HAct: Out-of-Distribution Detection with Neural Net Activation Histograms

Sudeepta Mondal
Raytheon Technologies Research Center
411 Silver Lane, East Hartford, CT 06118
sudeepta.mondal2@rtx.com

Ganesh Sundaramoorthi
Raytheon Technologies Research Center
411 Silver Lane, East Hartford, CT 06118
ganesh.sundaramoorthi@rtx.com

Abstract

We propose a simple, efficient, and accurate method for detecting out-of-distribution (OOD) data for trained neural networks, a potential first step in methods for OOD generalization. We propose a novel descriptor, HAct - *activation histograms, for OOD detection, that is, probability distributions (approximated by histograms) of output values of neural network layers under the influence of incoming data. We demonstrate that HAct is significantly more accurate than state-of-the-art on multiple OOD image classification benchmarks. For instance, our approach achieves a true positive rate (TPR) of 95% with only 0.05% false-positives using Resnet-50 on standard OOD benchmarks, outperforming previous state-of-the-art by 20.66% in the false positive rate (at the same TPR of 95%). The low computational complexity and the ease of implementation makes HAct suitable for online implementation in monitoring deployed neural networks in practice at scale.*

1. Introduction

Machine learning (ML) systems are typically constructed under the assumption that the training and test sets are sampled from the same statistical distribution. However, in practice, that is often not the case. For instance, data from new classes different from training may appear in the test set. In these cases, the ML system would perform unreliably, with possible high confidence on erroneous outputs [4]. It is thus desired to construct techniques so that the ML system can generalize to such *out-of-distribution* (OOD) data. To generalize to OOD data, a first step in a system to adapt to such OOD data may involve the detection of the OOD data, called the *OOD detection problem*. This has become a problem of significant recent interest in machine learning and computer vision [21], as it is important in the deployment of systems in practice.

Recent state-of-the-art (SoA)[15, 5, 1, 16] in OOD detection for neural networks has focused on identifying descriptors of the data that can distinguish between OOD and in-distribution (ID) data. Descriptors from the data that are sufficiently different from corresponding training data descriptors are considered to be from OOD data. Because the network computes statistics of the data layer by layer to determine features that are relevant for the ML task, many recent works have hypothesized that computing functions of such statistics can be used to identify OOD data. Indeed, such approaches have led to SoA performance. One such popular approach [15] determines that a threshold of the output of activations in the penultimate layer of classification convolutional neural networks (CNNs) is an effective descriptor. This approach has been generalized to other layers [5] and several other recent works [16, 1] have built upon this idea of building functions formed by thresholds of network activations. While these approaches have demonstrated SoA performance on large-scale datasets in an efficient manner that has the potential to be deployed in real-world systems, performance still needs to be advanced for use in applications such as safety-critical systems.

In this work, we introduce novel descriptors for OOD detection that are simple and efficient to compute and lead to a significant improvement in performance compared to state-of-the-art. We show that effective descriptors for OOD are probability distributions of the output values of neural network layers. We show how these descriptors can be incorporated within an efficient OOD detection algorithm from an existing trained network. Our specific contribution is that we introduce a novel descriptor (HAct) that can be used in OOD detection, i.e., the probability distribution of the output of a layer within a neural network. When combining this descriptor over multiple layers in an OOD detection framework, the resulting technique out-performs existing state-of-the-art as demonstrated on multiple benchmark datasets.

2. Related work

We highlight related work in OOD detection and refer the reader to Yang et al. [21] for a detailed survey. Several methods determine OOD data by comparing the data at test-time to the training dataset. For efficiency, this is typically implemented with auto-encoders [23], in which the auto-encoder is trained with ID data, which allows them to learn

