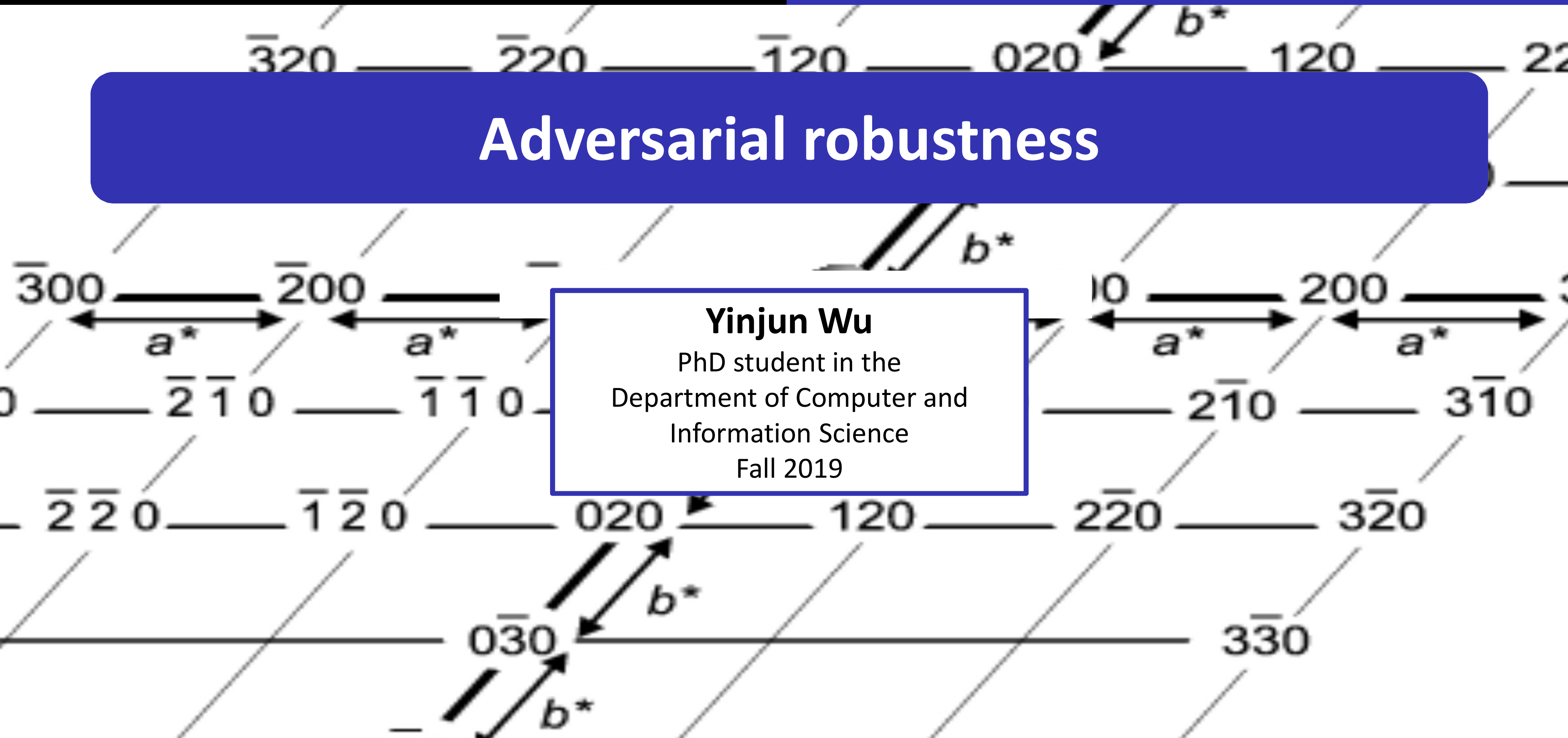
# Adversarial robustness

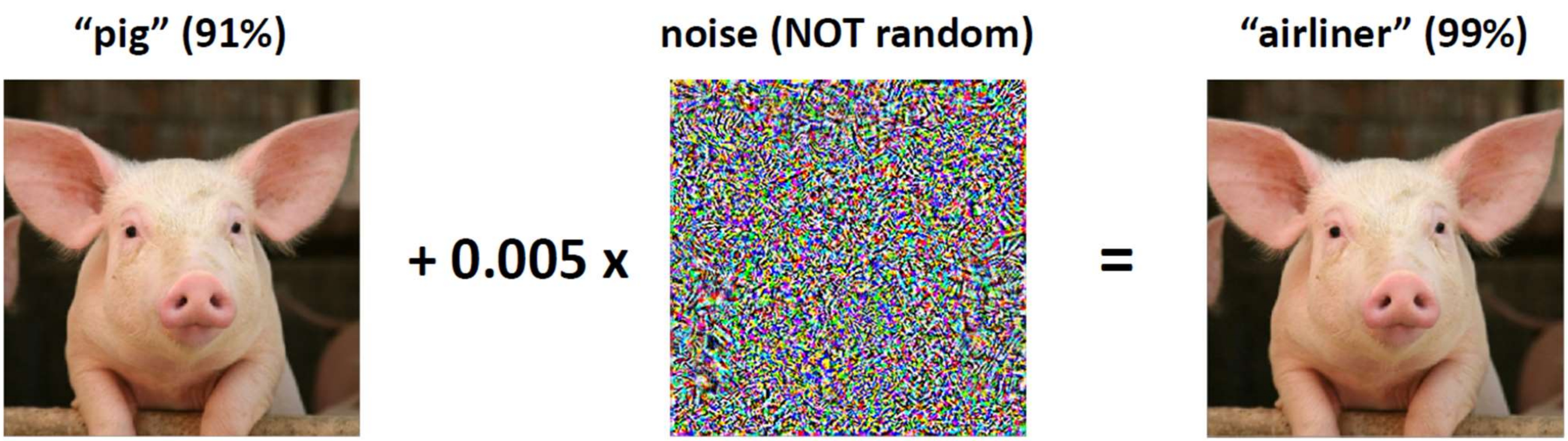

**Yinjun Wu**

PhD student in the
Department of Computer and
Information Science

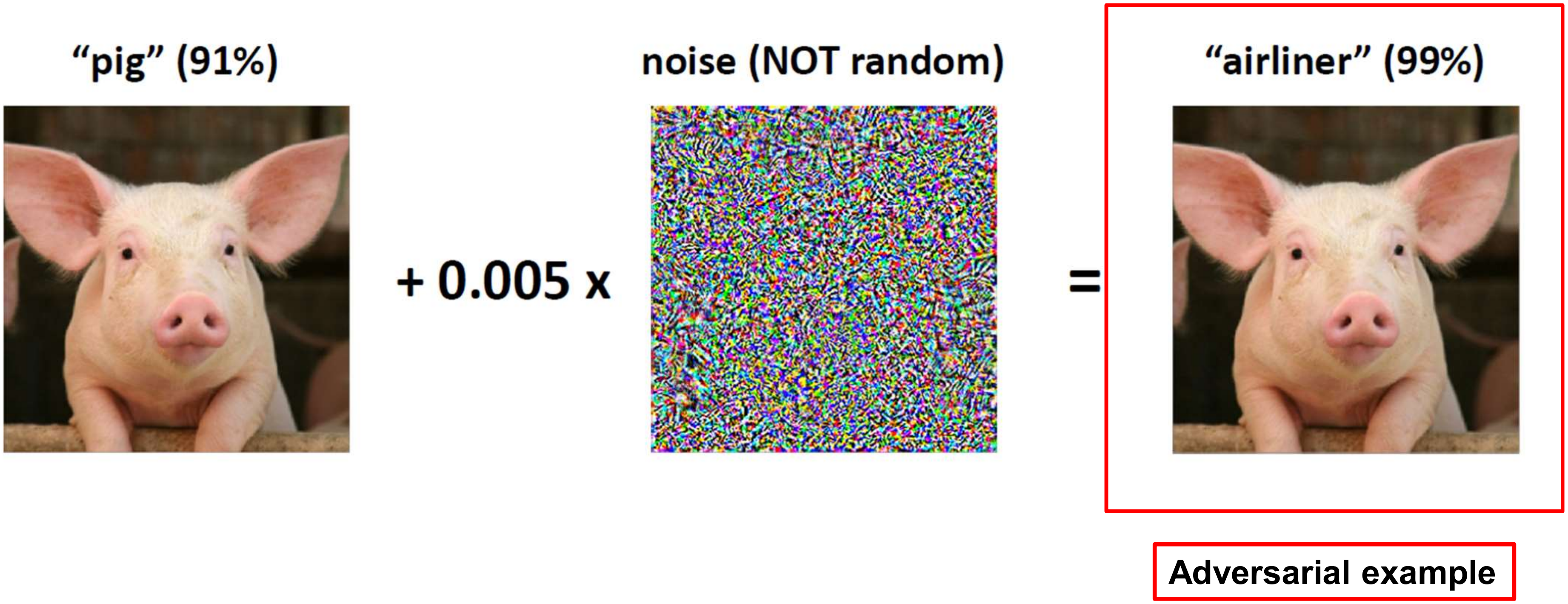

Fall 2019



# ML Predictions Are Accurate but Brittle



# ML Predictions Are Accurate but Brittle



# Typical research problems

- How to attack an ML model ? (attack)
- How to train a robust model with the existence of adversarial examples? (defense)
- How to find out whether adversarial examples exist? (Certifying robustness)
