Controlling Over-generalization and its Effects on Adversarial Examples Generation and Detection

Mahdieh Abbasi[1], Arezoo Rajabi[2], Azadeh S. Mozafari[1], Rakesh B. Bobba[2], Christian Gagné[1]

1. Université Laval, 2. Oregon State University

1 PROBLEM: INSECURITY of CNN

1.1 Adversarial examples

Adding **small** but **smart** perturbations to an input image generates another image, called *adversarial examples*, that is perceptually similar to the original image.

CNNs **confidently misclassifies** such benign-looking adversaries.

In hostile situations, the CNN-based systems can be managed to break silently (without any visual clues) by adversarial examples

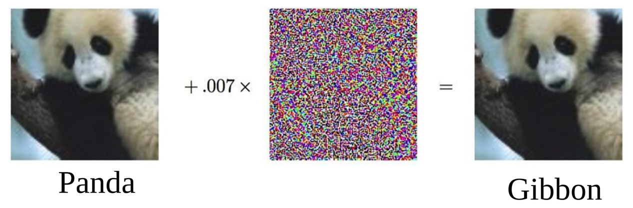


Figure 1: An adversarial example generated by Fast Gradient Sign (FGS) [Goodfellow et al. 2014].

1.2 Out-distribution samples

Despite of notable performance of CNNs on task-related samples (i.e. in-distribution), they **classify confidently** out-distribution samples into the task-related classes, instead of classifying them with low confidence.

For example, CNN trained for recognizing hand-written digits (from MNIST set) misclassifies printed letters (from NotMNIST set) as digits with high confidence.

In regular situations, confident wrong decisions made by naïve CNN in the presence of out-distribution samples can lead to some life catastrophes

Figure 2: NotMNIST (first row) and CIFAR-10 (second row) are confidently misclassified as digits by a CNN trained on MNIST.

2 CONTRIBUTIONS

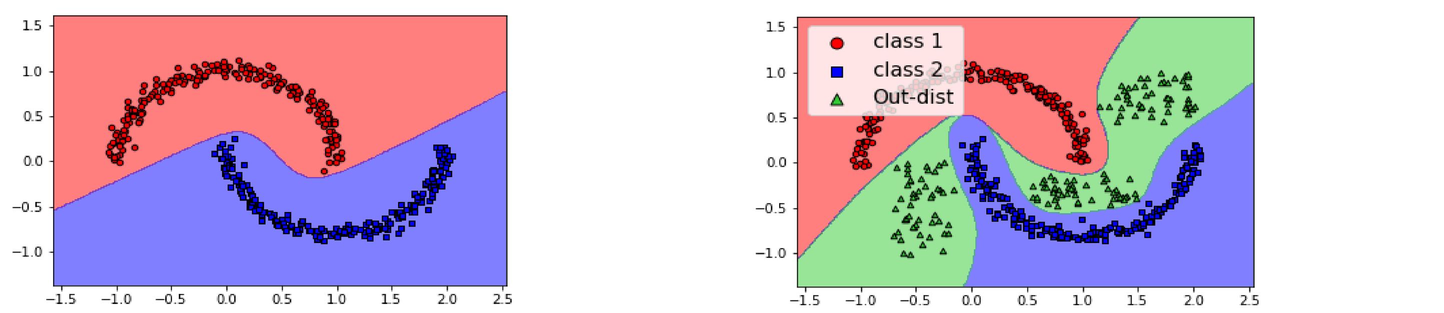
1) Drawing a relationship between these two unrelated issues (i.e. lack of robustness to adversaries and lack of suitable predictions on out-distribution samples) through **over-generalization.**

2) Effectively controlling over-generalization in input space by **our simple yet effective approach, i.e. augmented CNN,** leads to a significant drop in misclassification rates on both black-box adversaries and a wide range of out-distribution (unseen) sets.

3) Without training the augmented CNNs an any **adversaries**, generation of white-box attacks (adversaries) using augmented CNNs can become harder.

3 OVER-GENERALIZATION

A plain neural network divides an input space entirely to some pre-defined classes, while in-distribution samples occupy a small portion of this space Thus, **over-generalization** happens in the vacant regions that are empty of in-distribution samples.



a) Extension of decision boundary to the vacant regions by a naïve MLP, makes it very vulnerable to adversaries and out-distribution samples .

b) Reduction of over-generalization in such vacant regions by training the MLP on **some appropriate** out-distribution samples.

Figure 3: Illustration of over-generalization

4 PROPOSED METHOD

To alleviate **over-generalization**, we propose to augment CNN's output with an extra class (a.k.a dustbin class) to allow the samples from out of the learned concepts (i.e. out-distribution) classified as "dustbin".

