Boundary Aware Learning for Out-of-distribution Detection

Sen Pei\textsuperscript{1}, Xin Zhang\textsuperscript{1}, Richard Yi Da Xu\textsuperscript{2} and Gaofeng Meng\textsuperscript{1}

\textsuperscript{1}NLPR, Institute of Automation, Chinese Academy of Sciences  
\textsuperscript{2}Department of Mathematics, Hong Kong Baptist University  
peisen2020@ia.ac.cn, \{xin.zhang2018, gfmeng\}@nlpr.ia.ac.cn, xuyida@hkbu.edu.hk

Abstract

This paper focuses on the problem of detecting out-of-distribution (OOD) samples with neural nets. In image recognition tasks, the trained classifier often gives high confidence scores for input images that are remote from the in-distribution (ID) data, and this has greatly limited its application in real world. For alleviating this problem, we propose a boundary aware learning framework and instantiate a GAN based boundary aware classifier (GBAC) for generating a closed hyperspace that only contains most ID data. Our method is based on the fact that the traditional neural net separates the feature space as several unclosed regions which are not suitable for OOD detection. With GBAC, the OOD data distributed outside the closed hyperspace will be assigned with much lower confidence score, allowing more effective OOD detection while maintaining the classification performance. Moreover, we present a fast sampling method for generating hard OOD representations which lie on the boundary of the pre-mentioned closed hyperspace. Experiments taken on several datasets and neural net architectures promise the effectiveness of GBAC.

1 Introduction

Deep convolutional neural networks are one of the basic architectures of deep learning, and they have achieved great success in modern computer vision tasks. However, the over-confidence issue has always been with CNN which harms its generalization performance seriously. In previous research, neural networks have been proved to generalize well when the testing data are sampled from the same distribution as the training data (i.e., ID data), but in open world recognition, this prerequisite is hard to be guaranteed.

From the current situation, there are two major challenges need to be settled with an increasing need in recognition tasks: adversarial examples and OOD examples. In [Goodfellow et al., 2014a], it is found that even adding very small perturbations to the input can fool a well-trained classification net, and these modified inputs are the so-called adversarial examples. Another problem is detecting OOD examples that are drawn far away from the training data. The trained neural net often gives very high confidence to these OOD samples which has raised concerns for AI Safety [Amodei et al., 2016] in many applications, and that’s the so-called over-confidence issue [Nguyen et al., 2015]. As shown in Fig.1, a trained ResNet18 is used for extracting features from the MNIST dataset, and the blue points indicate the location of these images in feature space. It can be clearly found that in feature space, each category of ID data densely concentrates in a certain area with the mapping of CNN, but almost all the feature space is assigned with high confidence score.

Up to now, several studies have proposed different approaches for detecting OOD samples, and thus, improving the robustness of the classification net. In [Hendrycks and Gimpel, 2016], a softmax based baseline is proposed for identifying OOD samples, and the metrics are also defined properly in detail. Further, in ODIN [Liang et al., 2020], temperature scaling and input pre-processing are introduced for improving the confidence scores of ID samples. Most subsequent techniques can leverage ODIN since it is a post-processing method. In [Yang et al., 2018], convolutional prototype learning (CPL) is proposed for image classification which shows effectiveness in outlier rejection and class-incremental learning. Moreover, in [DeVries and Taylor, 2018], it points out that the outputs of softmax can not represent the confidence
of neural net actually, and thus, a new branch is separated for confidence estimation independently. All these previous works have brought many different perspectives and inspirations for solving the open world recognition tasks. However, they pay limited attention to the learning of hard OOD features which have a great influence on OOD detection.

In this paper, we attribute the reason of bad OOD detection performance to the fact that the traditional classification net can not perceive the boundary of ID data. And thus, this paper focuses on where and how to find these data that distribute at the boundary of ID and OOD, i.e., the hard OOD data. The key idea of our proposed GBAC is to make the CNN have the ability to perceive the boundary of ID samples. Motivated by the abnormal phenomenon shown in Fig.1, we uniformly sample feature points in feature space, and only those close to clusters are treated as positive samples for training a conditional boundary aware discriminator. With that, one can verify whether an input image belongs to ID data or not. This work mainly has the following contributions:

- A GAN based boundary aware classifier is proposed for improving the rejection ability of neural networks while maintaining the classification performance. This framework doesn’t change the weights of pre-trained model and can be combined with major classification nets.
- An efficient method named RSM (Representation Sampling Method) for sampling hard OOD feature representations is given. We argue that the hard OOD examples distribute at the boundary between ID and OOD data, and these OOD examples greatly influence the generalization ability of neural networks.
- We test the proposed method on several datasets with different neural net architectures, the results suggest that our proposed GBAC significantly improves the performance of OOD detection, allowing more robust classification in the open world.

