STOR566: Introduction to Deep Learning

Lecture 15: Adversarial Defense

Yao Li
UNC Chapel Hill

Oct 13, 2022

Attack vs. Defense

Adversarial Examples Are Not Easily Detected:
Bypassing Ten Detection Methods

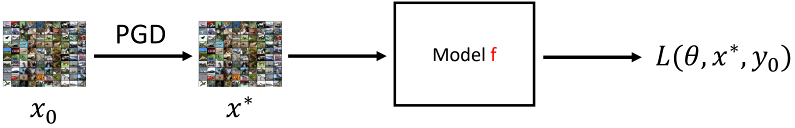
Nicholas Carlini David Wagner
University of California, Berkeley

Obfuscated Gradients Give a False Sense of Security:
Circumventing Defenses to Adversarial Examples

Anish Athalye [*1] **Nicholas Carlini** [*2] **David Wagner** [2]

Adversarial Training

Madry's adversarial training



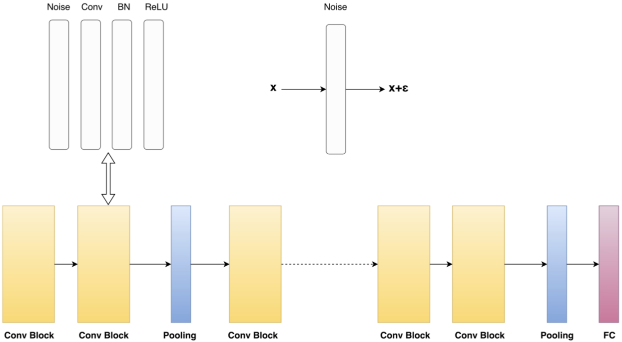
- **Madry's adversarial training:** Madry et al. (2018) proposed to incorporate the adversarial search inside the training process, by solving the following robust optimization problem:

$$\arg\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left\{ \max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} L\big(\boldsymbol{\theta}, \boldsymbol{x} + \boldsymbol{\delta}, y\big) \right\}$$

Madry et al. *Towards Deep Learning Models Resistant to Adversarial Attacks.* ICLR, 2018.

Randomization

Random Self-Ensemble



- **Random Self-Ensemble:** Liu et al. proposed a "noise layer", which fuses output of each layer with Gaussian noise.

Liu et al. *Towards Robust Neural Networks via Random Self-ensemble.* ECCV, 2018.

Projection

Embedding Regularized Classifier (ER-Classifier)

Observation 1

Levina et al., 2005: The only reason any methods work in very high dimensions is that, in fact, the data are not truly high-dimensional.

Observation 2

Tanay et al., 2016: Adversarial samples do not lie on the data manifold.

- How to use the two findings to improve robustness of Deep Neural Network?

Li et al. *Towards Robustness of Deep Neural Networks via Regularization.* ICCV, 2021.

Deep Classifier: Framework

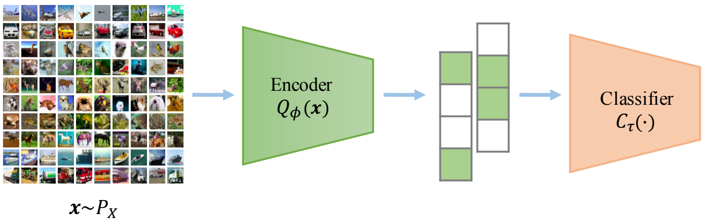


Figure: Deep Classifier

- P_X: Data distribution
- $\boldsymbol{Q}_\varphi$: Encoder (neural network)
- $\boldsymbol{C}_\tau$: Classifier (neural network)

ER-Classifier: Framework

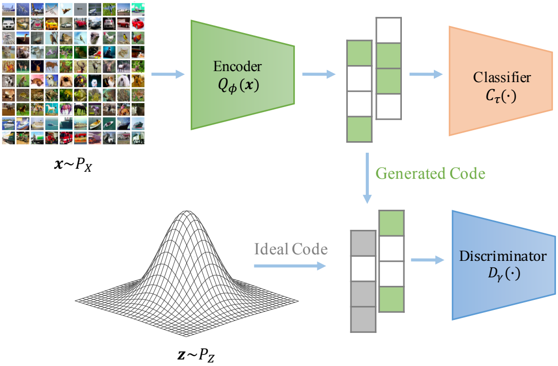


Figure: Embedding Regularized Classifier

- P_Z: Prior distribution
- $\boldsymbol{D}_\gamma$: Discriminator (neural network)

