Deep Anomaly Detection with Outlier Exposure

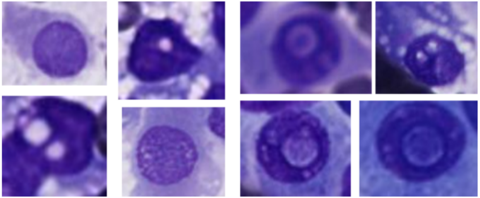
Hendrycks et al, 2019

Jean-Michel Begon

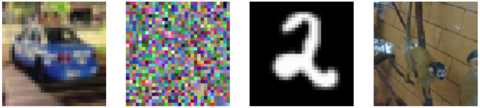
April 2nd, 2020

Motivation

Suppose you have a model to detect cancer cells (presence/absence) which was trained on data like :



What would happen with these inputs ?



Motivation

What would happen ?

- The model would make a prediction.
  - What if it predicts cancer ?
  - Might not be handy to double-check every prediction...

Other examples

- Same application, samples from different tissues
  - what if there are cancer cells, which are missed ?
- Autonomous driving
  - mistake something for a road sign and cause car crash.
- Biometric authentication, etc.

Such irrelevant samples are called out-of-distribution (OOD) samples. The goal is to detect them.

Out-of-distribution in the real world

Why would anyone feed the model with such erroneous inputs ?

- Malicious intent ;
- operator mistake ;
- error from an other (dispatching) model ;
- acquisition error ;
- variance in the pre-processing steps (*cf.* medical domain) ;
- bad lighting condition ;
- faulty sensor ;
- etc.

Actually, what is an OOD sample ? What is OODness ?

Outline

OODness

Informally

An out-of-distribution sample is a data point which is fed to a model but which does not belong to the originally-targeted distribution [1] $\mathcal{D}_{in}$ (called the in-distribution, ID).

Formally

???

1. I follow the notations of the paper, which are not ideal to distinguish between the joint probability distribution (samples and labels), the marginal probability distribution (samples whatever the labels) and the conditional probability distributions (samples knowing the label, labels knowing the sample).

OODness—comments

Dependence on the Model

The authors are focusing on detecting out-of-distribution samples via the network.

- Rationale : the model is supposed to have captured some information about the learning distribution ;
- also, training data might no longer be available.

The notion of membership

Binary classification problem

- In supervised learning, assign the class with highest density at x.
- Only one class, ill-posed problem (in the support ? Sufficient density—provided it exists ? Part of the minimum volume which encompassed most of the mass ?)

OODness—in practice

Practical (but arguable) solution regarding the membership :

- Other label space (*i.e.* other dataset), other distribution.

Evaluation protocol

1. Learn a model f as usual on the base task with $LS \sim \mathcal{D}_{in}^{n}$;
2. Derive some OOD detection $g(\cdot; f)$
   - Typically, g outputs a probability of a sample being OOD.
3. Test OOD prediction on TS_{ood}
   - $TS_{ood} = TS_{in} \cup TS_{out}$;
   - $TS_{in} \sim \mathcal{D}_{in}^{n_1}$, $TS_{in} \cap LS = \emptyset$;
   - $TS_{out} \sim \mathcal{D}_{out}^{n_2}$;
   - $\mathcal{D}_{in} \neq \mathcal{D}_{out}$.

Evaluation metrics are those of binary classification problems (accuracy, confusion matrix, area under the ROC curve, area under the precision-recall curve, etc.)

Baseline

Hendrycks & Gimpel (2016) proposed to use the maximum softmax probability MSP of a prediction to determine whether a sample x is in- or out-of-distribution :

$$z(x;\theta) = \begin{bmatrix} z_1 \\ \vdots \\ z_K \end{bmatrix} = f(x;\theta) \tag{1}$$

$$\hat{p}_k(x;\theta) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} \tag{2}$$

$$\text{MSP} = \max_k \hat{p}_k \tag{3}$$

where $f(\textbf{.};\theta)$ is a neural network parametrized by θ, and z is the logit vector. The dependence on x and θ is omitted when obvious.

