CodNN – Robust Neural Networks From Coded Classification

Netanel Raviv*, **Siddharth Jain**†, **Pulakesh Upadhyaya**‡, **Jehoshua Bruck**†, and **Anxiao (Andrew) Jiang**‡

*Department of Computer Science and Engineering, Washington University in St. Louis, St. Louis 63130, MO, USA

†Department of Electrical Engineering, California Institute of Technology, Pasadena 91125, CA, USA

‡Department of Computer Science and Engineering, Texas A&M University, College Station 77843, TX, USA

***Abstract*—Deep Neural Networks (DNNs) are a revolutionary force in the ongoing information revolution, and yet their intrinsic properties remain a mystery. In particular, it is widely known that DNNs are highly sensitive to noise, whether adversarial or random. This poses a fundamental challenge for hardware implementations of DNNs, and for their deployment in critical applications such as autonomous driving.**

In this paper we construct robust DNNs via error correcting codes. By our approach, either the data or internal layers of the DNN are coded with error correcting codes, and successful computation under noise is guaranteed. Since DNNs can be seen as a layered concatenation of classification tasks, our research begins with the core task of classifying noisy coded inputs, and progresses towards robust DNNs.

We focus on binary data and linear codes. Our main result is that the prevalent *parity code* can guarantee robustness for a large family of DNNs, which includes the recently popularized *binarized neural networks*. Further, we show that the coded classification problem has a deep connection to Fourier analysis of Boolean functions.

In contrast to existing solutions in the literature, our results do not rely on altering the training process of the DNN, and provide mathematically rigorous guarantees rather than experimental evidence.

I. Introduction

Deep Neural Networks (DNNs) have become a dominating force in Artificial Intelligence (AI), bringing revolutions in science and technology. A massive amount of academic and industrial research is being devoted to implementing DNNs in hardware [4]. Hardware-implemented DNNs are appearing in phones, sensors, healthcare devices, and more, which will revolutionize every sector of society [10], and make AI systems increasingly energy-efficient and ubiquitous.

In parallel, DNNs are known to be highly susceptible to adversarial interventions. In a recent line of works which followed [16], it was shown that by adding a small (and often indistinguishable to humans) amount of noise to the *inputs* of a DNN, one can cause it to reach nonsensical conclusions. More recently, it was shown [15] that in some DNN architectures one can attain similar effects by changing as little as one or two entries of the input. This reveals an orthogonal concern of a similar nature from the adversarial machine learning perspective: performance degradation due to malicious attacks.

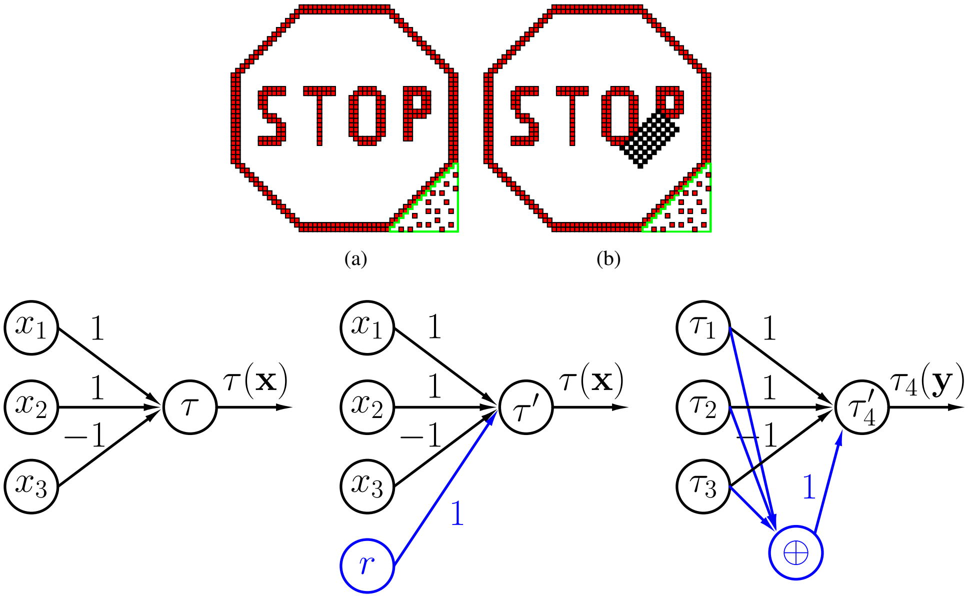
There exists a rich body of research which studies how to make DNNs robust to noise. This includes noise that is injected into the neurons/synapses, or into the inputs. Even though computation under noise has been studied since the 1950's [12], solutions have been almost exclusively heuristic.

To combat adversarial attacks to the inputs, much focus was given on adjusting the *training* process to produce more robust DNNs, e.g., by adjusting the regularization expression [3], or the loss function [7]. These approaches usually involve intractable optimization problems, and succeed insofar as the underlying optimization succeeds.

Combating noise in neurons/synapses has also enjoyed a recent surge of interest [17], which builds upon the previous wave of interest in DNNs in the early 1990's [14]. Most of this line of research focuses on replication methods (called augmentation), retraining, and providing statistical frameworks for testing fault tolerance of DNNs (e.g., training a DNN to remember a coded version of all possible outputs [6]). It is also worth mentioning that to a certain degree, DNNs tend to present some natural fault-tolerance without any intervention. This phenomenon is conjectured to be connected to *over-provisioning* [1], i.e., the fact that in most cases one uses more neurons than necessary, but rigorous guarantees remain elusive.

