Adversarial robustness through randomization
And some more...

Lucas GNECCO HEREDIA

CNRS, LAMSADE - Université Paris Dauphine - PSL

November 20, 2024

Table of Contents

Table of Contents

Problem setting: Standard classification

- Classification task: input space $\mathcal{X} \subset \mathbb{R}^d$ equipped with a distance d, typically ℓ_2 or ℓ_∞, and labels $\mathcal{Y} = \{1, \ldots, K\}$

Problem setting: Standard classification

- Classification task: input space $\mathcal{X} \subset \mathbb{R}^d$ equipped with a distance d, typically ℓ_2 or ℓ_∞, and labels $\mathcal{Y} = \{1, \ldots, K\}$
- True data distribution $\rho \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$

Problem setting: Standard classification

- Classification task: input space $\mathcal{X} \subset \mathbb{R}^d$ equipped with a distance d, typically ℓ_2 or ℓ_∞, and labels $\mathcal{Y} = \{1, \ldots, K\}$
- True data distribution $\rho \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$
- Error function, or 0-1 loss of a classifier $h : \mathcal{X} \to \mathcal{Y}$ defined as

$$\ell^{0\text{-}1}((x, y), h) = \mathbb{1}\{h(x) \neq y\}$$

Problem setting: Standard classification

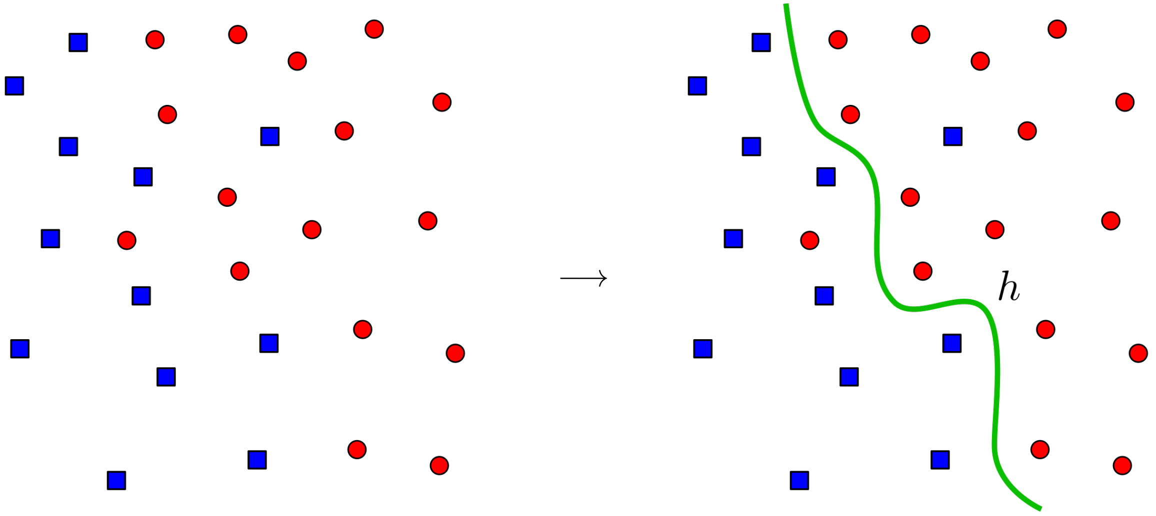
- Classification task: input space $\mathcal{X} \subset \mathbb{R}^d$ equipped with a distance d, typically ℓ_2 or ℓ_∞, and labels $\mathcal{Y} = \{1, \ldots, K\}$
- True data distribution $\rho \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$
- Error function, or 0-1 loss of a classifier $h : \mathcal{X} \to \mathcal{Y}$ defined as

$$\ell^{0\text{-}1}((x, y), h) = \mathbb{1}\{h(x) \neq y\}$$

- **Goal** Find $h : \mathcal{X} \to \mathcal{Y}$ within some family $\mathcal{H}$ with the lowest risk (highest accuracy).

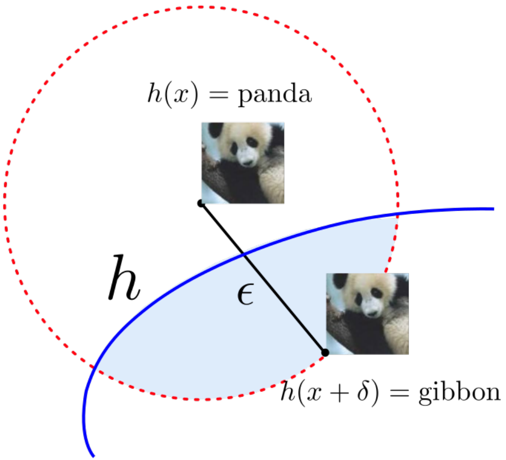
$$\mathcal{R}(h) = \mathbb{E}_{(x,y)\sim\rho}\left[\ell^{0\text{-}1}((x, y), h)\right] \qquad \text{(risk)}$$

