Gradient-Based Novelty Detection
Boosted by Self-Supervised Binary Classification

Jingbo Sun¹, Li Yang¹, Jiaxin Zhang², Frank Liu², Mahantesh Halappanavar³,
Deliang Fan¹, Yu Cao¹

¹ Arizona State University
² Oak Ridge National Laboratory
³ Pacific Northwest National Laboratory

jsun127@asu.edu, lyang166@asu.edu, zhangj@ornl.gov, liufy@ornl.gov, Mahantesh.Halappanavar@pnnl.gov,
dfan@asu.edu, Yu.Cao@asu.edu

Abstract

Novelty detection aims to automatically identify out-of-distribution (OOD) data, without any prior knowledge of them. It is a critical step in data monitoring, behavior analysis and other applications, helping enable continual learning in the field. Conventional methods of OOD detection perform multi-variate analysis on an ensemble of data or features, and usually resort to the supervision with OOD data to improve the accuracy. In reality, such supervision is impractical as one cannot anticipate the anomalous data. In this paper, we propose a novel, self-supervised approach that does not rely on any pre-defined OOD data: (1) The new method evaluates the Mahalanobis distance of the gradients between the in-distribution and OOD data. (2) It is assisted by a self-supervised binary classifier to guide the label selection to generate the gradients, and maximize the Mahalanobis distance. In the evaluation with multiple datasets, such as CIFAR-10, CIFAR-100, SVHN and TinyImageNet, the proposed approach consistently outperforms state-of-the-art supervised and unsupervised methods in the area under the receiver operating characteristic (AUROC) and area under the precision-recall curve (AUPR) metrics. We further demonstrate that this detector is able to accurately learn one OOD class in continual learning.

Introduction

Deep neural networks (DNNs) have achieved high accuracy in many fields, such as image classification, natural language processing, and speech recognition. Their success is built upon carefully handcrafted DNN architectures, big data collection and expensive model training. A well-trained model promises high inference accuracy if the input falls into the distribution of the training data. However, in many real-world scenarios, there is no guarantee that the input is always in the distribution. The encounter with out-of-distribution (OOD) input is inevitable due to the difficulty in data collection, unforeseeable user scenarios, and complex dynamics.

To manage the emergence of OOD data at the first moment, it is vitally important to have an accurate novelty detector that continuously evaluates the data stream and alarms the system once OOD data arrives. Upon the detection of OOD arrival, the system can then manage the situation with three possible methods: (1) It can rely on the OOD detector to collect new data and send them back to the data center, such that OOD data can be combined with previous in-distribution data (IDD) to re-train the model; (2) It can temporally utilize the detector as a one-class classifier to recognize the new class of OOD, in addition to existing IDD classes; and (3) It can activate a continual learning method at the edge to adapt the model in the field, such as the multi-head method with the assistance of the OOD detector. In all three cases, the accuracy and robustness of the novelty detector guard the success of continual model adaptation and knowledge update.

In this work, we propose a self-supervised approach which generates a set of OOD data from unsupervised statistical analysis, instead of supervised labeling. This set of OOD data is then combined with IDD to train a binary classifier, which in turn, helps boost the performance of the unsupervised detector that collects more OOD data for training. As this mutual process continues, our novelty detector achieves higher and higher confidence in OOD detection. The contributions of this paper are as follows:

- Gradient-based novelty detection that employs the Mahalanobis distance as the metric to differentiate in-distribution and OOD input. The gradients are generated from a pre-trained classifier for IDD only, without any pre-knowledge of OOD.
- A self-supervised binary classifier. As previous works demonstrated, the availability of a binary classifier helps boost the accuracy. Yet distinguished from them, we don’t rely on any labelled OOD data to train the classifier. The training set is initialized by the gradient-based detector. In turn, the binary classifier pre-screens OOD and IDD, and guides the selection of labels in the gradient-based detector to maximize the distance calculation. Through such mutual assistance, our approach is unsupervised and continually improves the accuracy.
- High accuracy in OOD detection and one-class classification. We evaluate our methodology in a comprehensive set of benchmarks. As shown in Fig. 1, our self-supervised method consistently achieves higher AUROC than other supervised and unsupervised results, confirming the advantages in gradient-based novelty detection.
Figure 1: Comparison of AUROC between our proposed method and other state-of-the-art methods.