In this paper we propose a fundamentally new approach for handling noise in DNNs. In this approach, the inputs to neurons are coded by using an error correcting code, and the neurons are modified accordingly, so that correct output is guaranteed as long as the noise level is below a certain threshold. Clearly, the encoding function must be simpler than conducting the computation itself, and should apply universally to a large family of neurons; we also aim for an efficient end-to-end design that does not require decoding.

Our approach is depicted in Fig. 1 for the famous case-study of stop-sign classification [2]. In this case-study, an autonomous vehicle uses a DNN to classify a stop sign as such, exposing the passenger to the perils of misclassification due to noise (e.g., a sticker). In our approach we envision addition of redundancy *to the actual physical object* (Fig. 1a), which aids the DNN in classification under noise (Fig. 1b). To facilitate this vision, the neurons inside the network must be revised accordingly, e.g., by adding weighted synapses (Fig. 1c and Fig. 1d). Alternatively, the inputs to some neurons might come from other neurons inside the network, rather than from the physical world; this case better encapsulates hardware failures, and redundancy is computed by additional components inside the DNN (Fig. 1e).



(a) (b)

(c) (d) (e)

Fig. 1: (a) a stop sign with added redundancy; (b) the added redundancy guarantees correct classification under noise, e.g., a sticker; (c) an uncoded neuron; (d) a coded neuron with one extra bit of redundancy, guaranteed to compute $\tau(\mathbf{x})$ correctly, even if *any* synapse is erased (see Example 2); (e) added redundancy *inside* the DNN–the coded neuron τ_4' computes $\tau_4(\mathbf{y})$ even if any of its incoming synapses is erased, where $\mathbf{y}$ is the output of the neurons τ_1, τ_2, and τ_3 in the previous layer.

Preliminaries are discussed in Section II, where a tight connection to the ℓ_1-metric is revealed. Simple but inefficient solutions, that rely on replication or Fourier analysis, are discussed in Section III. Finally, the main result of this paper is given in Section IV, where it is shown that the well-known parity code can guarantee successful classification under noise, and applications to a large family of DNNs are discussed.

II. Preliminaries

A DNN is a layered and directed acyclic graph, in which the first layer is called *the input layer*. Each edge (or *synapse*) corresponds to a real number called *weight*, and nodes in intermediate layers (also called *hidden layers*) are called *neurons*. Each neuron acts as computation unit, that applies some *activation function* on its inputs, weighted by the respective synapses. The result of the computation in the last layer are the outputs of the DNN.

Traditionally, the activation function is $\text{sign}(\mathbf{x}\mathbf{w}^\top - \theta)$, where $\mathbf{x}$ is a vector of inputs, $\mathbf{w}$ is a vector of weights, θ is a constant called *bias*, and

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}.$$

However, contemporary DNNs often employ continuous approximations of $\text{sign}(\cdot)$, known as *sigmoid* functions (such as $\text{logistic}(x) = \frac{1}{1+\exp(-x)}$ or $\tanh(x)$) or piecewise linear alternatives (such as $\text{ReLU}(x) = \max\{0, x\}$) in order to enable analytic learning algorithms (such as *backpropagation*). In our work, in order to establish rigorous and discrete guarantees, we focus on $\text{sign}(\cdot)$. Further, we focus on *binary* ± 1 inputs, which correspond to the binary field $\mathbb{F}_2$ by identifying 0 as 1 and 1 as -1, and exclusive or as product.

At the computational level, faults in DNN hardware appear as bit-errors, bit-erasures or analog noise, which can be permanent or transient. In this work we focus on bit-errors and bit-erasures, that are formally defined shortly. We denote scalars by lowercase letters $a, b, \ldots$, vectors by lowercase boldface letters $\mathbf{a}, \mathbf{b}, \ldots$, and use the same letter to refer to a vector and its entries (e.g., $\mathbf{a} = (a_1, a_2, \ldots)$). We use $d_i(\cdot,\cdot)$, $\|\cdot\|_i$, and $\mathcal{B}_i(\mathbf{z}, r)$ to denote the ℓ_i-distance, ℓ_i-norm, and ℓ_i-ball cantered at $\mathbf{z}$ with radius r, respectively, for $i \in \{0, 1, 2, \ldots, \infty\}$. We use $d_H(\cdot,\cdot)$ to denote Hamming distance, and let $\mathbb{1}_m$ be a vector of m ones.

A. Framework and problem definition

For a given neuron $\tau : \mathbb{F}_2^n \to \mathbb{F}_2$, where $\tau(\mathbf{x}) = \text{sign}(\mathbf{x} \cdot \mathbf{w}^\top - \theta)$ for some $\mathbf{w} \in \mathbb{R}^n$ and $\theta \in \mathbb{R}$, a triple $(E, \mathbf{v}, \mu)$ is called a *solution*, where $E : \mathbb{F}_2^n \to \mathbb{F}_2^m$, $\mathbf{v} \in \mathbb{R}^m$, and $\mu \in \mathbb{R}$. The respective *coded neuron* is $\tau'(E(\mathbf{x})) = \text{sign}(E(\mathbf{x})\mathbf{v}^\top - \mu)$. For nonnegative integers t and s, the coded neuron τ' is robust against t erasures and s errors ((t,s)-robust, for short) if for every disjoint t-subset $\mathcal{T} \subseteq [m]$, and s-subset $\mathcal{S} \subseteq [m]$, we have that

$$\text{sign}(\mathbf{x} \cdot \mathbf{w}^\top - \theta) = \text{sign}\left(\sum_{j \in [m] \setminus (\mathcal{T} \cup \mathcal{S})} E(\mathbf{x})_j v_j - \sum_{j \in \mathcal{S}} E(\mathbf{x})_j v_j - \mu \right) \quad (1)$$

for every $\mathbf{x} \in \mathbb{F}_2^n$.