Problem setting: Standard classification



Problem setting: Adversarial classification

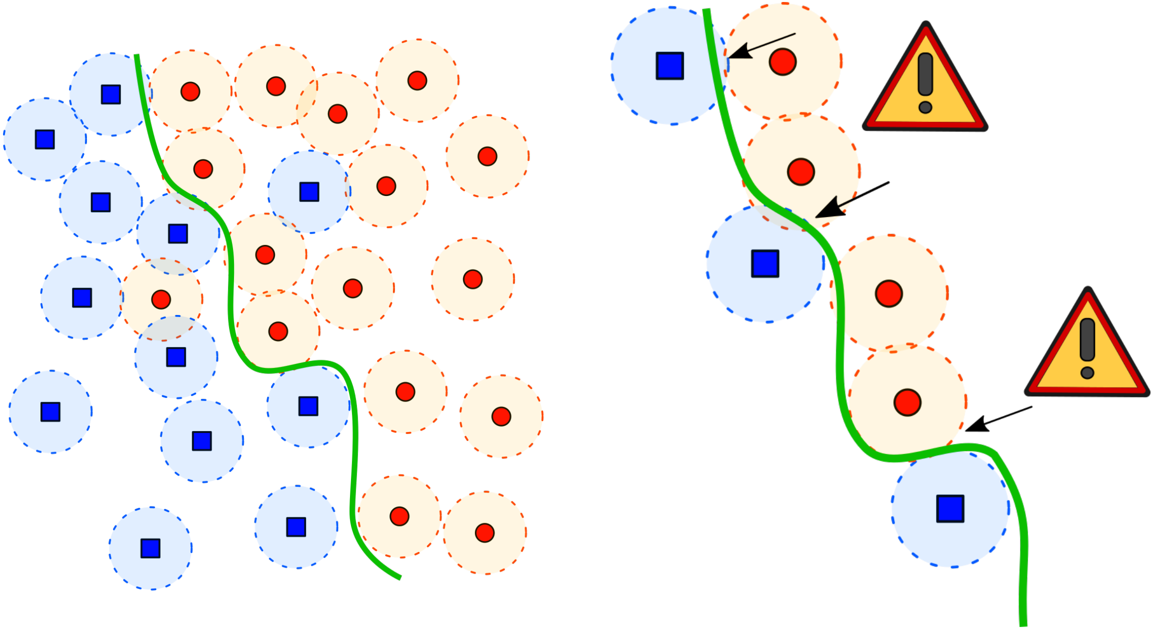
- Data perturbing adversary with budget ϵ can transport any x to $x' \in B_\epsilon(x) = \{x' \in \mathcal{X} \mid d(x, x') \leq \epsilon\}$ to induce an error.



Goal Find $h : \mathcal{X} \to \mathcal{Y}$ within some family $\mathcal{H}$ with the lowest **adversarial** risk (highest **robust** accuracy)

$$\mathcal{R}_\epsilon(h) = \mathbb{E}_{(x,y)\sim\rho}\left[\sup_{x' \in B_\epsilon(x)} \ell^{0\text{-}1}((x', y), h)\right]$$

Visualization (adversarial classification)



Randomized classifiers in the literature

Many previous works have proposed *stochastic* or *randomized* models as a way to improve robustness to adversarial attacks.

RANDOMIZED CLASSIFIERS IN THEORY

Intuitively, the output of a **randomized classifier** is not a label, but a *probability distribution* over labels.

Randomized classifiers in theory

Intuitively, the output of a **randomized classifier** is not a label, but a *probability distribution* over labels.

- Randomized: $\mathbf{h} : \mathcal{X} \to \Delta^K$.
- Deterministic: $h : \mathcal{X} \to \{1, \ldots, K\} \cong \{e_1, \ldots, e_K\} \subset \Delta^K$.

Randomized classifiers in practice

In practice, randomized classifiers involve randomized transformations of the **input** or **model**.

- Input noise injection [HRF19; Pin+19; Yu+21]

$$x \rightarrow \boxed{\text{sample noise } \eta \sim \mu} \rightarrow h(x+\eta)$$

- Weight noise injection or model sampling [HRF19; Pin+20; DS22; Wic+21; Dhi+18]

$$x \rightarrow \boxed{\text{sample model } h \sim \mu} \rightarrow h(x)$$

Randomized classifiers in practice

In practice, randomized classifiers involve randomized transformations of the **input** or **model**.

