Learning non-linear invariants for unsupervised out-of-distribution detection

Lars Doorenbos, Raphael Sznitman, Pablo Márquez-Neila

ARTORG Center for Biomedical Engineering Research, Artificial Intelligence in Medical Imaging Laboratory, Universität Bern, Switzerland

Abstract

An important hurdle to overcome before machine learning models can be reliably deployed in practice is identifying when samples are different from those seen during training, as the output for unexpected samples are often confidently incorrect, while not being identifiable as such. This problem is known as out-of-distribution (OOD) detection. A popular approach for the unsupervised OOD case is to reject samples with a high Mahalanobis distance with regards to the mean features of the training data. Recent work showed that the Mahalanobis distance can be thought of as finding the training data invariants, and rejecting OOD samples that violate them [1]. A key limitation to this approach is that it is limited to linear relations only. Here, we present a novel method capable of identifying non-linear invariants in the data. These are learned using a reversible neural network, consisting of alternating rotation and coupling layers. Results on a varied number of tasks show it to be the best method overall, and achieving state-of-the-art results on some of the experiments.

Background

Given a training set $\{\mathbf{x}_i\}_{i=1}^N$ with $\mathbf{x}_i \in \mathbf{X}$, we define an invariant as a non-constant function $g : \mathbf{X} \to \mathbb{R}$, such that $g(\mathbf{x}_i) = 0,\ \forall i$. That is, g is an invariant if it computes a constant value for the elements of the training set, but in general may not compute the same constant value for other elements (*e.g.*, elements of a test set). Our goal then is to find a set of invariants, $G = \{g_1, \dots, g_K\}$, over the set of training samples.

As noisy real-world data rarely lies in an exact manifold, this is unfeasible in practice even for a small number of invariants K. Instead, we relax the above goal and express it as a minimization problem to find a set of soft invariants:

$$\min_{g_k} \frac{1}{N} \sum_i g_k(\mathbf{x}_i)^2, \qquad \text{s.t.} \quad \|\nabla g_k(\mathbf{x}_i)\|_2 = 1 \quad \forall i. \tag{1}$$

The second equality prevents g_k from becoming a trivial projection (*i.e.*, effectively making it non-constant). Once $G = \{g_1, \dots, g_K\}$ is established, any test sample $\mathbf{x}$ can be scored by computing the ratios between the test error and the average training error,

$$s^2(\mathbf{x}) = \sum_k \frac{g_k(\mathbf{x})^2}{e_k}, \tag{2}$$

where e_k is the training mean squared error (MSE) of the soft invariant g_k,

$$e_k = \frac{1}{N} \sum_i g_k(\mathbf{x}_i)^2. \tag{3}$$

We can further simplify the optimization problem of Eq. (1) by constraining the invariants to the family of affine functions $g_k(\mathbf{x}) = \mathbf{a}_k^T \mathbf{f} + b_k$ with unitary $\mathbf{a}_k$. Under these conditions, Eq. (1) reduces to a PCA problem. Its solution sets $\mathbf{a}_k$ to the k-th smallest principal component and the squared error e_k is set to its corresponding eigenvalue. Moreover, the score function Eq. (2) can be re-written as the square of the Mahalanobis distance using the mean and the covariance of the training feature vectors [1].

Method

Restricting the invariants to affine functions will only allow us to capture linear relationships. Nonetheless, datasets might contain non-linear invariants important to OOD detection as well. Here, we propose a novel framework capable of learning them, which we dub the Volume Preserving Network (VPN).

By Eq. (1), we have to restrict ourselves to operations with a determinant of 1, to prevent learning trivial solutions. For this, we borrow from the literature of normalizing flows (*e.g. [2]*), and build a bijective, reversible neural network consisting of a combination of rotation and coupling layers (Our architecture is shown in Figure 1).

The rotation layer learns the weights of a rotation matrix with an optional bias. This layer does not learn non-linear relations, but is for example capable of emulating PCA. The coupling layers on the other hand are capable of learning non-linearities. They split the input features into two parts, x_1 and x_2. Then, x_2 is fed through a multi-layer perceptron (MLP), which outputs a vector of equal size that is used to translate x_1. Its output is concatenated with x_2 to produce the output y. Formally,

$$y = (x_1 + \text{MLP}(x_2), x_2). \tag{4}$$

See Figure 2 for a visual explanation of the coupling layer. Both of these layers are easily reversible: the rotation layer by using the transpose of the rotation matrix, and the coupling layer by using $y = (x_1 - \text{MLP}(x_2), x2)$.

