog codes. The results are shown in Fig. 1 a) in dotted lines. The results show an improvement in accuracy in all DNN models. However, significant improvement is seen for the accuracy of the CIFAR-10 dataset, whose performance was affected the most by noise. We also explore the mean squared error (MSE) distortion for symbols of the non-linear ECC, and compare their performance with linear analog codes. We consider the LMMSE decoder as shown in [4] for the linear code and both the MAP and ML decoders for the non-linear code. We consider the ResNet50 model, which is a 50 layer residual network with 25.6 million weights and VGG-16 which is a 16 layer CNN with 138 million weights. Fig. 1 b) and c) show that the MAP decoder shows the minimum distortion.

B. Extended Work

Our research so far studies analog weights protected by an analog ECC. An alternative is to use binarized neural networks (BNNs), where all weights are in $\{-1, 1\}$ so that binary ECC becomes a natural choice. BNNs have two advantages : smaller network sizes and ease of hardware implementation. To use binary ECC to protect the weights of BNNs, it is important to know how noise in different layers can affect the performance. We show our initial results in Fig. 2 a) and b), which are for MNIST and CIFAR-10 datasets respectively. The x-axis represents the bit error probability of the weights in each layer (only a single layer is made noisy each time), and the y-axis represents the accuracy of the DNNs. It can be seen that the noise in initial layers deteriorates the performance of DNN more substantially than subsequent layers. This observation is helpful for optimizing the code rate of ECCs for different layers. More details can be shown in the full presentation of the paper.

ACKNOWLEDGMENT: This work was supported in part by NSF Grant CCF-1718886.

References

[1] Brian Chen and Gregory W. Wornell, "Analog error-correcting codes based on chaotic dynamical systems", in *IEEE Transactions on Communications*, vol. 46, no. 6, pp. 881-890, 1998.

[2] F. Chollet, *Deep Learning with Python*, Manning Publications Co. 2017.

[3] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio, "Binarized Neural Networks", in *Advances in Neural Information Processing Systems*, pp. 4107-4115, 2016.

[4] Kai Xie, Jing Li and Yang Liu, "Analysis of Performance of Linear Analog Codes", in *arXiv preprint arXiv:1511.05509*, 2015.

[5] Hu Xing, Lin-Hua Ma, Le Ru,Song Zhang, "Analog Error Correction Codes based on Chaotic System: the 2-dimensional Tent Codes", in *Proc. International Conference on Wireless Communication and Sensor Network*, Wuhan, China, March 2012.