- Input noise injection [HRF19; Pin+19; Yu+21]

$$x \to \boxed{\text{sample noise } \eta \sim \mu} \to h(x+\eta)$$

- Weight noise injection or model sampling [HRF19; Pin+20; DS22; Wic+21; Dhi+18]

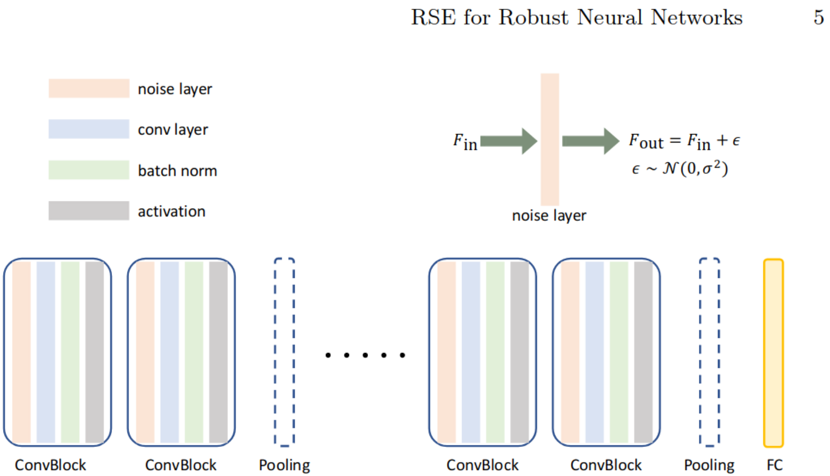
$$x \to \boxed{\text{sample model } h \sim \mu} \to h(x)$$

Most methods can be though as a distribution over some family of models...

Table of Contents

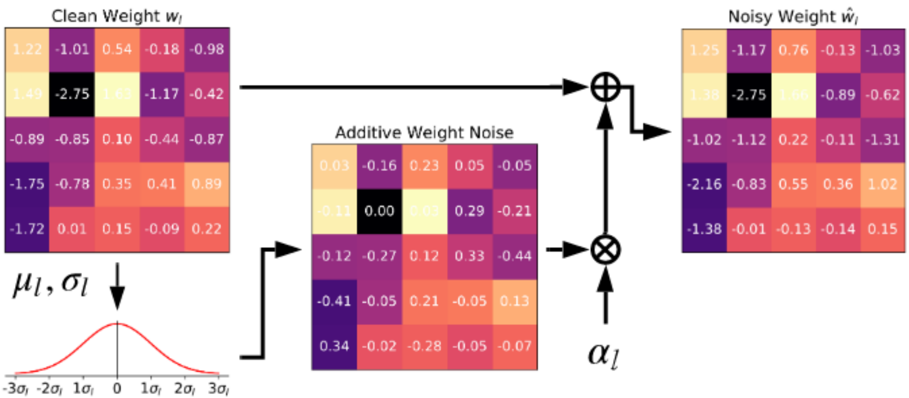
Random Self Ensemble [ECCV 2018] [Liu+18]

Basically Noise layers + Avg prediction

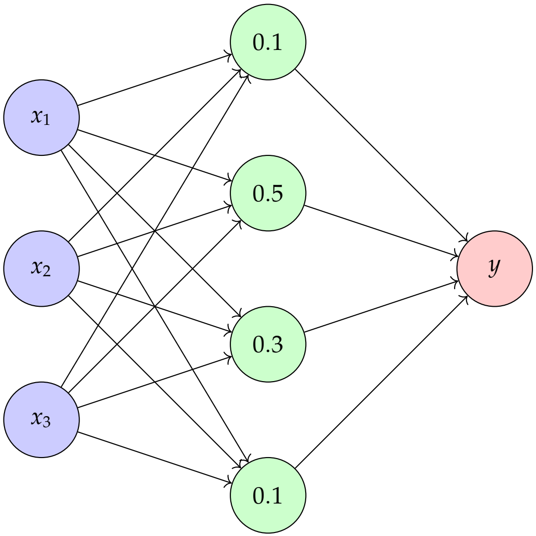


Parametric Noise Injection [CVPR 2019] [HRF19]

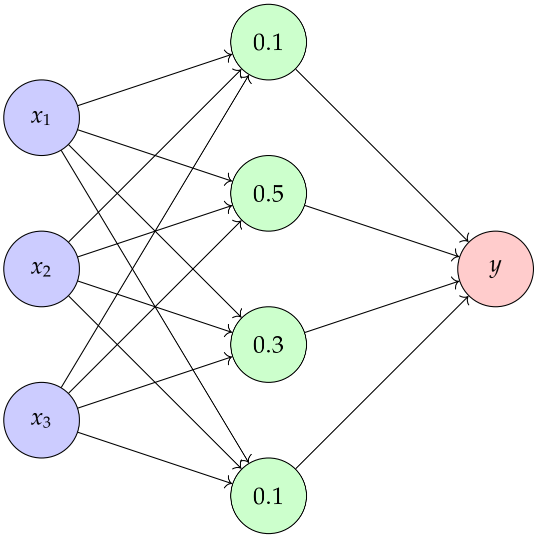
Weight or input noise injection + Adv training.