- …

# Table of contents

# Table of contents

# Notations

- Data: $(x, y) \in D(x \in \mathcal{X}, y \in \mathbb{Z})$
- Model: $h_\theta : \mathcal{X} \rightarrow \mathbb{R}^k$
- Loss function: $l : \mathbb{R}^k \times \mathbb{Z} \rightarrow \mathbb{R}_+$
  - Given $(x, y)$, $l(h_\theta(x), y)$
  - Cross-entropy loss: $l(h_\theta(x), y) = \log(\sum_{j=1}^{k} \exp(h_\theta(x)_j)) - h_\theta(x)_y$
- Empirical risk: $R(h_\theta) = \mathbf{E}_{(x,y)\sim\mathcal{D}}[\ell(h_\theta(x)), y)]$
  - Given training set $\{x_i \in \mathcal{X}, y_i \in \mathbb{Z}\}_{i=1}^{m}$, $\min_\theta \frac{1}{m} \sum_{i=1}^{m} l(h_\theta(x_i), y_i)$
  - (stochastic) gradient descent: $\theta^{(t+1)} = \theta^{(t)} - \frac{\alpha}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \nabla_\theta l(h_\theta(x_i), y_i)$

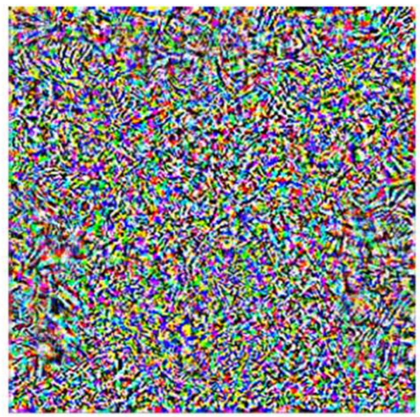
# Mathematical formulation for adversarial example

"pig" (91%)

$x$

**+ 0.005 x**

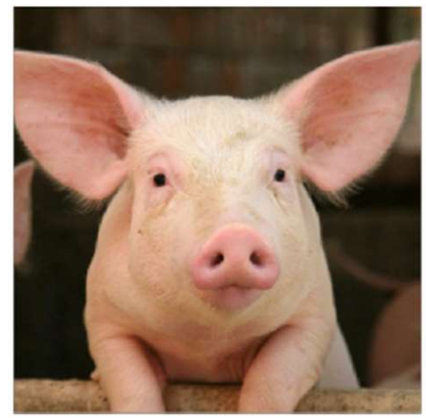
noise (NOT random)



$\delta$

**=**

"airliner" (99%)



