16. Theories of Deep Learning: Robustness

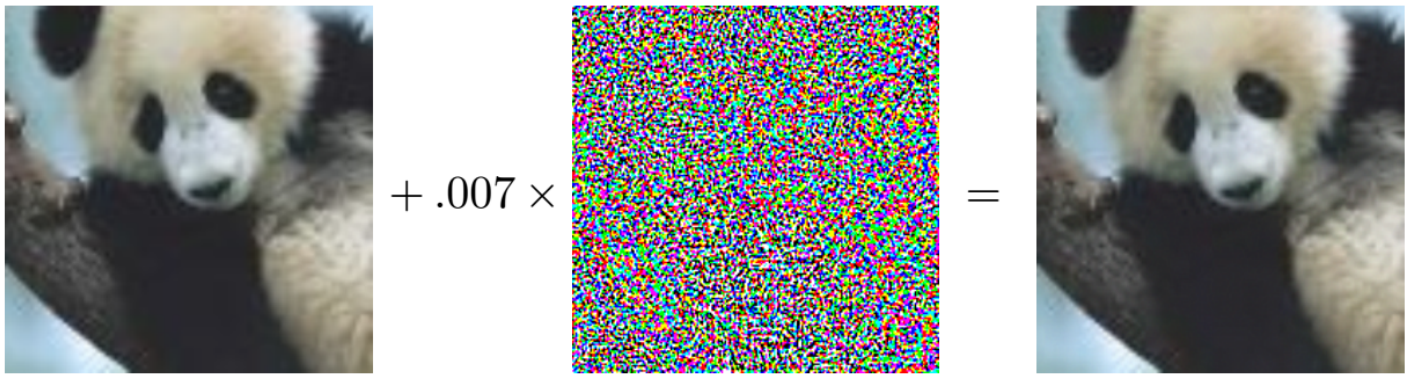
Instructor: Ruixuan Wang
wangruix5@mail.sysu.edu.cn

School of Computer Science and Engineering
Sun Yat-Sen University

June 20, 2022

Attacking models with adversarial examples

- Adversarial examples: input with imperceptible perturbations, resulting in incorrect output with high confidence



- Adversarial example $\mathbf{x}$ can be obtained by optimizing the loss

$$\max_{\mathbf{x}} L(\mathbf{x}, y_0; \boldsymbol{\theta}) \quad \text{or} \quad \min_{\mathbf{x}} L(\mathbf{x}, y_t; \boldsymbol{\theta})$$
$$s.t. \quad \|\mathbf{x} - \mathbf{x}_0\|_p < \epsilon$$

  - $(\mathbf{x}_0, y_0)$: clean input data and associated source class
  - $\boldsymbol{\theta}$: model parameters (fixed); y_t: target class
  - ℓ_p norm fewer than ϵ controls perturbation!
  - white-box attacks: knowing model parameters

Figure from Goodfellow et al., “Explaining and harnessing adversarial examples”, ICLR, 2015

Model defense

- Model defense: to make models robust to adversarial attacks
- Approach 1: defense by suppressing adversarial noise
  - denoising: construct a model to remove adversarial noise
  - projection: map data to feature manifold of clean data
- Approach 2: defense by adding randomness
  - random perturbation in data: e.g., randomly resize and padding, replace pixel by one of its neighbors, etc.
- Approach 3: gradient masking or obfuscation
  - make gradients vanishing, noisy, not differentiable, etc.
- Approach 4: adversarial training
- Approach 5: certified defense

Liao et al., 'Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser'", CVPR, 2018; Xie et al., "Mitigating Adversarial Effects Through Randomization", ICLR, 2018; Athalye et al., "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples", ICLR, 2018; Samangouei et al., "Defense-GAN: protecting classifiers against adversarial attacks using generative models", ICLR, 2018

Model defense by adversarial training

- One way to defend attacks: adversarial training
  - train the model by minimizing the loss on adversarial examples

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \Big[\max_{\boldsymbol{\delta}\in\mathcal{S}} L(\boldsymbol{\theta}, \mathbf{x}+\boldsymbol{\delta}, y) \Big]$$

    where $\boldsymbol{\delta}$ represents the perturbation (variable) within the allowed set $\mathcal{S}$ (e. g.,ℓ_∞ ball).
- The saddle point (min-max) problem can be solved because
  - inner max by mult-step projected gradient descent (PGD) from multiple random starts results in similar loss values $L(\cdot)$;

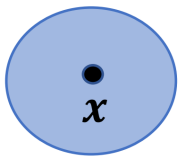
$$\mathbf{x}_{t+1} \quad = \quad \text{Project}_{\mathbf{x}_0+\mathcal{S}} \{\mathbf{x}_t + \alpha \ \text{sign}(\nabla_{\mathbf{x}} L(\boldsymbol{\theta}, \mathbf{x}_t, y))\}$$

  - Danskin's theorem: gradients at inner maximizers corresponds to descent directions for the saddle point problem.
- It suggests: a network robust against PGD adversaries yields robustness against all first-order (gradient-based) adversaries!