Background

Current OOD detectors usually use the information extracted from either the data itself or features projected by the feedforward path in the IDD engine. (Hendrycks and Gimpel 2016) demonstrated that the softmax score of the outlier tended to be higher compared with IDD and thus, thresholding inputs based on this score is feasible to detect the OOD. (Liang, Li, and Srikant 2017) improved this idea by introducing input preprocessing and temperature scaling. (Lee et al. 2018) proved that the in-distribution samples formed a multivariate Gaussian distribution in the high dimensional feature space, and proposed to use the Mahalanobis distance to measure how far an outlier is away from this in-distribution. There also exist autoencoder based OOD detectors (Abati et al. 2019; Chen et al. 2017; Hawkins et al. 2002; Kwon et al. 2020; Sakurada and Yairi 2014; Zhou and Paffenroth 2017) which use the reconstruction loss from the decoder to characterize the novelty. These data or activation-based methods demonstrated the value in OOD detection. On the other side, they have not explored one important step in the development of DNNs, the gradients back-propagated from the classification layer. These gradients present the first-order derivative to adapt the model and improve the separation of multiple classes. They contain a rich body of information to differentiate OOD from IDD.

To collect the gradients from the classifier that is prepared for the IDD, a label is required for cross-entropy loss and back-propagation. However, one challenge in the gradient-based approach is that labels of OOD data are not available in the process of novelty detection. To address this issue, a recent work by (Lee and AlRegib 2020) introduced the confounding label, which only triggers small gradients for the IDD input. The gradients of the OOD input would be larger since they introduce many new features that are different from IDD. (Kwon et al. 2020) explored the gradient-based representations of the novelty but they avoid the label issue by proposing a directional gradient constraint to the loss function so that the gradient direction of the OOD input does not align with the ones of the IDD. However, this method requires re-training of the model. In contrast, in this work, we utilize the gradient-based approach and the pre-trained model without any modification. In addition, we only use the training labels to collect the gradient.

Note that to boost the accuracy in novelty detection, many prior approaches utilize supervised training. They adopted a small amount of OOD samples to pre-train the novelty detector. For example, (Lee et al. 2018) trained a logistic regression detector to estimate the weights for feature ensemble. (Lee and AliRegib 2020) trained a binary detector to distinguish the gradient representation of the in-distribution and OOD input. However, this type of supervised training requires the availability of labelled OOD data up front, which restricts its application in reality.

Methodology

The overarching goal of our methodology is to accurately identify OOD data from IDD. A successful OOD detection is equivalent to correctly classify the OOD input as one new class (i.e., one-class classification). For IDD inputs, they will be classified to the previous known classes. To achieve this goal, we propose a closed-loop methodology that interleaves the unsupervised ODD detector based on the Mahalanobis distance, with a binary IDD/OOD classifier. Fig. 2 illustrates our methodology, consisting of these two main components:

- A gradient-based novelty detector: Distinguished from previous works that analyzed either the input data or the features, our detector exploits the gradients propagated backward from the classification layer for statistical analysis. The classifier in this step is designed for IDD only and all labels are from the IDD classes. To maximize the Mahalanobis distance between OOD and IDD classes, we select the appropriate class to label the OOD data, which is predicted by a binary classifier, as in Fig.2.
- A self-supervised binary classifier. This classifier is designed to pre-screen IDD and OOD data, in order to assist label selection to generate the gradients. While the structure of this binary classifier is similar as that in (Goodfellow et al. 2014), the training does not rely on any labeled data, but from a balanced set that is selected by the detector based on the Mahalanobis distance.