2 Related Works

There have been many techniques proposed for detecting OOD examples. To the best of our knowledge, the current approaches can be divided into four branches: softmax-based methods, generative model-based methods, classifiers-based methods and ensembles-based methods.

**Softmax based methods** In [Hendrycks and Gimpel, 2016], a baseline approach to detect OOD inputs named max-softmax is proposed, also in this work, the metrics of evaluating OOD detectors are defined properly. The max-softmax method is based on the observation that the predicted scores assigned to ID examples are higher than that of OOD examples. Following this, inspired by [Goodfellow et al., 2014a], ODIN [Liang et al., 2020] is proposed for improving the detection ability of max-softmax using temperature scaling and input pre-processing. ODIN can increase the confidence scores of ID examples greatly while marginally of OOD examples. In [Abdelzad et al., 2019; Lee et al., 2018b], these studies argue that the feature maps from the penultimate layer of neural networks are not suitable for detecting OOD examples, and thus, they use the features from a well-chosen layer and adopt some metrics such as Euclidean distance, Mahalanobis distance and OSVM classifier. In [DeVries and Taylor, 2018], a branch is separated out for confidence regression since the outputs of softmax can not well represent the confidence of neural networks.

**Generative model based methods** This kind of methods usually uses the ID samples to generate fake OOD samples, and then, train a \((C + 1)\) classifier which can improve the rejection ability of neural nets. [Vernekar et al., 2019] treats the OOD samples as two types, one indicates these samples that are close to but outside the ID manifold, and the other is these samples which lie on the ID boundary. This work uses Variational AutoEncoder [Sohn et al., 2015] to generate such data for training. In [Lee et al., 2018a], the authors argue that samples lie on the boundary of ID manifold can be treated as OOD samples, and they use GAN [Goodfellow et al., 2014b] to generate these data. The proposed joint training method of confident classifier and adversarial generator inspire our work. It can not be ignored that the methods mentioned above are only suitable for small toy datasets, and the joint training method harms the classification performance of neural net to some extent. Further, in [Denouden et al., 2018], the study points out that AutoEncoder [Kingma and Welling, 2014] can reconstruct the ID samples with much less error than OOD examples, allowing more effective detection with taking reconstruction error into consideration.

**Ensembles based methods** This kind of methods are similar to bagging in machine learning. In [Lakshminarayanan et al., 2017], the authors initialize different parameters for neural networks randomly, and the bagging sampling method is used for generating training data. This method enjoys the benefits of bagging method. Similarly, in [Sastry and Oore, 2019], the features from different layers of neural network are used for identifying OOD samples. The defined higher order Gram Matrices in this work yield better OOD detection performance. More recently, [Shalev et al., 2019] converts the labels of training data into different word embeddings using GloVe or FastText as the supervision to gain diversity and redundancy, the semantic structure improves the robustness of neural networks.

**Some new tendency** In [Ren et al., 2019], the authors argue that the likelihood score is heavily affected by the population level background statistics, and thus, they propose a likelihood ratio method to deal with background and semantic targets in image data. In [Hendrycks et al., 2019], the study finds that self-supervision can benefit the robustness of recognition tasks in a variety of ways. In [Zisselman and Tamar, 2020], a residual flow method is proposed for learning the distribution of feature space of a pre-trained deep neural network which can help to detect OOD examples. The latest work in [Winkens and Bunel, 2020] treats ood samples as near-OOD and far-OOD samples, it argues that contrastive learning can capture much richer features which improve performance in detecting near-OOD samples.