Madry et al., "Towards deep learning models resistant to adversarial attacks", ICLR, 2018

Model defense by certified defense

- Certified adversarial defense (in ℓ_2 norm)
  - returns both prediction and a certificate that the prediction is constant within an neighborhood around the input.



Certify that every prediction inside this ball will be "panda."

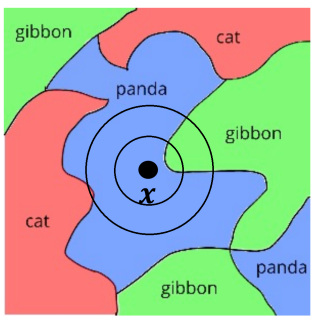
- How: randomized smoothing for certified defense
  1. train a base classifier f with Gaussian data augmentation
  2. smooth f into a new classifier g such that: $g(\mathbf{x}) =$ the most likely prediction by f on random Gaussian corruptions of $\mathbf{x}$

Example: consider the input $\boldsymbol{x}$ = 

Suppose that when f classifies $\mathcal{N}(\boldsymbol{x}, \sigma^2 I)$  ,

- 🐼 is returned with probability 0.80
- 🐻 is returned with probability 0.15
- 🐱 is returned with probability 0.05

Then $g(\boldsymbol{x}) =$ 🐼



  - note: Gaussian noise (e.g., $\sigma = 0.5$) is much larger than adversarial perturbation to which g is provably robust
  - large random perturb. "drown out" small adver. perturb.

Cohen et al., "Certified Adversarial Robustness via Randomized Smoothing", ICML, 2019

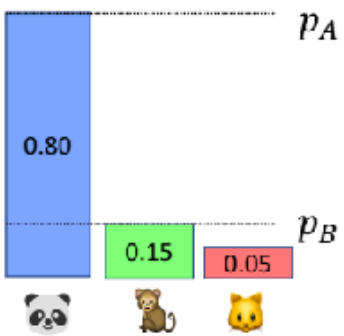
Model defense by certified defense

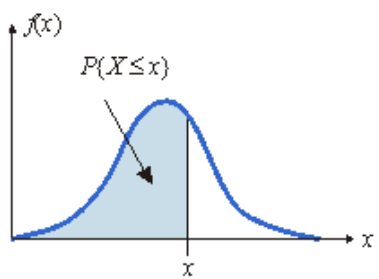
Theorem (Certified defense, Cohen et al. 2019)

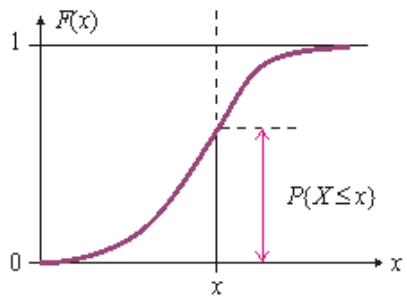
Given test data $\mathbf{x}$ *and samples of random Guassian corruptions from* $\mathcal{N}(\mathbf{x}, \sigma^2\mathbf{I})$*, let* p_A *and* p_B *respectively be the probability of the top and runner-up class predicted by* f *on random Gaussian corruptions of* $\mathbf{x}$*, then the smoothed classifier* g *is robust around* $\mathbf{x}$ *within the* ℓ_2 *radius* R *(significantly smaller than* σ*),*

$$R = \frac{\sigma}{2}\left(\Phi^{-1}(p_A) - \Phi^{-1}(p_B)\right),$$

where Φ^{-1} *is the inverse standard Gaussian CDF.*







Cohen et al., "Certified Adversarial Robustness via Randomized Smoothing", ICML, 2019

Is there any classifier that cannot be attacked?