Overall, the mutual assistance between these two components helps accomplish OOD detection without the need of external supervision. In the following subsections, we first introduce our main processing path in OOD detection: the Mahalanobis distance in the gradient space. We then introduce the binary classifier as a necessary assistant to boost the performance of the main path, as well as the self-supervised training of it. Finally, we describe how these two components mutually assist each other with a thorough case study.

Gradient-Based Novelty Detector

Our approach starts from a given dataset of in-distribution data (IDD), and a deep neural network (DNN) based classifier trained by this IDD dataset. Note that at this step, only labels for IDD classes are accessible. In our examples of the image classification task, previous works have demonstrated that such a DNN is capable of separating the manifolds of each in-distribution class and achieve high clas-
Figure 2: The flow of our self-supervised OOD detector. (1) A DNN-based classifier is first prepared for IDD data. (2) The binary classifier predicts IDD or OOD. If the result is OOD, then in the IDD classifier, it selects the appropriate label to maximize the distance. (3) Based on the selected label, gradients are generated by the IDD classifier for the calculation of Mahalanobis distance. (4) From each patch of data, a balanced set of IDD and OOD data is selected based on the Mahalanobis distance. (5) This balanced IDD/OOD data set is used to continuously train the binary classifier.

Figure 3: The distribution of the Mahalanobis distance of in-distribution data using predicted label (highlighted in red), and OOD using both predicted label (highlighted in blue) and selected label (highlighted in green). By using the selected label for all the outliers, their novelty score distribution shifts further away farther away from blue to green, as shown in the main figure, with less overlap with IDD data (the red distribution, as shown in the inset.)

In Equation (1), $\nabla_c f(x)$ is the back-propagated gradient with respect to the class $c$. This raises a question of how to calculate the gradient of the OOD input since their ground-truth label $Y_{ood}^{true}$ is not in the IDD label space $Y_{in}$. To solve this problem, we propose to use the predicted label $Y_{ood}^{Pred} \in Y_{in}$ to calculate and cross-entropy loss $L(Y_{ood}^{Pred}; X_{ood}; \Theta)$ and do the back-propagation. From the manifold perspective, this means that we select the manifold closest to the input sample. This manifold requires the minimal amount of adaptation to the new input, i.e., the minimal gradient and Mahalanobis distance. If such Mahalanobis distance is still large, the input sample has a high probability to be an outlier.

As shown in Table 1, if only using the predicted label by the IDD classifier, our proposed method is not competitive to the state-of-the-art method. This is due to the high overlap of the novelty score distribution between the IDD and OOD.
Table 1: Comparison of AUROC results between the state-of-the-art (Lee et al. 2018; Sehwag, Chiang, and Mittal 2021) and our proposed novelty detector. The middle column is the performance of our novelty detector using the predicted label from the DNN for IDD to calculate the loss, gradients and Mahalanobis distance. The right column shows the performance boosting if we intentionally select the label to maximize the distance between IDD and OOD.

<table>
<thead>
<tr>
<th>OOD</th>
<th>State-of-the-art</th>
<th>Ours (with predicted label)</th>
<th>Ours (with selected label)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TinyImageNet</td>
<td>99.50</td>
<td>91.10</td>
<td>99.92</td>
</tr>
<tr>
<td>SVHN</td>
<td>99.90</td>
<td>91.63</td>
<td>99.99</td>
</tr>
<tr>
<td>LSUN</td>
<td>99.70</td>
<td>90.08</td>
<td>99.99</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>93.40</td>
<td>86.03</td>
<td>97.99</td>
</tr>
</tbody>
</table>

To guide our novelty detector to use either the predicted or selected label in case of in-distribution or OOD input, we introduce a simple self-supervised binary classifier to screen IDD and OOD. The output from this binary classifier will be used to guide label selection in the pre-trained IDD classifier for gradient generation:

- For the predicted IDD input, the pre-trained model uses the predicted label \( y_{\text{pred}} \) for back-propagation.
- For the predicted OOD input, the model uses the selected label \( y_{\text{opt}} \).