3 Problem Statement

This work considers the problem of separating ID and OOD samples. Suppose \(P_{in}\) and \(P_{out}\) are distributions of ID and
4 GAN Based Boundary Aware Classifier

In traditional neural networks, the confidence scores outputted by softmax are usually very high even for OOD samples. We argue that the reason of this abnormal phenomenon is because the neural networks can not perceive the boundary of ID data. Traditionally, the feature in different layers of neural networks is assumed to follow a multivariate gaussian distribution, and thus, as in works [Abdelzad et al., 2019; Lee et al., 2018b; Vernek et al., 2019], they claim that the area with lower probability density of gaussian distribution can be treated as the boundary of ID data. But these methods exist three main nonnegligible drawbacks as follows:

- **Reasonability** It is not guaranteed that the features from each layer follow a multivariate gaussian distribution, although it is seemingly straightforward sometimes.

- **Curse of dimensionality** The neural network usually has extremely deep or wide architectures, and the number of channels is huge in feature maps. This will result in very high dimensional feature vectors which are expensive for calculating the mean vector $\mu$ and covariance matrix $\Sigma$ of multivariate gaussian distribution.

- **Inefficient sampling method** With a given multivariate gaussian distribution, it is in low efficiency if each sampling needs to compute the probability density.

For evading these drawbacks, we make no assumption on the distribution of features from the penultimate layer. We use a generator to capture the feature distribution while a discriminator to identify whether a data point is ID or OOD. In the following parts of this section, we will talk about the details of our proposed GBAC and give the method for sampling OOD representations from a trained generator with high efficiency.

4.1 Methodology

For a given image $x$, its corresponding feature representation $f$ can be got by the pre-trained classification net, and based on the total probability theorem, we have:

$$P(w|f) = P(w|f \in \mathcal{M}_f) \cdot P(f \in \mathcal{M}_f) + P(w|f \notin \mathcal{M}_f) \cdot P(f \notin \mathcal{M}_f)$$

(1)

here we use $w$ to represent the category label of ID data, and $\mathcal{M}_f$ represents the set of ID features. In traditional neural networks, we have no access to OOD data, and thus the softmax output is actually the conditional probability assuming the inputs are ID data, which means $P(w|f \in \mathcal{M}_f)$ is the output of pre-trained classification net. Empirically, the OOD data usually has quite different semantic meanings compared with ID data, and thus it is reasonable to approximate $P(w|f \notin \mathcal{M}_f)$ to 0. Then, we have:

$$P(w|f) \approx P(w|f \in \mathcal{M}_f) \cdot P(f \in \mathcal{M}_f)$$

(2)

It tells that the approximation of posterior can be formulated as the product of outputs from pre-trained classification net and the probability that $f$ belongs to $\mathcal{M}_f$. Therefore, what we expected to get is $P(f \in \mathcal{M}_f|w)$ via proposed GBAC.

4.2 Architecture

The proposed framework has three basic components: the Representation Extraction Module (REM), the Representation Generation Module (RGM), and the Representation Sampling Module (RSM).

- **REM** The pre-trained neural network is used for feature extraction. In [Abdelzad et al., 2019], the authors argue that the features from the penultimate layer of trained neural nets are not suitable for detecting OOD examples, but nowadays, the neural net has extremely deep architectures which makes searching the optimal layer a very burdensome task. In this work, we use the features extracted from the penultimate layer which still bring a markable improvement in OOD detection. In the following parts, $H$ and $h$ are used to indicate the pre-trained classification net with and without the top classification layer, and $\theta$ is the pre-trained weights. With these notations, the feature $f$ of an input image $x$ can be described as:

$$f = h(x; \theta)$$

(3)

- **RGM** This module contains a conditional generator $G$ and a conditional discriminator $D$. The generator is used for generating feature representations of ID data while the discriminator is used for giving quality scores to them. The latent variable $z$ is sampled from a normal distribution $P_z$. The features of training images from REM follow a distribution $P_f$. For learning the boundary of ID data via discriminator, we propose **shuffle loss** and **uniform loss**. In each batch of the training data, we get feature-label pairs like $(f, c)$. In a conditional GAN, these $(f, c)$ pairs are treated as positive samples. With a shuffle function $T(\cdot)$, the positive pair $(f, c)$ is transformed to a negative pair $(f, \tilde{c})$ where $\tilde{c} = T(c)$ is a mismatched label with feature $f$. The discriminator is expected to identify these samples as OOD data, and the classification loss is the so called **shuffle loss** as below:

$$L_s = \mathbb{E}_{P_f} (\log D(f; T(c)) - \log D(f; c))$$

(4)
In feature space, each category of ID data concentrates in some specific regions densely, therefore the space beyond each cluster area should be treated as OOD region. Given a batch features \( \{ f_1, f_2, f_3, \ldots, f_k \} \), the length of each feature \( f_i \) is \( m \). For uniformly sampling, we first calculate the minimal and maximal bound in \( m \)-dimensional space that contains all features within this batch. Thus, for all \( j \), we have:

\[
R_{\min}^{(j)} = \min_{1 \leq i \leq k} f_i^{(j)} \\
R_{\max}^{(j)} = \max_{1 \leq i \leq k} f_i^{(j)}
\]

therefore, the lower and upper bound of feature vectors are got as follows:

\[
a = (R_{\min}^{(1)}, R_{\min}^{(2)}, \ldots, R_{\min}^{(m)})^T \]  
\[
b = (R_{\max}^{(1)}, R_{\max}^{(2)}, \ldots, R_{\max}^{(m)})^T
\]

We use \( U(a, b) \) to indicate a uniform distribution in feature space based on this batch of training data. Randomly sampled features \( \tilde{f} \) from \( U(a, b) \) are treated as negative samples with randomly generated labels \( \tilde{c} \). Conflict happens when \( \tilde{f} \) is sampled near one cluster and its randomly generated label \( \tilde{c} \) is consistent with that. For tackling these conflicts, we strengthen the memory of discriminator about positive pairs.

The uniform loss is defined as follows:

\[
L_u = \mathbb{E}_{P_f} \log D(f; \hat{c}) - \mathbb{E}_{P_{f,c}} \log D(f; c)
\]  

With notations above, the loss function \( L_d \) for discriminator can be formulated as below:

\[
L_t = -\mathbb{E}_{P_{f,c}} \log D(f; c) - \mathbb{E}_{P_{f}} \log(1 - D(G(z; c)))
\]  

\[
L_d = L_t + L_s + L_u
\]

where \( L_d \) is the loss of discriminator in traditional cGAN. A well trained discriminator is a binary classifier for separating ID and OOD data. The loss function for training generator is almost unchanged, we add a regularization term to accelerate the convergence. For positive feature-label pairs \( \langle f, c \rangle \), the loss function of generator is formulated as follows:

\[
L_g = \mathbb{E}_{P_c} \log(1 - D(G(z; c); c)) + \lambda ||f - G(z; c)||_1
\]  

where \( || \cdot ||_1 \) indicates the L1 norm, and \( \lambda \) is a balance hyperparameter. We set \( \lambda \) to 1 in our experiments.

**RSM** Gradient descent method is used to sample hard OOD examples. For \( z \) sampled from \( P_z \), the corresponding feature representation \( f \) can be got by \( G(z) \), but usually the confidence score \( D(f) \) will be high since the generator \( G \) is trained for generating ID data. With Fast Gradient Sign Method [Goodfellow et al., 2014a], we can push the feature \( f \) towards the boundary of ID manifold which gets a much lower score from discriminator.

\[
\tilde{f} = f - \epsilon \frac{\partial D(f; c)}{\partial f} \approx f - \epsilon \text{sgn}(\partial D(f; c))
\]

\[
\tilde{z} = z - \epsilon \frac{\partial D(G(z; c); c)}{\partial z}
\]

where \( \tilde{f} \) represents the calibrated feature which scatters at the low density area of \( P_f \). \( \tilde{z} \) can be used for generating OOD features by \( G(\tilde{z}; c) \). \( \epsilon \) is assumed a random variable which follows a gaussian distribution for improving diversity of sampling.

### 4.3 Training Phase

There are two kinds of calibration methods: patched style and inplace style.

**Patched Style** This style of framework only need to train the cGAN. The discriminator is a binary classifier for separating ID and OOD data as mentioned before, and the confidence score outputed by a well trained discriminator can be treated as \( P(f \in M_f) \). Based on Eq.(2), the approximation of posteriori can be formulated as the product of outputs from pre-trained classification net and discriminator. With this framework style, there is no need to finetune the top classification layer, and the only drawback is to save the discriminator additionally.