- Assume data $\mathbf{x}$ can be generated by certain generator $\mathbf{x} = g(\mathbf{z})$, where $\mathbf{z}$ is i.i.d. Gaussian.
- Robustness $r(\mathbf{x})$ for data $\mathbf{x} = g(\mathbf{z})$ is defined as

$$r(\mathbf{x}) = \min_{\mathbf{r}} \|g(\mathbf{z}+\mathbf{r}) - \mathbf{x}\| \qquad \text{s.t. } f(g(\mathbf{z}+\mathbf{r})) \neq f(\mathbf{x}),$$

where $\|\cdot\|$ is any norm, $f : \mathbb{R}^d \to \{1, \ldots, K\}$ is any classifier.

Theorem (Fawzi et al. 2018)

Let $f : \mathbb{R}^d \to \{1, \ldots, K\}$ be an arbitrary classifier. For any η,

$$p(r(\mathbf{x}) \leq \eta) \geq 1 - \sqrt{\frac{\pi}{2}} \exp\Big\{ - \frac{\eta^2}{2L^2} \Big\},$$

where L is the Lipschitz constant of generator function $g(\mathbf{t})$.

- In general $L \ll \sqrt{d}$, where d is dimension of data $\mathbf{x}$;
- If $\eta = 2L$, then $p(r(\mathbf{x}) \leq 2L) \geq 0.8$, i.e., robustness is less than $2L$ (very small perturb.) with probability exceeding 0.8!

Fawzi et al., "Adversarial vulnerability for any classifier", NeurIPS, 2018

Is there any classifier that cannot be attacked?

Theorem (Existence of adversarial examples, Shafahi et al. 2019)

(Under certain conditions) input data $\mathbf{x}$ *will either (1) be originally mis-classified or (2) can be attacked within an* ϵ*-ball, with the probably at least*

$$1 - V_c \sqrt{\frac{\pi}{8}} \exp\left\{ - \frac{d-1}{2} \epsilon^2 \right\},$$

where V_c *is constant relevant to the class of data* $\mathbf{x}$*, and* d *is the dimension of* $\mathbf{x}$*.*

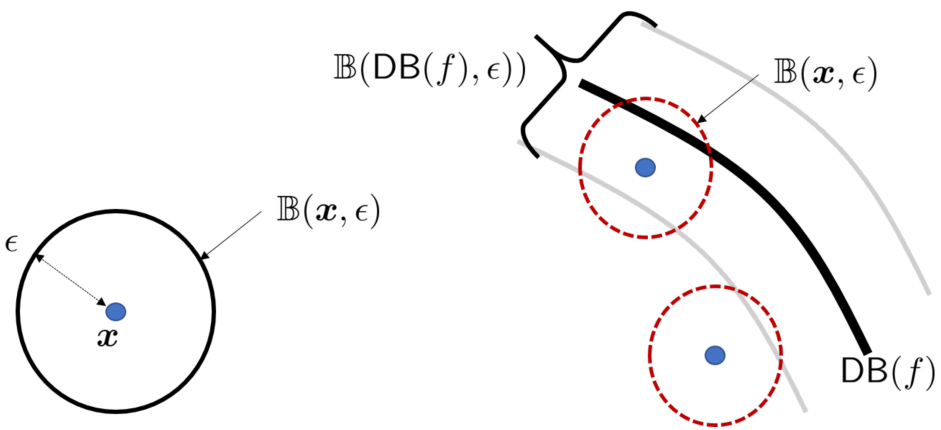
So: when dimension d grows, the probability will go to 1. Or, for larger-size images, the probability of being attacked will be higher.

If attacks not avoidable, what can we do?

Shafahi et al., "Are adversarial examples inevitable?", ICLR, 2019

Trade-off between accuracy and robustness

- There is natural trade-off between accuracy and robustness
  - e.g., always classify data into one specific class: ultimately robust but not accurate
  - May trade off accuracy and robustness during training?
- Notations
  - Input data $\mathbf{x} \in \mathcal{X}$, label $y \in \{+1, -1\}$, classifier $f : \mathcal{X} \to \mathbb{R}$
  - $\mathbb{B}(\mathbf{x}, \epsilon) = \{\mathbf{x}' \in \mathcal{X} : \|\mathbf{x}' - \mathbf{x}\| \leq \epsilon\}$: ϵ-ball surrounding data $\mathbf{x}$
  - $\mathsf{DB}(f) = \{\mathbf{x} \in \mathcal{X} : f(\mathbf{x}) = 0\}$: decision boundary of classifier
  - $\mathbb{B}(\mathsf{DB}(f), \epsilon) = \{\mathbf{x} \in \mathcal{X} : \exists \mathbf{x}' \in \mathbb{B}(\mathbf{x}, \epsilon) \text{ s.t. } f(\mathbf{x}) f(\mathbf{x}') \leq 0\}$: neighborhood of (or band surrounding) the decision boundary.
  - For any data $\mathbf{x}$ outside the band $\mathbb{B}(\mathsf{DB}(f), \epsilon)$, all data in its ϵ-ball $\mathbb{B}(\mathbf{x}, \epsilon)$ has the same prediction label from classifier f.



