Adversarial Robustness through Randomization by Diversifying Vulnerabilities (DVERGE)

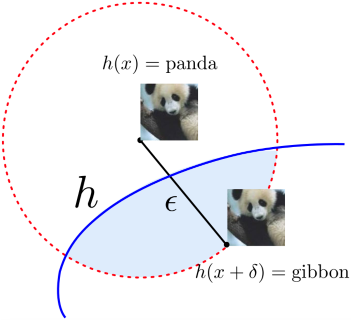
Mathilde Kretz, Alexandre Ngau

PSL Research University

December 5, 2023

Problem Setting: Adversarial Classification

- Train a classifier robust to l_∞ and l_2 attacks
- The attacks are aiming to perturb the correctly classified data to possibly find an overlapping class.



DVERGE (1) - Theoretical Approach of Vulnerability Diversity

Given the i-th model and its l-th layer, the **distilled feature** of a target (x, y) and a source pair (x_s, y_s) (where x, x_s are inputs and y, y_s are labels).

$$x'_{f_i^l}(x, x_s) = \arg\min_z \left\| f_i^l(z) - f_i^l(x) \right\|_2^2 \quad \text{s.t.} \quad \|z - x_s\|_\infty \leq \epsilon$$

$x'_{f_i^l}$ is the image that looks the most like the *source* image x_s and whose features are pushed towards those of x (the *target*, eg. y if we are on the last layer). It is high if $f_i^l(x)$ is a non-robust feature.

DVERGE (1) - Theoretical Approach of Vulnerability Diversity

The **vulnerability diversity metric** between two models i and j is then:

$$d(f_i, f_j) := \frac{1}{2}\mathbb{E}_{(x,y),(x_s,y_s),l}\left[\mathcal{L}_{f_i}(x'_{f_j^l}(x, x_s), y) \ + \mathcal{L}_{f_j}(x'_{f_i^l}(x, x_s), y)\right]$$

$d(f_i, f_j)$ effectively measures the vulnerability overlap between the two models.

The **learning objective** aiming to minimize the classification loss and maximizing the diversity toward the target y is the following:

$$\min_{f_i} \mathbb{E}_{(x,y)}\left[\mathcal{L}_{f_i}(x, y)\right] - \alpha \sum_{j \neq i} d(f_i, f_j)$$

DVERGE (1) - Practical Objective Function

The paper sheds light onto the possible divergence of the previous objective, they propose the following reformulation:

$$\min_{f_i} \mathbb{E}_{(x,y)} \left[\mathcal{L}_{f_i}(x, y) + \alpha \sum_{j \neq i} \mathbb{E}_{(x_s, y_s), l} \left[\mathcal{L}_{f_i}(x'_{f_j^l}(x, x_s), y_s) \right] \right]$$

The objective is now to minimize the *natural* loss and minimize the diversity towards the source output y_s ie. maximize the diversity not towards y_s.

Implementation

- Pre-trained three submodels with a clean dataset, resulting in diversified weak features throughout the submodels
- Trained the submodels using DVERGE (1) method
- Ensemble model that outputs the mean of the submodels' outputs is used for inference

Algorithm 1 DVERGE training routine for a N-sub-model ensemble.

`# initialization and pretraining`
for $i = 1, \dots, N$ **do**
    Randomly initialize sub-model f_i
    Pretrain f_i with clean dataset
`# round-robin feature diversification`
for $e = 1, \dots, E$ **do**
    Uniformly randomly choose layer l for feature distillation
    for $b = 1, \dots, B$ **do**
        $(X, Y) \leftarrow$ get batched input-label pairs
        $(X_s, Y_s) \leftarrow$ uniformly sample batched source input-label pairs
        `# get distilled batch for each model`
        for $i = 1, \dots, N$ **do**
            $X_i' := x'_{f_i^l}(X, X_s) \leftarrow$ non-robust feature distillation with Equation (1)
        `# calculate loss and perform SGD update for all sub-models`
        for $i = 1, \dots, N$ **do**
            $\nabla_{f_i} \leftarrow \nabla[\sum_{j \neq i} \mathcal{L}_{f_i}(f_i(X_j'), Y_s)]$
            $f_i \leftarrow f_i - lr \cdot \nabla_{f_i}$

Figure 1: DVERGE (1) Algorithm

Pretraining of the Baseline Models

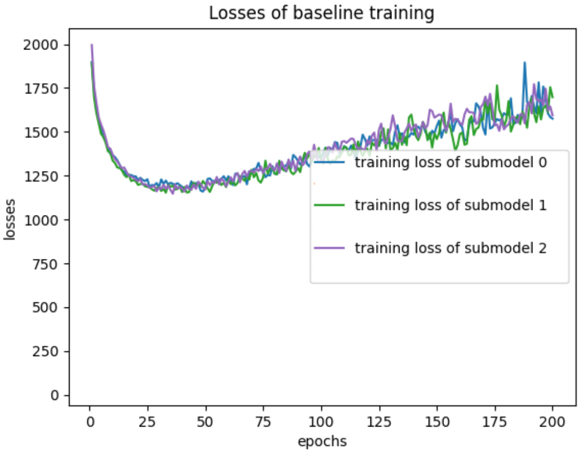


Figure 2: Submodels Pretraining

DVERGE Training for Three Submodels

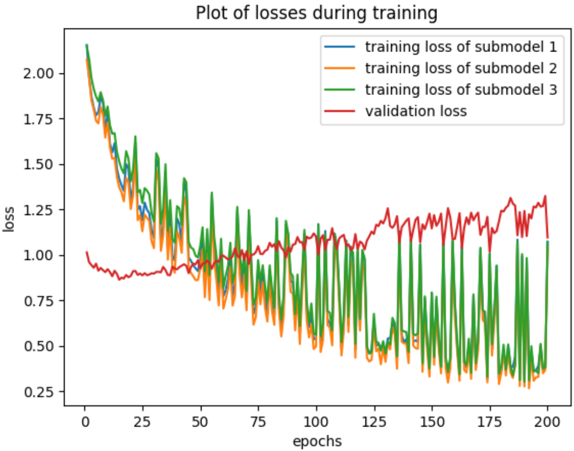


Figure 3: DVERGE (1) Training of Three Submodels Pretrained for 50 epochs (8 hours)

First Results

Using the three submodels trained following the DVERGE (1) method for 200 epochs, the ensemble model has the following performance :

- Natural Accuracy : 66.40
- FGSM Attack Accuracy : 31.44
- PGD L2 Norm Attack Accuracy : 43.94
- PGD Linf Norm Attack Accuracy : 6.44

N.B.: we had $\epsilon = 0.03$ *for all attacks,* $\alpha = 0.01$ *and* $num\_iter = 5$ *for the PGD attacks*

Next Steps

Different ideas to improve the results:

- Optimize the pretraining to have better results and diversify the features (train the base models for the optimal amount of epochs)
- Include adversarial training as proposed in the paper for the pretraining of submodels or the DVERGE training
- Perform a hyperparameter search to optimize the training in general

References

[1] H. Yang, J. Zhang, H. Dong, N. Inkawhich, A. Gardner, A. Touchet, W. Wilkes, H. Berry, and H. Li, "Dverge: Diversifying vulnerabilities for enhanced robust generation of ensembles," 2020.