Code

1. Prerequisites

1.1. Environment

Please set up the environment as follows: tensorflow == 1.14.0, numpy == 1.20.1 and python == 3.7.1.

1.2. Datasets

Before running the codes, you need to download all the datasets from their corresponding links provided in the following. For the OoD test sets, we directly use the ones officially released by [2] in their repository in Github.

CIFAR-10: http://www.cs.toronto.edu/ kriz/cifar.html
CIFAR-100: http://www.cs.toronto.edu/ kriz/cifar.html
OoD datasets: https://github.com/ShiyuLiang/odin-pytorch

2. Running

2.1. Benchmarks and Robustness Exploration

The codes necessary for replicating our experimental results in benchmark and robustness exploration are placed in 'Code_LSR/Method/'. The default model architecture is the Wide-ResNet-28-10 provided in https://github.com/akshaymehra24/WideResnet. To use Dense-BC please manually replace the Wide-ResNet-28-10 with the implementation of Dense-BC in https://github.com/taki0112/Densenet-Tensorflow. After downloading the datasets, please transform them into the format of TF-record by running:

$$python \quad Make\_tfrecord.py$$

Here, we only provide the one to transform CIFAR-10. To transform CIFAR-100 and OoD datasets into the same organized TF-record, you only need to do slight modifications on this script. Next, to train the classifier serving as AV feature extractor, please directly run:

$$python \quad Train\_CIFAR10.py$$

Next, run the following command to train the OoD detection module of the proposed method:

$$python \quad Train\_CIFAR10\_OOD.py$$

Please run 'Evaluator_CIFAR10.ipynb' box by box to reproduce the results in Benchmark experiments. As for the results in robustness exploration, please simply modify the number of layer channels of classifier in above codes.

2.2. Ablation Study

Codes for replicating our experimental results in ablation study are placed in 'Code_LSR/Ablation_study/'. After downloading all the datasets, please transform them into the format of TF-record by modifying 'Make_tfrecord.py'. For the experimental results of latent space autoregression [1], we directly use the code officially released in https://github.com/aimagelab/novelty-detection. For the remaining models in ablation study, please first train an classifier using:

$$python \quad Train\_CIFAR100.py$$

Next, for the model of column 2^{nd} and 3^{rd} in Table.3 in main paper, please run

$$python \quad autoregressor\_AVfeature\_CIFAR100.py$$

'Evaluator_autoregressor_AVfeature.ipynb' is the corresponding evaluation script. A box-by-box running of it will show all the experimental results in column 3^{rd}. For the results in column 2^{nd}, please modify this script of assessing reconstruction accuracy by normalized L2 distance to L2 distance.

For the results in column 4^{th}, please run 'Evaluator_CE_AVfeature.ipynb' and 'CE_AVfeature_CIFAR100.py' in a similar way as above. The results in column 5^{th} and 6^{th} can be obtained by running the codes for benchmark.

References

[1] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara. Latent space autoregression for novelty detection. 2018. 1

[2] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017. 1