Baseline

Soft OOD detection

$$g(x; f(\cdot; \theta)) = 1 - \max_k \hat{p}_k(x; \theta) \tag{4}$$

Hard OOD detection

$$g_{\text{hard}}(x; f(\cdot; \theta)) = \begin{cases} 0, & \text{if } 1 - \max_k \hat{p}_k(x; \theta) \leq t \\ 1, & \text{otherwise} \end{cases} \tag{5}$$

Rationale

A network should be confident of itself for ID samples, and not so confident on OOD's.

Related problems

The fuzziness shrouding OOD detection makes it close to other problems, mainly adversarial sample detection and misclassification prediction.

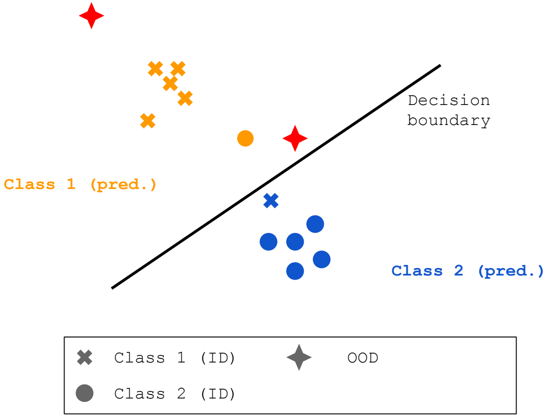
Also, depending on the data available and on how g is derived from f, OOD detection can fall into one or several categories :

- Novelty detection ;
- density estimation ;
- minimum volume set selection ;
- purely-supervised learning ;
- samplefree rejection ;
- etc.

Outline

The problem with the baseline

What happens in the pre-logit space



$$z(x;\theta) = W\,h(x;\theta') + b \tag{6}$$

where h is the pre-linear, feature extraction part of the network and $\theta = [\theta', W, b]$

Outlier exposure (OE)

Let $\mathcal{D}_{in}$ be the (joint) "in" distribution, and let $\mathcal{D}_{out}^{OE}$ and $\mathcal{D}_{out}^{test}$ be disjoint out-of-distributions.

Outlier exposure is a technique to improve OOD detection based on a model f :

1. Learn a model $f(\cdot;\theta^*)$ for the task on training data from $\mathcal{D}_{in}$;
   - typically by minimizing some loss $\mathcal{L}(f(x;\theta), y)$ by gradient descent for many epochs ;
2. fine-tune f for OOD detection by adding a second term in $\mathcal{L}_{OE}$ to the loss function ;
   - use additional data from $\mathcal{D}_{out}^{OE}$ and optimize for a few epochs ;
3. use f for OOD detection as usual ;
   - test the method on $\mathcal{D}_{out}^{test}$.

This begs the question of the relevance of $\mathcal{D}_{out}^{OE}$ for detecting $\mathcal{D}_{out}^{test}$ samples.

Outlier exposure

General form of fine-tuning objective

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}_{in}} \left[\mathcal{L}\left(f(x;\theta), y\right) + \lambda \mathbb{E}_{x'\sim\mathcal{D}_{out}^{OE}} \left[\mathcal{L}_{OE}\left(f(x'), f(x), y\right) \right] \right] \quad (7)$$

where $\mathcal{L}$ is the loss for the main task (*e.g.* cross-entropy for classification), λ is a hyper-parameter weighing the two components of the loss, and $\mathcal{L}_{OE}$ is a loss help detect OOD samples.

The form of $\mathcal{L}_{OE}$ is dependent on both the main task (*e.g.* classification) and how OOD detection is usually carried out.

