IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) | 978-1-6654-9427-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/INFOCOMWKSHPS57453.2023.10225774

Achieving Certified Robustness for Brain-Inspired Low-Dimensional Computing Classifiers

Fangfang Yang
UC Riverside

Shijin Duan
Northeastern University

Xiaolin Xu
Northeastern University

Shaolei Ren
UC Riverside

***Abstract*—Brain-inspired hyperdimensional computing (HDC) in machine learning applications has been achieving great success in terms of energy efficiency and low latency. The proposal of low-dimensional computing (LDC) classification model not only improves the inference accuracy of existing HDC-based models but also gets rid of the ultra-high dimension in them. However, the security part of LDC model to adversarial perturbations has not been touched. In this paper, we adopt the bounding technique, interval bound propagation (IBP), to train a LDC classification model that is provably robust against L_∞ norm-bounded adversarial attacks. Specifically, we propagate the L_∞ norm-bounded bounding box around the original input through layers of LDC model using interval arithmetic. After propagation, the worst case prediction logits can be computed based on the upper bound and the lower bound of the output bounding box. By minimizing the loss between the worst case prediction and the true label, the predicted label could be kept invariant over all possible adversarial perturbations within L_∞ norm-bounded ball. We evaluate the algorithm on both MNIST and fashion MNIST datasets. The experiment results corroborate that our trained models with IBP exhibit robustness against strong projected gradient descent (PGD) attacks and memory errors.**

***Index Terms*—Low-dimensional computing, adversarial attack, certified robustness, interval bound propagation**

I. Introduction

Brain-inspired hyperdimensional computing (HDC) classifiers have been emerging as light-weight machine learning alternatives to deep learning models [1]–[4]. More recently, a low-dimensional computing (LDC) classification framework has been proposed which, compared to traditional HDC-based classification models, improves the inference accuracy and meanwhile dramatically reduces the model size, inference latency and energy consumption by orders of magnitude [5].

Nonetheless, recent studies have shown that HDC-based classifiers are vulnerable to carefully crafted adversarial attacks in both white-box and black-box settings [6]–[9]. In such attacks, the adversarial perturbations introduced to the original input are visually indistinguishable but could make the output label deviate from the ground truth. There has been significant interest in the literature in constructing defenses to protect classification models against adversarial attacks, like obfuscating gradients, defensive distillation and retraining technique [8]–[13]. Unfortunately, these defenses are typically targeted to specific attacks. For example, obfuscating gradient technique takes advantage of gradient masking method and provides certain robustness against white-box iterative optimization attacks [13]–[16]. Thus, these defense techniques were broken soon by subsequent attacking schemes, which then drives the emergence of *certified* robustness for defenses [17], [18].

Certified defenses provide guarantee of robustness against *all* possible norm-bounded perturbation attacks. For example, methods proposed in work [19]–[21] alter the network configurations such as the network structure and activation function, which makes them struggle to generalize across different types of networks. Provable robustness technique via random smoothing requires taking the mean of the output vectors, which is susceptible to the outliers and lead to ambiguous outputs [22]–[26]. The study in [18] leverages differential privacy and provides a scheme which requires extra model structure like the separate auto-encoder. Interval bound propagation (IBP) technique bypasses the challenges of these methods. It is comparable to two forward passes through the network, without changing the original network and inducing extra structure [27]–[31].

In this paper, we make the first effort to study provable robustness of LDC models with IBP for classification against adversarial attacks. To obtain a certifiably robust LDC model against L_∞ perturbation, the minimum difference between logits of the true class and any other class, called minimum margin, has to be larger than zero for any input perturbation within L_∞ norm-bound ball. To this end, IBP is adopted to calculate the lower bound of the minimum margin. An appropriate loss function is defined to guarantee a non-negative value of the lower bound and thus a correct labelling over L_∞ norm-bounded perturbed inputs. For evaluation, we train LDC models across a wide range of L_∞ perturbation radii, referred to as training perturbation radius, based on both MNIST and fashion MNIST dataset. We also employ the elision technique to make the lower bound of the minimum margin tighter and compare the performance of the trained models in terms of nominal accuracy and verified accuracy. Besides, we implement a powerful white box attacking method, projected gradient descent (PGD), to each of the trained models and demonstrate a drastic reduction in attack success rate from 100% to below 0.1% with IBP robust training. The trained models also exhibit high performance with memory errors existing.

II. Preliminaries and Formulation

A. LDC classifier

In a nutshell, an LDC classifier maps the encoding and inference process of HDC classifier into an equivalent *compact* neural network that includes a non-binary neural network for

978-1-6654-9427-4/23/$31.00 ©2023 IEEE

value representation followed by a binary neural network layer for sample encoding and another binary layer for inference. After training, it can extract optimized low-dimensional binary vectors to represent features and values for efficient inference.