**Inplace Style** This style of framework does not need to save the additional discriminator, but the pre-trained classification net has to be finetuned. We use \( P_{\theta}(y = j|f_i) \) to represent the probability of category label \( j \) outputed by softmax given feature \( f_i \). The positive features are generated via REM, and the negative hard OOD data \( \tilde{f} \) is generated by RSM. The softmax outputs for negative hard OOD data are expected to follow a uniform distribution which means a much lower confidence score. Suppose we have \( C \) classes, therefore the cross-entropy loss \( L_1 \) and KL-divergence loss \( L_2 \) can be formulated as follows:

\[
L_1 = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} \mathbb{I}(y_i = j) \log P_{\theta}(y = j|f_i)
\]

\[
L_2 = -\frac{1}{NC} \sum_{i=1}^{N} \sum_{j=1}^{C} \log Q(y = j|\tilde{f}_i) P_{\theta}(y = j|\tilde{f}_i)
\]

where \( \mathbb{I}(\cdot) \) is the indicator function, \( y_i \) is the target label of feature \( f_i \), \( Q(\cdot) \) is a classes-wise uniform distribution which means \( Q(\cdot) = \frac{1}{C} \) is always hold. Therefore, the Eq.(16) can be simplified to:

\[
L_2 = -\frac{1}{NC} \sum_{i=1}^{N} \sum_{j=1}^{C} \log C \cdot P_{\theta}(y = j|\tilde{f}_i)
\]

The objective of calibration framework is a weighted sum of cross-entropy loss and KL-divergence loss, and we set \( \lambda \) to 1 in experiments:

\[
L = L_1 + \lambda L_2
\]

### 4.4 Inference Phase

The inference process is corresponding to the training process, and we give the inference pipeline below.

**Patched Style** Recall that \( H, h \) and \( D \) are classification net, feature extractor and discriminator correspondingly. For a given image \( x \), we have:

\[
h(x; \theta_h) = f
\]

\[
H(x; \theta_H) = P(w|f \in M_f)
\]

\[
D(f; \theta_D, \hat{c}) = P(f \in M_f|f)
\]
this confidence score can be used for evaluating the performance of OOD detection.

\[
P(w|f) \approx \mathcal{H}(x; \theta_H) \cdot D(f; \theta_D, \hat{c}) \quad (22)
\]

\[
= \mathcal{H}(x; \theta_H) \cdot D(h(x; \theta_h); \theta_D, \hat{c}) \quad (23)
\]

**Inplace Style** In this style of framework, since the classification net \( \mathcal{H} \) has been calibrated, the output \( \mathcal{H}(x; \theta_H) \) of a given image is the approximation of posterior, and it can be used for detecting OOD examples directly.

## 5 Experiments

In this section, we first introduce the datasets and evaluation metrics, and then, we give the details of training and parameter setting in the experimental setup. The proposed GBAC in patched manner is compared with some current techniques such as max-softmax (baseline) [Hendrycks and Gimpel, 2016], ODIN [Liang et al., 2020], GCPL [Yang et al., 2018], Mahalanobis distance-based approach (MD) [Lee et al., 2018b] and so on.

### 5.1 Datasets

We keep ID and OOD examples the same size in evaluation. **MNIST and Fashion-MNIST** These are two basic datasets in deep learning. MNIST is a handwritten digit dataset released in [LeCun, 1998] and Fashion-MNIST contains some clothing images which is released in [Xiao et al., 2017]. **Omniglot** is another handwritten dataset which contains 50 different alphabets released in [Lake et al., 2015]. This dataset is often used in few-shot learning or meta-learning tasks. In this work, we treat Omniglot as OOD data. **CIFAR-10 and CIFAR-100** These two tiny image datasets are released in [Krizhevsky et al., 2009] as much more challenging image recognition benchmark. **TinyImageNet** is similar to ImageNet images [Deng et al., 2009] which contains 10k images belong to 200 categories. **SVHN** is a dataset which contains street view house numbers from real world released in [Netzer et al., 2011]. **LSUN** is a large-scale scene understanding dataset including bedroom, classroom, kitchen and many other scene images. This dataset is released in [Yu et al., 2015].

