Exploring Adversarial Training for Out-of-Distribution Detection

Irena Gao, David Yue, and Ryan Han
{igao, davidyue, ryanhan}@stanford.edu

Problem

Classifiers behave unreliably when test-data is drawn from a different distribution than train-data, *e.g.* an example from an unknown class or an adversarial attack.

We need to detect these *out-of-distribution* examples.

Problem: Not as easy as supervised learning, because classifier does not know *a priori* what out-of-distribution examples it will encounter. Collecting an auxiliary outlier dataset is also expensive.

Proposal

The Mahalanobis Method (mostly unsupervised approach)**:** Model in-data as class-conditional Gaussians in feature space. Classify a new example based on its Mahalanobis distance to the nearest Gaussian. See equation (1) and citations.

Problem: Supervised learning is still required to weight distances in different feature spaces (*i.e.* distances calculated on different layers of a neural network), and to set the binary decision threshold.

Proposal: In any supervised learning, train on adversarially-attacked in-data, rather than training on an auxiliary outlier dataset.

Hypothesis.
Adversarial examples lie close to --- but just off of --- the data manifold, while outlier datasets lie far from the manifold (Figure 1). Training on adversarial examples should perform as well as, if not better, than training on outlier datasets.

Significance.
If this hypothesis holds, OOD detection can be done without requiring additional outlier data.

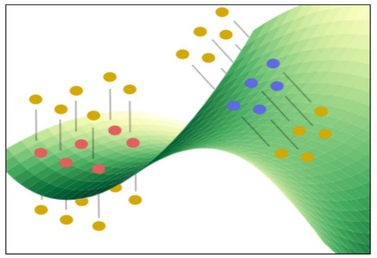

Figure 1: The Data Manifold hypothesis suggests that in-data lies on a low-dimensional surface in high-dimensional feature space. Classes live in clusters separated by low-density regions. OOD examples lie off the manifold.

Method

1. Divide ResNet-34 into five blocks. Compute Mahalanobis scores in output space of each block.
   Note: Σ, μ are Gaussian parameters of the in-dataset at block ℓ.

$$M_\ell(x) = \max_c -(f_\ell(x) - \mu_\ell^c)^T \Sigma_\ell^{c-1} (f_\ell(x) - \mu_\ell^c)$$

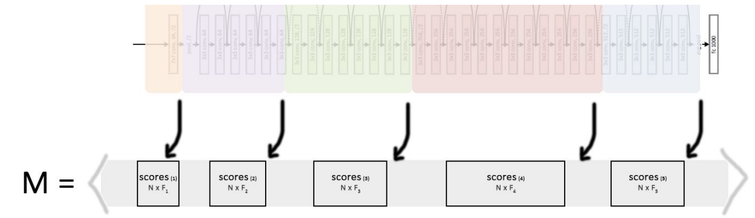

Figure 2: Method schematic. The original Mahalanobis Method is used to compute tensor M, which contains Mahalanobis scores for each test example at five positions. This is fed into a logistic regression, which weights the block scores and sets a decision threshold.

2. Use tensor M as input to a logistic regression.
3. **Experiment with the data used to train the logistic regression.**

Training Data Composition	
in-dataset (fixed)	LABEL: 1 (in)
out-dataset (experimental variable) OPTION A: use in-dataset but perturbed with an adversarial attack OPTION B: use a large outlier dataset (control)	LABEL: 0 (out)

Datasets and Attacks

Outlier Datasets	
SVHN	32-by-32 color images of street view house numbers split into 73,257 training digits and 26,032 test digits.
CIFAR-10	32-by-32 color images in 10 classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images.
TinyImageNet	64-by-64 color images in 200 classes, with 500 training images per class. There are 100,000 training images and 10,000 test images.
LSUN	32-by-32 color images in 10 scene classes. There around 1 million labeled images.

Adversarial Attacks	
Fast Gradient Sign Method (FGSM)	$x^A = x + \epsilon \operatorname{sign}(\nabla J(x))$
Basic Iterative Method (BIM)	$x^A := \operatorname{clip}_\epsilon[x^A + \alpha \operatorname{sign}(\nabla J(x^A))]$
Carlini Wagner L2 Attack (CWL-2)	$\text{minimize} \ \ \|\frac{1}{2}(\tanh(w)+1) - x\|_2^2 + c \cdot f(\frac{1}{2}(\tanh(w)+1)$
DeepFool Attack	$\arg\min_{\boldsymbol{r}} \|\boldsymbol{r}\|_2$ s.t. $\exists k : \boldsymbol{w}_k^\top(\boldsymbol{x}_0+\boldsymbol{r}) + b_k \geq \boldsymbol{w}_{\hat{k}(\boldsymbol{x}_0)}^\top(\boldsymbol{x}_0+\boldsymbol{r}) + b_{\hat{k}(\boldsymbol{x}_0)}$,

Results

In-Distribution	Training Distribution	Test Distribution	Validation on Test Distribution	
			TNR @ 90% TPR	AUROC
CIFAR-10	FGSM	Adversarial Attacks	0.4077	0.6978
		Out Datasets	0.4801	0.6161
	BIM	Adversarial Attacks	0.3645	0.7122
		Out Datasets	0.2987	0.4485
	DeepFool	Adversarial Attacks	0.4008	**0.7807**
		Out Datasets	0.6253	**0.8051**
	CWL-2	Adversarial Attacks	0.3409	0.7167
		Out Datasets	0.3552	0.5808
	TinyImageNet	Adversarial Attacks	**0.4210**	0.7332
		Out Datasets	**0.6407**	0.7246
SVHN	**FGSM**	Adversarial Attacks	0.5856	0.8461
		Out Datasets	**0.9272**	**0.9520**
	BIM	Adversarial Attacks	0.6823	0.8758
		Out Datasets	0.2014	0.2729
	DeepFool	Adversarial Attacks	**0.7436**	**0.9095**
		Out Datasets	0.4405	0.5186
	CWL-2	Adversarial Attacks	**0.7429**	**0.9070**
		Out Datasets	0.3250	0.3876
	TinyImageNet	Adversarial Attacks	0.6206	0.8622
		Out Datasets	**0.9957**	**0.9957**

Figure 3: Performances for ten {in-dataset, train-dataset} setups on two tasks: detecting adversarial attacks (FGSM, BIM, DeepFool, CWL-2) and detecting outlier datasets (ImageNet, LSUN, CIFAR-10, SVHN). Best performances are bolded. Training on FGSM and ImageNet perform consistently well on both tasks, while BIM, DeepFool, and CWL-2 perform much better at detecting adversarial examples than detecting outlier examples.

- Training on FGSM and ImageNet give stable performances on both tasks. Training on BIM, DeepFool, and CWL-2 better detect adversarial attacks than outlier datasets.
- DeepFool and CWL-2 detect adversarial attacks well.
- FGSM and ImageNet have similar accuracies.

Conclusions

New Hypothesis.
BIM, CWL-2, and DeepFool perturb data along the same axes, giving an incomplete picture of the data manifold. Attacks like FGSM disperse data along more axes, giving a better picture of the manifold boundary. When the axis conditions are met, adversarial training can replace out-distributions.

Selected Work:

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In Advances in Neural Information Processing Systems, pages 7167–7177, 2018.