4.1. On selection of training samples for dustbin class

A computationally simple yet effective way for acquiring dustbin class training samples.

Interpolated set

Why: an adversarial example happens near (on the margin) of the decision boundaries that separate two classes. By some interpolated samples, we aim at tightening the decision boundaries.

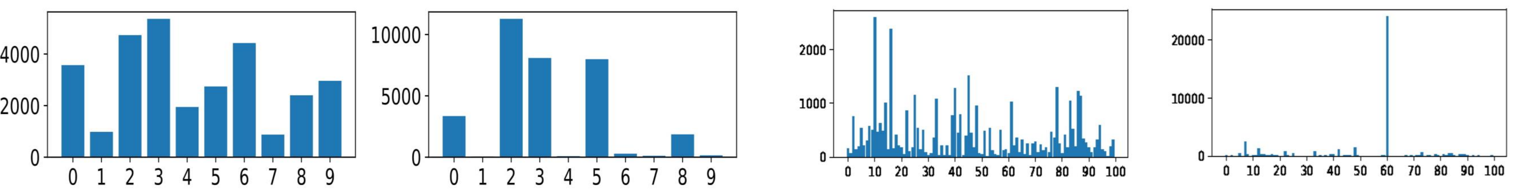
How to generate: a sample $\boldsymbol{x}$ and its nearest neighbor in the feature space from a different class $\boldsymbol{x}'$ are linearly combined in input space as follows $\boldsymbol{x}'' = \alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{x}'$.

Figure 4: Interpolated data for MNIST (left) and CIFAR-10 (right)

Natural out-distribution dataset

Why: without computational overhead for generating synthetic samples for dustbin class, taking advantage from available training samples from other task-irrelevant datasets.

How to select an out-distribution set: the more uniform misclassification of out-distribution samples over the classes of the in-distribution set, the more appropriate they are as dustbin class for training the augmented CNNs.