Trade-off between accuracy and robustness

- Natural classification error: average of 0-1 loss over all data

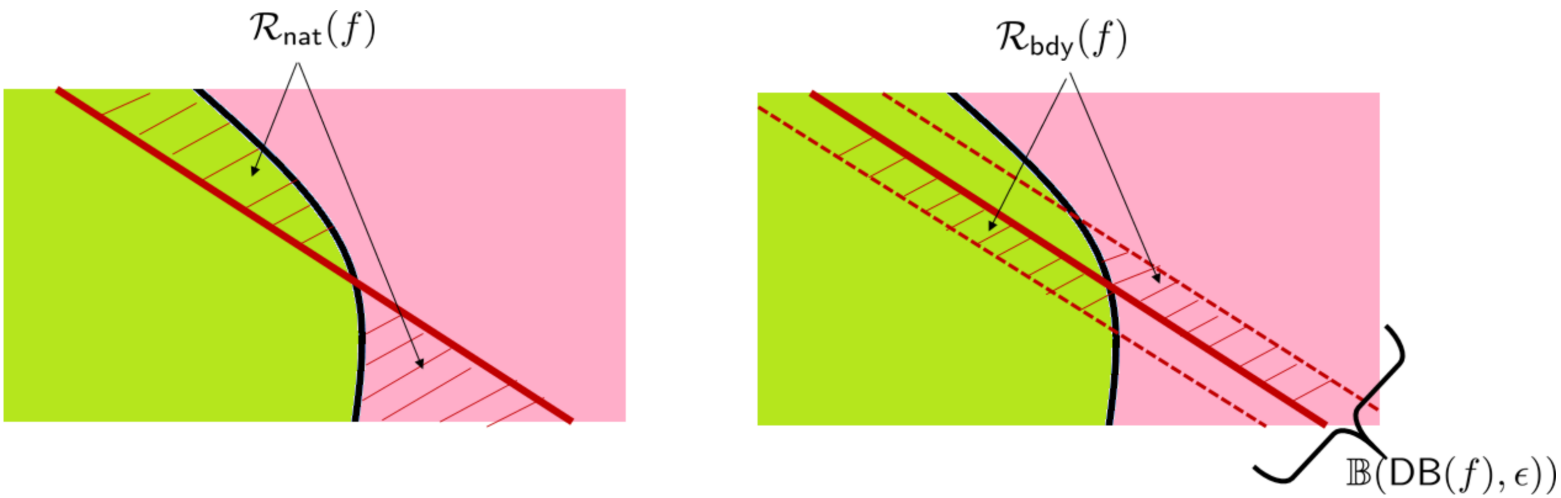
$$\mathcal{R}_{\text{nat}}(f) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathsf{I}\big(f(\mathbf{x})y \le 0\big)\,,$$

  - about the data $\mathbf{x}$'s which are mis-classified

- Boundary classification error

$$\mathcal{R}_{\text{bdy}}(f) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\mathsf{I}\big(\mathbf{x} \in \mathbb{B}(\text{DB}(f),\epsilon) \text{ and } f(\mathbf{x})y > 0\big)\,,$$

  - about the data $\mathbf{x}$'s which are inside the decision band and correctly classified (but likely mis-classified)



Figures here and in prev+next slides are from Stanley Chan, "Machine learning I: Robustness and accuracy trade off", Lecture 37, 2020

Trade-off between accuracy and robustness

- Robust classification error (which should be minimized)

$$\mathcal{R}_{\text{rob}}(f) = \mathcal{R}_{\text{nat}}(f) + \mathcal{R}_{\text{bdy}}(f)$$

- The $0-1$ loss is not differentiable, often replaced by differentiable 'classification-calibration surrogate loss' ϕ

Theorem (Upper bound of robust error, Zhang et al. 2019)

For any non-negative loss function ϕ and classifier $f : \mathcal{X} \to \mathbb{R}$, let $\mathcal{R}_\phi(f) := \mathbb{E}\phi(f(\mathbf{x})y)$ and $\mathcal{R}^*_\phi = \min_f \mathcal{R}_\phi(f)$. Then, upper bound of robust error can be estimated as below,

$$\mathcal{R}_{rob}(f) - \mathcal{R}^*_{nat} \leq \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi) + \mathbb{E} \max_{\mathbf{x}' \in \mathbb{B}(\mathbf{x},\epsilon)} \phi(f(\mathbf{x}')f(\mathbf{x})/\lambda)\,,$$

where $\mathcal{R}^*_{nat} = \min_f \mathcal{R}_{nat}(f)$, ψ is the ψ-transform of ϕ, and $\lambda > 0$ is a balancing constant.

