Impact of adversarial sparsity as an auxiliary metric in adversarial robustness

Виконав: студент 2-го року навчання
Освітньо-наукової програми
“Прикладна математика”, 113
Кузьменко Дмитро Олександрович

Керівник Швай Н. О.
кандидат фіз.-мат. наук

Introduction

- Olivier and Raj, in their work "How many perturbations break this model? Evaluating robustness beyond adversarial accuracy", arXiv:2207.04129, Aug. 2022, propose an auxiliary metric named Adversarial Sparsity (AS) for evaluating robustness of deep learning models.
- AS is a metric that gauges the usual size required for a subset of adversarial examples Δ to include at least one adversarial perturbation that causes that model to misclassify the sample.
- More formally, we consider a series of expanding subsets of Δ: with $m_1 < m_2$, $\emptyset \subseteq \Delta^{m1} \subseteq \Delta^{m2} \subseteq \Delta$. We can define adversarial sparsity relative to Δ^m as:

$$AS(f, x, \varepsilon, \Delta^m) := inf\{m, \Delta^m \cap Adv(f, x, \varepsilon) \neq \emptyset\} \quad (1)$$

- If a distribution D of such sequences is provided, we can define adversarial sparsity as the expected value of AS:

$$AS_{full}(f, x, \varepsilon) := E_{(\Delta m)\sim D}[AS(f, x, \varepsilon, \Delta^m)] \quad (2)$$

The goal of the research

- Adversarial accuracy (AA) is traditionally considered the main metric for evaluating adversarial defenses, which involves calculating the fraction of correctly classified perturbed samples to the total number of samples.
- We incorporate evaluation of both adversarial accuracy and adversarial sparsity in out research, with the main focus on the latter.
- **The goal of this work** is to study sparsity in-depth, tend to the authors' remarks on areas for improvement, and attempt to produce novel valuable observations for the domain of adversarial robustness.

L_∞ Sparsity Algorithm

Algorithm 2 L_∞ sparsity, binary search

Require: model f, point $(x, y), K \in \mathbb{N}, N \in \mathbb{N}$

```
k ← 0
i ← 0
while i ≤ N do
    m_0^i ← n_pixels * 0.5ε, m_1^i ← n_pixels * 6ε, (u_i, σ_i), ~ U(S^n)
    i ← i + 1
end while
while k ≤ K do
    i ← 0
    while i ≤ N do
        m^i ← (m_0^i + m_1^i)/2                          ▷ binary search
        x_adv ← PGD_{m^i}(f, x, y, (u_i, σ_i))
        if f(x_adv) ≠ y then
            m_1^i ← m^i
        else
            m_0^i ← m^i
        end if
        i ← i + 1
    end while
    k ← k + 1
end while
return (1/n) Σ_0^n m_i
```

Algorithm 2. The L_∞ sparsity computation using a binary search step is suboptimal in terms of time complexity, according to the authors.

Proposed approach: L_∞ Sparsity with n-Ary hybrid search

The authors mention that while attacks over multiple directions can be computed in batches, binary search constitutes the major bottleneck of this algorithm.

We tested the speed and approximation accuracy of traditional L_∞ sparsity computation, and attempted to modify the bottle-neck part of the algorithm with hybrid n-Ary search to speed up the process while retaining the qualitative approximation of sparsity.

Algorithm 3. Our approach involves two phases - n-Ary search phase that uses higher arity to quickly narrow down the range of parameters *m*; and the binary search phase which uses more directions for refined sparsity approximation.

Algorithm 3 (Proposed) L_∞ sparsity, n-Ary hybrid search

Require: model f, point (x, y), $K \in \mathbb{N}, N \in \mathbb{N}, arity \in \mathbb{N}, 1 \leq steps_{arity} < 10$