Namely, when computing over data encoded by E, correct output for *every* $\mathbf{x} \in \mathbb{F}_2^n$ is guaranteed, even if at most t of the inputs to τ' are omitted (erasures), and at most s are negated (errors). Further, for a nonnegative integer r we say that τ' is r-robust if it is (t, s)-robust for every nonnegative t and s such that $t + 2s \leq r$.

For $\mathbf{v} \in \mathbb{R}^m$ and $\mu \in \mathbb{R}$, let $\mathcal{H}(\mathbf{v}, \mu) = \{\mathbf{y} \in \mathbb{R}^m | \mathbf{y}\mathbf{v}^\top = \mu\}$. For a given solution $(E, \mathbf{v}, \mu)$, we say that the *minimum distance* of the respective coded neuron is

$$\begin{aligned} d = d(E, \mathbf{v}, \mu) &= d_1(E(\mathbb{F}_2^n), \mathcal{H}(\mathbf{v}, \mu)) \\ &= \min_{\mathbf{x} \in \mathbb{F}_2^n} d_1(E(\mathbf{x}), \mathcal{H}(\mathbf{v}, \mu)). \end{aligned}$$

The choice of the ℓ_1-metric will be made clear in the sequel. The figure of merit by which we measure a given solution is its *relative distance* d/m.

Example 1. *For a given neuron τ, and an integer m, let*

$$E(\mathbf{x}) = \begin{cases} \mathbb{1}_m & \text{if } \tau(\mathbf{x}) = 1 \\ -\mathbb{1}_m & \text{if } \tau(\mathbf{x}) = -1 \end{cases}.$$

It is readily verified that the solution $(E, \mathbb{1}_m, 0)$ is $(m-1)$-robust.

Since layers in DNNs normally contain multiple neurons, the solution in Example 1 is useless for constructing robust DNNs. Instead, one would like to have *joint coding* E for a large family of neurons.

Problem Definition: For a given set of neurons $\{\tau_i(\mathbf{x}) = \text{sign}(\mathbf{x}\mathbf{w}_i^\top - \theta_i)\}_{i=1}^{\ell}$ find a joint coding function E and $\{\mathbf{v}_i, \mu_i\}_{i=1}^{\ell}$ which maximize $d_{\min}/m$, where $d_{\min} = \min_{i \in [\ell]} d(E, \mathbf{v}_i, \mu_i)$.

Furthermore, we restrict our attention to functions E which encode binary linear codes, due to their prevalence in hardware and ease of analysis. Since we use the $\{\pm 1\}$-representation of $\mathbb{F}_2$, every entry of $E(\mathbf{x})$ is a multilinear monomial in the entries of $\mathbf{x}$.

Example 2. *Fig. 1c depicts the uncoded neuron $\tau(\mathbf{x}) = \text{sign}(x_1 + x_2 - x_3)$, and Fig. 1d depicts its coded version $\tau'(\mathbf{x}) = \text{sign}(x_1 + x_2 - x_3 + r)$, where $r = x_1 x_2 x_3$. Table I shows two examples of robustness to any 1-erasure.*

(x_1, x_2, x_3, r)	Erasure	$\tau'(\text{noisy } E(\mathbf{x})) = \tau(\mathbf{x})$
$(1, -1, 1, -1)$	x_1	$\text{sign}(0 - 1 - 1 - 1) = -1$
	x_2	$\text{sign}(1 - 0 - 1 - 1) = -1$
	x_3	$\text{sign}(1 - 1 - 0 - 1) = -1$
	r	$\text{sign}(1 - 1 - 1 - 0) = -1$
$(-1, 1, -1, 1)$	x_1	$\text{sign}(-0 + 1 + 1 + 1) = 1$
	x_2	$\text{sign}(-1 + 0 + 1 + 1) = 1$
	x_3	$\text{sign}(-1 + 1 + 0 + 1) = 1$
	r	$\text{sign}(-1 + 1 + 1 + 0) = 1$

TABLE I: Two examples of correct computation of $\tau(\cdot)$ (Figure 1c) by $\tau'(E(\cdot))$ (Figure 1d). This holds for the remaining six inputs as well.

B. Robustness and the ℓ_1-metric

In this section we justify the above definitions, and in particular, the use of the ℓ_1-metric to obtain robustness. First, notice that errors and erasures while evaluating τ' can be seen as changes in $E(\mathbf{x})$. For example, let $\mathbf{v} = (v_1, v_2, v_3)$ and $E(\mathbf{x}) = (y_1, y_2, y_3)$, and then an erasure at entry 1 is equivalent to evaluating τ' at the point $(0, y_2, y_3)$. Similarly, an error in entry 2 is equivalent to evaluating τ' at $(y_1, -y_2, y_3)$.

