Certified Robustness to Label Flipping Attacks via Randomized Smoothing

ICML 2020

Elan Rosenfeld, Ezra Winston, Pradeep Ravikumar, J. Zico Kolter

{elan, ezrawinston}@cmu.edu

{pradeep, zkolter}@cs.cmu.edu

Carnegie Mellon University

BOSCH

Motivation

Adversarial Input Manipulation

- Modern classifiers are susceptible to many types of input manipulation.

Motivation

Adversarial Input Manipulation

- Modern classifiers are susceptible to many types of input manipulation.
- **Label flipping attack**: a *train-time* attack where training labels are manipulated.
    - Objective is to cause resulting trained classifier to perform poorly.
    - E.g., mislabeling spam emails or fake reviews to cause detector to fail in production.

Motivation

Adversarial Input Manipulation

- Modern classifiers are susceptible to many types of input manipulation.
- **Label flipping attack**: a *train-time* attack where training labels are manipulated.
    - Objective is to cause resulting trained classifier to perform poorly.
    - E.g., mislabeling spam emails or fake reviews to cause detector to fail in production.
- In comparison, adversarial examples are *test-time* attacks.

Motivation

Provable Defenses

- Recent shift to provable defenses against adversarial perturbations.
    - Certify robustness of each classification.
    - We call such certifications **pointwise**.

Motivation

Provable Defenses

- Recent shift to provable defenses against adversarial perturbations.
  - Certify robustness of each classification.
  - We call such certifications **pointwise**.
- For train-time attacks, *no pointwise-certified defenses exist.*
  - Pointwise is preferable when we care about each distinct classification (loans, parole grants, etc.)

Motivation

Provable Defenses

- Recent shift to provable defenses against adversarial perturbations.
    - Certify robustness of each classification.
    - We call such certifications **pointwise**.
- For train-time attacks, *no pointwise-certified defenses exist.*
    - Pointwise is preferable when we care about each distinct classification (loans, parole grants, etc.)
- We present the first **pointwise-certified** linear classifier, with **no data assumptions**.
    - Makes a prediction by outputting an expectation over predictions with respect to a distribution over training labels.

Randomized Smoothing for Test-Time Attacks

Given a classifier *f* and input *x*, don't directly certify *f*. Certify weighted majority vote of *f* applied to *x* perturbed by noise: $g(x) = \mathbb{E}_{\epsilon \sim p(\epsilon)}[f(x+\epsilon)]$

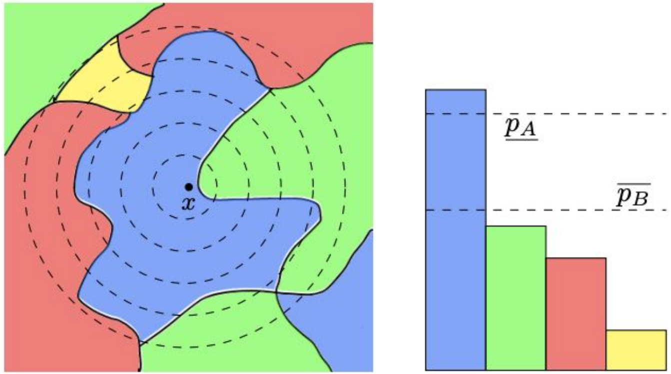
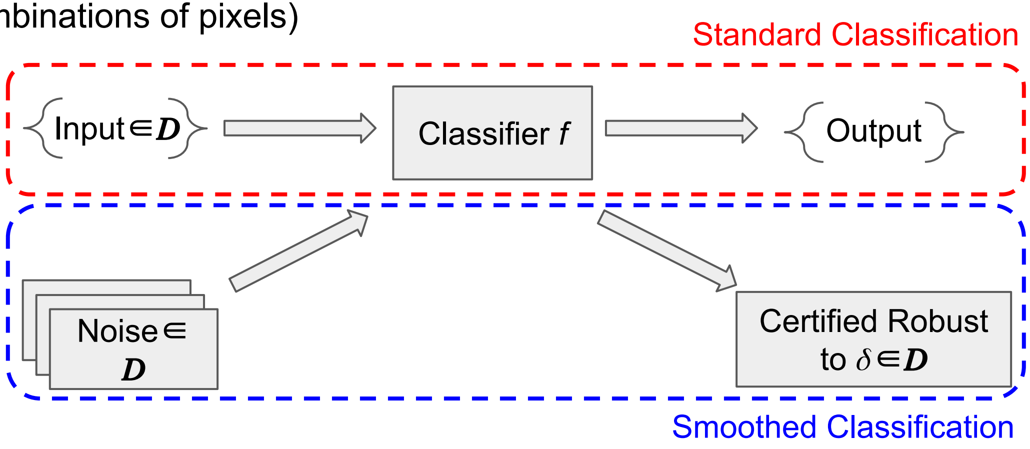