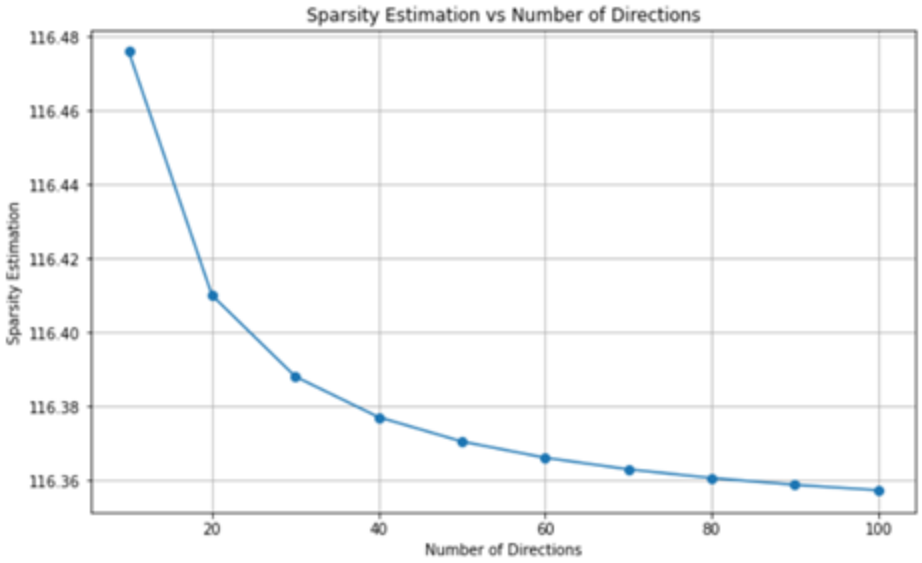
$i, j, k \leftarrow 0$
$N_{ary} = \frac{N}{arity}$
while $i \leq N, j \leq N_{ary}$ **do**
  $m_0^i, m_0^j \leftarrow n_{pixels} * 0.5\varepsilon, m_1^i, m_1^j \leftarrow n_{pixels} * 6\varepsilon, (u_i, \sigma_i), (u_j, \sigma_j) \sim U(S^n)$
  $i \leftarrow i + 1$
  $j \leftarrow j + 1$
end while
while $k \leq steps_{arity}$ **do**
  $j \leftarrow 0$
  while $j \leq N_{ary}$ **do**
    while $1 \leq arity_i \leq arity$ **do**
      $m_{arity_i}^i \leftarrow m_0^j + \frac{arity_i(m_1^j - m_0^j)}{arity}$ ▷ n-Ary search phase
    end while
    $x_{adv} \leftarrow PGD_{m^j}(f, x, y, (u_j, \sigma_j))$
    if $f(x_{adv}) \neq y$ **then**
      $m_1^j \leftarrow m^j$
    else
      $m_0^j \leftarrow m^j$
    end if
    $j \leftarrow j + 1$
  end while
  $k \leftarrow k + 1$
end while
while $k + steps_{arity} \leq K$ **do**
  $i \leftarrow 0$
  while $i \leq N$ **do**
    $m^i \leftarrow \frac{m_0^i + m_1^i}{2}$ ▷ binary search phase
    $x_{adv} \leftarrow PGD_{m^i}(f, x, y, (u_i, \sigma_i))$
    if $f(x_{adv}) \neq y$ **then**
      $m_1^i \leftarrow m^i$
    else
      $m_0^i \leftarrow m^i$
    end if
    $i \leftarrow i + 1$
  end while
  $k \leftarrow k + 1$
end while
return $\frac{1}{n} \sum_0^n m_i$

Experiments

- **Dataset and threat-model selection:** 50 class-balanced samples from CIFAR-10 with L_∞ distance and ε of 8/255.
- **Computational power**: a single RTX 3060 GPU.
- **Adversarial model repositories:** RobustBench and timm.
- **Models used**: a lightweight untrained vision transformer MobileViTv2 (1.12M); XCiT-M12 (46M) a thoroughly trained cross-variance image transformer; and a state-of-the-art CIFAR-10 benchmark Wang2023Better_WRN_70-16 (267M), is a diffusion probabilistic model with wide ResNet backbone trained with TRADES.