We focus on certified robustness of an LDC model for classification tasks. The LDC model can be formulated as a function f_θ: $x \rightarrow \mathbb{R}^C$, where the input data is in a normalized N-dimensional subspace $x \subseteq [0,1]^N$. The model provides confidence scores $f_\theta(x) \subseteq [0,1]^C$ for all C classes. $F_\theta(x) = \arg\max_{i\in[C]} f_\theta(x)_i$ is the predicted class label of model f_θ given input x. θ is the set of trainable parameters of the model, which is trained to minimize the cross-entropy loss.

Specifically, an LDC classifier f_θ can be equivalently mapped to a 3-stage neural network, including value layer, feature layer and class layer, respectively, as shown in Fig. 1. It can be mathematically represented as follows:

$$\begin{cases} z_0 = x_0 \\ z_1 = Concat(W_0 z_0^i + b_0) \\ z_1 = Bin(Tanh(z_1)) \\ z_2 = Bin(W_1^b z_1) \\ z_3 = W_2^b z_2 \end{cases} \tag{1}$$

where $x_0 \subseteq [0,1]^N$ is the input. z_0^i is the ith dimension of z_0 for $i = 1$ to N. $Bin(z) = sign(z)$ and $Concat(z)$ is concatenating operation which joins the weighted sum of each item in the input vector into a single output vector. The trainable parameters $\theta = \{W_0, b_0, W_1^b, W_2^b\}$. The shape of W_0, W_1^b, W_2^b is $(D_v, D_p), (D_f, N \times D_v), (D_c, D_f)$ respectively. D_p represents the dimension of each input feature value and D_c is the dimension of final output vector. Take MNIST classification as an example, $D_p = 1$ since each pixel value could be represented as a single scalar. $D_c = 10$ because there are 10 classes in total. D_v and D_f are the hyperparameters of the model representing dimension of value vector and dimension of feature vector in HDC context. Thus, in LDC model there are only affine transformations, $Wz + b$, and monotonic activation functions, $Concat(z)$, $Bin(z)$ and $Tanh(z)$.

B. Adversarial attacks

Adversarial attacks can be categorized into two settings, targeted attack and untargeted attack. The goal of targeted attack is to mislead the model to classify the adversarial example to an intended target class, y_{tg}, instead of the true class, y_{true}. On the other hand, an untargeted attacker makes the model misclassify the perturbed image as any class, y', other than the original true class, y_{true}. Both attacks can be formulated as bounded perturbation. For given input (x_0, y_{true}), the attacker would like to generate a perturbed input $A_{p,\epsilon}(x_0) = \{x : \|x - x_0\|_p < \epsilon\}$ such that $F_\theta(x) \neq y_{true}$. We use $A_{p,\epsilon}(x_0)$ to denote the perturbed input which is sampled from the region centered at x_0 with ϵ radius, where ϵ represents the perturbation magnitude measured by L_p norm for $p \geq 1$. Common choices of L_p are L_1, L_2 and L_∞.

C. Robustness verification

To certify the robustness of a classifier against norm-bound perturbation, $A_{p,\epsilon}(x_0)$, we need to verify that for any possible perturbed input $x \in A_{p,\epsilon}(x_0)$ the predicted class is always the true label y_{true}. To achieving this purpose, we define a minimum margin, $M(y_{true}, y')$, as the minimum prediction logit difference between the true class label y_{true} and any other class y', when the input x is within the L_p norm-bounded ball by ϵ. $\forall x \in A_{p,\epsilon}(x_0)$ and $y' \neq y_{true}$, we have

$$\begin{aligned} M(y_{true}, y') &= min_x(f_\theta(x)_{y_{true}} - f_\theta(x)_{y'}) \\ &= min_x(e_{y_{true}} - e_{y'}) f_\theta(x) \end{aligned} \tag{2}$$

where e_i is the i^{th} standard basis vector. For any $y' \neq y_{true}$, if we can verify that $M(y_{true}, y') > 0$, which means the true label will always has the highest confidence score, f_θ is certifiably robust at x_0 within radius ϵ with respect to L_p norm.