Zhang et al., "Theoretically principled trade-off between robustness and accuracy", ICML, 2019

Trade-off between accuracy and robustness

- Theorem suggest: minimize the right to achieve minimum left

$$\min_f \left\{ \underbrace{\psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}_\phi^*)}_{\text{for accuracy}} + \underbrace{\mathbb{E} \max_{\mathbf{x}' \in \mathbb{B}(\mathbf{x}, \epsilon)} \phi(f(\mathbf{x}')f(\mathbf{x})/\lambda)}_{\text{regularization for robustness}} \right\},$$

- Replace the first term by the empirical risk $\mathbb{E}\phi(f(\mathbf{x})y)$, then

$$\min_f \mathbb{E} \left\{ \underbrace{\phi(f(\mathbf{x})y)}_{\text{for accuracy}} + \underbrace{\max_{\mathbf{x}' \in \mathbb{B}(\mathbf{x}, \epsilon)} \phi(f(\mathbf{x}')f(\mathbf{x})/\lambda)}_{\text{regularization for robustness}} \right\},$$

- Optimization is a trade-off between accuracy and robustness!
  - 1st term: general classifier loss
  - 2nd term: encourage classifier pred to be smooth such that pred of data $\mathbf{x}$ and its adversarial neighbor $\mathbf{x}'$ have same sign
  - more accurate $f(\mathbf{x})$: smaller first term, but larger difference between $f(\mathbf{x})$ and $f(\mathbf{x}')$ would cause larger second term

Zhang et al., "Theoretically principled trade-off between robustness and accuracy", ICML, 2019

Trade-off between accuracy and robustness

- Optimization can be extended to mult-class classification

$$\min_f \mathbb{E}\Big\{ L(f(\mathbf{x}), y) + \max_{\mathbf{x}' \in \mathbb{B}(\mathbf{x}, \epsilon)} L(f(\mathbf{x}'), f(\mathbf{x}))/\lambda \Big\},$$

- Recall the adversarial training loss (Madry et al., 2018)

$$\min_f \mathbb{E} \max_{\mathbf{x}' \in \mathbb{B}(\mathbf{x}, \epsilon)} L(f(\mathbf{x}'), y),$$

  - not consider first term in trade-off formulation
  - the upper bound from the trade-off formulation is tighter
- How to optimize the trade-off formulation?
  - similar to adversarial training with multi-step projected gradient descent (PGD), but combining (first) classifier loss
- Trade-off formulation (TRADES) achieves SOTA of defense

Zhang et al., "Theoretically principled trade-off between robustness and accuracy", ICML, 2019

Beyond trade-off

- Always need to trade off accuracy and robustness?
  - observed in real dataset: different classes show pixel-based separation larger than general perturbation level (Yang et al.)
  - so both accuracy and robustness may be achieved in practice
  - proved: there exist locally Lipschitz function to achieve both
- Observe the trade-off optimization formulation again