  (discriminator $\boldsymbol{D}$ used to separate "true" $\boldsymbol{z} \sim P_Z$ and "fake" $\tilde{\boldsymbol{z}} \sim \boldsymbol{Q}_Z$)

ER-Classifier: Objective Function

Minimize:

$$\mathbb{E}_{P_X}\mathbb{E}_{\boldsymbol{Q}(\boldsymbol{z}|\boldsymbol{x})}\left[\ell(y, \boldsymbol{C}(\boldsymbol{z}))\right] + \lambda\mathcal{D}(\boldsymbol{Q}_Z, P_Z) \tag{1}$$

Notations:

- $\boldsymbol{Q}(\boldsymbol{z}|\boldsymbol{x})$: encoder, projecting $\boldsymbol{x} \sim P_X$ to embedding vector $\boldsymbol{z}$
- $\boldsymbol{C}(\cdot)$: classifier, taking $\boldsymbol{z}$ as input and performing prediction
- ℓ: classification loss function
- y: label of $\boldsymbol{x}$

ER-Classifier: Objective Function

Minimize:

$$\mathbb{E}_{P_X}\mathbb{E}_{\boldsymbol{Q}(\boldsymbol{z}|\boldsymbol{x})}\left[\ell(y, \boldsymbol{C}(\boldsymbol{z}))\right] + \lambda\mathcal{D}(\boldsymbol{Q}_Z, P_Z) \quad (1)$$

Notations:

- $\boldsymbol{Q}(\boldsymbol{z}|\boldsymbol{x})$: encoder, projecting $\boldsymbol{x} \sim P_X$ to embedding vector $\boldsymbol{z}$
- $\boldsymbol{C}(\cdot)$: classifier, taking $\boldsymbol{z}$ as input and performing prediction
- ℓ: classification loss function
- y: label of $\boldsymbol{x}$

- $\boldsymbol{Q}_Z$: marginal distribution of $\boldsymbol{z}$ when $\boldsymbol{x} \sim P_X$ and $\boldsymbol{z} \sim \boldsymbol{Q}(\boldsymbol{z}|\boldsymbol{x})$.
- P_Z: prior distribution
- $\mathcal{D}$: arbitrary divergence between $\boldsymbol{Q}_Z$ and P_Z

  ER-Classifier: Combined with **adversarial training**, optimizing over adversarial examples

ER-Classifier: Methods Compared

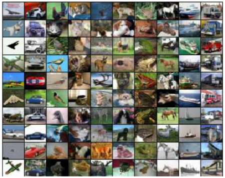
1. No defense: Models without any defense
2. ER-Classifier: ER-Classifier (with adversarial training)
3. ER-Classifier-: ER-Classifier without adversarial training
4. Madry's Adv: Madry's adversarial training
5. RSE: Random Self-ensemble.