Outlier exposure—classification

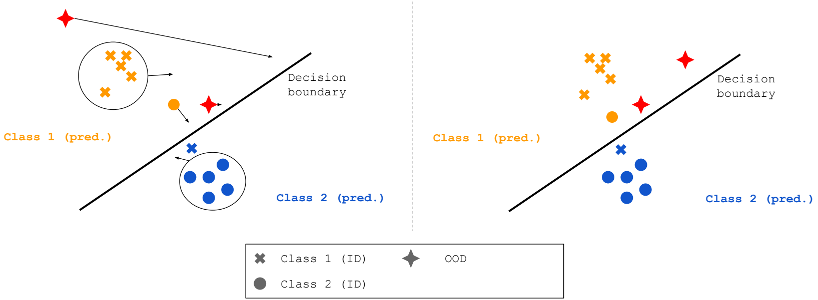
In classification, with MSP as metric to detect OOD samples, the fine-tuning becomes[2] :

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}_{in}} \left[-\log \hat{p}_y(x,\theta)\right] + \lambda \mathbb{E}_{x'\sim\mathcal{D}_{out}^{OE}} \left[-\sum_{k=1}^{K} \frac{1}{K} \log \hat{p}_k(x',\theta)\right] \tag{8}$$

Forcing the predictions of OOD samples to look more like a uniform distribution enhances the reliability of MSP for detection.

2. The actual implementation is a bit more involved for numerical reasons.

Outlier exposure—geometrical intuition



Outline

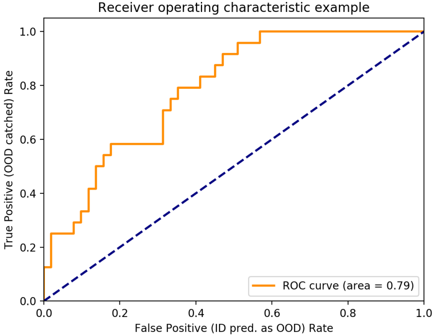
Reminder on binary assessment metrics

Truth	Prediction OOD	Prediction ID	
OOD	TP (catched)	FN (missed)	P
ID	FP (confused)	TN (identified)	N

$$FPR = \frac{FP}{FP + TN} \qquad (9)$$

$$TPR = \frac{TP}{TP + FN} \qquad (10)$$

$FPR95$ is the ratio of ID samples confused for OOD samples when 95% of the OOD are correctly catched.



adapted from `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html`

Raw classification

$\mathcal{D}_{\text{in}}$	$\mathcal{D}_{\text{out}}^{\text{test}}$	FPR95 ↓ MSP	FPR95 ↓ +OE	AUROC ↑ MSP	AUROC ↑ +OE	AUPR ↑ MSP	AUPR ↑ +OE
SVHN	Gaussian	5.4	0.0	98.2	100.	90.5	100.
	Bernoulli	4.4	0.0	98.6	100.	91.9	100.
	Blobs	3.7	0.0	98.9	100.	93.5	100.
	Icons-50	11.4	0.3	96.4	99.8	87.2	99.2
	Textures	7.2	0.2	97.5	100.	90.9	99.7
	Places365	5.6	0.1	98.1	100.	92.5	99.9
	LSUN	6.4	0.1	97.8	100.	91.0	99.9
	CIFAR-10	6.0	0.1	98.0	100.	91.2	99.9
	Chars74K	6.4	0.1	97.9	100.	91.5	99.9
	Mean	6.28	**0.07**	97.95	**99.96**	91.12	**99.85**
CIFAR-10	Gaussian	14.4	0.7	94.7	99.6	70.0	94.3
	Rademacher	47.6	0.5	79.9	99.8	32.3	97.4
	Blobs	16.2	0.6	94.5	99.8	73.7	98.9
	Textures	42.8	12.2	88.4	97.7	58.4	91.0
	SVHN	28.8	4.8	91.8	98.4	66.9	89.4
	Places365	47.5	17.3	87.8	96.2	57.5	87.3
	LSUN	38.7	12.1	89.1	97.6	58.6	89.4
	CIFAR-100	43.5	28.0	87.9	93.3	55.8	76.2
	Mean	34.94	**9.50**	89.27	**97.81**	59.16	**90.48**
CIFAR-100	Gaussian	54.3	12.1	64.7	95.7	19.7	71.1
	Rademacher	39.0	17.1	79.4	93.0	30.1	56.9
	Blobs	58.0	12.1	75.3	97.2	29.7	86.2
	Textures	71.5	54.4	73.8	84.8	33.3	56.3
	SVHN	69.3	42.9	71.4	86.9	30.7	52.9
	Places365	70.4	49.8	74.2	86.5	33.8	57.9
	LSUN	74.0	57.5	70.7	83.4	28.8	51.4
	CIFAR-10	64.9	62.1	75.4	75.7	34.3	32.6
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Subset of vision OOD example detection for the maximum softmax probability (MSP) with and without OE. $\mathcal{D}_{out}^{OE}$ is 80M Tiny Images. All results are percentages and the result of 10 runs (Hendrycks et al, 2019).