Unsupervised Preparation of OOD Samples for Initial Training: One key feature of our proposed method is to be self-supervised, which means no OOD sample is available in advance. Therefore to create a dataset for the training of the binary classifier, we utilize the predicted IDD/OOD samples selected by the gradient-based novelty detector. For example, assuming the binary classifier is randomly initialized and a batch \( X^1 \) mixed with IDD and OOD data comes in, our novelty detector will first calculate the novelty confidence score \( S^1 = \{s_1, s_2, \ldots, s_N\} \) using the predicted label. We select \( N/2 \) samples that correspond to the highest and lowest confidence scores in the \( S^1 \) as the binary classifier training data set \( X_{\text{pred}} = \{X_{\text{in}}^{\text{pred}}, X_{\text{ood}}^{\text{pred}}\} \). This step helps select the best possible in-distribution and OOD data from the current batch so that the binary classifier’s training inputs are reasonable. We use the \( X_{\text{pred}}^{\text{opt}} \) to select the label and use the \( X_{\text{pred}}^{\text{pred}} \) to train the binary classifier. Once training is done, the following input batches will involve the cooperation from both the gradient-based novelty detector and the binary classifier.

**Self-Supervised Binary Classifier**

**Label Selection**: To select a label \( y_{\text{opt}} \) that gives the maximum gradient distance, we propose to use the predicted softmax class probability:

\[
y_{\text{Custom}} = \arg\min_{c} (\sum_{i} \text{Softmax}(f(x_{i}^{\text{OOD}}, \Theta)))
\]  

(4)

We use Equation (4) to select the class that has the minimal average softmax probability for a batch of OOD data. This is equivalent to finding the least likely class for an OOD data to fall into. By using \( y_{\text{opt}} \), its cross-entropy loss \( L(y_{\text{opt}}; X_{\text{ood}}; \Theta) \) becomes the maximum which results in the largest gradient and Mahalanobis distance. The only question left is where to find the OOD samples to estimate this label. We will address this issue in subsection.

**Calculation of the Mahalanobis Score**: The gradient-based novelty detector takes the prediction result from the binary classifier and the concatenated batch. For each sample \( x_i \) in batch, our novelty detector first classifies it to one of the in-distribution class \( y_i^{\text{pred}} \in Y_{\text{in}} \), then checks...
the binary classifier prediction $\hat{y}^t_i$. If $y^t_i$ is 0 (predicted in-distribution), use $y^t_i^{\text{red}}$ as the ground truth label in the loss function for back-propagation, otherwise, use the pre-selected label $y^{\text{Opt}}$. After the gradient is available, we calculate the Mahalanobis distance $s_i$ as the novelty confident score.

**Re-Training of the Binary Classifier:** Given novelty confident score $S^{\text{all}} = \{s_1, s_2, \ldots, s_{N \times K}\}$ from the gradient-based novelty detector, we select $(N \times k)/2$ samples from the concatenated batch that correspond to the highest and lowest scores in $S^{\text{all}}$ as a new training dataset for the binary classifier. Before training, we re-initialize the binary classifier to make sure the previous model will not be inherited into the current stage. Once training is done, our system is updated with the knowledge of all previous batches and is ready to process next available inputs with higher detection accuracy.

As shown in Fig. 2, this mutual assistance continues with more unlabelled data, which keeps improving the accuracy of both the unsupervised detection engine and the binary classifier in this closed loop.