III. Interval Bound Propagation

A. Interval Bound Propagation

It is highly non-trivial to find the exact minimum margin $M(y_{true}, y')$ (hereafter $M_{y'}$) and prove $M_{y'} > 0$. Instead, we could look for a lower bound of $M_{y'}$ and control the value inside this bound. To this end, we consider the framework of IBP [27], [32] to train a provably robust LDC classifier to L_∞ adversarial perturbation of size ϵ (which is stronger than L_p adversarial perturbation with $1 \leq p < \infty$). IBP is an algorithm that can be used to find a lower bound of the minimum margin $M_{y'}$ by bounding the activation z_k of each layer. Specifically, it propagates the axis aligned bounding box from layer to layer using interval arithmetic. For L_∞ norm-bounded perturbation by ϵ, lower bounds and upper bounds of each layer can be represented by the following equations.

$$\begin{aligned} \bar{z}_{0,i}(\epsilon) &= x_{0,i} + \epsilon \\ \underline{z}_{0,i}(\epsilon) &= x_{0,i} - \epsilon \\ &\cdots \\ \bar{z}_{k,i}(\epsilon) &= max_{\underline{z}_{k-1}(\epsilon) \leq z_{k-1} \leq \bar{z}_{k-1}(\epsilon)} h_{k,i}(z_{k-1}) \\ \underline{z}_{k,i}(\epsilon) &= min_{\underline{z}_{k-1}(\epsilon) \leq z_{k-1} \leq \bar{z}_{k-1}(\epsilon)} h_{k,i}(z_{k-1}) \\ &\cdots \end{aligned} \tag{3}$$

where $z_{k,i}$ is the i-th coordinate of z_k and $z_k = h_k(z_{k-1})$. In LDC, there are three layers, thus $k = 0$ to 3. $h_k(z)$ is the transformation function of k-th layer, which is either affine transformation or element-wise monotonic activation function, $Concat(\cdot)$, $Bin(\cdot)$ and $Tanh(\cdot)$. For the affine layer $h_k(z_{k-1}) = Wz_{k-1} + b$, obtaining the upper bound and lower bound, i.e. solving the above optimization problem, can be done efficiently with two matrix multiplication as follows:

$$\begin{aligned} \mu_{k-1} &= \frac{\bar{z}_{k-1} + \underline{z}_{k-1}}{2} \\ r_{k-1} &= \frac{\bar{z}_{k-1} - \underline{z}_{k-1}}{2} \\ \mu_k &= W\mu_{k-1} + b \\ r_k &= |W| r_{k-1} \\ \bar{z}_k &= \mu_k + r_k \\ \underline{z}_k &= \mu_k - r_k \end{aligned} \tag{4}$$

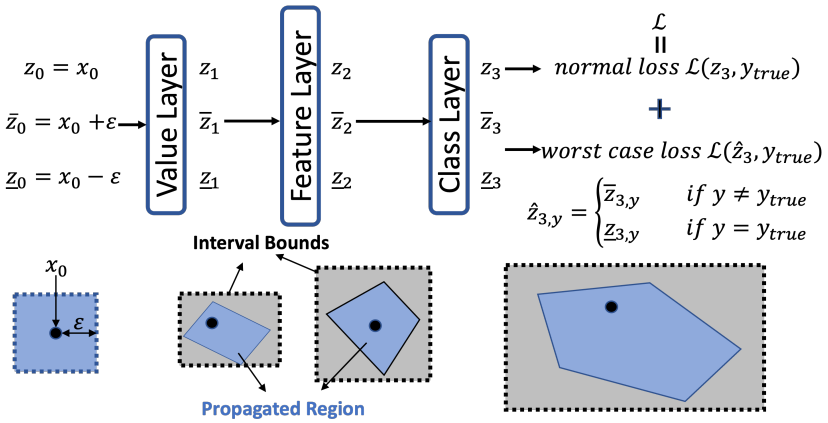


Fig. 1. Illustration of LDC model with IBP method. The L_∞ norm-bounded perturbation with radius ϵ (in blue) is propagated through layers of LDC model. The interval bound (in gray), represented as $|\bar{z}_k, \underline{z}_k|$, is propagated simultaneously through layers, which always encompasses the blue region.

where $|\cdot|$ is element-wise absolute value operator. When $h_k(z_{k-1})$ is element-wise monotonic activation function, such as $Concat(\cdot)$, $Bin(\cdot)$ and $Tanh(\cdot)$, we have:

$$\begin{aligned} \bar{z}_k &= h_k(\bar{z}_{k-1}) \\ \underline{z}_k &= h_k(\underline{z}_{k-1}) \end{aligned} \tag{5}$$