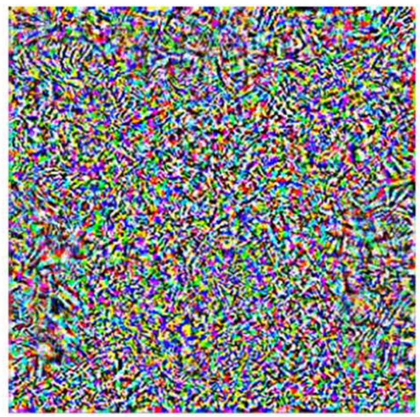
$x + \delta$

$$\max_{\delta \in \Delta} l(h_\theta(x + \delta), y)$$
$$(\Delta = \{\delta : ||\delta||_\infty < \epsilon\})$$

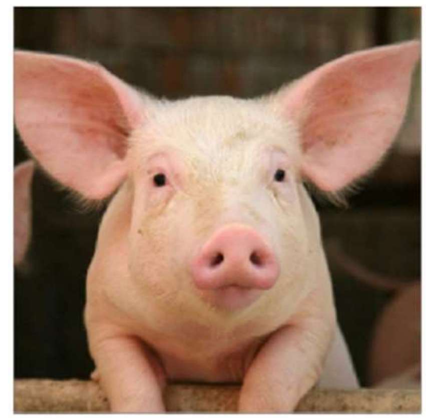
# Mathematical formulation for adversarial example

“pig” (91%) noise (NOT random) “airliner” (99%)

**+ 0.005 x**



**=**



$x$ $\delta$ $x + \delta$

$$\max_{\delta \in \Delta} l(h_\theta(x+\delta), y)$$
$$(\Delta = \{\delta : ||\delta||_\infty < \epsilon\})$$

$$\max_{\delta \in \Delta} [l(h_\theta(x+\delta), y)$$
$$-l(h_\theta(x+\delta), y_{target})]$$

# Empirical risk with adversarial examples

- "Worst case" loss:

$$R_{\text{adv}}(h_\theta) = \mathbf{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\delta\in\Delta(x)} \ell(h_\theta(x+\delta)), y)\right]$$

- Empirical risk given training set $D_{\text{train}}$ :

$$\hat{R}_{\text{adv}}(h_\theta, D_{\text{train}}) = \frac{1}{|D_{\text{train}}|}\sum_{(x,y)\in D_{\text{train}}} \max_{\delta\in\Delta(x)} \ell(h_\theta(x+\delta)), y).$$

- Objective function:

$$\min_\theta \hat{R}_{\text{adv}}(h_\theta, D_{\text{train}}) = \min_\theta \frac{1}{|D_{\text{train}}|}\sum_{(x,y)\in D_{\text{train}}} \max_{\delta\in\Delta(x)} \ell(h_\theta(x+\delta)), y).$$

# Empirical risk with adversarial examples

- "Worst case" loss:

$$R_{\text{adv}}(h_\theta) = \mathbf{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\delta\in\Delta(x)} \ell(h_\theta(x+\delta)), y)\right]$$

- Empirical risk given training set $D_{\text{train}}$ :

$$\hat{R}_{\text{adv}}(h_\theta, D_{\text{train}}) = \frac{1}{|D_{\text{train}}|}\sum_{(x,y)\in D_{\text{train}}} \max_{\delta\in\Delta(x)} \ell(h_\theta(x+\delta)), y).$$

- Objective function:

$$\min_\theta \hat{R}_{\text{adv}}(h_\theta, D_{\text{train}}) = \min_\theta \frac{1}{|D_{\text{train}}|}\sum_{(x,y)\in D_{\text{train}}} \max_{\delta\in\Delta(x)} \ell(h_\theta(x+\delta)), y).$$

- (stochastic) gradient descent:

$$\theta \leftarrow \theta - \frac{\alpha}{|B|}\sum_{(x,y)\in B} \nabla_\theta \max_{\delta\in\Delta} \ell(h_\theta(x+\delta)), y).$$

How to compute this?