**Experiments**

We use three pre-trained ResNet-34 networks (He et al. 2016) provided by (Lee et al. 2018) as the base of our gradient-based novelty detector. Each model is trained on CIFAR-10, CIFAR-100 and SVHN (Netzer et al. 2011) with the testing accuracy of 93.67%, 78.34% and 96.68% accordingly. To calculate the gradient-based Mahalanobis distance, we only use the gradient extracted from the last layer of the feature extractor. Our simple binary classifier has the structure of three convolution layers and one batch normalization layer with a Sigmoid classifier. It is trained by minimizing the cross-entropy loss using Adam (Kingma and Ba 2014). The initial learning rate is set to 0.0002 and the decay rate is controlled by $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We train it for 500 epochs in both initial training and re-training process. Regarding the batch size, we find that the amount of IDD and OOD samples in each batch, and the batch size itself have a strong impact on the performance of the binary classifier and the overall system, as discussed in subsection

Our experiment includes three in-distribution datasets: CIFAR-10, CIFAR-100 and SVHN. We test each of them using the other two as the OOD. In addition, we use two more OOD datasets: the resized version of the ImageNet (Russakovsky et al. 2015) and LSUN (Yu et al. 2016) provided by (Liang, Li, and Srikant 2017). Each of these two datasets contains 10,000 images with size 32 by 32. For all three in-distribution datasets, we use only the testing portion of the images because the training portion has already been used to train the novelty detector. For each IDD/OOD pair, we randomly select 5,000/5,000 (IDD/OOD) as the training dataset and further divide them into mini batches with size of 100/100 (IDD/OOD) to emulate the data streaming, the rest 5,000/5,000 (IDD/OOD) samples are used to test the binary classifier accuracy and the overall novelty detection performance after each new batch has been taken into the system. We evaluate our detector with two performance metrics: AUROC and AUPR. All experiments are performed with PyTorch (Paszke et al. 2019) on one NVIDIA GeForce RTX 2080 platform.

**Algorithm 1: Gradient-based Novelty Detection Boosted by Self-supervised Binary Classification**

**Input:** In-distribution gradient distributions $\{\mu, \Sigma^{-1}\}$, batches for testing $[X^1, X^2, \ldots, X^K]$

1: **function** noveltyScore $(X, \mu, \Sigma^{-1}, \hat{Y}, y^{\text{Opt}})$
2: ** Mahalanobis Distance** = $\text{list}()$
3: **for each** $x$ in $X$ **do**
4: $c = y^{\text{Opt}}$ //Label for back-propagation
5: **if** $\hat{Y}$ is None or Binary Classifier predict x as IDD **then**
6: $c = \text{Novelty detector predicted label}$
7: **end if**
8: $\text{Score} = (\nabla_c f(x) - \mu_c)^T \hat{\Sigma}^{-1} (\nabla_c f(x) - \mu_c)$
9: Mahalanobis_Distance.append(Score)
10: **end for**
11: **return** Mahalanobis_Distance
12: **end function**
13: $X^{\text{all}} = \text{None}$
14: **while** new batch $X^k$ is available **do**
15: $X^{\text{all}} = X^{\text{all}} + X^k$ //batch concatenation
16: **if** $X^k$ is the first batch **then**
17: $S^{\text{all}} = \text{noveltyScore}(X^{\text{all}}, \mu, \Sigma^{-1})$
18: **else**
19: $\hat{Y} = \text{binary classifier prediction on } X^{\text{all}}$
20: $S^{\text{all}} = \text{noveltyScore}(X^{\text{all}}, \hat{\mu}, \Sigma^{-1}, \hat{Y}, y^{\text{Opt}})$
21: **end if**
22: Based on $S^{\text{all}}$, select samples from $X^{\text{all}}$
23: with the highest/lowest score as predicted $X_{\text{in}}/X_{\text{ood}}$
24: Use $\{X_{\text{in}}, X_{\text{ood}}\}$ to train binary classifier
25: **if** $y^{\text{Opt}}$ is None **then**
26: $y^{\text{Opt}} = \arg \min_i (\sum \text{Softmax}(f(x^{\text{ood}}_i)))$
27: **end if**
28: **end while**
29: **return** $X_i \in X^{\text{all}}$ is an outlier if $S_i^{\text{all}} > \text{threshold}$
Table 2: Comprehensive evaluation of AUROC and AUPR on multiple IDD and OOD benchmarks.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>OOD</th>
<th>AUROC / AUPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDD</td>
<td>OOD</td>
<td></td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>TinyImageNet</td>
<td>99.90/99.91</td>
</tr>
<tr>
<td></td>
<td>CIFAR-100</td>
<td>84.79/81.46</td>
</tr>
<tr>
<td></td>
<td>SVHN</td>
<td>99.94/99.98</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>TinyImageNet</td>
<td>99.49/99.43</td>
</tr>
<tr>
<td></td>
<td>CIFAR-10</td>
<td>72.61/70.07</td>
</tr>
<tr>
<td></td>
<td>LSUN</td>
<td>99.57/99.53</td>
</tr>
<tr>
<td></td>
<td>SVHN</td>
<td>99.79/99.95</td>
</tr>
<tr>
<td>SVHN</td>
<td>CIFAR-10</td>
<td>99.94/99.72</td>
</tr>
<tr>
<td></td>
<td>TinyImageNet</td>
<td>99.95/99.82</td>
</tr>
<tr>
<td></td>
<td>CIFAR-100</td>
<td>99.65/99.25</td>
</tr>
</tbody>
</table>