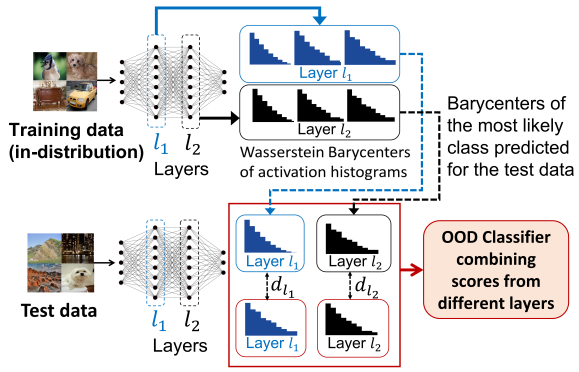

Figure 1: **Schematic of Activation Histogram (HAct) based OOD detection.** HAct involves a preparation phase in which barycenters of activation histograms for each class in the training dataset are calculated. During online operation, the HAct is computed for incoming data, and the distance to the most likely barycenter is thresholded.

the distribution of the training data. Thus, any data that is OOD would have a high reconstruction error, and thus the reconstruction error is thresholded to identify OOD data. While effective, this approach may characterize data that is different from training data as OOD, yet the network might still be able to generalize to that data. Other approaches aim to model uncertainty of a network on data, and characterize high-uncertainty data as OOD. There are several methods to determine uncertainty, e.g., ensemble methods [14] that measure the divergence of an ensemble of networks on data, test-time augmentation approaches [19] that measure the divergence of the network from augmented versions, uncertainty of network confidence scores [13], and Bayesian approaches [7] that treat weights in the network as probability distributions and calculate the output as the resulting distribution.

More recently, current SoA [15, 5, 1, 16] has sought to construct descriptors of the trained network and the incoming data to distinguish between OOD and ID data. In [15], a threshold of activation outputs in the penultimate layer of a network is used as a descriptor for OOD detection. This is used to compute the energy score [11] (see [12] for an alternative score); large values of the score are OOD. Djurisic et al. [5] generalizes the approach of Sun et al. [15] by thresholding not just in the penultimate layer but multiple feature layers, improving performance. Ahn et al. [1] also thresholds activation outputs, but uses the total number of activated features as a descriptor, and uses pruning to remove un-important parts of the network (see also [16] for a related idea of sparsification) for OOD and this outperforms the results of Djurisic et al [5]. Other descriptors of the trained network and the incoming data for OOD detection are topological descriptors [10], which are computed at dense layers. While effectively demonstrated on small networks, so far this approach has not scaled to large networks and datasets demonstrated in state-of-the-art (e.g., [15, 5]). Our approach computes a descriptor as a function of the network and incoming data, but instead, we show that the distributions of outputs of layers in a network are an alternative and effective descriptor for OOD detection, improving state-of-the-art while being simple and efficient.

3. Method for OOD Detection

In this section, we present our novel approach for OOD detection. We start by presenting our novel descriptor, called *Activation Histograms* (HAct), for OOD detection, computed from the trained network and the incoming data. We then show how this descriptor can be integrated within an OOD detection procedure. To illustrate the principles, we focus on OOD detection within the classification task.

3.1. Activation Histogram (HAct) Descriptors

Given a trained neural network F, we define a descriptor for a layer l described by a linear operation within the network (e.g., linear or convolutional layers). Let x be the input tensor to layer l, and W be the weight tensor for layer l. We consider the layer to be formed by a linear operation as defined by $y_i = \sum_j W_{ij} x_j$ where i, j represent (multi-dimensional) indices of the tensors (biases are not used in our approach). We consider the *activation weights*, formed from scalar components of intermediate computations of layer l, i.e., $A_{ij} := |W_{ij} \cdot x_j|$, which describes how much the j-th coordinate of the input observation x activates the ij-th unit of the weight tensor. Our descriptor, called the *activation histogram*, for layer l is the probability distribution of the elements of A, i.e., A_{ij}, which considers x_j to be a random variable. We approximate the probability distribution of A_{ij} by computing a histogram of *all* the activation weights, which we denote as h, defined as follows:

$$h_k = \frac{1}{n} \sum_{i,j} \mathbb{1}_{[\alpha_k, \alpha_{k+1})}(A_{ij}), \tag{1}$$

where n is the number of activation weights (elements of A), $0 \leq \alpha_0 < \alpha_1 < \ldots < \alpha_m$ is the partition of the range space of the weights, and $\mathbb{1}$ denotes the indicator function. The partition is fixed, except $\varepsilon := \alpha_0$, which we consider a hyper-parameter and is tuned for OOD accuracy. The vector $h = (h_k)$ is considered the OOD descriptor for layer l. For the sake of simplicity of notation, we do not indicate the dependence of h on l, but that is understood. In the next section, we will make use of multiple histograms at several layers for OOD detection. For CNNs, we will use both the dense layer used for classification and convolutional layers.