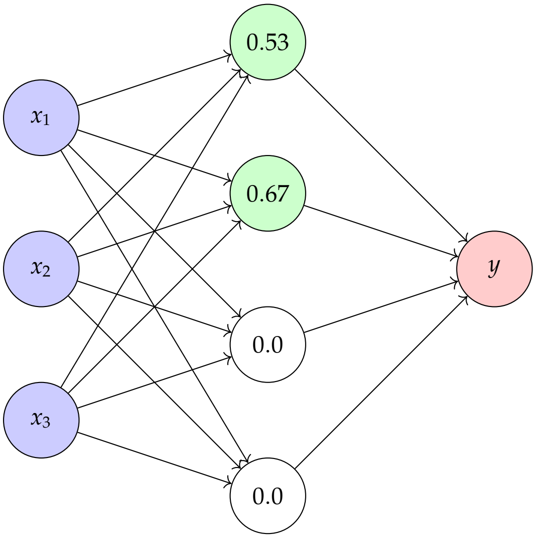


Activation Pruning [ICLR 2018] [Dhi+18]

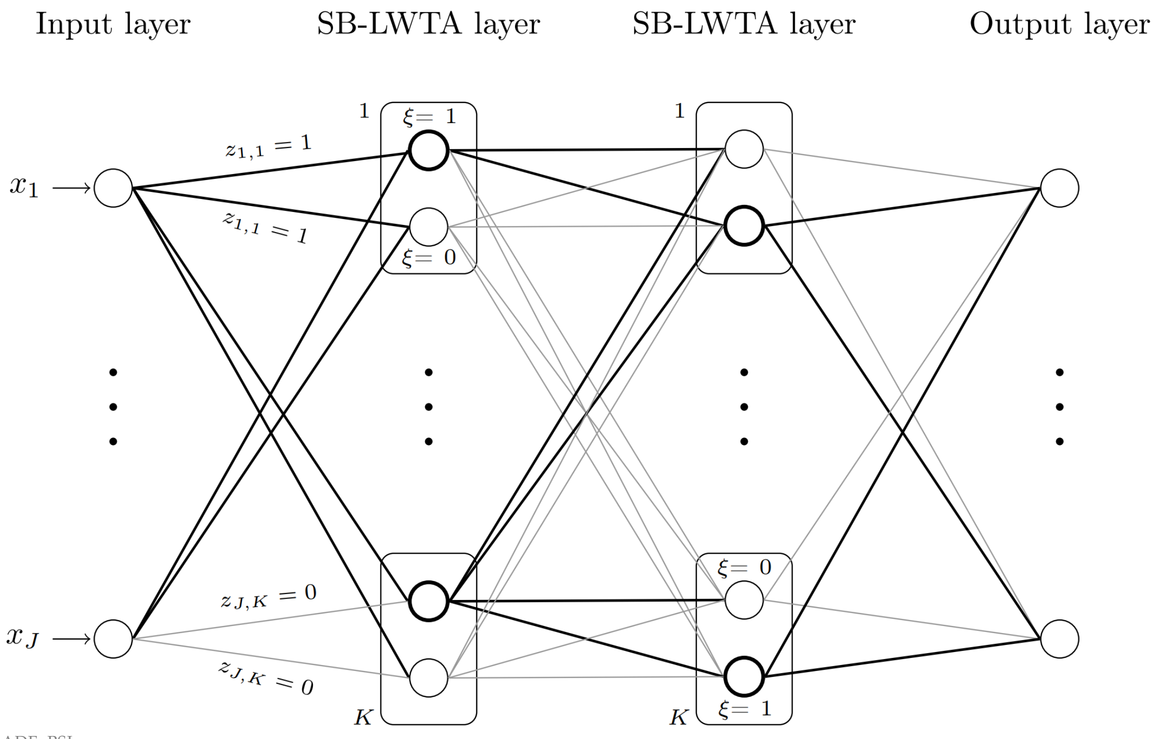


Activation Pruning [ICLR 2018] [Dhi+18]





Stochastic Local-Winner-Takes-All [PCT21; Pan+21]



Other approaches

- Random resize and padding [Xie+17]
- Simple and Effective Stochastic Neural Networks [Yu+21]

Obfuscated gradients

Many (if not all) of the methods do not provide *real* robustness. They just make it harder to find an attack with the usual gradient methods [ACW18].