Datasets

MNIST

28×28, 10

Train: 60,000; Test: 10,000



CIFAR10

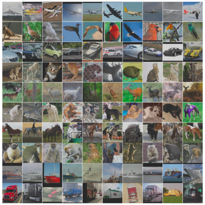
32×32, 10

Train: 50,000; Test: 10,000

ImageNet 143

64×64, 143

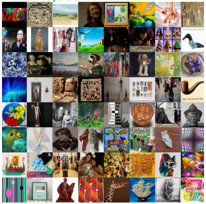
Train: 18,073; Test: 7,105



STL10

96×96, 10

Train: 5,000; Test: 8,000



Tiny ImageNet

64×64, 200

Train: 10,000; Test: 1,000

Experiment Results

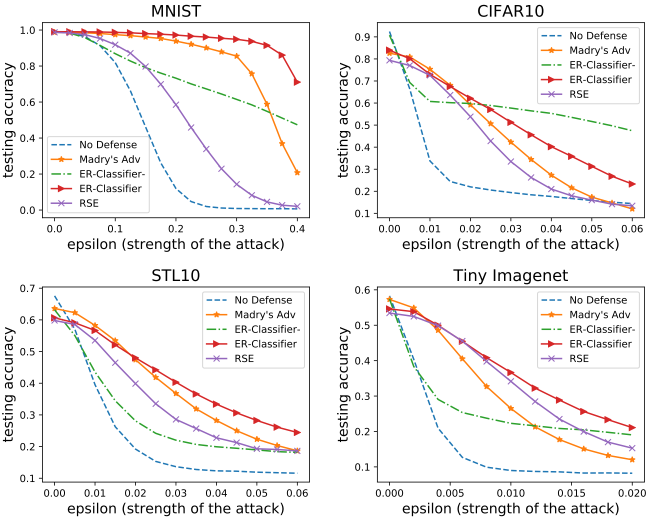


Figure: Testing accuracy under ℓ_∞-PGD attack on four different datasets.

Evaluate Discriminator

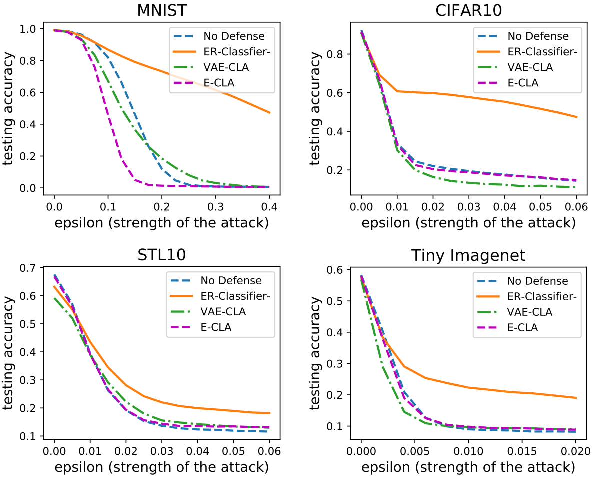


Figure: Testing accuracy of E-CLA, VAE-CLA and ER-Classsifier^{-} under ℓ_{∞}-PGD attack on four different datasets: MNIST, CIFAR10, STL10 and Tiny Imagenet.

Embedding Visualization

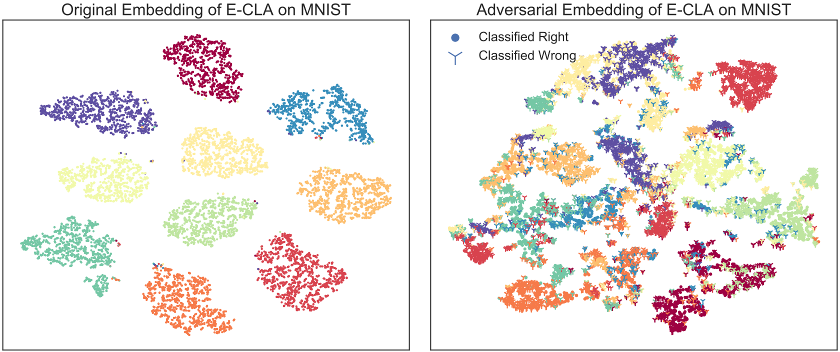


Figure: 2D embeddings for E-CLA on MNIST.

Embedding Visualization

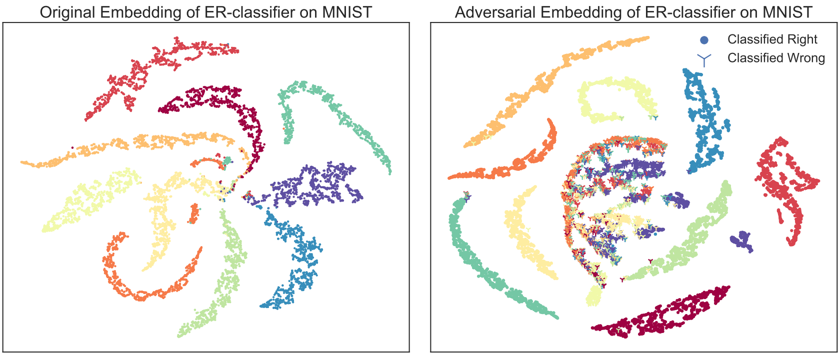


Figure: 2D embeddings for ER-Classifier on MNIST.