In the case of an LDC classifier, after propagation, we obtain the upper and lower bounds of the output logits, $\bar{z}_3$ and $\underline{z}_3$. With the bounds of z_3 and the IBP method for affine layer, a lower bound of minimum margin $M_{y'}$ can be computed as

$$\begin{aligned} \underline{M_{y'}} &= min_{\underline{z}_3 \leq z_3 \leq \bar{z}_3}(e_{y_{true}} - e_{y'})z_3 \\ &= e_{y_{true}}\underline{z}_3 - e_{y'}\bar{z}_3 \\ &= \underline{z}_{3,y_{true}} - \bar{z}_{3,y'} \\ &\leq min_{\underline{z}_0 \leq x \leq \bar{z}_0}(e_{y_{true}} - e_{y'})f_\theta(x) = M_{y'} \end{aligned} \tag{6}$$

For any class label y' other than the true label y_{true}, to make the lower bound of minimum margin, $\underline{M_{y'}}$, larger than 0, we can construct worst case prediction $\hat{z}_k$, where the logit of the true class is equal to its lower bound and the other logits are equal to their upper bound. Note that, if $\epsilon = 0$, $\hat{z}_k = z_k$.

$$\hat{z}_{k,y}(\epsilon) = \begin{cases} \bar{z}_{k,y}(\epsilon) & y \neq y_{true} \\ \underline{z}_{k,y}(\epsilon) & y = y_{true} \end{cases} \tag{7}$$

We then minimize a worst-case cross entropy loss $\mathcal{L}(\hat{z}_k, y_{true})$ during the training procedure. However, a direct application of worst-case cross entropy loss alone does not work since the propagated bounds are too loose. In reality, As shown in Fig. 1, during training stage, we feed the network with both original training input, z_0, its upper bound, $\bar{z}_0$, and lower bound, $\underline{z}_0$, then minimize a combination of normal cross-entropy loss and worst case cross-entropy loss.

$$\mathcal{L} = k\mathcal{L}(z_k, y_{true}) + (1-k)\mathcal{L}(\hat{z}_k, y_{true}) \tag{8}$$

where k is a trade-off parameter, which controls the relative weight of robust training versus fitting to the original input images.

B. Elision of Last Layer

Considering the fact that the last layer in LDC network is a linear layer, $z_3 = W_2^b z_2$, to make the calculated lower bound of minimum margin, $\underline{M_{y'}}$, tighter, we elide the bound propagation of the last linear layer:

$$\begin{aligned} \underline{M_{y'}} &= min_{\underline{z}_3 \leq z_3 \leq \bar{z}_3}(e_{y_{true}} - e_{y'})z_3 \\ &\leq min_{\underline{z}_2 \leq z_2 \leq \bar{z}_2}(e_{y_{true}} - e_{y'})W_2^b z_2 \\ &= min_{\underline{z}_2 \leq z_2 \leq \bar{z}_2}\hat{W}z_2 = \underline{M_{y'}^e} \\ &\leq min_{\underline{z}_0 \leq x \leq \bar{z}_0}(e_{y_{true}} - e_{y'})f_\theta(x) = M_{y'}. \end{aligned} \tag{9}$$

Thus, minimizing $\hat{W}z_2$ over $\underline{z}_2 \leq z_2 \leq \bar{z}_2$, with $\hat{W} = (e_{y_{true}} - e_{y'})W_2^b$, gives a tighter lower bound, $\underline{M_{y'}^e}$, of minimum margin $M_{y'}$. By doing so, we could bypass the additional relaxation induced by the last linear layer.

IV. Results

We will present our evaluation results based on MNIST and fashion MNIST datasets in this section. We first discuss the experiment setup. In what follows, the nominal accuracy and verified accuracy are shown for each trained model with different training epsilon. Besides, the robustness results of our trained models against PGD attack and memory cell errors are also presented.

A. Experiment Setup

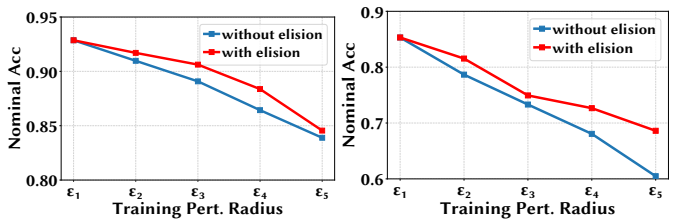
a) Hyperparameters: The hyperparameters related to the LDC model architecture follow those in [5], where D_v/D_f is set to 4/64 to get a good trade-off between good accuracy and a sufficiently small model size. The loss function of the training process uses $CrossEntropyLoss(\cdot)$, and $Adam(\cdot)$ method is adopted as the optimizer following SOTA training strategy for BNN [33]. Even if the $Adam(\cdot)$ method intrinsically adapts the learning rate to each parameter, tuning the initial learning rate and decay scheme for $Adam(\cdot)$ yield significant performance improvement [34]. Thus, we implement grid-search mechanism to find the best initial learning rate and weight decay. Besides, we also adopt exponential learning rate decay with decay rate

TABLE I
CONFIGURATION OF TRAINING PROCEDURE FOR EACH DATASET WITH DIFFERENT TRAINING PERTURBATION RADII.