Defense	Dataset	Distance	Accuracy
Buckman et al. (2018)	CIFAR	0.031 (ℓ_∞)	0%*
Ma et al. (2018)	CIFAR	0.031 (ℓ_∞)	5%
Guo et al. (2018)	ImageNet	0.005 (ℓ_2)	0%*
Dhillon et al. (2018)	CIFAR	0.031 (ℓ_∞)	0%
Xie et al. (2018)	ImageNet	0.031 (ℓ_∞)	0%*
Song et al. (2018)	CIFAR	0.031 (ℓ_∞)	9%*
Samangouei et al. (2018)	MNIST	0.005 (ℓ_2)	55%**
Madry et al. (2018)	CIFAR	0.031 (ℓ_∞)	47%
Na et al. (2018)	CIFAR	0.015 (ℓ_∞)	15%

Evaluation of randomized models

See also this issue.

3 Description of `RobustBench`

We start by providing a detailed layout of our proposed leaderboards for ℓ_∞, ℓ_2, and common corruption threat models. Next, we present the Model Zoo, which provides unified access to most networks from our leaderboards.

3.1 Leaderboard

Restrictions. We argue that accurate benchmarking adversarial robustness in a standardized way requires some restrictions on the type of considered models. The goal of these restrictions is to prevent submissions of defenses that cause some standard attacks to fail without truly improving robustness. Specifically, we consider only classifiers $f : \mathbb{R}^d \to \mathbb{R}^C$ that

- have in general *non-zero gradients* with respect to the inputs. Models with zero gradients, e.g., that rely on quantization of inputs [13, 53], make gradient-based methods ineffective thus requiring zeroth-order attacks, which do not perform as well as gradient-based attacks. Alternatively, specific adaptive evaluations, e.g. with Backward Pass Differentiable Approximation [5], can be used which, however, can hardly be standardized. Moreover, we are not aware of existing defenses solely based on having zero gradients for large parts of the input space which would achieve competitive robustness.
- have a *fully deterministic forward pass*. To evaluate defenses with stochastic components, it is a common practice to combine standard gradient-based attacks with Expectation over Transformations [5]. While often effective it might be not sufficient, as shown by Tramèr et al. [142]. Moreover, the classification decision of randomized models may vary over different runs for the same input, hence even the definition of robust accuracy differs from that of deterministic networks. We note that randomization *can* be useful for improving robustness and deriving robustness certificates [82, 25], but it also introduces variance in the gradient estimators (both white- and black-box) making standard attacks much less effective.
- do not have an *optimization loop* in the forward pass. This makes backpropagation through it very difficult or extremely expensive. Usually, such defenses [118, 84] need to be evaluated adaptively with attacks that rely on a combination of hand-crafted losses.

On Adaptive Attacks to Adversarial Example Defenses [Tra+20]

On Adaptive Attacks
to Adversarial Example Defenses

Florian Tramèr*
Stanford University
tramer@cs.stanford.edu

Nicholas Carlini*
Google
nicholas@carlini.com

Wieland Brendel*
University of Tübingen
wieland.brendel@uni-tuebingen.de

Aleksander Mądry
MIT
madry@mit.edu

Table of Contents

Expected risk

Suppose that the randomness of the model can be described by some distribution μ over a family of classifiers $\mathcal{H}$.

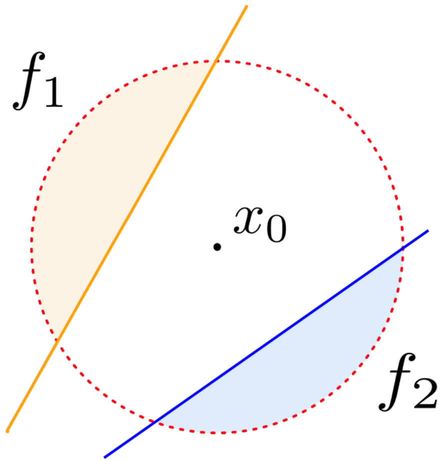
$$x \rightarrow \boxed{\text{sample model } h \sim \mu} \rightarrow h(x)$$

Expected risk

Deterministic	Randomized
$\mathcal{R}(h) = \mathbb{E}_{x,y}[\ell(h,x,y)]$	$\mathcal{R}(\mathbf{h}_\mu) = \mathbb{E}_{x,y}[\mathbb{E}_{h\sim\mu}[\ell(h,x,y)]]$

Matching pennies of classifiers

Mixing classifiers that are **vulnerable** but not **simultaneously vulnerable** creates a situation reminiscent of the game of *matching pennies*.