# What's next [adml]

- Three types of methods to solve the inner maximization
  - Lower bounding the inner maximization
    - Fast Gradient Sign Method (FGSM) [GSS14]
    - Projected gradient descent (PGD)
  - Combinatorial optimization (mixed integer programming) [Tjeng et al. 2018]
  - Upper bounding the inner maximization (convex relaxations)
- Overall training algorithm

# What’s next [adml]

- Three types of methods to solve the inner maximization
  - Lower bounding the inner maximization
    - Fast Gradient Sign Method (FGSM) [GSS14]
    - Projected gradient descent (PGD)
  - Combinatorial optimization (mixed integer programming) [Tjeng et al. 2018]
  - Upper bounding the inner maximization (convex relaxations)
- Overall training algorithm

# Fast Gradient Sign Method (FGSM) – simple model

- Binary logistic regression (exact solution) [GSS14]:

$$l(h_\theta(x), y) = \log(1 + \exp(-y\theta^T x))$$

$$\text{argmax}_{||\delta||_\infty \leq \epsilon} \ell(h_\theta(x + \delta)), y) = -y\epsilon \cdot \text{sign}(\theta)$$

# Fast Gradient Sign Method (FGSM) – general form

- Updating $\delta$ by single projected gradient step:

$$\delta := \epsilon \cdot \underbrace{\mathrm{sign}(\nabla_\delta \ell(h_\theta(x+\delta), y))}_{\text{steepest descent}}$$

# Projected gradient descent

- Updating $\delta$ iteratively:

$$\delta := \delta + \alpha \nabla_\delta \ell(h_\theta(x + \delta), y)$$

# Projected gradient descent

- Updating $\delta$ iteratively:

Remember that $\delta \in \Delta$

$$\delta := \delta + \alpha \nabla_\delta \ell(h_\theta(x+\delta), y)$$

- Project into the bound:

$$\delta := \mathcal{P}(\delta + \alpha \nabla_\delta \ell(h_\theta(x+\delta), y))$$

- With steepest descent:

$$\delta := \mathcal{P}(\delta + \alpha \cdot \mathrm{sign}(\nabla_\delta \ell(h_\theta(x+\delta), y)))$$

# What’s next [adml]

- Three types of methods to solve the inner maximization
  - Lower bounding the inner maximization
    - Fast Gradient Sign Method (FGSM) [GSS14]
    - Projected gradient descent (PGD)
  - Combinatorial optimization (mixed integer programming) [Tjeng et al. 2018]
  - Upper bounding the inner maximization (convex relaxations)
- Overall training algorithm

# Combinatorial optimization

- Consider DNN with ReLU-activation ($\theta = \{W_i, b_i\}_{i=1,\ldots,d}$)

$$
\begin{aligned}
z_1 &= x \\
z_{i+1} &= f_i(W_i z_i + b_i), \quad i = 1, \ldots, d \\
h_\theta(x) &= z_{d+1}
\end{aligned}
$$

- Explicitly write the problem as:

$$
\begin{aligned}
\min_{z_{1,\ldots,d+1}} \; & z_{d+1,y} - z_{d+1,y_{\text{targ}}} \\
\text{subject to} \; & \|z_1 - x\|_\infty \le \epsilon \\
& z_{i+1} = \max\{0, W_i z_i + b_i\}, i = 1, \ldots, d-1 \\
& z_{d+1} = W_d z_d + b_d
\end{aligned}
$$

# Combinatorial optimization

- Consider DNN with ReLU-activation ($\theta = \{W_i, b_i\}_{i=1,\ldots,d}$)

$$
\begin{aligned}
z_1 &= x \\
z_{i+1} &= f_i(W_i z_i + b_i), \quad i = 1, \ldots, d \\
h_\theta(x) &= z_{d+1}
\end{aligned}
$$

- Explicitly write the problem as:

linearization

$$
\begin{aligned}
\min_{z_{1,\ldots,d+1}} \quad & z_{d+1,y} - z_{d+1,y_{\text{targ}}} \\
\text{subject to} \quad & \|z_1 - x\|_\infty \le \epsilon \\
& \boxed{z_{i+1} = \max\{0, W_i z_i + b_i\}}, \; i = 1, \ldots, d-1 \\
& z_{d+1} = W_d z_d + b_d
\end{aligned}
$$