As such, both errors and erasure can be seen as evaluation of the same coded neuron τ' on a data point which is shifted along axis-parallel lines. Therefore, the encoded points must be far away from $\mathcal{H}(\mathbf{v}, \mu)$ in ℓ_1-distance. More precisely, since error and erasures do not cause any point to shift outside the closed hypercube $[-1, 1]^m$, it is only necessary to have large ℓ_1-distance from $\mathcal{H}' = \mathcal{H}'(\mathbf{v}, \mu) = \mathcal{H}(\mathbf{v}, \mu) \cap [-1, 1]^m$.

First, we provide the formula for the ℓ_1-distance of a point from a hyperplane.

Lemma 1. *[8, Sec. 5] For every $\mathbf{z} \in \mathbb{R}^m$ we have that $d_1(\mathbf{z}, \mathcal{H}) = \frac{|\mathbf{z} \cdot \mathbf{v}^\top - \mu|}{\|\mathbf{v}\|_\infty}$.*

Second, we provide a necessary and sufficient condition for the robustness of a coded neuron $\tau'(E(\mathbf{x})) = \text{sign}(E(\mathbf{x})\mathbf{v}^\top - \mu)$. We denote the positive points of τ by $\mathcal{F}^+$, and the negative points by $\mathcal{F}^-$.

Theorem 1. *For a positive integer r and a neuron $\tau(\mathbf{x}) = \text{sign}(\mathbf{x}\mathbf{w}^\top - \theta)$, a coded neuron $\tau'(\mathbf{x}) = \text{sign}(E(\mathbf{x})\mathbf{v}^\top - \mu)$ is r-robust if and only if*

$$\begin{aligned} \text{sign}(\mathbf{x}\mathbf{w}^\top - \theta) = \text{sign}(E(\mathbf{x})\mathbf{v}^\top - \mu) \textit{ for every } \mathbf{x} \in \mathbb{F}_2^n, \\ r \leq d_1(E(\mathcal{F}^+), \mathcal{H}'), \textit{ and} \\ r < d_1(E(\mathcal{F}^-), \mathcal{H}'). \end{aligned} \tag{2}$$

Proof. Assume that the conditions in (2) hold. To show that τ' is r-robust we must show that (1) holds for every $\mathbf{x} \in \mathbb{F}_2^n$ and every mutually disjoint $\mathcal{S}$ and $\mathcal{T}$ such that $|\mathcal{T}| + 2|\mathcal{S}| \leq r$. Assuming for contradiction that τ' is not r-robust, there exists some $\mathbf{x} \in \mathbb{F}_2^n$ and corresponding sets $\mathcal{S}$ and $\mathcal{T}$ such that $E(\mathbf{x})$ is misclassified under erasures in $\mathcal{T}$ and errors in $\mathcal{S}$. Since any set of errors or erasures keeps $E(\mathbf{x})$ inside $[-1, 1]^m$, it follows that this misclassification of $E(\mathbf{x})$ corresponds to moving it along an axis-parallel path P of length $|P| = |\mathcal{T}| + 2|\mathcal{S}|$, which crosses $\mathcal{H}'$.

If $\mathbf{x} \in \mathcal{F}^+$, then to attain misclassification we must have $|P| > d_1(E(\mathbf{x}), \mathcal{H}')$. However, this implies that $r \geq |\mathcal{T}| + 2|\mathcal{S}| = |P| > d_1(E(\mathbf{x}), \mathcal{H}') \geq r$, a contradiction. If $\mathbf{x} \in \mathcal{F}^-$, then to attain misclassification we must have $|P| \geq d_1(E(\mathbf{x}), \mathcal{H}')$. However, this implies that $r \geq |\mathcal{T}| + 2|\mathcal{S}| = |P| \geq d_1(E(\mathbf{x}), \mathcal{H}') > r$, another contradiction.

Conversely, assume that τ' is r-robust. Since $r \geq 0$, it follows that τ' is in particular $(0, 0)$-robust, and hence according to (1) if follows that $\text{sign}(\mathbf{x}\mathbf{w}^\top - \theta) = \text{sign}(E(\mathbf{x})\mathbf{v}^\top - \mu)$ for every $\mathbf{x} \in \mathbb{F}_2^n$. Assume for contradiction that $r > d_1(E(\mathcal{F}^+), \mathcal{H}')$, which implies that there exists $\mathbf{x} \in \mathcal{F}^+$ such that $r > d_1(E(\mathbf{x}), \mathcal{H}')$, and let $\mathcal{B}_{\mathbf{x}} \triangleq \mathcal{B}_1(E(\mathbf{x}), r) \cap [-1, 1]^m$. This readily implies that some vertex $\mathbf{y}$ of $\mathcal{B}_{\mathbf{x}}$ lies on the opposite side of $\mathcal{H}$. It can be proved (full proof will be given in future versions of this paper) that $\mathbf{y}$ is an integer point, and that all such points correspond to erasures in some set $\mathcal{T}$ and errors in some set $\mathcal{S}$ such that $|\mathcal{T}|+2|\mathcal{S}|\leq r$. Therefore, the existence of $\mathbf{y}$ contradicts the r-robustness of τ'. The proof that $r<d_1(E(\mathcal{F}^-),\mathcal{H}')$ is similar. □

We conclude this section by showing that redundancy is *necessary* for any nontrivial robustness. Since any non-constant neuron must have a positive point $\mathbf{x}$ and a negative point $\mathbf{y}$ such that $d_H(\mathbf{x},\mathbf{y})=1$, and since any hyperplane must cross the convex hull of $\mathbf{x}$ and $\mathbf{y}$, the following is immediate.