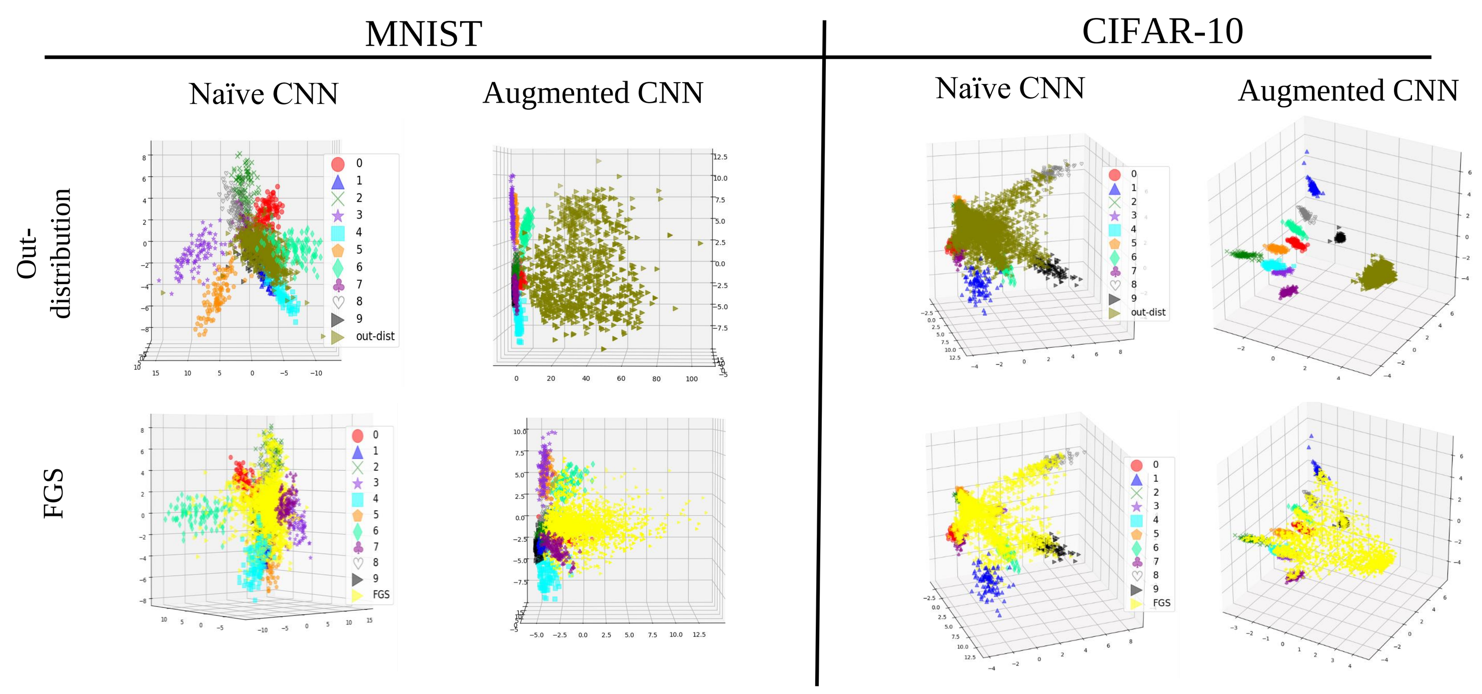


a) Two out-distribution sets for **CIFAR-10**: CIFAR-100 (left) vs SVHN (right)

b) Two out-distribution sets for **CIFAR-100**: TinyImageNet (left) vs LSUN (right)

5 THE FEATURE SPACE

Adversarial examples are automatically disentangled from in-distribution samples in the feature space (the last convolutional layer) of augmented CNNs, **while they never trained on any adversaries.**



6 EVALUATION

6.1. Black-box adversaries

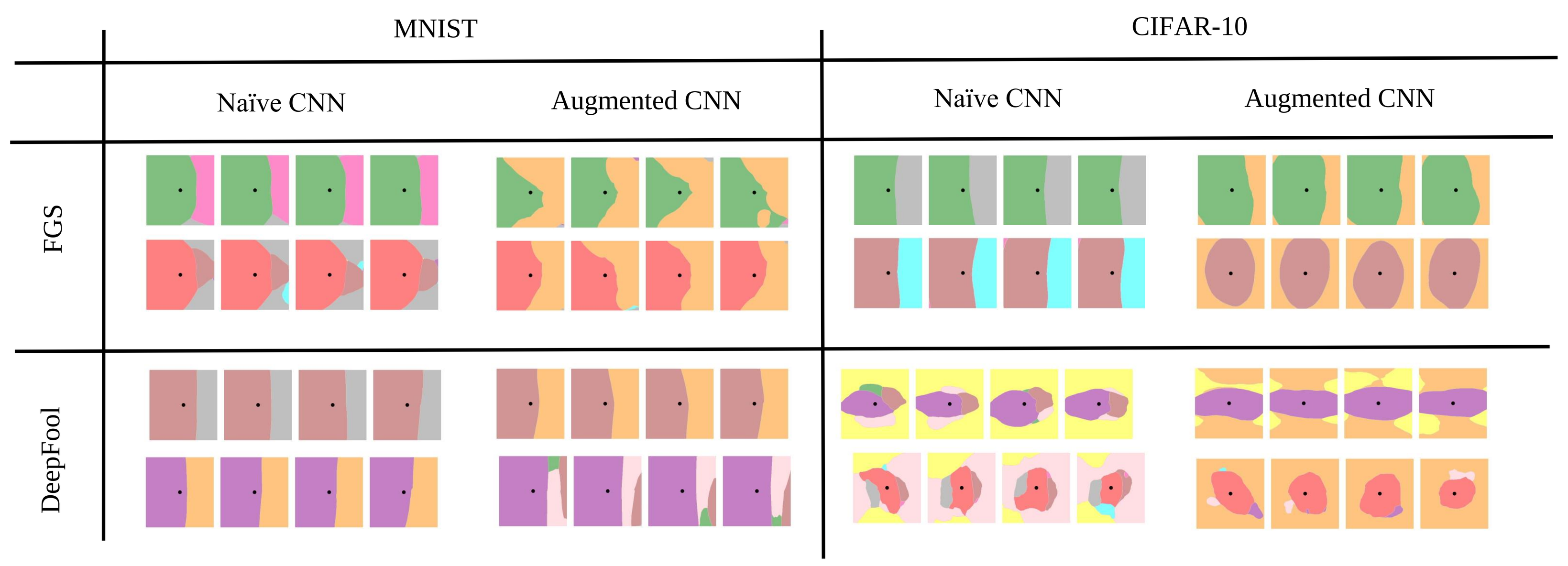
While the accuracy on in-distribution sets are maintained by augmented CNNs, their misclassification (error) rates on different types of strong adversaries are reduced noticeably.

Note augmented CNNs attempts to classify some of these black-box adversaries as dustbin (i.e. equal as rejection) meanwhile correctly classify the remaining that can not fool (transferred) to the augmented CNNs.