Randomization can improve robustness: The Matching Penny Gap

Definition 3.1 (Matching penny gap)

The matching penny gap of $\mathbf{h}_\mu$ *at* (x, y) *is:*

$$\pi_{\mathbf{h}_\mu}(x,y) = \underbrace{\mu(\mathcal{H}_{vb}(x,y))}_{\textit{ind. vul}} - \underbrace{\mu^{\max}(x,y)}_{\textit{simult. vul}}$$

where

$$\begin{aligned}
\mathcal{H}_{vb}(x,y) &= \{h \in \mathcal{H}_b : \exists x'_h \in B_\epsilon(x) \text{ such that } h(x'_h) \neq y\}, && \textit{individually vulnerable} \\
\mathfrak{H}_{svb}(x,y) &= \{\mathcal{H}' \subseteq \mathcal{H}_b : \exists x' \in B_\epsilon(x) \text{ such that } \forall h \in \mathcal{H}', h(x') \neq y\}, && \textit{families of sim. vulnerable} \\
\mu^{\max}(x,y) &= \sup_{\mathcal{H}' \in \mathfrak{H}_{svb}(x,y)} \mu(\mathcal{H}'). && \textit{max simultaneously vulnerable}
\end{aligned}$$

If $\pi_{\mathbf{h}_\mu}(x,y) > 0$*, we say that* $\mathbf{h}_\mu$ *is in* matching penny configuration *at* (x,y).

EXAMPLE

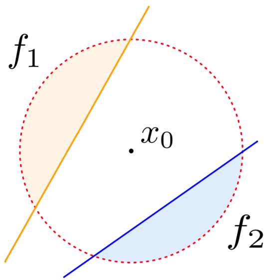


Figure. Let us $\pi_{\mathbf{h}_\mu}$ at the point (x_0, y) for this toy example. Both f_1, f_2 correctly predict the class y for x_0 in the white area, but they are fooled in the orange and blue areas, respectively.

$$\mathcal{H}_b = \{f_1, f_2\}, \qquad \mu = \left(\tfrac{1}{2}, \tfrac{1}{2}\right)$$

$$\mathcal{H}_{vb}(x_0, y) = \{f_1, f_2\} \implies \mu(\mathcal{H}_{vb}(x_0, y)) = 1$$

$$\mathfrak{H}_{svb}(x_0, y) = \{\{f_1\}, \{f_2\}\} \implies \mu^{\max}(x_0, y) = \tfrac{1}{2}$$

$$\therefore \quad \pi_{\mathbf{h}_\mu}(x_0, y) = 1 - \tfrac{1}{2} = \tfrac{1}{2}$$

Two vulnerable classifiers can be mixed to obtain $\frac{1}{2}$ expected adversarial risk !

Main result

Theorem 1

For a mixture $\mathbf{h}_\mu : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$ constructed from $\mathcal{H}_b$ using distribution μ, we have that,

$$\mathcal{R}_\epsilon(\mathbf{h}_\mu) = \mathbb{E}_{h\sim\mu}\left[\mathcal{R}_\epsilon(h)\right] - \mathbb{E}_{(x,y)\sim\rho}[\pi_{\mathbf{h}_\mu}(x,y)]. \quad (1)$$

This theorem shows the link between the risk of a mixture $\mathbf{h}_\mu$ and the average risk. The gap is exactly the expected *matching penny gap*.

Corollary 1

$\mathcal{R}_\epsilon(\mathbf{h}_\mu) < \inf_{h \in \mathcal{H}_b} \mathcal{R}_\epsilon(h)$ *if and only if the following condition holds.*

$$\mathbb{E}_{(x,y)\sim\rho}[\pi_{\mathbf{h}_\mu}(x,y)] > \mathbb{E}_{h\sim\mu}[\mathcal{R}_\epsilon(h)] - \inf_{h\in\mathcal{H}_b} \mathcal{R}_\epsilon(h)$$

When does randomization improve robustness

Corollary 1

$\mathcal{R}_\epsilon(\mathbf{h}_\mu) < \inf_{h \in \mathcal{H}_b} \mathcal{R}_\epsilon(h)$ *if and only if the following condition holds.*

$$\boxed{\mathbb{E}_{(x,y)\sim\rho}[\pi_{\mathbf{h}_\mu}(x,y)]} > \mathbb{E}_{h\sim\mu}[\mathcal{R}_\epsilon(h)] - \inf_{h\in\mathcal{H}_b} \mathcal{R}_\epsilon(h)$$

- Randomized classifiers are better if their expected matching penny gap is high.

WHEN DOES RANDOMIZATION IMPROVE ROBUSTNESS

Corollary 1

$\mathcal{R}_\epsilon(\mathbf{h}_\mu) < \inf_{h\in\mathcal{H}_b} \mathcal{R}_\epsilon(h)$ *if and only if the following condition holds.*

$$\mathbb{E}_{(x,y)\sim\rho}[\pi_{\mathbf{h}_\mu}(x,y)] > \boxed{\mathbb{E}_{h\sim\mu}[\mathcal{R}_\epsilon(h)] - \inf_{h\in\mathcal{H}_b} \mathcal{R}_\epsilon(h)}$$

- Randomized classifiers are better if their expected matching penny gap is high.
- RHS tells us that the individual $h \in \mathcal{H}_b$ should have similar robustness.