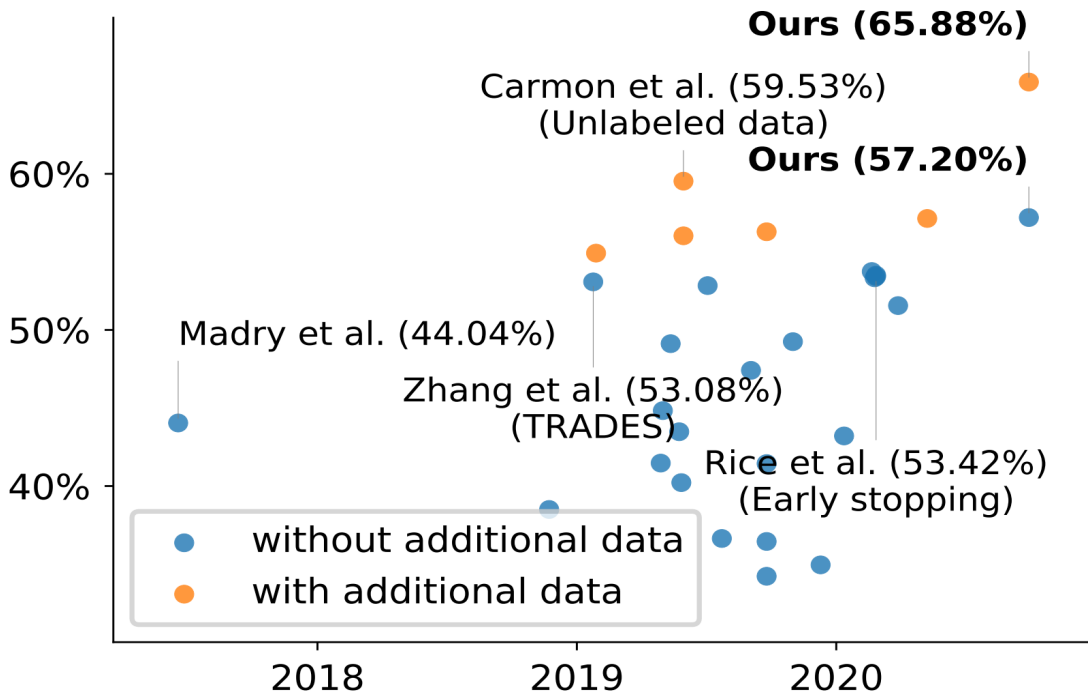
$$\min_f \mathbb{E}\Big\{ \underbrace{L(f(\mathbf{x}), y)}_{\text{require labeled data}} + \underbrace{\max_{\mathbf{x}' \in \mathbb{B}(\mathbf{x}, \epsilon)} L(f(\mathbf{x}'), f(\mathbf{x}))/\lambda}_{\text{just need un-labelled data}} \Big\},$$

- With the property, designed semi-supervised method
  - Zhai et al. developed similar decomposition theorem
  - labelled data for first term
  - un-labelled data for second term: get pseudo-labels first, then apply multi-step PGD for adversarial training
  - vs: prior study claimed requiring more labelled data

Yang et al., "A Closer Look at Accuracy vs. Robustness", NeurIPS, 2020; Zhai et al., "Adversarially Robust Generalization Just Requires More Unlabeled Data", arXiv, 2019; Schmidt et al., "Adversarially robust generalization requires more data", NeurIPS, 2018

Beyond model training and testing factors

- More factors have been found to affect model robustness
  - early stopping during training improves robustness
  - increasing the capacity of models improves robustness
  - Swish/SiLU activation functions perform better
  - additional unlabeled data (with pseudo-label) helps a lot
  - model weight averaging (WA) consistently improves robustness