Image: Cohen et al. 2019

- Robustness certificate is a function of margin between first and second class.
  - This means we need a lower bound on probability assigned to first class $\underline{p_A}$.

Randomized Smoothing for Test-Time Attacks

"Input" is an image to be classified.
(domain $\boldsymbol{D}$ is combinations of pixels)



Certified robust to adversarial pixel perturbations.

"Noise" is pixel perturbations.

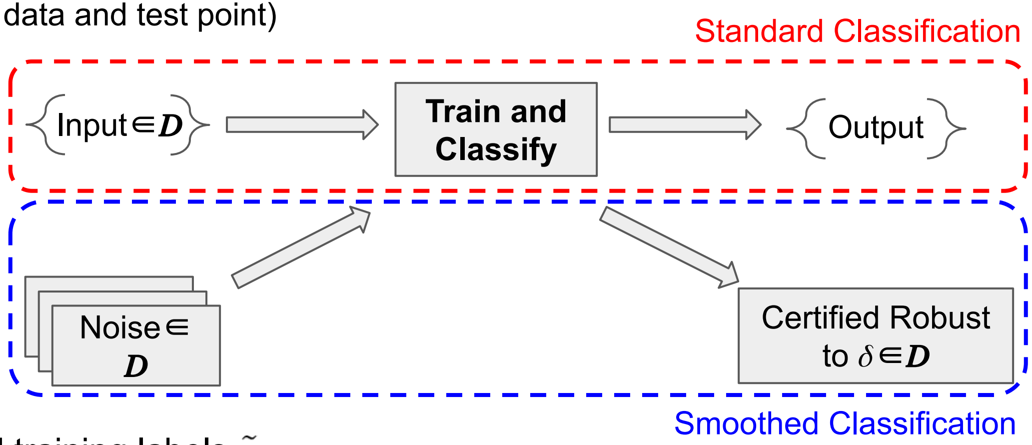
Our Key Observation:

The theory behind randomized smoothing applies to **arbitrary functions *f***, not just classifiers.

Recast *f* as the Whole Training Procedure

"Input" is *n* training points X, labels $\mathbf{y}$, test point x_{n+1}

($\boldsymbol{D}$ is now training data and test point)



Certified robust to adversarial label-flipping attacks.

"Noise" is flipped training labels $\tilde{\mathbf{y}}$.

(smoothing over input pixels)

$$g(x) = \mathbb{E}_{\epsilon \sim p(\epsilon)}[f(x+\epsilon)]$$

→

(smoothing over training labels)

$$g(X, \mathbf{y}, x_{n+1}) = \mathbb{E}_{\tilde{\mathbf{y}} \sim p(\mathbf{y})}[f(X, \tilde{\mathbf{y}}, x_{n+1})]$$

One Major Caveat

Smoothed Classifier: $$g(X, \mathbf{y}, x_{n+1}) = \mathbb{E}_{\tilde{\mathbf{y}} \sim p(\mathbf{y})}[f(X, \tilde{\mathbf{y}}, x_{n+1})]$$

- Previous randomized smoothing applications probabilistically bound integral with sampling.
    - Implementing naively would require training *thousands of classifiers* per test point.
- We develop an algorithm to make classification tractable and **guaranteed**.
    - Certificate applies for **any features** with **no data assumptions**.