Dimension Selection

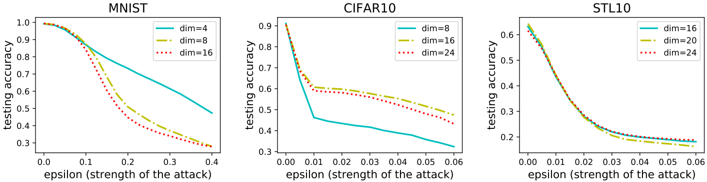


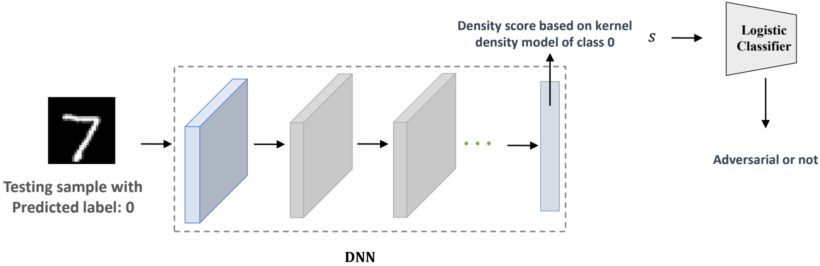
Figure: Testing accuracy of models with different embedding dimensions under ℓ_∞-PGD attack.

Data	Data dim.	Estimated Intrinsic dim.	Embedding dim.
MNIST	$1 \times 28 \times 28$	13	4
CIFAR10	$3 \times 32 \times 32$	17	16
STL10	$3 \times 96 \times 96$	20	16
Tiny Imagenet	$3 \times 64 \times 64$	19	20

Table: Pixel space dimension, intrinsic dimension estimated by the method proposed by Levina et al., and final embedding dimension used.

Detection

KD adversarial detection



- **KD Adversarial Detection:** Feinman et al. proposed to detect adversarial examples by performing Kernel Density (KD) estimation in the subspace of deep classifier.

Feinman et al. *Detecting Adversarial Samples from Artifacts.* ICML, 2017.

Bayesian Adversarial Detector (BATer)

Observation 1

Tanay et al., 2016: Adversarial samples do not lie on the data manifold.

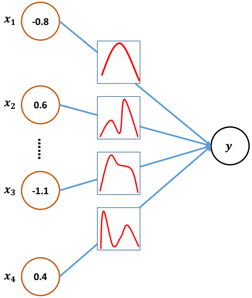
Observation 2

A randomized network can lead to a distribution of hidden features, making it easier for detecting an out-of-manifold example.

- Our approach: detecting adversarial examples with Bayesian neural network
  - Bayesian neural network
  - Hidden output distributional differences

Li et al. *Detecting Adversarial Examples with Bayesian Neural Network.* 2021.

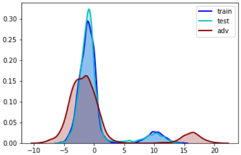
Bayesian Neural Network (BNN)



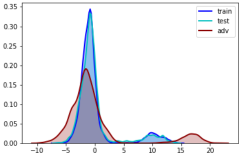
- All weights are represented by probability distributions over possible values
- BNN: a probabilistic model $p(y|\boldsymbol{x}, \boldsymbol{w})$

(Blundell et al., 2015)

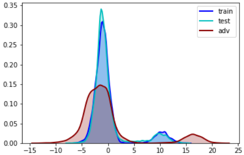
Why BNN not DNN?