Dataset	Pert. Radius	Without Elision		With Elision	
		Learning Rate	Weight Decay	Learning Rate	Weight Decay
MNIST	ϵ_1	0.0001	0.0001	0.0001	0.0001
	ϵ_2	0.0001	0	0.0001	0.0001
	ϵ_3	0.0001	0	0.0001	0.0001
	ϵ_4	0.0001	0.0001	0.0001	0.001
	ϵ_5	0.0001	0.001	0.0001	0.01
Fashion MNIST	ϵ_1	0.0001	0.0001	0.0001	0.0001
	ϵ_2	0.0001	1e-5	0.001	0.01
	ϵ_3	0.001	0.01	0.001	0.001
	ϵ_4	0.0001	0.0001	0.0001	0.01
	ϵ_5	0.001	1e-5	0.001	0.01

of 0.95 to the provided initial learning rate for a better convergence. Table I shows the best choice of initial learning rate and weight decay for different dataset with different training perturbation radii. The 5 different perturbation radii ϵ_1, ϵ_2, ϵ_3, ϵ_4, ϵ_5 represent the L_∞ perturbation of 0, 0.02, 0.05, 0.08, and 0.1 associated to the normalized input $x \subseteq [0,1]^N$.

b) Training: According to Section III, the final loss function to minimize is a combination of normal cross entropy loss and worst case cross entropy loss. The relative importance of the worst case loss is determined by the hyperparameter k. According to literature, it achieves better results by slowly reducing k starting from 1 until 0.5. The same strategy is used for training perturbation radius, staring with 0 and slowly being raised up to the target value. In reality, the total iteration in our experiment is set to 120000 with batch size of 64. During the first 2000 iterations, the model is trained to reduce nominal loss alone, which can be regarded as a warm up period. Starting from the $2000th$ iteration, the model entered a linearly ramp up phase by gradually decreasing parameter k and increasing the perturbation radius. After 10000 iterations, parameter k and the training perturbation radius settle to 0.5 and the target radius respectively.



(a) (b)

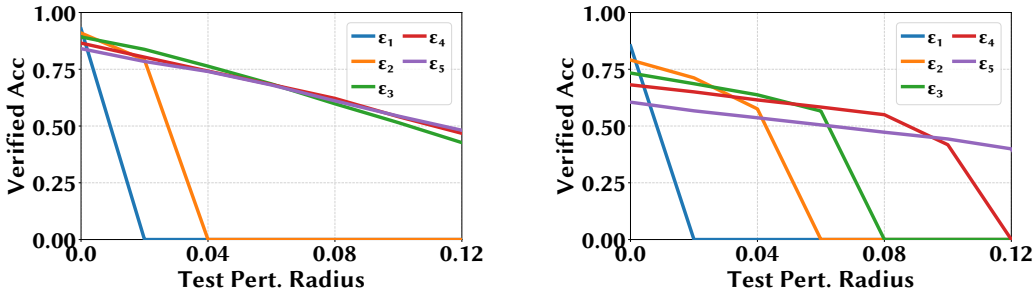
Fig. 2. Nominal accuracy of LDC models with different training perturbation radii, ϵ_1, ϵ_2, ϵ_3, ϵ_4, and ϵ_5, representing perturbation radius of 0, 0.02, 0.05, 0.08, and 0.1 respectively. The blue line are the results without elision technique and the red line represents the one with eliding the last layer. (a) MNIST dataset. (b) Fashion MNIST dataset.

B. Nominal Accuracy and Verified Accuracy

We train LDC classifiers using a range of perturbation radii $\epsilon \subseteq \{0, 0.02, 0.05, 0.08, 0.1\}$ on both MNIST and fashion MNIST datasets. Note that when training perturbation radius $\epsilon = 0$, the normal training with standard cross-entropy loss is performed. After training, we obtain 5 robust models for each dataset. During testing, we test each of the trained models against adversarial perturbation from 0 to 0.12. We add the test adversarial perturbation to each test image and compute the worst case prediction, based on which we obtain the inference accuracy over the test set, which is called verified accuracy. Note that when test perturbation radius is 0, the nominal test accuracy is obtained, which is called nominal accuracy.