Raw classification without MSP

Instead of using MSP, we can use the following metric to detect OOD samples :

$$\mathrm{uce}(x;\theta) = -H\left(\mathcal{U}; \hat{p}(x;\theta)\right) = \sum_{k=1}^{K} \frac{1}{K} \log \hat{p}_k(x;\theta) \tag{11}$$

- use more information to discriminate ;
- closer to what OE encourages ;

$\mathcal{D}_{\text{in}}$	FPR95 ↓ MSP	FPR95 ↓ $H(\mathcal{U};p)$	AUROC ↑ MSP	AUROC ↑ $H(\mathcal{U};p)$	AUPR ↑ MSP	AUPR ↑ $H(\mathcal{U};p)$
CIFAR-10	9.50	9.04	97.81	97.92	90.48	90.85
CIFAR-100	38.50	33.31	87.89	88.46	58.15	58.30
Tiny ImageNet	13.99	7.45	92.18	95.45	79.26	85.71
Places365	28.21	19.58	90.57	92.53	71.04	74.39

Comparison between the maximum softmax probability (MSP) and *uce* OOD scoring methods on a network fine-tuned with OE. Results are percentages and an average of 10 runs. (Hendrycks et al, 2019).

GAN-synthesized OOD samples

Lee et al. (2018) proposed to use a GAN (Goodfellow et al. 2014) to generate synthetic data and concurrently train an OOD detector with the following optimization procedure :

$$\begin{aligned} \min_{G} \max_{D} \min_{\theta} \quad & \mathbb{E}_{x \sim \mathcal{D}_{in}} [\log D(x)] + \mathbb{E}_{x \sim G} [\log(1 - D(x))] \\ & + \mathbb{E}_{(x,y) \sim \mathcal{D}_{in}} [- \log \hat{p}_y(x; \theta)] \\ & + \beta \, \mathbb{E}_{x \sim G} [KL (\mathcal{U} \,||\, \hat{p}(x; \theta))] \end{aligned} \tag{12}$$

where KL is Kullback–Leibler divergence, $\mathcal{U}$ is the uniform distribution, and β is some weighing parameter.

GAN-synthesized OOD samples and outlier exposure

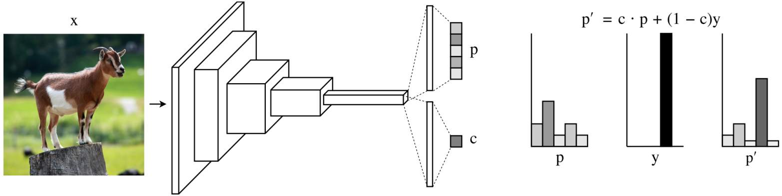
What happens if we fine-tune the discriminator with OE ?

	FPR95 ↓			AUROC ↑			AUPR ↑		
$\mathcal{D}_{\text{in}}$	MSP	+GAN	+OE	MSP	+GAN	+OE	MSP	+GAN	+OE
CIFAR-10	32.3	37.3	11.8	88.1	89.6	97.2	51.1	59.0	88.5
CIFAR-100	66.6	66.2	49.0	67.2	69.3	77.9	27.4	33.0	44.7

Comparison among the maximum softmax probability (MSP), MSP + GAN, and MSP + GAN + OE OOD detectors. The same network architecture is used for all three detectors. All results are percentages and averaged across all $\mathcal{D}_{out}^{OE}$ datasets (Hendrycks et al, 2019).