We typically choose $\varepsilon > 0$ as in the (deep) layers we consider, the input x is usually sparse, which would mean the histogram is overly weighted in the first bin. We thus

use a simple thresholding operation with ε to decrease this influence of zero outputs. We study the choice in the experiments, which do not show sensitivity for reasonable choices.

3.2. OOD Detection With Activation Histograms

We now specify how one can use the activation histograms in the previous section to perform OOD detection. We assume a trained neural network, F. For simplicity, we will assume F is trained for classification. Our procedure is similar to the framework of Lacombe et al. [10], but we will use our activation histograms rather than topological descriptors to demonstrate the effectiveness of our new descriptors. The simple observation underlying our approach is that activation histograms change for OOD data compared to ID data, and therefore, our approach seeks to detect such changes in activation histograms. Figure 1 gives a schematic overview of our approach.

The procedure first consists of preparing the OOD detector using the training data set (or a subset of it) of the trained network, and then after the preparation and during online operation, the OOD detector no-longer requires the training set. In the preparation step, an average activation histogram $\bar{h}^c$ is computed for each classification category c by computing the average of activation histograms over all data in that category, i.e.,

$$\bar{h}^c = \arg\min_{h} \sum_{\{x_{train}:F(x_{train})=c\}} D(h, h(x_{train})), \quad (2)$$

where $h(x_{train})$ is the activation histogram of x_{train}, and D is a metric between probability distributions. During online operation for OOD detection, the most likely category c^* for the test data x_{test} is chosen, the activation histogram for x_{test}, $h(x_{test})$, is computed, and if the distance between $\bar{h}^{c^*}$ and $h(x_{test})$ exceeds a threshold, the data x_{test} is considered OOD for the network F. More formally, our OOD detection using activation histograms for a given layer l is given by

$$d_l(x) = \begin{cases} \text{OOD} & D(\bar{h}^{F(x)}, h(x)) > \tau \\ \text{ID} & D(\bar{h}^{F(x)}, h(x)) \leq \tau \end{cases}. \quad (3)$$

To combine information from multiple layers for an OOD detector, we define the overall detector to return ID if all the OOD detectors for all layers return ID, otherwise, the detector returns OOD.

In the above formulation, one needs to choose an appropriate metric D to define the average descriptors for training categories, and the distance between test and training histograms. Following [10], we choose the Wasserstein metric, which was shown to be effective. In this case, $\bar{h}^c$ are referred to as Wasserstein barycenters. Algorithm 1 shows the pseudo-code of our proposed OOD detector inference and preparation.

Algorithm 1 Pseudo-Code for HAct OOD detection

Inputs : Training dataset (X, y), where X denotes inputs, and $y \in \{1, 2, \cdots, k_0\}$ are class label outputs. Neural network F trained on (X, y).
Preparation step:
for layers $l_1, l_2, \cdots l_n$ **do**
    for $k = 1$ to k_0 **do**
        Calculate the Wasserstein barycenter $\bar{h}^c$ (2)
    end for
end for
Online Operation: For a test observation x
for layers $l_1, l_2, \cdots l_n$ **do**
    Calculate HAct, $h(x)$ (1)
    Calculate the detector $d_l(x)$ using (3)
end for
Define an overall OOD detector $d(x)$ by combining $d_l(x)$ over layers, which returns ID if $d_l(x)$ for all l return ID, otherwise, $d(x)$ returns OOD.

4. Experiments

Datasets: We test our method on benchmark datasets for OOD detection. The first benchmark is derived from CIFAR-10 [9]. Networks are trained on 8 categories of CIFAR-10 training set and the remaining two categories are considered OOD (denoted by CIFAR-2). The test set used is the CIFAR-10 test set. The second set of datasets (used in [15]) involves ImageNet [3] for ID and the Places [22], SUN [20], iNaturalist [18] and Textures [2] for OOD data. For all our experiments, we use 100 examples from each training class in the ID data for the HAct preparation step.