### 5.2 Evaluation metrics

We adopt the following commonly used metrics to measure the performance of detecting OOD examples in this work. **FPR at 95% TPR** is the probability of an OOD example being misclassified as ID examples when the True Positive Rate is as high as 95%. True positive Rate and False Positive Rate are the same as defined in ROC curve. Lower is better. In some cases, it is abbreviated to FPR 95%.

**Detection Error** measures the misclassification probability when True Positive Rate is as high as 95%. It is defined as 0.5(1 − TPR) + 0.5FPR. Lower is better. **AUROC** represents the area under ROC curve. Greater AUROC indicates that the neural network is more confident to assign higher score to ID data than OOD data. An ideal classifier has an AUROC score of 1. Greater is better. **AUPR** represents the area under Precision-Recall curve. AUPR indicates the ability of detecting ID examples while AUPRx10 indicates that of OOD examples. Both are expected to be higher.

### 5.3 Experimental setup

**GCPL** We use distance-based cross-entropy loss and prototype loss as mentioned in [Yang et al., 2018] for generalized convolutional prototype learning. The hyper-parameter \( \lambda \) that controls the weights of prototype loss is set to 0.01. **ODIN** Official code in ODIN is used. For any other methods, we do not adopt this technique for fair comparison. **MD** is a mahalanobis distance-based method without retraining the neural networks. The official code can be found at [deepMahalanobis detector]. It’s worth pointing out that this method uses data explicitly labeled as OOD for training, therefore it’s unfair for comparison with other techniques. We list the metric results from its original paper as an upper bound.

### 5.4 Detection results

We detail the experimental results on several datasets with ResNet18, ResNet34 and DenseNet-BC.

#### Results on MNIST, Fashion-MNIST and Omniglot

In this experiment, we set two groups for observing the effects of GBAC. The first group takes MNIST as ID data, and the mixture of Fashion-MNIST and Omniglot is treated as OOD data. It’s worth noting that in all experiments, we use no OOD data for training or tuning the model, and no access to OOD data is our basic setting. The second group takes Fashion-MNIST as ID data while MNIST and Omniglot as OOD data. For simplicity, Cls Acc and Det Err are used to represent Classification Accuracy and Detection Error. The results listed in Table.1 promise that GBAC improves the softmax baseline and outperforms the other techniques with a remarkable gap. For ODIN, we use a temperature 10 and magnitude 5e-4.

| ID | F-MNIST & Omniglot | F-MNIST & Omniglot |
|----|--------------------|--------------------|
| OOD | MNIST | MNIST |
| Cls Acc | 99.43/99.43/99.23/99.43 | 91.51/91.51/90.93/91.51 |
| Det Err | 4.14/5.01/4.77/3.06 | 32.42/19.14/30.73/7.10 |
| FPR 95% | 3.29/5.03/4.54/1.11 | 59.84/33.27/56.45/9.20 |
| AUROC | 97.66/97.94/97.96/99.32 | 89.44/93.45/81.79/97.82 |
| AUPRcl | 92.22/97.42/98.14/99.46 | 90.80/94.28/72.40/98.31 |
| AUPRout | 97.24/97.64/97.35/99.09 | 86.20/91.36/82.38/96.95 |

Table 1: Detecting OOD samples on MNIST, Fashion-MNIST and Omniglot with ResNet18.

#### Results on CIFAR-10, CIFAR-100 and SVHN

We set sufficient experiments in this part for testing the generalization ability of the proposed GBAC. The pre-trained ResNet-34 and DenseNet-BC on CIFAR-10, CIFAR-100 and SVHN come from ODIN. Since Detection Error is consistent with FPR at 95% TPR, therefore we just give the results of the latter. From Table.2 we can see that GBAC improves the metric results significantly no matter combined with softmax baseline or ODIN which promises its generalization ability. Optimal temperature and magnitude are searched for ODIN in each group.
correspondingly. The details of pre-mentioned loss functions proposed in GBAC. DenseNet-BC is used for feature extraction. CIFAR-10 is set as ID data while TinyImageNet is set as OOD data. We set four groups that use \( L_s, L_t + L_s, L_t + L_u \) and \( L_t + L_s + L_u \) as loss functions corresponding. The details of pre-mentioned loss functions can be found in Eqs.(4), (9) and (10). It is seen in Tab.3 that both shuffle loss and uniform loss we proposed can boost the performance of OOD detection.

