ReAct: Out-of-distribution Detection With Rectified Activations

Yiyou Sun
UW-Madison

Chuan Guo
Facebook AI Research

Sharon Yixuan Li
UW-Madison
Problem Introduction
**Closed-world:**
Training and testing distributions **match**

**Open-world:**
Training and testing distributions **differ**
Training examples: traffic signs

\[ p_i(x) = \frac{\exp(f_i(x))}{\sum_{j=1}^{N} \exp(f_j(x))}, \]

confidence: \( \max_{i} p_i \)

Why is OOD Detection hard?
Test time: **out-of-distribution example**

*Ideally: Low confidence in predicting as traffic sign*
Neural Networks Can Be Over-confident to \textit{Out-of-distribution} Examples

[Nguyen et al. 2015]

\textbf{In-Distribution (ID) Samples}

\begin{itemize}
  \item 0.99
  \item 0.98
  \item 0.94
  \item \ldots
  \item 0.97
\end{itemize}

\textbf{Out-of-distribution (OOD) Samples}

\begin{itemize}
  \item 0.85
  \item 0.89
  \item 0.92
  \item \ldots
  \item 0.82
\end{itemize}
Prior literature on post hoc OOD detection

ODIN
[Liakos et al. ICLR 2018]

Energy score
[Liu et al. NeurIPS 2020]

MSP
[Hendrycks et al. ICLR 2017]

Mahalanobis
[Lee et al. NeurIPS 2018]

They all use original activations!
We need to look deep inside the networks!

What’s the fundamental cause and mitigation of model overconfidence on OOD data?

Answering this requires carefully examining the **internal mechanism** by which the network is trained and evaluated.
Methodology
Problem Statement

- Supervised learning setting
  - input space: $X = \mathbb{R}^d$
  - label space: $Y = \{1, 2, \ldots, K\}$
  - a training dataset drawn i.i.d. from a joint distribution: $P = X \times Y$
  - training a neural network: $f(x; \theta)$

- In test time, test data can come from a different distribution
  - whose label set has not interaction with $Y$
  - should not be predicted by model

- OOD detection can be formulated by a binary classification problem

$$G(x; f) = \begin{cases} 0 & \text{if } x \sim D_{\text{out}} \\ 1 & \text{if } x \sim D_{\text{in}} \end{cases}$$
Set up

ID: ImageNet

Convolutional Layers

Feature Layer

Output layer $f(x)$

$z = h(x) \in \mathbb{R}^m$
Visualization of ID’s Activation

ID: ImageNet

$x$ → Convolutional Layers → $z = h(x) \in \mathbb{R}^m$

Feature Layer

Unit Indices
Visualization of OOD’s Activation

OOD: iNaturalist

feature layer $z = h(x) \in \mathbb{R}^m$
Comparison of Unit Activation

OOD data produces overly high activations
**ReAct: Rectified Activation**

1. We perform post hoc truncation to the unit activation at a certain threshold $c$.

   \[
   \bar{h}(x) = \text{ReAct}(h(x); c) \\
   \text{ReAct}(x; c) = \min(x, c)
   \]

2. The model output after ReACT is produced by a fully connected layer.

   \[
   f^{\text{ReAct}}(x) = W^\top \bar{h}(x) + b
   \]

3. During test, ReAct can be used by many OOD scoring functions.

   \[
   G_\lambda(x; f^{\text{ReAct}}) = \begin{cases} 
   \text{in} & S(x; f^{\text{ReAct}}) \geq \lambda \\
   \text{out} & S(x; f^{\text{ReAct}}) < \lambda 
   \end{cases}
   \]
ReAct: Rectified Activation

Simple and Effective

Before Rectification
FPR95: 55.72%

After Rectification
FPR95: 20.38%

Unit Indices

$$\bar{h}(x) = \text{ReAct}(h(x); c)$$

$$\text{ReAct}(x; c) = \min(x, c)$$
Experiments
Experiments Setting

Dataset

- ImageNet-1k
- iNaturalist
- SUN
- Places
- Textures

ImageNet-based OOD detection benchmark is more challenging than CIFAR, high-resolution images