Model	Model size, mil params	Search configuration	Robust Acc., %	Sparsity, pixels	Time, it/s	Time saved, %	Sparsity deviation, %
MobileViTV2	1.12	binary	0.12	62.4	13.1	-	-
MobileViTV2	1.12	3-ary, 7 steps	0.12	71.5	9.8	**34**	15
MobileViTV2	1.12	5-ary, 5 steps	0.10	64.9	11.6	12	**4**
WRN-70-16	267	binary	0.84	142.4	24.5		-
WRN-70-16	267	3-ary, 7 steps	0.84	150.4	18.8	**30**	6
WRN-70-16	267	5-ary, 5 steps	0.84	142.8	22.0	11	**0.3**
XCiT-M	46	binary	0.56	115.2	21.1	-	-
XCiT-M	46	3-ary, 7 steps	0.56	119.7	16.1	**31**	4
XCiT-M	46	5-ary, 5 steps	0.56	116.4	19.7	7	**1**

Table 1. The comparison of binary- and n-Ary-based sparsity computation configurations between MobileViTv2, WRN-70-16, and XCiT-M on 50 CIFAR-10 samples. Time consumption decrease is best with 3-ary, 7 steps and the overestimation of sparsity gained - with 5-ary, 5 steps (bold); the baseline, most accurate approximation of sparsity is considered to be achieved with binary configuration (underlined).



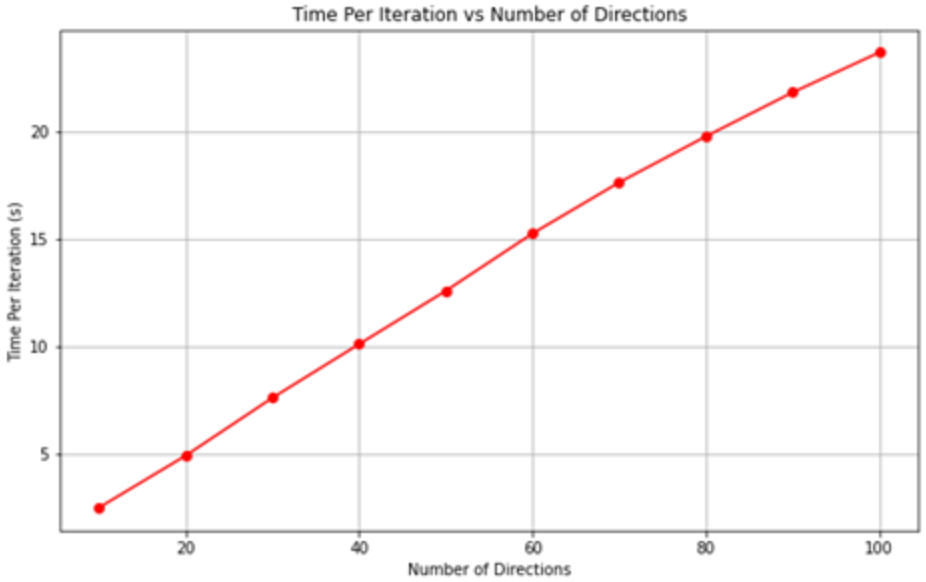


Figure 1. The plot of sparsity estimation against the number of directions (**left**). We can observe a sparsity overestimation pattern similar to hyperbolic decay. Choosing the right number of directions may yield an optimal trade-off between speed and estimation accuracy. The plot of computational time spent against the number of directions (**right**) is, expectedly, linear, therefore the selection of directions revolves around the number of samples in O(*N)* time complexity.

Main contribution and scientific novelty

- We have proposed the **novel hybrid n-Ary search-based approach** to approximating adversarial sparsity which yields significant increase in computation time, while not affecting the approximation to the high degree.
- In our analysis, we studied time preservation by comparing binary and n-Ary search setups on single sample under different numbers of directions. The average time gain for these experiments comprised **6.84%.**
- As number of directions *n* increases, the delta change in approximation of sparsity over time decreases. Therefore, it may be a potent goal to find an optimal value for *n* to optimize each experiment.
- We also observed that the impact of sparsity overestimation is more pronounced in less trained models. MobileViTv2 exhibited a **15%** overestimation, while WRN-70-16 showed only a **6%** overestimation.

Practical input and future work

- **Practical input:** the findings of this work shed light on the trade-offs involved in the hybrid n-Ary search approach and its impact on sparsity approximation and computational efficiency. The insights gained from this research can contribute to the development of more efficient and accurate algorithms for adversarial sparsity calculation, ultimately improving the understanding and robustness of models against adversarial attacks.
- **Future work**: experiments on finding the optimal value of absolute or relative decrease in margin (e.g. delta of sparsity change) can be conducted in order to pinpoint a well-balanced setup of parameters for a more convenient and easier testing of new hypotheses concerning adversarial sparsity.

Thank you for your attention