We train the reversible VPN with two loss functions. First, the VPN is trained to find a set number K of invariants, which comes down to minimizing the L_2 norm of K dimensions (Eq. (1)). This loss function by itself is not sufficient, as the network can simply ignore the other $N - K$ dimensions, and might learn uninformative invariants.

To remedy this, we make use of the reversibility of the VPN. By going back from the invariant representation to the original data space, we obtain a reconstruction x' of the original data point x. The second loss aims to minimize this reconstruction error, *i.e.* $L_{rec} = (x - x')^2$.

We set the number of invariants equal to the number of linear invariants (a lower bound), which we define as the largest number of principal components, starting from the smallest, that together explain less than 2% of the variance.

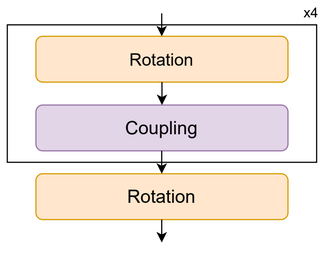

Figure 1: The VPN architecture used.

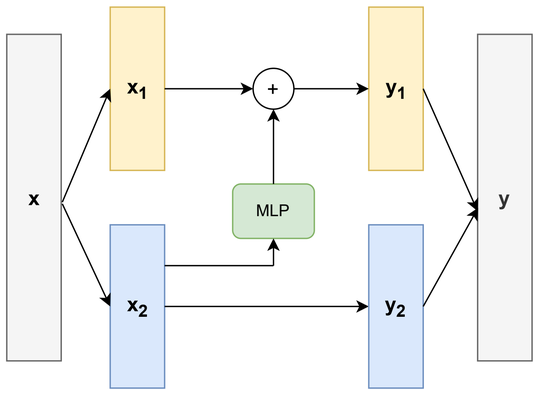

Figure 2: The coupling layer.

An example application of our VPN on two-dimensional toy data can be seen in Figure 3. The toy data shown in Figure 3(a) has no linear invariant, *i.e.* there exists no affine g_k for which $\frac{1}{N}\sum_i g_k(\mathbf{x}_i)^2$ is reasonably close to 0. However, it does have a non-linear invariant, in that all samples lie on part of the unit circle (with some noise).

After training, passing the data forward through the network gives the invariant representation shown in Figure 3(b). It can be seen that the network has learned a dimension that is almost constant, which is the non-linear invariant, and the variability is encoded in the other dimension. Here, the out-of-distributionness of test samples can be scored using Eq. 2, that is computing the ratio between its value in the invariant dimension to that of the mean of the training set.

From this invariant representation the data can be reconstructed by reversing through the network (Figure 3(c)). Despite the fact that one of the dimensions has almost no variability, the network is still capable of reconstructing the original input data to a large degree.

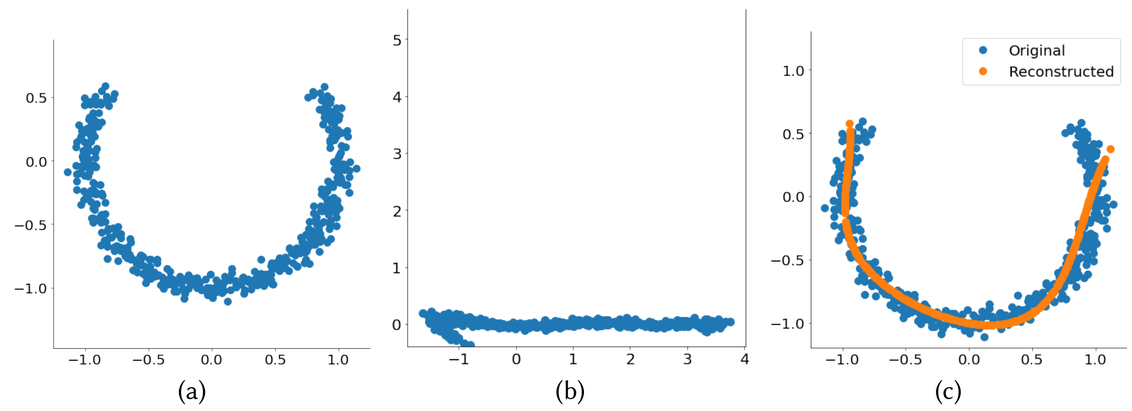

Figure 3: Results on toy data. (a) shows the data, (b) the invariant representation, and (c) the reconstruction from the invariant representation together with the original data.