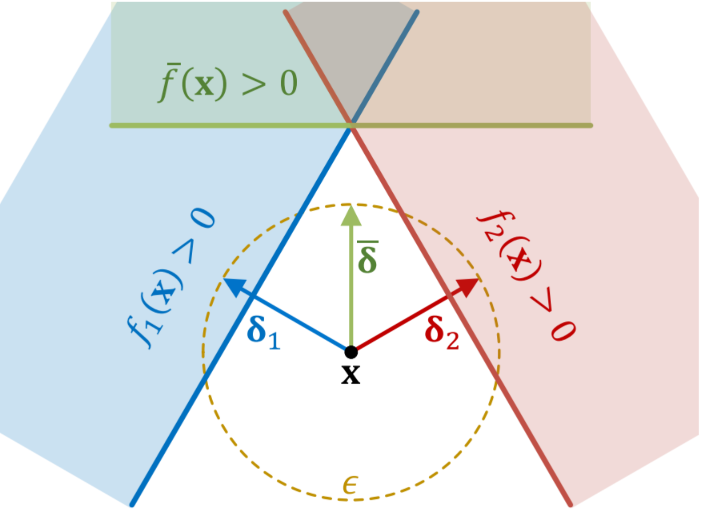
Attacking a mixture

We have seen the importance of using adaptive attacks to evaluate robustness.

Attacking a mixture

We have seen the importance of using adaptive attacks to evaluate robustness.

Dbouk & Shanbhag [DS22] show that attacking a mixture of classifiers is not as trivial as it was believed!



Training a mixture in practice

Only one method has been proposed in the literature. It trains classifiers sequentially in a boosting fashion, while adapting the weights of the mixture. Inside, AT is applied inside using their attack named ARC.

On the Robustness of Randomized Ensembles to Adversarial Perturbations

Hassan Dbouk [1] **Naresh R. Shanbhag** [1]

TRAINING A MIXTURE IN PRACTICE

Only one method has been proposed in the literature. It trains classifiers sequentially in a boosting fashion, while adapting the weights of the mixture. Inside, AT is applied inside using their attack named ARC.

On the Robustness of Randomized Ensembles to Adversarial Perturbations

Hassan Dbouk [1] **Naresh R. Shanbhag** [1]

Training a mixture to leverage the idea of non-simultaneously vulnerable classifiers is still an open problem, and the limits of this approach are still unknown!

Training a mixture in practice

Only one method has been proposed in the literature. It trains classifiers sequentially in a boosting fashion, while adapting the weights of the mixture. Inside, AT is applied inside using their attack named ARC.

On the Robustness of Randomized Ensembles to Adversarial Perturbations

Hassan Dbouk[1] **Naresh R. Shanbhag**[1]

Training a mixture to leverage the idea of non-simultaneously vulnerable classifiers is still an open problem, and the limits of this approach are still unknown!

How can we train models that behave nicely together and increase the matching penny gap?

Table of Contents

DIVERSE ENSEMBLES

The intuition of using different models that can compensate their vulnerabilities to produce a better model is closely related to ensembles!

Improving Adversarial Robustness via Promoting Ensemble Diversity

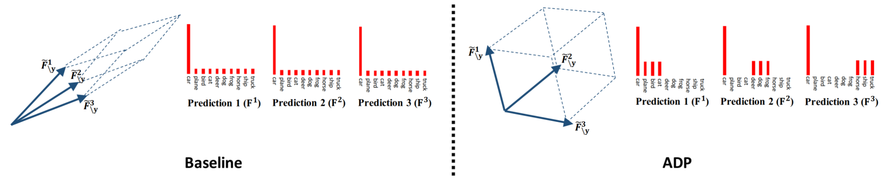


Figure 1. Illustration of the ensemble diversity. **Baseline:** Individually training each member of the ensemble. **ADP:** Simultaneously training all the members of the ensemble with the ADP regularizer. The left part of each panel is the normalized non-maximal predictions.

Improving Adversarial Robustness of Ensembles with Diversity Training

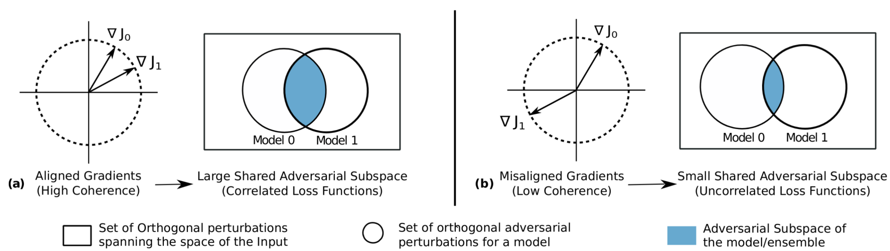


Table of Contents

Bayesian Neural Networks

Let f_θ be a deep neural network with parameters θ and $\mathcal{D}$ be a training data set. Instead of learning θ with empirical risk minimization, BNNs consider:

- A prior $p(\theta)$ over the parameters of the model (often uniform or Gaussian)
- A likelihood $p(y|f_\theta(x))$.