		MNIST (in-dist) / NotMNIST (out-dist)		CIFAR-10 (in-dist) / CIFAR-100 (out-dist)		CIFAR-100 (in-dist) / ImageNet (out-dist)	
		Naive AlexNet	Augm. AlexNet	Naive VGG	Augm. VGG	Naive Resnet164	Augm. Resnet164
In-dist. test	Acc.	99.50	99.48	90.50	86.65	75.52	73.37
	Rej.	—	0.08	—	8.47	—	5.02
	Err.	0.50	**0.44**	9.47	**4.61**	24.48	**21.61**
FGS	Acc	35.14	0.35	36.16	29.50	67.67	50.03
	Rej	–	99.59	—	45.11	—	36.87
	Err	65.86	**0.07**	63.84	**25.39**	32.33	**13.10**
I-FGS	Acc	25.90	0.01	51.19	50.28	22.20	16.80
	Rej	—	99.90	—	24.76	—	45.75
	Err	74.10	**0.09**	48.81	**24.96**	77.80	**37.45**
T-FGS	Acc	19.99	0.00	36.24	24.35	59.93	37.07
	Rej	—	100	—	51.33	–	46.87
	Err	80.01	**0.0**	63.76	**24.32**	40.07	**16.06**
DeepFool	Acc	1.89	5.36	56.82	42.81	77.20	66.27
	Rej	—	89.84	—	40.26	—	15.33
	Err	98.11	**4.80**	43.18	**16.93**	22.80	**18.41**
C&W (L_2)	Acc	22.49	7.50	42.50	39.00	74.50	60.50
	Rej	—	77.49	—	39.50	—	25.50
	Err	77.51	**15.01**	57.50	**21.50**	25.50	**14.00**

Table 1 :Comparison of the augmented CNNs with their naïve counterparts on clean samples (i.e. in-distribution) and various types of strong adversarial examples.

6.2. Moving in adversarial directions



6.3. Unseen out-distribution sets

While the augmented CNNs are trained on a single out-distribution set, they are able to reject a wide-range of **unseen** out-distribution sets.

In-distribution train	Out-distribution test	Naive model Error (%)	Augmented model Error (%)	Augmented model Rejection (%)
MNIST	NotMNIST (seen)	93.15	0.01	99.98
	Omniglot (unseen)	95.19	0.00	100
	CIFAR-10(gc) (unseen)	64.26	0.00	100
CIFAR-10	CIFAR-100* (seen)	97.05	3.71	96.21
	ImageNet* (unseen)	96.62	12.20	87.49
	SVHN* (unseen)	95.56	7.61	92.29
	LSUN* (unseen)	96.12	14.31	84.80
CIFAR-100	ImageNet* (seen)	79.34	1.52	98.35
	SVHN (unseen)	81.19	67.75	16.25
	LSUN* (unseen)	96.12	0.01	99.99

Table 2: Error rate of naïve models with their augmented counterparts on a wide-range of out-distribution sets.

6.4. White-box adversarial examples

The probability of visiting dustbin regions are higher than other fooling classes when one tries to generate adaptive adversaries using the augmented CNNs.

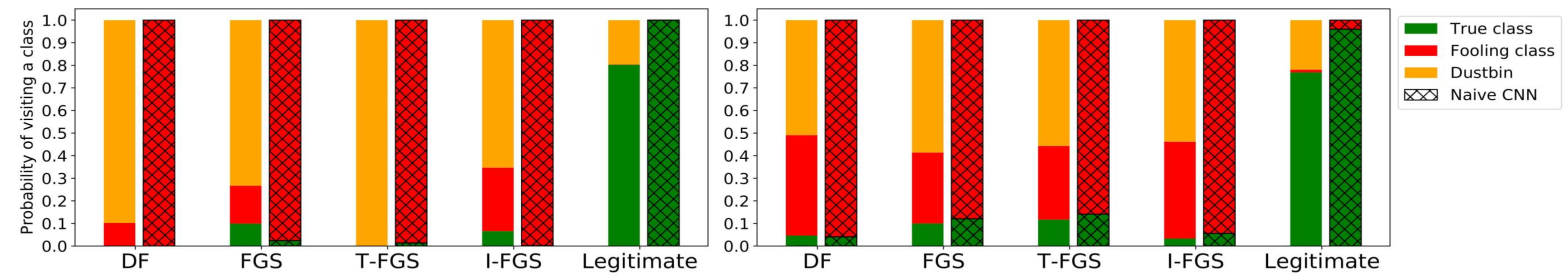


Figure 6: Generating various types of adversaries for MNIST (left) and CIFAR-10 (right) using augmented CNNs and its naïve counterpart.