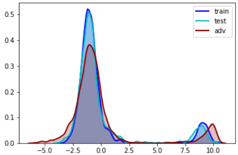
(a) BNN: Layer 23



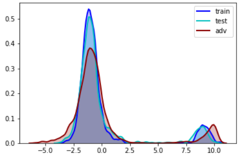
(b) BNN: Layer 33



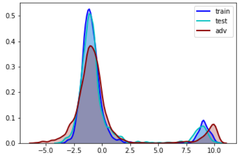
(c) BNN: Layer 43



(d) DNN: Layer 23



(e) DNN: Layer 33



(f) DNN: Layer 43

Figure: Hidden Layer Output Distributions (HODs) of BNN on automobile class of CIFAR10. train: HODs of training examples from automobile class. test: HODs of test examples from automobile class. adv: HODs of adversarial examples predicted as automobile. The adversarial examples are generated by PGD.

BATer: Framework

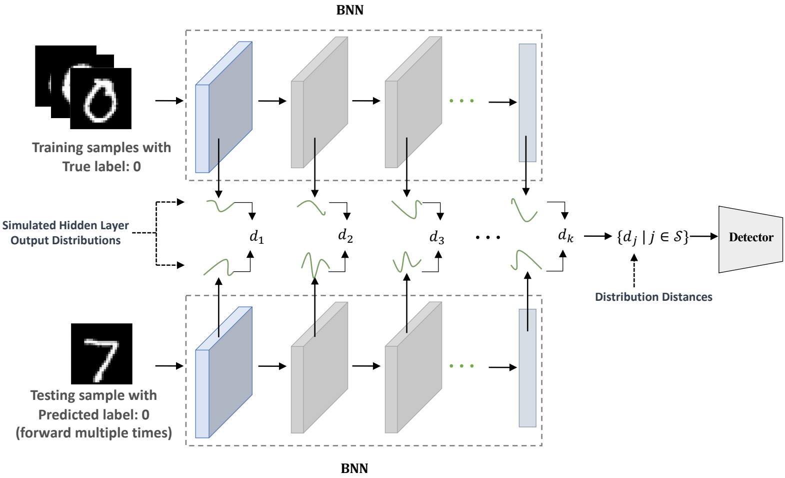


Figure: Framework of BATER. k is the total number of hidden layers. d_j represents the distribution distance measured from j-th hidden layer. $\mathcal{S}$ is the index set of selected hidden layers.

BATer: Methods Compared

1. **KD**: R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner. *Detecting adversarial samples from artifacts.* ICML, 2017
2. **LID**: X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, G. Schoenebeck, D. Song, M. E. Houle, andJ. Bailey. *Characterizing adversarial subspaces using local intrinsic dimensionality.* ICLR, 2018.
3. **ODD**: K. Roth, Y. Kilcher, and T. Hofmann. *The odds are odd: A statistical test for detecting adversarial examples.* ICML, 2019.
4. **ReBeL**: Raghuram, Jayaram, et al. *A General Framework For Detecting Anomalous Inputs to DNN Classifiers.* ICML, 2021.
5. **BATer**: Our proposed method

BATer: Comparison With Other Methods

Data	Metric	C&W					FGSM					PGD				
		KD	LID	ODD	ReBeL	BATER	KD	LID	ODD	ReBeL	BATER	KD	LID	ODD	ReBeL	BATER
CIFAR10	AUC	0.945	0.947	0.955	0.968	**0.980**	0.873	0.957	0.968	0.990	**0.995**	0.791	0.777	0.963	0.962	**0.971**
	TPR(FPR@0.01)	0.068	0.220	0.591	0.309	**0.606**	0.136	0.385	0.224	0.698	**0.878**	0.018	0.093	0.059	0.191	**0.813**
	TPR(FPR@0.05)	0.464	0.668	0.839	0.726	**0.881**	0.401	0.753	0.709	0.974	**0.991**	0.148	0.317	0.819	0.789	**0.881**
	TPR(FPR@0.10)	0.911	0.856	0.901	0.954	**0.965**	0.572	0.875	**1.000**	**1.000**	0.998	0.285	0.448	**0.999**	**0.999**	0.917
MNIST	AUC	0.932	0.785	0.968	0.980	**0.999**	0.933	0.888	0.952	0.992	**0.999**	0.801	0.861	0.967	0.975	**0.989**
	TPR(FPR@0.01)	0.196	0.079	0.212	0.630	**0.974**	0.421	0.152	0.898	0.885	**0.972**	0.062	0.170	0.607	0.382	**0.733**
	TPR(FPR@0.05)	0.616	0.263	0.911	0.900	**0.997**	0.692	0.503	0.908	0.990	**0.998**	0.275	0.396	0.934	0.851	**0.957**
	TPR(FPR@0.10)	0.818	0.397	**1.000**	0.972	**1.000**	0.796	0.678	0.917	**1.000**	**1.000**	0.429	0.552	0.945	0.956	**0.999**
Imagenet -sub	AUC	0.811	0.905	0.886	0.834	**0.941**	0.914	0.983	0.844	0.842	**0.989**	0.989	**0.991**	0.777	0.824	0.976
	TPR(FPR@0.01)	0.193	**0.401**	0.185	0.035	0.146	0.460	**0.772**	0.042	0.045	0.569	**0.930**	0.829	0.010	0.028	0.729
	TPR(FPR@0.05)	0.452	**0.653**	0.398	0.167	0.538	0.727	0.952	0.188	0.197	**0.989**	**0.966**	0.961	0.054	0.139	0.904
	TPR(FPR@0.10)	0.584	0.754	0.566	0.312	**0.815**	0.822	0.987	0.364	0.358	**1.000**	0.979	**0.984**	0.121	0.280	0.947

