原文链接：2402.03502 (arxiv.org)

文章的核心想法：通过使用野生数据集（ID和OOD混杂）来模拟实际情境，从而加强模型对OOD数据的检测能力。

OOD检测的一个重要问题是--带标记的OOD数据训练样本太少了。（之前很多情况，例如ID和OOD数据集都打上对应的ID或者OOD标签情境太理想）我们不妨考虑直接拿野生数据集过来用，好处有这几个：

- 量大，容易获取，而且不需要人工剔除出OOD数据打上标签
- 贴合实际情境，练出来的效果好

文章就此剔除使用未标记的野生数据集训练的方法。

训练好ID分类器

这一步与之前的模型没有差别。先对模型进行预训练能够识别并分类ID样本，易知这一步使用的数据集是纯净的ID数据集，使用的损失函数选用ERM 经验损失最小化，应用到本模型里是这个：

$w_{S_{in}} \in arg_{w \in W} \quad min R_{S_{in}}(h_W)$

其中$R_{S_{in}}(h_W)$就是ERM：

$$R_{\mathcal{S}^{\text{in}}}(\mathbf{h_w}) = \frac{1}{n} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{S}^{\text{in}}} \ell(\mathbf{h_w}(\mathbf{x}_i), y_i), \tag{3}$$

依次我们预训练好网络。方便起见我们再提前定义好之后的部分要用的概念参考梯度：

$$\bar{\nabla} = \frac{1}{n} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{S}^{\text{in}}} \nabla \ell(\mathbf{h}_{\mathbf{w}_{\mathcal{S}^{\text{in}}}}(\mathbf{x}_i), y_i), \tag{4}$$

进行过滤分类得出candidate outliers

这里就是这篇论文的核心--既然我们做的是OOD检测，但是又没有现成的OOD标签数据，那我们就从现成的数据中找出outliers作为模拟OOD，以此再来学习OOD和ID之间的差别。而我们找出

candidate outliers的方法就是通过一个新定义的指标--filter score。

对于每一个数据点我们先定义好它的参考梯度向量（其实就是数据点的梯度向量减掉之前我们提到的参考梯度向量）写作：

$vector_i = [\nabla \ell(h_{w_{S_{in}}}(\tilde{x}_i), \hat{y}_{\tilde{x}_i}) - \bar{\nabla}]$

其中$\bar{\nabla}$是刚才我们定义的参考梯度向量，也就是ERM对参数求导组成的参考向量。

然后我们把所有数据点的参考梯度向量合起来就是这个模型的梯度矩阵G：

$$\mathbf{G} = \begin{bmatrix} \nabla \ell(\mathbf{h}_{\mathbf{w}_{\mathcal{S}^{\text{in}}}}(\tilde{\mathbf{x}}_1), \widehat{y}_{\tilde{\mathbf{x}}_1}) - \bar{\nabla} \\ \cdots \\ \nabla \ell(\mathbf{h}_{\mathbf{w}_{\mathcal{S}^{\text{in}}}}(\tilde{\mathbf{x}}_m), \widehat{y}_{\tilde{\mathbf{x}}_m}) - \bar{\nabla} \end{bmatrix}^{\top},$$

对这个矩阵求奇异向量v，接着我们就可以得到每个样本点对应的filter score:

$$\tau_i = \left\langle \nabla \ell(\mathbf{h}_{\mathbf{w}_{\mathcal{S}^{\text{in}}}}(\tilde{\mathbf{x}}_i), \widehat{y}_{\tilde{\mathbf{x}}_i}) - \bar{\nabla}, \mathbf{v} \right\rangle^2,$$

至于奇异向量的计算方法就是这个：

$$\mathbf{v} \in \underset{\|\mathbf{u}\|_2=1}{\arg\max} \sum_{\tilde{\mathbf{x}}_i \in \mathcal{S}_{\text{wild}}} \left\langle \mathbf{u}, \nabla \ell(\mathbf{h}_{\mathbf{w}_{\mathcal{S}^{\text{in}}}}(\tilde{\mathbf{x}}_i), \widehat{y}_{\tilde{\mathbf{x}}_i}) - \bar{\nabla} \right\rangle^2.$$

下一步我们的任务就是寻找一个合适的阈值T，阈值T令ID数据的分数小于这个阈值，OOD的分数大于这个阈值。

得到阈值T之后我们可以得到训练过程中的模拟OOD--candidate outliers，接着我们把得到的模拟OOD送进训练将要用上的OOD分类器即可。

使用ID和candidate outliers训练OOD分类器

这一步很简单了，相当于先前使用标签数据训练一个二分类器，分类目标是：

$$
\begin{aligned}
R_{\mathcal{S}^{\text{in}},\mathcal{S}_T}(\mathbf{g}_{\boldsymbol{\theta}}) &= R^{+}_{\mathcal{S}^{\text{in}}}(\mathbf{g}_{\boldsymbol{\theta}}) + R^{-}_{\mathcal{S}_T}(\mathbf{g}_{\boldsymbol{\theta}}) \\
&= \mathbb{E}_{\mathbf{x}\in\mathcal{S}^{\text{in}}}\ \mathbb{1}\{\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{x}) \leq 0\} + \mathbb{E}_{\tilde{\mathbf{x}}\in\mathcal{S}_T}\ \mathbb{1}\{\mathbf{g}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}) > 0\}.
\end{aligned} \tag{8}
$$

至此方法结束。

从中我们可以看到该方法主要好处有：

- 无需人工手动标记大量OOD样本，简单易用
- 所用的数据集来自野生数据集，OOD分布情况贴合实际，因此训练的模型对实际情境预测效果很好

方法的核心就是克服了传统方法中标记OOD样本不具代表性，而且数量少的缺点，既然我们训练的数据集中缺少OOD，那我们可以先模拟一堆OOD样本出来，然后将问题转化为监督学习。