Lemma 2. *Unless $\tau(\mathbf{x})=\operatorname{sign}(\mathbf{x}\mathbf{w}^\top-\theta)$ is constant, the solution $(E,\mathbf{v},\mu)=(\operatorname{Id},\mathbf{w},\theta)$ is 0-robust.*

Further, by denoting $\delta=d_1(\mathbb{F}_2^n,\mathcal{H}(\mathbf{w},\theta))$, we have that the relative distance of the solution $(\operatorname{Id},\mathbf{w},\theta)$ is δ/n. However, computing δ for a given neuron τ is in general NP-complete, by a reduction to PARTITION [9].

III. A Few Elementary Solutions

A. Robustness by replication

For a vector $\mathbf{v}$, let $\mathbf{v}_{(\ell)}$ be the result of concatenating $\mathbf{v}$ with itself ℓ times, and for $E:\mathbb{F}_2^n\to\mathbb{F}_2^m$ let $E_{(\ell)}:\mathbb{F}_2^n\to\mathbb{F}_2^{\ell m}$ be the function $E_{(\ell)}(\mathbf{x})=E(\mathbf{x})_{(\ell)}$.

Lemma 3. *Let $(E,\mathbf{v},\mu)$ be a solution with minimum distance d. Then, for every positive integer ℓ, the solution $(E_{(\ell)},\mathbf{v}_{(\ell)},\ell\mu)$ has distance ℓd and identical relative distance d/m.*

Proof. According to Lemma 1, and since $\|\mathbf{v}_{(\ell)}\|_\infty=\|\mathbf{v}\|_\infty$, we have that

$$d_1\left(E_{(\ell)}(\mathbb{F}_2^n),\mathcal{H}(\mathbf{v}_{(\ell)},\ell\mu)\right)=\frac{\min_{\mathbf{x}\in\mathbb{F}_2^n}|E_{(\ell)}(\mathbf{x})\mathbf{v}_{(\ell)}^\top-\ell\mu|}{\|\mathbf{v}\|_\infty},$$

and since $E_{(\ell)}(\mathbf{x})\mathbf{v}_{(\ell)}^\top=\ell\cdot E(\mathbf{x})\mathbf{v}^\top$, it follows that this equals

$$\ell\cdot\frac{\min_{\mathbf{x}\in\mathbb{F}_2^n}|E(\mathbf{x})\mathbf{v}^\top-\mu|}{\|\mathbf{v}\|_\infty}=\ell d,$$

and thus the relative distance is $\ell d/\ell m=d/m$. □

Therefore, by applying the ℓ-replication code $E(\mathbf{x})=\mathbf{x}_{(\ell)}$, one can obtain robustness but not increase the relative distance. Moreover, since computing the aforementioned δ is NP-hard, explicit robustness guarantees are hard to come by.

B. Robustness from the Fourier spectrum

Recall that every function $f:\mathbb{F}_2^n\to\mathbb{R}$ (and in particular, every neuron τ) can be written as a linear combination $f(\mathbf{x})=\sum_{\mathcal{S}\subseteq[n]}\hat{f}(\mathcal{S})\chi_{\mathcal{S}}(\mathbf{x})$, where $\chi_{\mathcal{S}}(\mathbf{x})\triangleq\prod_{s\in\mathcal{S}}x_s$ and $\hat{f}(\mathcal{S})=\mathbb{E}_{\mathbf{x}}\chi_{\mathcal{S}}(\mathbf{x})f(\mathbf{x})$ for every $\mathcal{S}\subseteq[n]$. The vector $\hat{\mathbf{f}}\triangleq(\hat{f}(\mathcal{S}))_{\mathcal{S}\subseteq[n]}$ is called the *Fourier spectrum* of f, and if f is Boolean then $\|\hat{\mathbf{f}}\|_2=1$. We denote by $\hat{\mathbf{f}}_\varnothing$ the vector $\hat{\mathbf{f}}$ with its $\varnothing$-entry omitted, i.e., $\hat{\mathbf{f}}_\varnothing\triangleq(\hat{f}(\mathcal{S}))_{\mathcal{S}\subseteq[n],\mathcal{S}\neq\varnothing}$. We refer to the following solution as the *Fourier solution*.

Lemma 4. *For a neuron τ, the coded neuron $\tau'(E(\mathbf{x}))\triangleq\operatorname{sign}(\sum_{\mathcal{S}\subseteq[n]}\hat{\tau}(\mathcal{S})\chi_{\mathcal{S}}(\mathbf{x}))$ has minimum distance $\|\hat{\boldsymbol{\tau}}_\varnothing\|_\infty^{-1}$.*