Metrics: We benchmark HAct using standard metrics for detection used in SoA [15]. The first is false positive rate (FPR) at a true positive rate (TPR) of 95%, denoted as FPR95 (lower is better), and the second is the area under the ROC curve, denoted as AUROC (higher is better).

Hyper-parameter tuning: The main hyper-parameter that needs to be tuned in our algorithm is ε, the threshold for zero values in the activation histograms. We performed a grid search using the values of $\{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ using the CIFAR-10 benchmark on the CNN-2 architecture described below, and determined the optimal value of 10^{-3} in terms of the FPR95 metric. Other values did not significantly change the FPR95. This can vary with different datasets and the architecture. However, we found that $\varepsilon = 10^{-3}$ performed well on other datasets and architectures so we used this in all experiments below.

Layer Choice for Activation Histograms: As discussed in the previous section, HAct descriptors can be computed at any linear layer. We explore this choice by experimenting with the last dense layer, the convolution layer just before the dense layer, and the combination of both us-

Table 1: Results on Resnet-50 with Imagenet-1k as ID and different benchmark OOD datasets.

Methods	OOD Datasets									
	Places		SUN		iNaturalist		Textures		Average	
	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
ReAct [15]	33.85	91.58	24.20	94.20	20.38	96.22	47.30	89.80	31.43	92.95
ASH-B [5]	33.45	92.31	22.08	95.10	14.21	97.32	21.17	95.50	22.73	95.06
DICE [16]	46.49	87.48	35.15	90.83	25.63	94.49	31.72	90.30	34.75	90.77
DICE + ReAct [16]	36.86	90.67	25.45	93.94	18.64	96.24	28.07	92.74	27.25	93.40
LINe [1]	28.52	92.85	19.48	95.26	12.26	97.56	22.54	94.44	20.70	95.03
HAct (Ours)	**0.08**	**99.91**	**0.03**	**99.96**	**0.04**	**99.95**	**0.02**	**99.31**	**0.04**	**99.78**

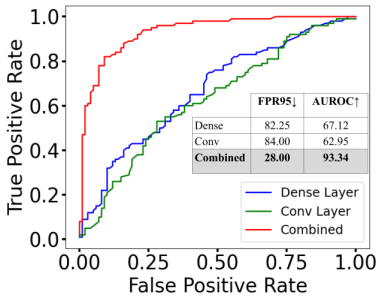

Figure 2: Using HAct descriptors over multiple layers improves OOD detection. Experiment on CIFAR OOD dataset and a shallow CNN (CNN-2 : see text for details).

ing the combined detector as described in the previous section. We experiment on the CIFAR OOD benchmark with the CNN-2 architecture (described below, further details in supplementary material). Results shown in Figure 2 indicate that the combination of HAct from both layers leads to better OOD detection performance, which we have observed on multiple other datasets.

Comparison to SoA on Large-Scale OOD Datasets: We experiment with ResNet-50 [8] trained on ImageNet-1k data, and benchmark on the OOD datasets used in [15]. We compare to the most SoA - ReAct [15], ASH-B [5], DICE [16], DICE+ReAct[16] and LINe [1]. Table 1 shows the results of the comparison (note that HAct is computed at the last classification layer which is a dense layer and the convolutional layer preceding the dense layer). We consistently out-perform all competing methods by a wide margin, nearly achieving zero FPR95 and nearly 100% AUC. Our result is a 20.66% improvement in FPR95 over the next best method [1], demonstrating the utility of activation histograms. HAct takes ∼60ms for inference on an NVIDIA GeForce RTX 3080 GPU, which is comparable to SoA.