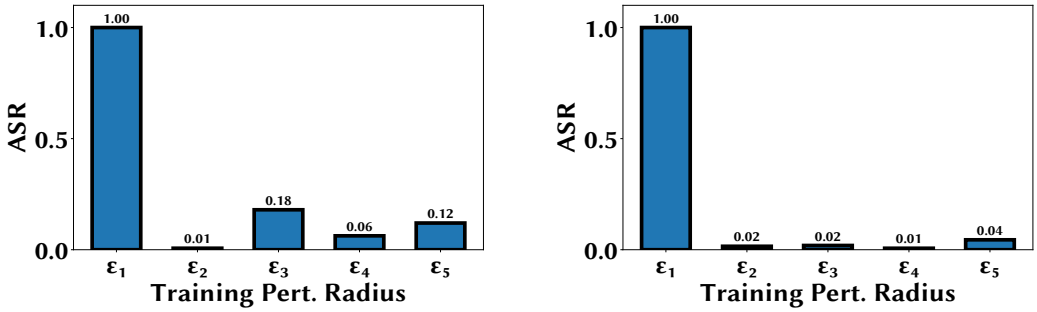
To test the effectiveness of elision technique, we compare the nominal accuracy of five trained models with and without eliding the last layer. Fig. 2(a) presents the results based on MNIST dataset. From the figure we can see the nominal test accuracy of standard LDC model with zero training perturbation radius is around 93%. The red line shows the results when eliding the last layer during IBP procedure. The blue line displays that without elision technique. In both lines, the nominal accuracy is decreasing with the increasing of training perturbation radius. This corroborate that the addition of verification loss deteriorate the ability of the model fitting to the dataset. However, the red line is sliding slower than the blue one. This is because that the elision of the last layer makes the calculated bound tighter and the penalty to the nominal accuracy becomes less severe compared to that of standard IBP method without elision scheme. Similarly, we give the experiment results of nominal accuracy on the basis of fashion MNIST dataset, referring to Fig. 2(b), revealing the akin results.

On the other hand, we demonstrate the verified accuracy of the trained models with standard IBP method without elision of the last layer. We choose a spectrum of test perturbation radii from 0 to 0.12 spaced by 0.02. Fig. 3(a) gives the results of MNIST dataset. As we can see from the figure, the standard model without robust training presents a zero verified accuracy when the test adversarial perturbation is above 0.02. The model trained with 0.02 training perturbation radius exhibits immunity to test adversarial perturbation of 0.02 and becomes vulnerable again when the test perturbation increases to 0.04. Models with training perturbation radius of 0.05, 0.08, and 0.1 show similar robustness. However, the verified accuracy of the model trained with smaller training perturbation radius degrades more quickly as the test perturbation radius increases. The effectiveness of increasing training perturbation radius becomes more obvious in the results of fashion MNIST dataset. As shown in Fig. 3(b), the model trained with higher training perturbation radius show higher verified accuracy especially for large test perturbation radius.



(a) (b)

Fig. 3. Verified accuracy against different test perturbation radii from 0 to 0.12. Five different models, ϵ_1, ϵ_2, ϵ_3, ϵ_4, and ϵ_5, associate with training perturbation radius of 0, 0.02, 0.05, 0.08, and 0.1 respectively. (a) MNIST dataset. (b) Fashion MNIST dataset.



(a) (b)

Fig. 4. Attack success rate (ASR) of PGD attacking method to five LDC models, ϵ_1, ϵ_2, ϵ_3, ϵ_4, and ϵ_5, trained with IBP with training perturbation radius of 0, 0.02, 0.05, 0.08, and 0.1 respectively. (a) MNIST dataset. (b) Fashion MNIST dataset.

C. Robustness Against PGD

We also assess the trained models' robustness against a powerful attack method called PGD. PGD is a white-box attack algorithm which means the attacker has full access to the model, including models' weights and gradients. It is actually an iterative version of FGSM [35]. In our experiment, we calculate the attack success rate (ASR) of PGD method under 200 iterations. Specifically, we use FGSM to introduce adversarial perturbation to each test image, which can be correctly classified by model. We calculate the percentage of images that can be crafted within 200 iterations to mislead the classifier, which is denoted as ASR. To attack a trained model with a specific training ϵ, we use the same amount of perturbation ϵ in PGD algorithm. Note that for the model without robust training, the PGD attacking perturbation ϵ is 0.02. We use ASR of PGD to indicate the model robustness.

Fig. 4(a) and fig. 4(b) are the results of PGD to the robust LDC models on the basis of MNIST and fashion MNIST dataset, respectively. From the figure, we can see that for both MNIST and fashion MNIST dataset, PGD can achieve 100% attack success rate attacking the models without robust training. However, to the models trained with IBP, the ASR decrease significantly. PGD fails to attack the models trained with IBP across the full training perturbation spectrum.