Proof. Notice that τ' is defined by the encoding function $E:\mathbb{F}_2^n\to\mathbb{F}_2^{2^n-1}$ such that $E(\mathbf{x})=(\chi_{\mathcal{S}}(\mathbf{x}))_{\mathcal{S}\subseteq[n],\mathcal{S}\neq\varnothing}$, known as the *punctured Hadamard* encoder. In addition, the respective halfspace is $\mathcal{H}=\mathcal{H}(\hat{\boldsymbol{\tau}}_\varnothing,-\hat{\tau}(\varnothing))\triangleq\{\mathbf{y}\in\mathbb{F}_2^{2^n-1}|\sum_{\mathcal{S}\neq\varnothing}y_{\mathcal{S}}\hat{\tau}(\mathcal{S})+\hat{\tau}(\varnothing)=0\}$, where the coordinates of $\mathbb{R}^{2^n-1}$ are indexed by all nonempty subsets of $[n]$. To find the minimum distance of the Fourier solution, we compute

$$\begin{aligned}d_1(E(\mathbb{F}_2^n),\mathcal{H})&=\frac{\min_{\mathbf{x}\in\mathbb{F}_2^n}|\hat{\boldsymbol{\tau}}_\varnothing\cdot E(\mathbf{x})+\hat{\tau}(\varnothing)|}{\|\hat{\boldsymbol{\tau}}_\varnothing\|_\infty}\\&=\frac{\min_{\mathbf{x}\in\mathbb{F}_2^n}|\sum_{\mathcal{S}\subseteq[n]}\hat{\tau}(\mathcal{S})\chi_{\mathcal{S}}(\mathbf{x})|}{\|\hat{\boldsymbol{\tau}}_\varnothing\|_\infty}=\|\hat{\boldsymbol{\tau}}_\varnothing\|_\infty^{-1},\end{aligned}$$

where the last equality follows since $\tau(\mathbf{x})=\sum_{\mathcal{S}\subseteq[n]}\hat{\tau}(\mathcal{S})\chi_{\mathcal{S}}(\mathbf{x})\in\{\pm1\}$. □

The relative distance of this solution is $\|\hat{\boldsymbol{\tau}}_\varnothing\|_\infty^{-1}/(2^n-1)$, and notice that unlike replication (Subsection III-A), it does not depend on the particular way in which τ is given. However, this solution involves exponentially many redundant bits, and is therefore impractical.

IV. Robustness from the Parity Code

The discussion in this section applies to DNNs that employ *binary neurons*, i.e., neurons in which $\mathbf{w}\in\mathbb{F}_2^n$. This family of DNNs includes, as a strict subset, the recently popularized *binarized neural networks* [5]. Later on, we generalize the solution to all neurons, and show its superiority over replication in cases where $\|\mathbf{w}\|_1$ is bounded. Specifically, we show that the parity code attains relative distance $2/(n+1)$, which outperforms replication.

We denote

$$\mathcal{U}\triangleq\{\tau:\mathbb{F}_2^n\to\mathbb{F}_2|\tau(\mathbf{x})=\operatorname{sign}(\mathbf{x}\mathbf{w}^\top-\theta)\text{ and }\mathbf{w}\in\mathbb{F}_2^n\},$$

and since $\mathbf{x}\mathbf{w}^\top\in\{-n,-n+2,\ldots,n\}$ for every $\mathbf{x}$ and $\mathbf{w}$ in $\mathbb{F}_2^n$, it follows that for every $\tau\in\mathcal{U}$ one can round the respective θ to the nearest value[1] in $\{-n-1,-n+1,\ldots,n+1\}$ without altering τ. Hence, we assume without loss of generality that all given θ's are in $\{-n-1,-n+1,\ldots,n+1\}$. With this choice of θ, any function $f\in\mathcal{U}$ has $\delta=1$, since

$$\begin{aligned}\delta=d_1(\mathbb{F}_2^n,\mathcal{H}(\mathbf{w},\theta))&=\frac{\min_{\mathbf{x}\in\mathbb{F}_2^n}|\mathbf{x}\mathbf{w}^\top-\theta|}{\|\mathbf{w}\|_\infty}\\&=\min_{\mathbf{x}\in\mathbb{F}_2^n}|\mathbf{x}\mathbf{w}^\top-\theta|=1\end{aligned}$$

by Lemma 1. Thus, replication achieves relative distance $1/n$ (e.g., 2-replication achieves 1-robustness with $m=2n$).

We also employ the following notations from Boolean algebra. For $\mathbf{x},\mathbf{w}\in\mathbb{R}^n$ let $\mathbf{x}\oplus\mathbf{w}$ denote their pointwise product, which amounts to Boolean sum if both $\mathbf{x}$ and $\mathbf{w}$ are in $\mathbb{F}_2^n$. Further, for $\mathbf{x}\in\mathbb{F}_2^n$ we let $w_H(\mathbf{x})$ be the number

[1] More precisely, if $\theta\in(-n+2t,-n+2t+2]$ for an integer $0\leq t\leq n$, then θ is replaced by $-n+2t+1$. If $\theta\leq-n$ it is replaced by $-n-1$, and if $\theta>n$ it is replaced by $n+1$.

of (-1)-entries in $\mathbf{x}$, known as *Hamming weight*. The next two lemmas, whose proofs will appear in future version of this paper, demonstrate that functions in $\mathcal{U}$ depend only on $w_H(\mathbf{x} \oplus \mathbf{w})$.

Lemma 5. *For every* $\mathbf{x}$ *and* $\mathbf{w}$ *in* $\mathbb{F}_2^n$ *we have* $\mathbf{x}\mathbf{w}^\top = n - 2w_H(\mathbf{x} \oplus \mathbf{w})$.