# Linearization

- Assume that each element of $W_i z_i + b_i$ is bounded by $[l_i, u_i]$

$$z_{i+1} = \max\{0, W_i z_i + b_i\} \quad \Rightarrow \quad \begin{aligned} z_{i+1} &\succcurlyeq W_i z_i + b_i \\ z_{i+1} &\succcurlyeq 0 \\ u_i \cdot v_i &\succcurlyeq z_{i+1} \\ W_i z_i + b_i &\succcurlyeq z_{i+1} + (1 - v_i) l_i \\ v_i &\in \{0, 1\}^{|v_i|} \end{aligned}$$

# Linearization

- Assume that each element of $W_i z_i + b_i$ is bounded by $[l_i, u_i]$

$$z_{i+1} = \max\{0, W_i z_i + b_i\}$$

$$\begin{aligned} z_{i+1} &\succcurlyeq W_i z_i + b_i \\ z_{i+1} &\succcurlyeq 0 \\ u_i \cdot v_i &\succcurlyeq z_{i+1} \\ W_i z_i + b_i &\succcurlyeq z_{i+1} + (1 - v_i) l_i \\ v_i &\in \{0, 1\}^{|v_i|} \end{aligned}$$

$$z_{i+1,j} = \begin{cases} W_{i,j} z_i + b_i & v_{i,j} = 1 \\ 0 & v_{i,j} = 0 \end{cases}$$

# Determine the upper and lower bound

- Assume that each element of $z_i$ is bounded by $[\hat{l}_i, \hat{u}_i]$

$$(W_i z_i + b_i)_j = \sum_k w_{i,jk} z_{i,k} + b_{i,j}$$

$$\sum_k w_{i,jk} z_{i,k} + b_{i,j} \geq \min(w_{i,jk}, 0)\hat{u}_i + \max(w_{i,jk}, 0)\hat{l}_i + b_{i,j}$$
$$\sum_k w_{i,jk} z_{i,k} + b_{i,j} \leq \max(w_{i,jk}, 0)\hat{u}_i + \min(w_{i,jk}, 0)\hat{l}_i + b_{i,j}$$

# Full picture of Combinatorial optimization

$$
\begin{aligned}
\min_{z_{1,\ldots,d+1}, v_{1,\ldots,d-1}} \quad & z_{d+1,y} - z_{d+1,y_{\text{targ}}} \\
\text{subject to} \quad & z_{i+1} \succcurlyeq W_i z_i + b_i, \;\; i = 1\ldots,d-1 \\
& z_{i+1} \succcurlyeq 0, \;\; i = 1\ldots,d-1 \\
& u_i \cdot v_i \succcurlyeq z_{i+1}, \;\; i = 1\ldots,d-1 \\
& W_i z_i + b_i \succcurlyeq z_{i+1} + (1 - v_i) l_i, \;\; i = 1\ldots,d-1 \\
& v_i \in \{0,1\}^{|v_i|}, \;\; i = 1\ldots,d-1 \\
& z_1 \preccurlyeq x + \delta \\
& z_1 \succcurlyeq x - \delta \\
& z_{d+1} = W_d z_d + b_d.
\end{aligned}
$$

# Full picture of Combinatorial optimization

$$
\begin{aligned}
\min_{z_{1,\ldots,d+1}, v_{1,\ldots,d-1}} \quad & z_{d+1,y} - z_{d+1,y_{\text{targ}}} \\
\text{subject to} \quad & z_{i+1} \succcurlyeq W_i z_i + b_i, \quad i = 1\ldots, d-1 \\
& z_{i+1} \succcurlyeq 0, \quad i = 1\ldots, d-1 \\
& u_i \cdot v_i \succcurlyeq z_{i+1}, \quad i = 1\ldots, d-1 \\
& W_i z_i + b_i \succcurlyeq z_{i+1} + (1 - v_i) l_i, \quad i = 1\ldots, d-1 \\
& \boxed{v_i \in \{0,1\}^{|v_i|}}, \quad i = 1\ldots, d-1 \\
& z_1 \preccurlyeq x + \delta \\
& z_1 \succcurlyeq x - \delta \\
& z_{d+1} = W_d z_d + b_d.
\end{aligned}
$$

Branch-and-Bound

# Certifying robustness

- Combinatorial optimization provides “exact” solutions
  - Can determine whether any adversarial example exists for a given example by looking at the sign of the objective function

$$\min_{z_{1,\ldots,d+1}, v_{1,\ldots,d-1}} z_{d+1,y} - z_{d+1,y_{\text{targ}}}$$