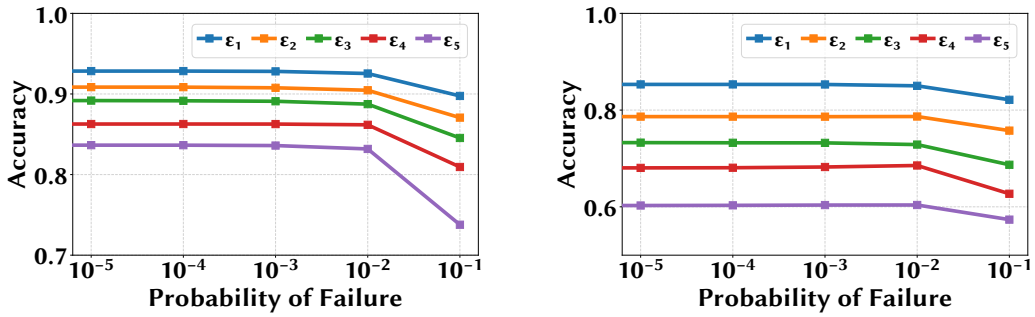
D. Robustness Against Memory Errors

LDC classifiers have orders-of-magnitude less dimensions than their HDC counterparts. While the prior study has shown that LDC classifiers are still robust to memory errors in the hardware [5], it is not clear if our novel IBP robust training method will break such robustness and introduce vulnerabilities for LDC classifiers. Thus, in addition to perturbation attacks, we also evaluate the performance of LDC model with IBP robust training in the presence of erroneous memory cells in the hardware. To demonstrate robustness against such hardware errors, we conduct RTL fault simulations where we inject memory bit flips during every clock cycle of execution. In the simulation, we set the probability of failure for each memory cell to be 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, respectively.

Fig. 5(a) and Fig. 5(b) present the test accuracy with memory errors. X axis displays the probability of failure for each memory cell in every clock cycle. From the figures we can see that for both MNIST and fashion MNIST dataset, the accuracy diminishes when the training perturbation radius increases. And for each model, the performance maintains a high accuracy when the probability is lower than 10^{-2} even if the accuracy starts to drop afterwards. Thus, with our robust training, LDC classifiers are still robust against memory errors.

V. Conclusion

In this paper, we propose IBP to achieve certified robustness for LDC classifiers against all possible L_∞ norm-bounded adversarial attacks. Specifically, based on IBP, the worst case prediction logits can be computed based on the upper bound and the lower bound of the output bounding box during training. Thus, by minimizing the loss between the worst case prediction and the true label, the predicted label could be kept invariant over all possible adversarial perturbations within L_∞ norm-bounded ball. We evaluate our algorithm on both MNIST and fashion MNIST datasets, demonstrating that it can withstand projected gradient descent (PGD) attacks.



(a) (b)

Fig. 5. Classification accuracy of five trained models, ϵ_1, ϵ_2, ϵ_3, ϵ_4, and ϵ_5, with faulty memory cells, when the probability of failure for each memory cell is 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2} and 10^{-1}. ϵ_1, ϵ_2, ϵ_3, ϵ_4, and ϵ_5 correspond to training perturbation radius of 0, 0.02, 0.05, 0.08, and 0.1 respectively. (a) MNIST dataset. (b) Fashion MNIST dataset.

ACKNOWLEDGEMENT

This work is supported in part by the U.S. National Science Foundation under grants CNS-1910208 and CNS-2153690.

REFERENCES

[1] L. Ge and K. K. Parhi, "Classification using hyperdimensional computing: A review," *IEEE Circuits and Systems Magazine*, vol. 20, no. 2, pp. 30–47, 2020.

[2] A. Rahimi, P. Kanerva, and J. M. Rabaey, "A robust and energy-efficient classifier using brain-inspired hyperdimensional computing," in *Proceedings of the 2016 International Symposium on Low Power Electronics and Design*, ISLPED '16, (New York, NY, USA), p. 64–69, Association for Computing Machinery, 2016.

[3] D. Kleyko, A. Rahimi, D. Rachkovskij, E. Osipov, and J. Rabaey, "Classification and recall with binary hyperdimensional computing: Tradeoffs in choice of density and mapping characteristics," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, pp. 1–19, 04 2018.

[4] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation,"

[5] S. Duan, X. Xu, and S. Ren, "A brain-inspired low-dimensional computing classifier for inference on tiny devices," in *TinyML*, 2022.