Lemma 6. *For every* $\tau \in \mathcal{U}$ *and every* $\mathbf{x} \in \mathbb{F}_2^n$ *we have*

$$\tau(\mathbf{x}) = \begin{cases} 1 & w_H(\mathbf{x} \oplus \mathbf{w}) \leq \frac{n-\theta-1}{2} \\ -1 & w_H(\mathbf{x} \oplus \mathbf{w}) \geq \frac{n-\theta+1}{2} \end{cases}.$$

In this section we let $m = n+1$ and define $E : \mathbb{F}_2^n \to \mathbb{F}_2^{n+1}$ as the *parity* encoder $E(\mathbf{x}) = (x_1, \ldots, x_n, \chi_{[n]}(\mathbf{x}))$. Then, we let $\theta' \triangleq \frac{n-\theta-1}{2}$, and define the *parity* solution $(E, \mathbf{v}, \mu)$, where

$$\mathbf{v} = (w_1, \ldots, w_n, (-1)^{\theta'} \chi_{[n]}(\mathbf{w})), \text{ and}$$
$$\mu = \theta.$$

Lemma 7. *The relative distance of the parity solution is* $2/(n+1)$.

Proof. We show that $d_1(E(\mathbb{F}_2^n), \mathcal{H}) = 2$, where $\mathcal{H} = \mathcal{H}(\mathbf{v}, \theta)$. Since $\|\mathbf{v}\|_\infty = 1$, Lemma 1 implies that

$$d_1(E(\mathbb{F}_2^n), \mathcal{H}) = \min_{\mathbf{x} \in \mathbb{F}_2^n} |E(\mathbf{x}) \cdot \mathbf{v}^\top - \theta|,$$

and hence it suffices to show that $|E(\mathbf{x})\mathbf{v}^\top - \theta| \geq 2$ for every $\mathbf{x} \in \mathbb{F}_2^n$ and that the coded neuron always correctly computes τ. Since $\chi_{[n]}(\mathbf{w}) \cdot \chi_{[n]}(\mathbf{x}) = \chi_{[n]}(\mathbf{w} \oplus \mathbf{x}) = (-1)^{w_H(\mathbf{w} \oplus \mathbf{x})}$, Lemma 5 implies that

$$\begin{aligned} E(\mathbf{x})\mathbf{v}^\top - \theta &= \mathbf{x}\mathbf{w}^\top + (-1)^{\theta'} \chi_{[n]}(\mathbf{w}) \cdot \chi_{[n]}(\mathbf{x}) - \theta \qquad (3) \\ &= n - 2w_H(\mathbf{x} \oplus \mathbf{w}) + (-1)^{\theta' + w_H(\mathbf{x} \oplus \mathbf{w})} - \theta, \end{aligned}$$

and we distinguish between the next four cases.

Case 1: If $w_H(\mathbf{x} \oplus \mathbf{w}) \leq \frac{n-\theta-1}{2} - 1$, then

$$\begin{aligned} (3) &\geq n - (n - \theta - 1) + 2 + (-1)^{\theta' + w_H(\mathbf{w} \oplus \mathbf{x})} - \theta \\ &= 3 + (-1)^{\theta' + w_H(\mathbf{w} \oplus \mathbf{x})} \geq 2. \end{aligned}$$

Case 2: If $w_H(\mathbf{x} \oplus \mathbf{w}) = \frac{n-\theta-1}{2}$, then

$$\begin{aligned} (3) &= n - (n - \theta - 1) + (-1)^{\theta' - \frac{n-\theta-1}{2}} - \theta \\ &= 1 + (-1)^0 = 2. \end{aligned}$$

Case 3: If $w_H(\mathbf{x} \oplus \mathbf{w}) = \frac{n-\theta+1}{2}$, then

$$\begin{aligned} (3) &= n - (n - \theta + 1) + (-1)^{\theta' + \frac{n-\theta+1}{2}} - \theta \\ &= -1 + (-1)^{n-\theta} = -2, \end{aligned}$$

where the last equality follows since $n - \theta$ is always odd.
Case 4: If $w_H(\mathbf{x} \oplus \mathbf{w}) \geq \frac{n-\theta+1}{2} + 1$, then

$$\begin{aligned} (3) &\leq n - (n - \theta + 1) - 2 + (-1)^{\theta' + w_H(\mathbf{x} \oplus \mathbf{w})} - \theta \\ &= -3 + (-1)^{\theta' + w_H(\mathbf{x} \oplus \mathbf{w})} \leq -2. \end{aligned}$$

Now, it follows from Lemma 6 and from the first two cases that $\operatorname{sign}(E(\mathbf{x})\mathbf{v}^\top - \theta) = 1$ whenever $\tau(\mathbf{x}) = 1$. Similarly, the latter two cases imply that $\operatorname{sign}(E(\mathbf{x})\mathbf{v}^\top - \theta) = -1$ whenever $\tau(\mathbf{x}) = -1$. Therefore, the coded neuron $\tau'(E(\mathbf{x})) = \operatorname{sign}(E(\mathbf{x})\mathbf{v}^\top - \theta)$ correctly computes τ on all inputs with minimum distance $d = 2$, and the claim follows. □

By using the parity solution, one can attain 1-robustness, i.e., robustness against *any single (adversarial) erasure*, by adding only one bit of redundancy. In contrast, to attain 1-robustness by using replication (Subsection III-A), one should add n bits of redundancy. Moreover, it is readily verified that due to Lemma 2, the suggested solution is optimal in terms of the length $m = n + 1$ among all 1-robust solutions.