Experiments and results

We evaluate our approach on two different tasks. The first is a standard OOD task involving 10 experiments on CIFAR-10, where each experiment takes one of the 10 classes as in-distribution and uses the remaining 9 as OOD. We extract features from the penultimate layer of a pretrained ResNet-18, and use those as our training set. Results are shown in Table 1.

	1	2	3	4	5	6	7	8	9	10	Mean
IF	84.5	94.4	71.5	72.6	87.1	67.2	85.5	81.4	87.6	93.5	82.5
kNN (k=2)	86.5	95.6	73.8	75.2	87.3	77.6	89.4	84.6	89.7	94.2	85.4
kNN (k=20)	84.8	95.5	70.5	73.3	86.1	76.0	87.7	83.5	89.0	94.3	84.1
LOF (k=2)	81.7	93	76.6	78.3	80.4	76.3	82.6	83.1	89.2	91.6	83.3
LOF (k=20)	87.7	95.7	79.7	79.8	88	78.5	90.6	83.6	92.1	95	87.1
Mahalanobis	90.7	96.4	79.9	**81.4**	91.7	85.4	92.3	89.5	**93.0**	96.0	89.6
NL-invs *(ours)*	**91.3**	**96.6**	**80.5**	81.3	**92.0**	**86.0**	**92.6**	**90.3**	92.6	**96.1**	**89.9**

Table 1: Results on CIFAR-10 in AUC.

For the second task we adapt 10 datasets initially selected for comparing unsupervised anomaly detectors ([3]) to our purpose. We split the outliers and an equal number of randomly selected inliers off into a test set, such that the training data only contains normal samples (sometimes also referred to as semi-supervised anomaly detection). See the results in Table 2.

	Aloi	Thyroid	B-cancer	Speech	Letter	Pen_g	Pen_l	Satellite	Shuttle	KDD99	Mean
IF	55.3	79.4	**100**	40.2	59.7	96.9	57.0	96.6	99.7	95.4	78
kNN (k=2)	**76.2**	71.9	**100**	76.7	**94.1**	99.9	99.0	97.1	**99.9**	**100**	**91.5**
kNN (k=20)	66.5	64.5	**100**	48.4	88.9	**100**	98.0	**98.2**	**99.9**	**100**	86.4
LOF (k=2)	75.5	73.3	96.0	**92.1**	91.2	99.6	**100**	91.3	99.1	97.2	**91.5**
LOF (k=20)	75.2	66.9	**100**	48.1	91.2	99.7	**100**	96.9	**99.9**	98.5	87.6
Mahalanobis	57.0	66.0	**100**	45.9	84.6	99.1	58.0	94.1	99.7	94.8	79.9
NL-invs *(ours)*	55.5	**84.8**	**100**	66.0	90.2	99.1	**100**	94.5	**99.9**	**100**	88.6

Table 2: Results on non-image datasets in AUC. B-cancer, Pen_l and Pen_g denote the breast-cancer, pen global and pen local datasets respectively.

Conclusions

We find that the methods behave erratically across tasks. For example, kNN and LOF with $k = 2$ are the best methods on the non-image datasets, where the Mahalanobis distance scores more than 10 AUC lower on average, while this trend is reversed for CIFAR-10, where the Mahalanobis distance outperforms kNN and LOF with $k = 2$ by over 4 points. Yet, in both cases our method achieves competitive results, scoring the highest on CIFAR-10, and third best on the non-image evaluations.

Overall, our method is best across the 20 experiments ran, indicating it to be a promising approach for unsupervised out-of-distribution detection.

References

[1] L. Doorenbos, R. Sznitman, and P. Márquez-Neila, "Data invariants to understand unsupervised out-of-distribution detection," *arXiv preprint arXiv:2111.13362*, 2021.

[2] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *Advances in neural information processing systems*, vol. 31, 2018.

[3] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PloS one*, vol. 11, no. 4, p. e0152173, 2016.

Funding: This work was funded by the Swiss National Science Foundation (SNSF), research grant 200021_192285 "Image data validation for AI systems".