Confidence branch

DeVries & Taylor (2018) proposed a conformal prediction approach for OOD detection :



Confidence branch (taken from (DeVries & Taylor, 2018)). The c value correspond to our b.

$$\hat{p}'_k(x, y; \theta) = b(x; \theta)\, \hat{p}_k(x; \theta) + (1 - b(x; \theta))\, y_k \qquad (13)$$

$$\mathcal{L}(x, y; \alpha, \theta) = -\log\left[\hat{p}'_y(x, y; \theta)\right] - \alpha \log\left[b(x; \theta)\right] \qquad (14)$$

Confidence branch and outlier exposure

fine-tuning objective function

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_{in}}\mathcal{L}(x,y;\alpha,\theta) + 0.5\,\mathbb{E}_{x'\sim\mathcal{D}_{out}^{OE}}\left[\log b(x';\theta)\right] \qquad (15)$$

the confidence b is encouraged to be low on $\mathcal{D}_{out}^{OE}$

	FPR95 ↓			AUROC ↑			AUPR ↑		
$\mathcal{D}_{\text{in}}$	MSP	Branch	+OE	MSP	Branch	+OE	MSP	Branch	+OE
CIFAR-10	49.3	38.7	20.8	84.4	86.9	93.7	51.9	48.6	66.6
CIFAR-100	55.6	47.9	42.0	77.6	81.2	85.5	36.5	44.4	54.7
Tiny ImageNet	64.3	66.9	20.1	65.3	63.4	90.6	30.3	25.7	75.2

Comparison among the maximum softmax probability, Confidence Branch, and Confidence Branch + OE OOD detectors. The same network architecture is used for all three detectors. All results are percentages, and averaged across all $\mathcal{D}_{out}^{test}$ datasets (Hendrycks et al, 2019).

Outline

Discussion

- Outlier exposure (OE) enhances other OOD detection methods
  - by fine-tuning the underlying method on OOD data ;
  - the objective must be selected based on the task and the underlying OOD detector.
- Authors claim that training from scratch with OE improves on original task
  - good regularizer ;
- Also works
  - on NLP tasks ;
  - for density estimation ;
  - to enhance conformal prediction.

Discussion—supervised approach

Does supervised approaches to OOD detection even make sense ?

The choice of $\mathcal{D}_{out}^{OE}$ with respect to $\mathcal{D}_{in}$ and $\mathcal{D}_{out}^{test}$ is crucial

- cannot use noisy version of $\mathcal{D}_{in}$ as $\mathcal{D}_{out}^{OE}$
  - too simple, easy to set aside ;
- $\mathcal{D}_{out}^{OE}$ must be close to $\mathcal{D}_{in}^{test}$
  - otherwise too obvious, easy to set aside ;
- $\mathcal{D}_{out}^{OE}$ must be revelvant for $\mathcal{D}_{out}^{test}$
  - otherwise, procedure is useless ;

Do not extrapolate results !

Valid so long as all distributions remain close.

Bibliography

- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).
- Hendrycks, D., & Gimpel, K. (2016). A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv preprint arXiv :1610.02136.
- DeVries, T., & Taylor, G. W. (2018). Learning confidence for out-of-distribution detection in neural networks. In Proceedings of the International Conference on Learning Representations
- Lee, K., Lee, H., Lee, K., & Shin, J. (2018). Training confidence-calibrated classifiers for detecting out-of-distribution samples. arXiv preprint arXiv :1711.09325.
- Hendrycks, D., Mazeika, M., & Dietterich, T. (2019). Deep anomaly detection with outlier exposure. In Proceedings of the International Conference on Learning Representations.