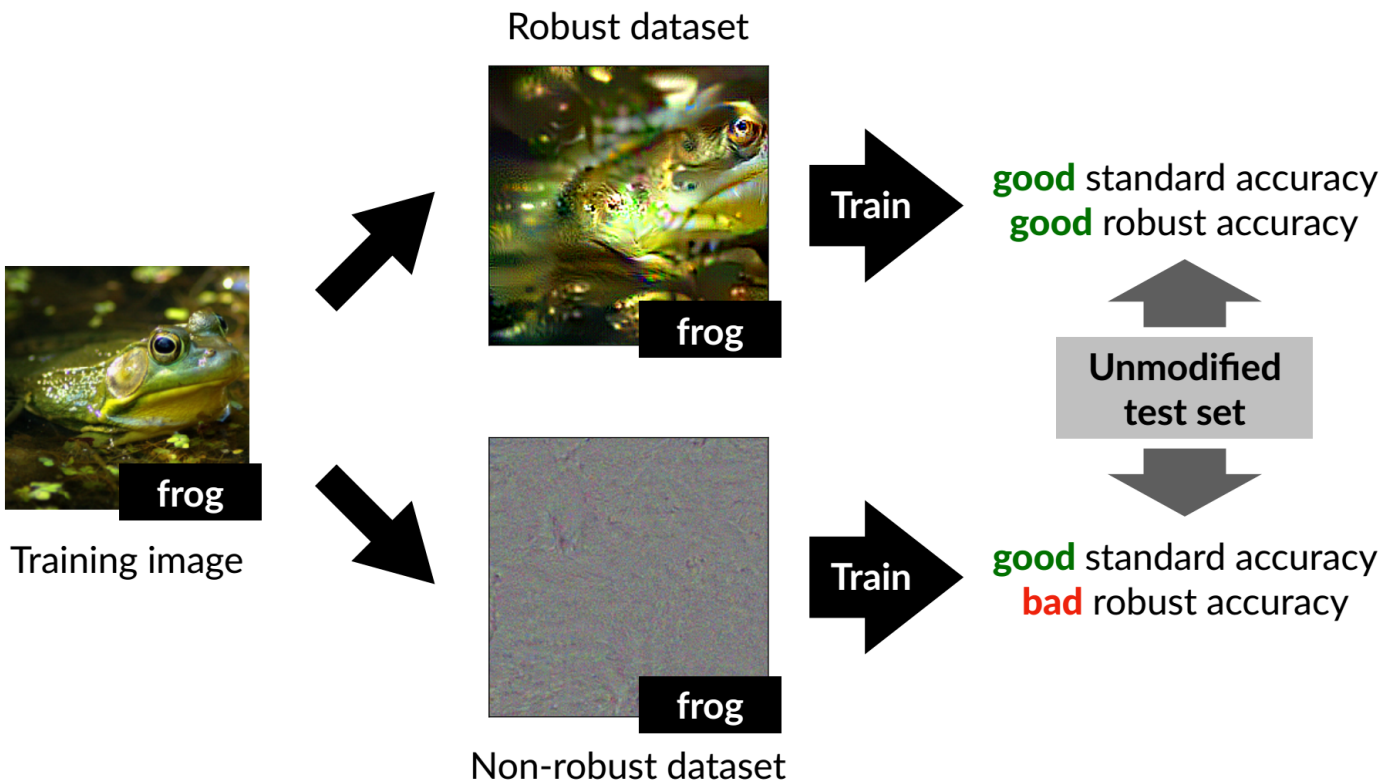


Accuracy against AutoAttack on Cifar-10 with ℓ_∞ perturbations of size 8/255.

Gowal et al., "Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples", arXiv, 2020

Interpretation of adversarial examples

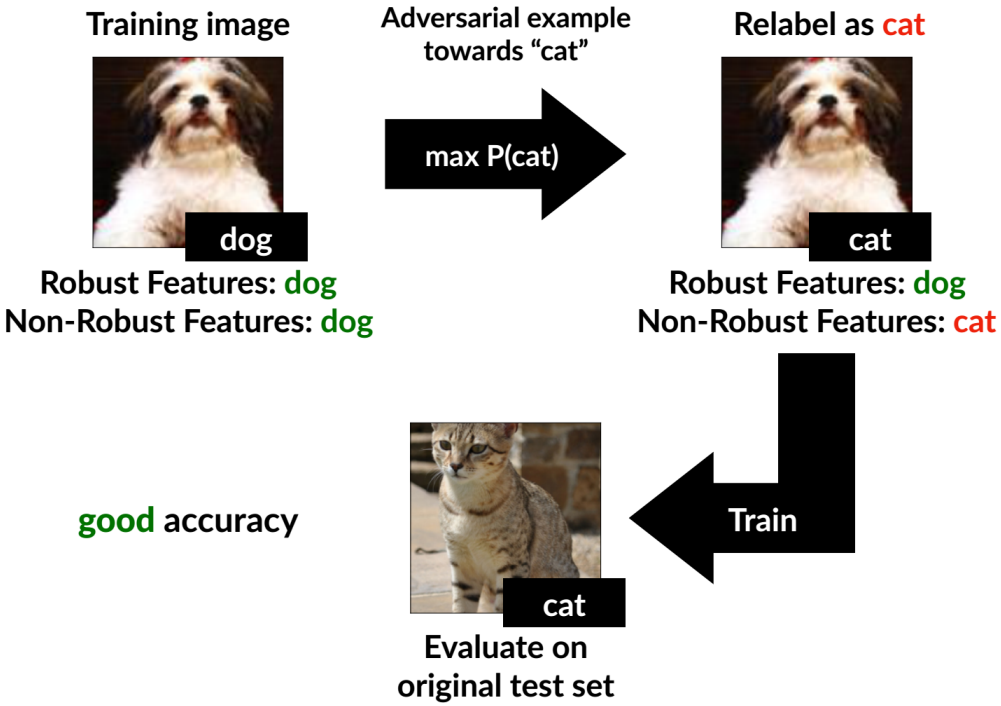
- Learned predictive features of each class by standard training
  - each learned class-specific feature correlates with the class
  - robust features: adding small perturbation to input does not change much correlation between feature and prediction label
  - non-robust features: perturb. makes correlation disappeared; exists in original data, explaining attack transferability



Ilyas et al., "Adversarial Examples are not Bugs, they are Features", NeurIPS, 2019

