Learning ODIN
Amir Jevnisek and Shai Avidan

Out-of-Distribution Detection Goal is to identify samples outside of a predetermined distribution, which is defined as In-Distribution. At training time only normal data is available. Area Under ROC Curve (AUROC) and FPR95 are the standard evaluation metrics.

Problem Statement

Introduction

ODIN OOD Detector:

1. \[ S_i(x; T) = \frac{\exp(f_i(x)/T)}{\sum_{j=1}^C \exp(f_j(x)/T)} \]
2. \[ \hat{x} = x - \text{sign}(-\nabla_x \log S_i(x; T)) \]
3. \[ g(x; \theta, T, \varepsilon) = \begin{cases} 1 & \text{if } \max_j g_j(x; \theta, T) \leq \delta, \\ 0 & \text{if } \max_j g_j(x; \theta, T) > \delta, \end{cases} \]

\[ L_{GQ} = \sum_{j \neq \max} \frac{||\nabla x S_j||_1}{||\nabla x S_{\max}||_1} \]

OOD Algorithm

Algorithm 1 Learning ODIN

Inputs: (1) In-Distribution dataset \( D_{ID} \), (2) classifier \( f_0(x) \) with parameters \( \theta \)

Outputs: (1) An OOD detector, (2) An In-Distribution classifier.

1. Train \( f_0(x) \) with the following loss: \( L = L_{CE} + \lambda \cdot L_{GQ} \)
2. Create \( g(x) \) an ODD OOD-Detector on \( f_0(x) \)’s logits.
   return \( g(x) \) is the OOD-detector, \( f_0 \) is the In-Distribution classifier

OOD Performance

CIFAR-10 as ID: Adding GQ to any of the architectures improves results. Our approach achieves best results for all metrics compared to baseline models.

High Res. Images

GQ also works for High Resolution images (ImageNet). It is possible to use GQ for fine tuning, which lets us work with pre-trained models.

Gradient Distribution

When our loss term is added, the distribution for In-Distribution samples is wider than in the case of classical training (CE-loss).

ODIN is a prevailing OOD framework. ODIN assumes that the logit gap is maximal for In-Distribution samples.

We encode this assumption into the loss function. So we learn what ODIN is expecting.

Implementation

Code is available!