Table 3: Ablation results. All experiments are performed with the training set of CIFAR-10, and no OOD data is used.

6 Conclusion and Discussion
In this paper, we propose GBAC for boosting OOD detection. GBAC is inspired by the fact that traditional classification nets only focus on the separation between classes and thus divide the feature space as several open regions without explicit boundary. This manner is not suitable for OOD detection since the classification net can not aware the boundary of each category in training data. With GBAC, we train a boundary aware discriminator as an auxiliary module which dramatically boosts the performance of OOD detection, and the discriminator of GBAC can be well combined with different classification nets. Besides, with our proposed framework we can get a generator simultaneously, the generator can be used for generating ID features and hard OOD features. The sufficient experiment results promise the generalization ability of GBAC, and the visualization results provide an intuitive way to realize the method and its effects.
References

[Abdelzad et al., 2019] Vahdat Abdelzad, Krzysztof Czarnecki, Rick Salay, Taylor Denouden, Sachin Vernekar, and Buu Phan. Detecting out-of-distribution inputs in deep neural networks using an early-layer output. CoRR, abs/1910.10307, 2019.

[Amodei et al., 2016] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. CoRR, abs/1606.06565, 2016.

[Deng et al., 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[Denouden et al., 2018] Taylor Denouden, Rick Salay, Krzysztof Czarnecki, Vahdat Abdelzad, Buu Phan, and Sachin Vernekar. Improving reconstruction autoencoder out-of-distribution detection with mahalanobis distance. CoRR, abs/1812.02765, 2018.

[DeVries and Taylor, 2018] Terrance DeVries and Graham W. Taylor. Learning confidence for out-of-distribution detection in neural networks, 2018.

[Goodfellow et al., 2014a] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. Computer Science, 2014.

[Goodfellow et al., 2014b] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[Hendrycks and Gimpel, 2016] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. CoRR, abs/1610.02136, 2016.

[Hendrycks et al., 2019] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. Advances in Neural Information Processing Systems (NeurIPS), 2019.

[Kingma and Welling, 2014] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

[Krizhevsky et al., 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[Lake et al., 2015] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. Science, 350(6266):1332–1338, 2015.

[Lakshminarayanan et al., 2017] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017.

[LeCun, 1998] Yann LeCun. The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/, 1998.

[Lee et al., 2018a] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples, 2018.

[Lee et al., 2018b] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks, 2018.

[Liang et al., 2020] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks, 2020.

[Netzer et al., 2011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Ng. Reading digits in natural images with unsupervised feature learning. NIPS, 01 2011.

[Nguyen et al., 2017] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, 2015.

[Ren et al., 2019] Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection, 2019.

[Sastry and Oore, 2019] Chandramouli Shama Sastry and Sageev Oore. Detecting out-of-distribution examples with in-distribution examples and gram matrices. CoRR, abs/1912.12510, 2019.

[Shalev et al., 2019] Gabi Shalev, Yossi Adi, and Joseph Keshet. Out-of-distribution detection using multiple semantic label representations, 2019.

[Sohn et al., 2015] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 28. Curran Associates, Inc., 2015.

[Vernekar et al., 2019] Sachin Vernekar, Ashish Gaurav, Vahdat Abdelzad, Taylor Denouden, Rick Salay, and Krzysztof Czarnecki. Out-of-distribution detection in classifiers via generation. CoRR, abs/1910.04241, 2019.

[Winkens and Bunel, 2020] Jim Winkens and Rudy Bunel. Contrastive training for improved out-of-distribution detection. CoRR, abs/2007.05566, 2020.

[Xiao et al., 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747, 2017.

[Yang et al., 2018] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. CoRR, abs/1805.03438, 2018.

[Yu et al., 2015] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. CoRR, abs/1506.03365, 2015.

[Zisselman and Tamar, 2020] Ev Zisselman and Aviv Tamar. Deep residual flow for out of distribution detection, 2020.