Since the parity function E is universal to all binary neurons, every DNN which comprises of binary neurons (and in particular, binarized DNNs) can be made robust to adversarial tampering in its input. Furthermore, to employ this technique for error *inside* the DNN, one should add a single parity gate in every layer (see Fig. 1e). If one wishes to construct DNNs by *only* using neurons, a classic result by Muroga [11] shows how to implement the parity function by using neurons.

A. Generalized parity for all neurons

The following generalizes the parity code, and requires *integer weights*. Since every neurons has a representation with only integer weights [13, Exercise. 5.1], it applies to all neurons. However, superiority to replication in terms of relative distance is guaranteed only if $\|\mathbf{w}\|_1 < \frac{2n}{\delta} - 1$. The proof will appear in future versions of this paper.

Theorem 2. *The relative distance of the solution* $(E_{\mathbf{w}}, \mathbf{v}, \theta)$ *is* $2/(\|\mathbf{w}\|_1 + 1)$, *where*

$$\begin{aligned} \mathbf{v} &= (\mathbb{1}_{\mathbf{w}}, (-1)^{\theta'} \chi_{[\|\mathbf{w}\|_1]}(\mathbb{1}_{\mathbf{w}})), \\ \mathbb{1}_{\mathbf{w}} &= (\underbrace{\operatorname{sign}(w_1), \ldots, \operatorname{sign}(w_1)}_{|w_1| \textit{ times}}, \ldots, \underbrace{\operatorname{sign}(w_n), \ldots, \operatorname{sign}(w_n)}_{|w_n| \textit{ times}}), \end{aligned}$$

and

$$E_{\mathbf{w}}(\mathbf{x}) = \left(\underbrace{x_1, \ldots, x_1}_{|w_1| \textit{ times}}, \ldots, \underbrace{x_n, \ldots, x_n}_{|w_n| \textit{ times}}, \prod_{i=1}^{n} x_i^{w_i \bmod 2} \right).$$

V. Discussion and Future Research

In this paper we studied a novel approach for combating noise in DNNs with error correcting codes, established basic framework, and presented several solutions. This work can be seen as an extension of the recently popularized *coded computation* topic, which concerns computation over coded data in distributed environments, into the realm of neural computation. A plethora of questions remain widely open:

1) Extend the above framework to other activation functions, and to sigmoid functions in particular.
2) Develop solutions with relative distance greater than $2/(n+1)$ for binarized neurons.
3) Find other families of neurons for which robustness can be guaranteed.
4) Extend Lemma 2 to other parameter regimes, i.e., establish fundamental trade-offs between the parameters n, m, and d.

References

[1] M. El-Mhamdi and R. Guerraoui, “When Neurons Fail,” *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2017.

[2] Eykholt *et al.*, “Robust physical-world attacks on deep learning visual classification,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1625–1634, 2018.

[3] I. Goodfellow, J. Shlens and C. Szegedy, “Explaining and Harnessing Adversarial Examples,” *International Conference on Learning Representations (ICLR)*, 2015.

[4] K. Guo, S. Han, S. Yao, Y. Wang, Y. Xie and H. Yang, “Software-hardware Codesign for Efficient Neural Network Acceleration”, *IEEE Micro*, vol. 37, no. 2, pp. 18–25, 2017.

[5] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv and Y. Bengio, “Binarized Neural Networks,” *Proc. Neural Information Processing Systems (NIPS)*, pp. 4107–4115, 2016

[6] H. Ito and T. Yagi, “Fault Tolerant Design using Error Correcting Code for Multilayer Neural Networks,” *IEEE International Workshop on Defect and Fault Tolerance in VLSI Systems*, pp. 177–184, 1994.

[7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards Deep Learning Models Resistant to Adversarial Attacks,” *International Conference on Learning Representations* (ICLR), 2018.

[8] E. Melachrinoudis, “An analytical solution to the minimum L_p-norm of a hyperplane,” *Journal of Mathematical Analysis and Applications*, vol. 211, no. 1, pp. 172–189, 1997.

[9] S. Mertens, “The easiest hard problem: Number partitioning,” *Computational Complexity and Statistical Physics*, vol. 125, no. 2, pp. 125–139, 2006.

[10] M. Mohammadi, A. Al-Fuqaha, S. Sorour and M. Guizani, “Deep Learning for IoT Big Data and Streaming Analytics: A Survey,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018.

[11] S. Muroga, “The Principle of Majority Decision Logical Elements and the Complexity of their Circuits,” *IFIP Congress*, 1959.

[12] J. von-Neumann, “Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components,” *Automata Studies*, vol. 34, pp. 43–98, 1956.

[13] R. O’Donnell. Analysis of Boolean functions. Cambridge University Press, 2014.

[14] D. S. Phatak and I. Koren, “Complete and Partial Fault Tolerance of Feedforward Neural Nets,” *IEEE Transactions on Neural Networks*, vol. 6, no. 2, pp. 446–456, 1995.

[15] A. Shamir, I. Safran, E. Ronen, and O. Dunkelman, “A simple explanation for the existence of adversarial examples with small hamming distance,” *arXiv:1901.10861*, 2019.

[16] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow and R. Fergus, “Intriguing Properties of Neural Networks,” *arXiv:1312.6199*, 2013.

[17] C. Torres-Huitzil and B. Girau, “Fault and Error Tolerance in Neural Networks: A Review,” *IEEE Access*, vol. 5, pp. 17322–17341, 2017.