  - For any alternative label, negative results mean the existence of adversarial examples
  - For all alternative labels, positive results mean that there is no adversarial examples

# What's next [adml]

- Three types of methods to solve the inner maximization
  - Lower bounding the inner maximization
    - Fast Gradient Sign Method (FGSM) [GSS14]
    - Projected gradient descent (PGD)
  - Combinatorial optimization (mixed integer programming) [Tjeng et al. 2018]
  - Upper bounding the inner maximization (convex relaxations)
- Overall training algorithm

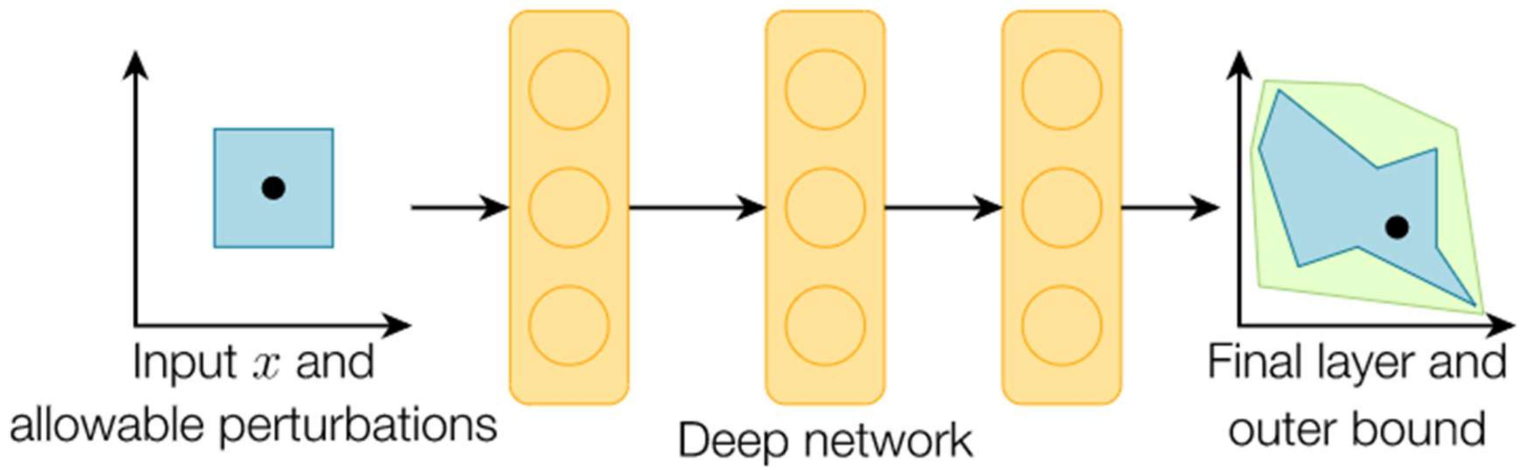
# convex relaxations

$$
\begin{aligned}
\min_{z_{1,\ldots,d+1}, v_{1,\ldots,d-1}} \quad & z_{d+1,y} - z_{d+1,y_{\text{targ}}} \\
\text{subject to} \quad & z_{i+1} \geq W_i z_i + b_i, \quad i = 1\ldots, d-1 \\
& z_{i+1} \geq 0, \quad i = 1\ldots, d-1 \\
& u_i \cdot v_i \geq z_{i+1}, \quad i = 1\ldots, d-1 \\
& W_i z_i + b_i \geq z_{i+1} + (1 - v_i) l_i, \quad i = 1\ldots, d-1 \\
& v_i \in [0,1]^{|v_i|}, \quad i = 1\ldots, d-1 \\
& z_1 \leq x + \delta \\
& z_1 \geq x - \delta \\
& z_{d+1} = W_d z_d + b_d.
\end{aligned}
$$