[6] W. Chen and H. Li, "Adversarial attacks on voice recognition based on hyper dimensional computing," *Journal of Signal Processing Systems*, vol. 93, no. 7, pp. 709–718, 2021.

[7] F. Yang and S. Ren, "Adversarial attacks on brain-inspired hyperdimensional computing-based classifiers," *arXiv preprint arXiv:2006.05594*, 2020.

[8] D. Ma, J. Guo, Y. Jiang, and X. Jiao, "Hdtest: Differential fuzz testing of brain-inspired hyperdimensional computing," 2021.

[9] R. Thapa, D. Ma, and X. Jiao, "Hdxplore: Automated blackbox testing of brain-inspired hyperdimensional computing," in *2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 90–95, IEEE, 2021.

[10] J. Uesato, B. O'Donoghue, A. v. d. Oord, and P. Kohli, "Adversarial risk and the dangers of evaluating against weak attacks," 2018.

[11] O. Gungor, T. Rosing, and B. Aksanli, "Res-hd: Resilient intelligent fault diagnosis against adversarial attacks using hyper-dimensional computing," 2022.

[12] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," 2015.

[13] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," 2018.

[14] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, ACM, nov 2017.

[15] X. Xu, J. Chen, J. Xiao, Z. Wang, Y. Yang, and H. T. Shen, "Learning optimization-based adversarial perturbations for attacking sequential recognition models," in *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 2802–2822, 2020.

[16] H. Qiu, Y. Zeng, Q. Zheng, T. Zhang, M. Qiu, and G. Memmi, "Mitigating advanced adversarial attacks with more advanced gradient obfuscation techniques," *arXiv preprint arXiv:2005.13712*, 2020.

[17] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," 2016.

[18] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, "Certified robustness to adversarial examples with differential privacy," 2018.

[19] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier, "Parseval networks: Improving robustness to adversarial examples," in *International Conference on Machine Learning*, pp. 854–863, PMLR, 2017.

[20] A. Raghunathan, J. Steinhardt, and P. Liang, "Certified defenses against adversarial examples," 2018.

[21] E. Wong and J. Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," 2017.

[22] J. M. Cohen, E. Rosenfeld, and J. Z. Kolter, "Certified adversarial robustness via randomized smoothing," 2019.

[23] J. Teng, G.-H. Lee, and Y. Yuan, "l_1 adversarial robustness certificates: a randomized smoothing approach," 2019.

[24] A. Levine, A. Kumar, T. Goldstein, and S. Feizi, "Tight second-order certificates for randomized smoothing," 2020.

[25] A. Levine and S. Feizi, "Robustness certificates for sparse adversarial attacks by randomized ablation," 2019.

[26] H. Salman, G. Yang, J. Li, P. Zhang, H. Zhang, I. Razenshteyn, and S. Bubeck, "Provably robust deep learning via adversarially trained smoothed classifiers," 2019.

[27] S. Gowal, K. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. A. Mann, and P. Kohli, "On the effectiveness of interval bound propagation for training verifiably robust models," *CoRR*, vol. abs/1810.12715, 2018.

[28] P. Morawiecki, P. Spurek, M. Smieja, and J. Tabor, "Fast and stable interval bounds propagation for training verifiably robust models," *CoRR*, vol. abs/1906.00628, 2019.

[29] Y. Wang, Z. Shi, Q. Gu, and C.-J. Hsieh, "On the convergence of certified robust training with interval bound propagation," in *International Conference on Learning Representations*, 2022.

[30] P.-S. Huang, R. Stanforth, J. Welbl, C. Dyer, D. Yogatama, S. Gowal, K. Dvijotham, and P. Kohli, "Achieving verified robustness to symbol substitutions via interval bound propagation," *arXiv preprint arXiv:1909.01492*, 2019.

[31] C. Wei and J. Z. Kolter, "Certified robustness for deep equilibrium models via interval bound propagation," in *International Conference on Learning Representations*, 2021.

[32] R. Ehlers, "Formal verification of piece-wise linear feed-forward neural networks," 2017.

[33] Z. Liu, Z. Shen, S. Li, K. Helwegen, D. Huang, and K.-T. Cheng, "How do adam and training strategies help bnns optimization," in *ICML*, pp. 6936–6946, 2021.

[34] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The marginal value of adaptive gradient methods in machine learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 4151–4161, Curran Associates Inc., 2017.

[35] F. Wu, R. Gazo, E. Haviarova, and B. Benes, "Efficient project gradient descent for ensemble adversarial attack," *arXiv preprint arXiv:1906.03333*, 2019.