Figure 4: AUROC in our novelty detector (top) and the testing accuracy of the binary classifier (bottom) during the mutual assistance (i.e., number of iterations).
Figure 5: (a) Comparison of AUROC with (right) and without (left) the binary classifier boosting. (b) Comparison of AUROC between the good (right) and bad (left) OOD prediction from the novelty detector. (c) AUROC by using different percentages of the selected images to train the binary classifier. (d) Comparison of AUROC with (left) and without (right) binary classifier re-initialization.

Ablation Study and Discussion

We test the 9+1 single-head accuracy on each class with different memory budgets (Table 3). Compared with (Du et al. 2020) which used memory budget of 2,000 IDD samples with the accuracy <90%, our proposed method achieves 91-96% accuracy in any class with only 1,000 IDD samples and competitive performance even with 500 IDD samples.

We further conduct two ablation studies regarding the training of the binary classifier: (1) With different percentage of the selected images as the training dataset of the binary classifier, what’s the influence on the overall performance? (2) If the binary classifier is not re-initialized after each iteration, will it still converge and provide good guidance for the novelty detection? As shown in Fig. 5(c), when fewer IDD/OOD images from the prediction of the novelty detector are selected to train the binary classifier, the overall AUROC result decreases. Even though the selection of fewer data better separates IDD and ODD samples, we also exclude the samples in which IDD and OOD are very similar. Therefore, the binary classifier receives imbalanced training and has a degraded prediction on the testing dataset.

Fig. 5(d) presents the CIFAR-10 vs. CIFAR-100 AUROC results with and without re-training the binary classifier after each iteration. In both experiments, the AUROC decreases when we choose not to re-initialize the binary classifier. This is because during the early stage, the novelty detector is not able to accurately detect the outlier and thus, the predicted training dataset for the binary classifier may suffer from inaccurate IDD/OOD labeling. This ambiguity on IDD/OOD affects the binary classifier in the following iterations during our training process. On the other hand, re-training the binary classifier helps improve the confidence of the novelty detector with more accurate IDD/OOD detection.

Conclusion

In this paper, we propose a new framework for novelty detection and one-class continual learning, which involves the cooperation of a gradient-based novelty detector and a self-supervised binary classifier with mutually assistance between these two components. Our proposed framework outperforms previous supervised and unsupervised novelty detectors, as well as one-class continual learning. For instance, our method achieves superior performance on CIFAR-100 vs. CIFAR-10 case by improving AUROC from 84% to 91% compared with the state-of-the-art. The success of the new approach promises accurate and robust novelty detection in continual learning and other applications.

Acknowledgements

This research was supported in part by the U.S. Department of Energy, through the Office of Advanced Scientific Computing Research’s “Data-Driven Decision Control for Complex Systems (DnC2S)” project. It was also partially supported by C-BRIC, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored...
References