[Huang and Li, CVPR 2021]
ReAct establishes the SOTA post hoc detection performance.
ReAct is compatible with various OOD scoring functions

Takeaway: ReAct can improve the performance on originally competitive methods.
Effect of Rectification Threshold

| Rectification percentile | FPR95 ↓ | AUROC ↑ | AUPR ↑ | ID ACC. ↑ | Rectification threshold $c$
|-------------------------|---------|---------|--------|-----------|------------------|
| No ReAct [32]           | 58.41   | 86.17   | 96.88  | 75.08     | $\infty$        |
| $p = 99$                | 44.57   | 90.45   | 97.96  | 75.12     | 2.25             |
| $p = 95$                | 35.39   | 92.39   | 98.37  | 74.76     | 1.50             |
| $p = 90$                | 31.43   | 92.95   | 98.50  | 73.75     | 1.00             |
| $p = 85$                | 34.08   | 92.05   | 98.35  | 72.91     | 0.84             |
| $p = 80$                | 41.51   | 89.54   | 97.91  | 71.93     | 0.72             |
| $p = 65$                | 74.62   | 74.14   | 94.39  | 67.14     | 0.50             |
| $p = 10$                | 74.70   | 57.55   | 86.06  | 1.22      | 0.06             |

Table 2: Effect of rectification threshold for inference. Model is trained on ImageNet using ResNet-50 [11]. All numbers are percentages and are averaged over 4 OOD test datasets.

**Notation $p$:** p% of the ID activations are less than the threshold we set.

**Takeaway:** Over-activation does compromise the ability to detect OOD data, and ReAct can effectively alleviate this problem.
Other Experiments

1. ReAct is effective on other network architectures

2. Applying ReAct on the penultimate layer is most effective

3. ReAct is effective on CIFAR Benchmarks

4. ReAct is effective on networks trained with different normalization mechanisms (GroupNorm and WeightNorm).

(See more details in the paper)
Discussion and Theoretical Insight
Why does OOD data trigger abnormal unit activation patterns?

Adding a BatchNorm layer (between weights and activation function)

\[ \text{BatchNorm}(z; \gamma, \beta, \epsilon) = \frac{z - \mathbb{E}_{\text{in}}[z]}{\sqrt{\text{Var}_{\text{in}}[z] + \epsilon}} \cdot \gamma + \beta \]
Why does OOD data trigger abnormal unit activation patterns?

**Takeaway:** Using mismatched BatchNorm statistics—that are estimated on ID data but blindly applied to the OOD data—can trigger abnormally high unit activations.
Why does OOD data trigger abnormal unit activation patterns?

**Oracle:** Using True Statistics

**Reality:** Using False Statistics

---

**Takeaway:** Our method favorably matches the oracle performance using the ground truth BN statistics.
Activation Reduction on OOD data is more than ID

\[ E_{in} [z_i - \bar{z}_i] = \phi \left( \frac{c - \mu}{\sigma_{in}} \right) \cdot \sigma_{in} - \left[ 1 - \Phi \left( \frac{c - \mu}{\sigma_{in}} \right) \right] \cdot (c - \mu) \]

\[ E_{out} [z_i - \bar{z}_i] = (1 - \epsilon)^2 \phi \left( \frac{c - \mu}{(1 - \epsilon) \sigma_{out}} \right) \cdot \sigma_{out} - (1 - \epsilon) \left[ 1 - \Phi \left( \frac{c - \mu}{(1 - \epsilon) \sigma_{out}} \right) \right] \cdot (c - \mu) \]

Key insight:

1. When \( \epsilon < 0 \) (OOD), ReAct has a greater reduction effect for activation.

2. If the sum of weight is positive in the final layer, we can prove ReAct leads more reduction for OOD’s output.
Summary

1. We introduce ReACT---a simple and effective OOD detection approach that utilizes activation truncation.

2. We provide important insights that abnormally high activations on OOD data can harm their detection and how ReACT effectively mitigates this issue.

3. We extensively evaluate ReACT on a suite of OOD detection tasks and establish a state-of-the-art performance among post hoc methods.
Thank you!

https://github.com/deeplearning-wisc/react