Comparison with Topological Descriptors for OOD: We compare our approach to topological descriptors [10]. Note that this approach has not been demonstrated on large-scale OOD benchmarks [15] for classification, because of the large computational complexity. Since we use a similar framework, but different descriptors, we compare against [10] to illustrate our computational advantage. The topological approach can scale to only dense layers in large CNNs due to speed limitations. Even in this case, HAct is ∼50 times faster on ResNet-50 trained on ImageNet-1k on the last dense layer. Due to the nonlinear increase of computational complexity of [10] as a function of the number of activation weights, it does not scale to convolutional layers. Besides the advantage of speed, our approach is also better in OOD detection. To demonstrate this, we compare against [10] on smaller datasets (FMNIST/MNIST benchmark and the CIFAR-8/CIFAR-2 benchmark) and architectures (CNN-1 [17] and CNN-2 [6] which are shallow CNNs - see supplementary material for full specifications) considered in [10]. The descriptors for HAct are only computed on the final dense layer for comparison. Results are shown in Table 2, and indicate that our method significantly outperforms the topological descriptors[10].

Table 2: Comparison of our HAct descriptors with topological descriptors for OOD detection [10] from dense layers.

Model	OOD	Method	Metric	
			FPR95 ↓	AUROC ↑
CNN-1	FMNIST	Topology	46.50	92.20
		HAct (Ours)	**12.50**	**95.50**
CNN-2	CIFAR-2	Topology	86.00	55.39
		HAct (Ours)	**82.25**	**67.12**

5. Conclusions

We introduced a new descriptor HAct (activation histograms of linear layer outputs) of incoming data to a neural network that are effective in distinguishing OOD from ID data. The combination of descriptors from the dense and the preceding convolutional layer in CNNs was shown effective for OOD detection, out-performing SoA. The simplicity and efficiency of HAct imply the potential to be deployed in practical systems. Given the generality, future work will involve application to other classes of neural nets.

A current limitation of our method is the need for accessing training data in the preparation step before inference, which may preclude some applications. However, there may be methods to approximate barycenters from the trained network without the need for accessing training data, which is a subject of future work.

References

[1] Yong Hyun Ahn, Gyeong-Moon Park, and Seong Tae Kim. Line: Out-of-distribution detection by leveraging important neurons, 2023.

[2] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.

[3] 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, 2009.

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

[5] Andrija Djurisic, Nebojsa Bozanic, Arjun Ashok, and Rosanne Liu. Extremely simple activation shaping for out-of-distribution detection. 2022.

[6] Pytorch examples. Cifar-10 classification example network. `https://www.tensorflow.org/tutorials/images/cnn`, Last accessed on 08-26-2023.

[7] Ethan Goan and Clinton Fookes. *Bayesian Neural Networks: An Introduction and Survey*, pages 45–87. Springer International Publishing, Cham, 2020.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 630–645, Cham, 2016. Springer International Publishing.

[9] Alex Krizhevsky. Learning multiple layers of features from tiny images.

[10] Théo Lacombe, Yuichi Ike, Mathieu Carriere, Frédéric Chazal, Marc Glisse, and Yuhei Umeda. Topological uncertainty: Monitoring trained neural networks through persistence of activation graphs, 2021.

[11] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020.

[12] Xixi Liu, Yaroslava Lochman, and Christopher Zach. Gen: Pushing the limits of softmax-based out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23946–23955, 2023.

[13] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks, 2018.

[14] Rahul Rahaman and alexandre thiery. Uncertainty quantification and deep ensembles. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 20063–20075. Curran Associates, Inc., 2021.

[15] Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations, 2021.

[16] Yiyou Sun and Yixuan Li. Dice: Leveraging sparsification for out-of-distribution detection. In *European Conference on Computer Vision*, 2022.

[17] TensorFlow tutorials. Mnist classification example network. `https://github.com/pytorch/examples/blob/main/mnist/main.py`, Last accessed on 08-26-2023.

[18] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8769–8778, 2018.

[19] Guotai Wang, Wenqi Li, Sébastien Ourselin, and Tom Vercauteren. Automatic brain tumor segmentation using convolutional neural networks with test-time augmentation. In Alessandro Crimi, Spyridon Bakas, Hugo Kuijf, Farahani Keyvan, Mauricio Reyes, and Theo van Walsum, editors, *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 61–72, Cham, 2019. Springer International Publishing.

[20] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010.

[21] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey, 2022.

[22] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018.

[23] Chong Zhou and Randy C. Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 665–674, New York, NY, USA, 2017. Association for Computing Machinery.