Table: Performance of detection methods against adversarial attacks.

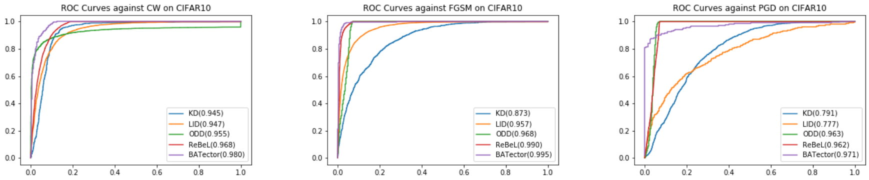
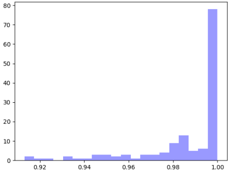


Figure: ROC Curves on CIFAR10. (See more figures in our paper)

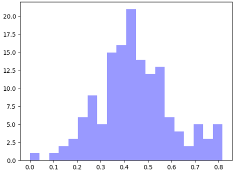
Ablation Study: BNN vs. DNN

Class	CIFAR10		MNIST	
	BNN	DNN	BNN	DNN
class1	0.978	0.489	0.929	0.901
class2	0.972	0.410	1.000	0.967
class3	0.973	0.501	0.993	0.892
class4	0.994	0.594	0.991	0.958
class5	0.955	0.477	1.000	0.883
class6	0.995	0.729	0.999	0.937
class7	0.976	0.584	0.989	0.878
class8	0.973	0.537	1.000	0.941
class9	0.915	0.493	0.959	0.874
lass10	0.949	0.567	0.982	0.917

Table: AUC of BATER with different structures (BNN vs. DNN) on CIFAR10 and MNIST of different classes.



(a) AUC of BNN



(b) AUC of DNN

Figure: AUC Histograms of BATER with different structures (BNN vs. DNN) on Imagenet-sub.

High Confidence Attack

- Athalye et al.: detection methods can fail when the confidence level of adversarial examples generated by C&W attack increases.

Data	Metric	C&W (Confidence)		
		0	10	20
CIFAR10	AUC	0.980	0.999	0.995
	TPR(FPR@0.01)	0.606	0.998	0.939
	TPR(FPR@0.05)	0.881	1.000	0.995
	TPR(FPR@0.10)	0.965	1.000	0.995
MNIST	AUC	0.999	0.995	0.995
	TPR(FPR@0.01)	0.974	0.913	0.919
	TPR(FPR@0.05)	0.997	0.993	0.994
	TPR(FPR@0.10)	1.000	0.998	0.999
Imagenet -sub	AUC	0.928	0.991	0.983
	TPR(FPR@0.01)	0.146	0.896	0.642
	TPR(FPR@0.05)	0.538	0.951	0.910
	TPR(FPR@0.10)	0.815	0.977	0.964

Table: Performance of BATER against high-confidence C&W.

Conclusions

- Different types of adversarial defense

Questions?