Binary Classification via Ordinary Least-Squares

- Training points: $X \in \mathbb{R}^{n \times k}$
- Binary labels: $\mathbf{y} \in \{0, 1\}^n$
- Test point: $x_{n+1} \in \mathbb{R}^k$

Solve: $\hat{\beta} = (X^\top X)^{-1} X^\top \mathbf{y}$

Predict: $f(X, \mathbf{y}, x_{n+1}) = \mathbf{1}\{x_{n+1}^\top \hat{\beta} \geq 1/2\}$

This can be precomputed and reused for each test point!

Evaluation per sampled $\mathbf{y}$ is just an inner product!

Equivalently, Solve: $\boldsymbol{\alpha} = \overbrace{X(X^\top X)^{-1}}\, x_{n+1}^\top$

Predict: $f(X, \mathbf{y}, x_{n+1}) = \mathbf{1}\{\boldsymbol{\alpha}^\top \mathbf{y} \geq 1/2\}$

From Probabilistic to Deterministic

With kernel representation $\boldsymbol{\alpha}$, *we don't need to even evaluate the classifier*.

Recall our smoothed classifier:

$$g(X, \mathbf{y}, x_{n+1}) = \mathbb{E}[f(X, \tilde{\mathbf{y}}, x_{n+1})] = \mathbb{E}[\mathbf{1}\{\boldsymbol{\alpha}^\top \tilde{\mathbf{y}} \geq 1/2\}]$$

Expectation of an indicator function is the probability it outputs 1:

$$g(X, \mathbf{y}, x_{n+1}) = P\left(\boldsymbol{\alpha}^\top \tilde{\mathbf{y}} \geq 1/2\right)$$

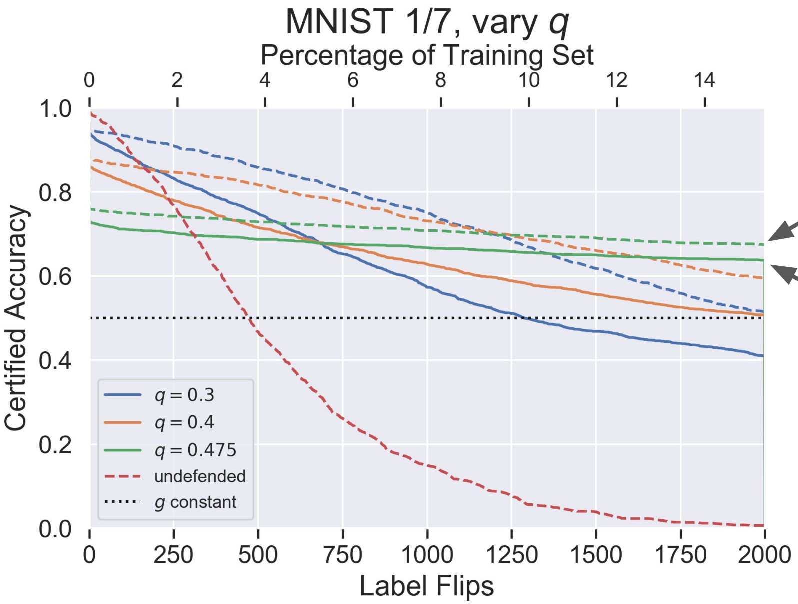
This is a sum of independent random variables. Apply the **Chernoff Bound**:

$$P(\boldsymbol{\alpha}^\top \tilde{\mathbf{y}} \leq 1/2) \leq \min_{t>0} \left\{ e^{t/2} \prod_{i=1}^{n} \mathbb{E}[e^{-t\boldsymbol{\alpha}_i \tilde{\mathbf{y}}_i}] \right\}$$

This is a one-dimensional optimization, solvable via Newton's method.

Experiments: Binary Classification

- Hyperparameter (q) represents probability of flipping label under noise distribution.
  - Controls tradeoff between accuracy and robustness.



For 68% of test points, our attack using 2000 flips was unable to cause misclassification.

64% certified accuracy against an adversary flipping 2000 labels *for each test point*.

Multi-Class Classification

- Our algorithm derives a guaranteed lower bound on the probability assigned to a class.
- We can repeat for each class, choose the class with the highest lower bound.
- This generalizes robust certification to the multi-class case!

Experiments: Multi-Class Classification

Features learned in an unsupervised fashion.