Interpretation of adversarial examples

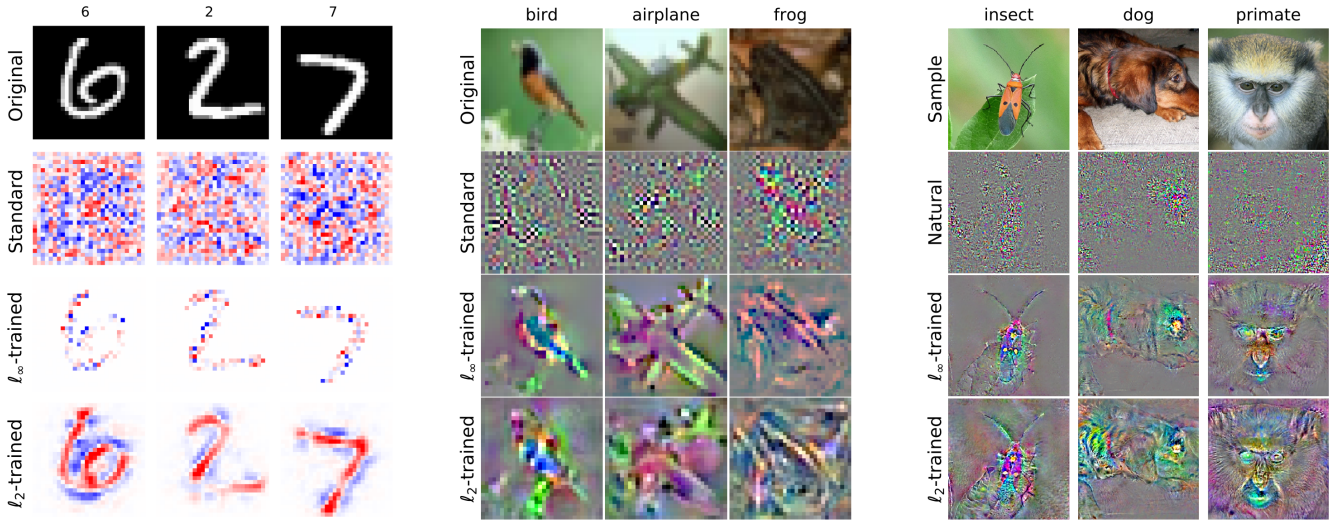
- Leaned predictive features of each class by standard training:
  - each learned class-specific feature correlates with the class
  - robust features: adding small perturbation to input does not change much correlation between feature and prediction label
  - non-robust features: perturb. makes correlation disappeared; exists in original data, explaining attack transferability



Construct a non-robust training set which appears mislabeled to humans (via adversarial examples) but results in good accuracy on the original test set

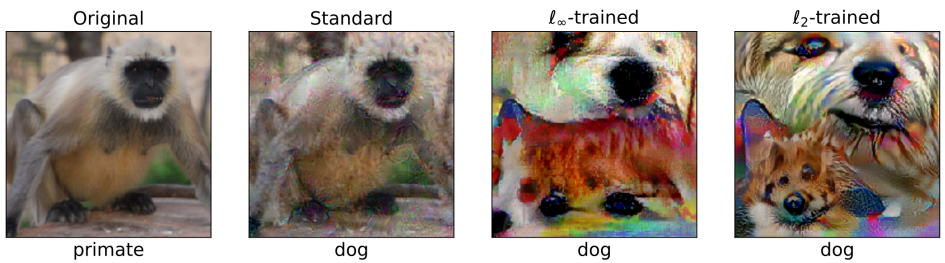
Interpretation of adversarial training

- Robust model by adversarial training learns semantic features
  - gradients are more interpretable for robust models (last 2 rows)
  - indicate: adversarial training learns human-perceptual features



Visualization of the loss gradient with respect to input pixels

- Adversarial examples of robust model exhibit salient features of target class (last 2 cols), while that of standard model not.



Tsipras et al., "Robustness May Be at Odds with Accuracy", ICLR, 2019

Summary

- Adversarial examples are unavoidable
- Various defense methods can improve model robustness
- There is often trade-off between accuracy and robustness
- Robust models more likely learn semantic features

References and further reading:

- Some course material adapted from
  - Stanley Chan, "Machine learning I: Robustness and accuracy trade off", Lecture 37, 2020
- Relevant interesting papers
  - Machiraju et al., 'Bio-inspired Robustness: A Review", arXiv, 2021
  - Tramer et al., 'On Adaptive Attacks to Adversarial Example Defenses", NeurIPS, 2020
- Interpretation of deep learning models
  - Chen et al., 'Neural Ordinary Differential Equations", NeurIPS, 2018