Bayesian Neural Networks

Let f_θ be a deep neural network with parameters θ and $\mathcal{D}$ be a training data set. Instead of learning θ with empirical risk minimization, BNNs consider:

- A prior $p(\theta)$ over the parameters of the model (often uniform or Gaussian)
- A likelihood $p(y|f_\theta(x))$.

Using Bayes' rule, the *postetior* distribution $p(\theta|\mathcal{D})$ is proportional to $p(\theta)p(\mathcal{D}|\theta)$.

Bayesian Neural Networks

Let f_θ be a deep neural network with parameters θ and $\mathcal{D}$ be a training data set. Instead of learning θ with empirical risk minimization, BNNs consider:

- A prior $p(\theta)$ over the parameters of the model (often uniform or Gaussian)
- A likelihood $p(y|f_\theta(x))$.

Using Bayes' rule, the *postetior* distribution $p(\theta|\mathcal{D})$ is proportional to $p(\theta)p(\mathcal{D}|\theta)$.

Predictions are now made using the *posterior predictive*:

$$p(y|x, \mathcal{D}) = \mathbb{E}_{p(\theta|\mathcal{D})}[p(y|f_\theta(x))]$$

Bayesian Neural Networks in practice

In practice, exact inference is intractable, so approximate inference is needed (Hamiltonian Monte Carlo, Stochastic Gradient Langevin Dynamics or Variational Inference).

Bayesian Neural Networks in practice

In practice, exact inference is intractable, so approximate inference is needed (Hamiltonian Monte Carlo, Stochastic Gradient Langevin Dynamics or Variational Inference).

In practice, an ensemble is approximately sampled from $p(\theta|\mathcal{D})$:

$$\sum_{i=1}^{m} f_{\theta_i}(x), \quad \theta_i \sim p(\theta|\mathcal{D}).$$

Carbone et al. (2020) [Car+20; Wic+21] provide a theoretical guarantee for over parametrized BNNs on the infinite data limit:

Theorem 1. *Let $f(\mathbf{x}, \mathbf{w})$ be a fully trained overparametrized BNN on a prediction problem with data manifold $\mathcal{M}_D \subset \mathbb{R}^d$ and posterior weight distribution $p(\mathbf{w}|D)$. Assuming $\mathcal{M}_D \in \mathcal{C}^\infty$ almost everywhere, in the large data limit we have a.e. on $\mathcal{M}_D$*

$$\left(\langle \nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{w}) \rangle_{p(\mathbf{w}|D)}\right) = \mathbf{0}. \tag{3}$$

References I

[ACW18] Anish Athalye, Nicholas Carlini, and David Wagner. **“Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples”.** In: *International conference on machine learning*. PMLR. 2018, pp. 274–283.

[Car+20] Ginevra Carbone et al. **“Robustness of bayesian neural networks to gradient-based attacks”.** In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 15602–15613.

[Dhi+18] Guneet S. Dhillon et al. **“Stochastic Activation Pruning for Robust Adversarial Defense”.** In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=H1uR4GZRZ.

[DS22] Hassan Dbouk and Naresh Shanbhag. **“Adversarial Vulnerability of Randomized Ensembles”.** In: *International Conference on Machine Learning*. PMLR. 2022, pp. 4890–4917.

[HRF19] Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. **“Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack”.** In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 588–597.

[Liu+18] Xuanqing Liu et al. **“Towards robust neural networks via random self-ensemble”.** In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 369–385.

References II

[Pan+21] Konstantinos Panousis et al. **"Local competition and stochasticity for adversarial robustness in deep learning".** In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 3862–3870.

[PCT21] Konstantinos P Panousis, Sotirios Chatzis, and Sergios Theodoridis. **"Stochastic local winner-takes-all networks enable profound adversarial robustness".** In: *arXiv preprint arXiv:2112.02671* (2021).

[Pin+19] Rafael Pinot et al. **"Theoretical evidence for adversarial robustness through randomization".** In: *Advances in Neural Information Processing Systems* 32 (2019).

[Pin+20] Rafael Pinot et al. **"Randomization matters how to defend against strong adversarial attacks".** In: *International Conference on Machine Learning*. PMLR. 2020, pp. 7717–7727.

[Tra+20] Florian Tramer et al. **"On adaptive attacks to adversarial example defenses".** In: *Advances in neural information processing systems* 33 (2020), pp. 1633–1645.

[Wic+21] Matthew Wicker et al. **"Bayesian inference with certifiable adversarial robustness".** In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 2431–2439.

[Xie+17] Cihang Xie et al. **"Mitigating adversarial effects through randomization".** In: *arXiv preprint arXiv:1711.01991* (2017).

[Yu+21] Tianyuan Yu et al. **"Simple and effective stochastic neural networks".** In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021, pp. 3252–3260.