# convex relaxations

$$
\begin{aligned}
\min_{z_{1,\ldots,d+1}, v_{1,\ldots,d-1}} \quad & z_{d+1,y} - z_{d+1,y_{\text{targ}}} \\
\text{subject to} \quad & z_{i+1} \geq W_i z_i + b_i, \quad i = 1\ldots,d-1 \\
& z_{i+1} \geq 0, \quad i = 1\ldots,d-1 \\
& u_i \cdot v_i \geq z_{i+1}, \quad i = 1\ldots,d-1 \\
& W_i z_i + b_i \geq z_{i+1} + (1 - v_i) l_i, \quad i = 1\ldots,d-1 \\
& \boxed{v_i \in [0,1]^{|v_i|},} \quad i = 1\ldots,d-1 \\
& z_1 \leq x + \delta \\
& z_1 \geq x - \delta \\
& z_{d+1} = W_d z_d + b_d.
\end{aligned}
$$

# The effect of relaxation



# Certifying robustness

- Can determine whether any adversarial example exists for a given example by looking at the sign of the objective function

$$\min_{z_{1,\ldots,d+1}, v_{1,\ldots,d-1}} z_{d+1,y} - z_{d+1,y_{\text{targ}}}$$

- For all alternative labels, positive results still mean that there is no adversarial examples
- But the converse is not true

# What's next [adml]

- Three types of methods to solve the inner maximization
  - Lower bounding the inner maximization
    - Fast Gradient Sign Method (FGSM) [Goodfellow et al. (2014b)]
    - Projected gradient descent (PGD)
  - Combinatorial optimization (mixed integer programming) [Tjeng et al. 2018]
  - Upper bounding the inner maximization (convex relaxations)
- Overall training algorithm

# Overall training algorithm

Repeat:

1. Select minibatch $B$, initialize gradient vector $g := 0$
2. For each $(x, y)$ in $B$:
   a. Find an attack perturbation $\delta^\star$ by (approximately) optimizing
   $$\boxed{\delta^\star = \underset{\|\delta\| \leq \epsilon}{\operatorname{argmax}} \, \ell(h_\theta(x + \delta), y)}$$
   b. Add gradient at $\delta^\star$
   $$g := g + \nabla_\theta \ell(h_\theta(x + \delta^\star), y)$$
3. Update parameters $\theta$
$$\theta := \theta - \frac{\alpha}{|B|} g$$

# Other variants

- Cascade adversarial training [Na et al., 2018]
  - Once generating one adversarial example, adding it to augment training set
- Thermometer encoding [Goodfellow et al. (2018)]
  - Discretize each input
  - Propose Logit-Space Projected Gradient Ascent to find adversarial examples

# Other variants – cont.

- Stochastic Activation Pruning [Dhillon et al., 2018]
- Mitigating through Randomization [Xie et al. 2018]
  - Adding one random layer before the input to the model
- ....

# Evaluation of various adversarial training

- Most of those defenses failed with a new attack technique called “Backward Pass Differentiable Approximation” in [Athalye et al. (2018)]
  - Replace non-differentiable layer with an approximate differentiable function for backward computation (but not in the forward computation)
- More re-evaluations are needed

# Table of contents

# What’s next

- How to attack ML model via poisoning existing training set [Mei et al. 2015]
- How to attack ML model via adding poisoned training samples [Steinhardt et al. 2017]

# What's next

- How to attack ML model via poisoning existing training set [Mei et al. 2015]
- How to attack ML model via adding poisoned training samples [Steinhardt et al. 2017]

# Notations

- Data: $(x, y) \in D(x \in \mathcal{X}, y \in \mathbb{Z})$
- Model: $h_\theta : \mathcal{X} \rightarrow \mathbb{R}^k$
- Loss function: $l : \mathbb{R}^k \times \mathbb{Z} \rightarrow \mathbb{R}_+$
  - Given $(x, y)$, $l(h_\theta(x), y)$
  - Cross entropy loss: $l(h_\theta(x), y) = \log(\sum_{j=1}^{k} \exp(h_\theta(x)_j)) - h_\theta(x)_y$
- Empirical risk: $R(h_\theta) = \mathbf{E}_{(x,y)\sim\mathcal{D}}[\ell(h_\theta(x)), y)]$
  - Given training set $\{x_i \in \mathcal{X}, y_i \in \mathbb{Z}\}_{i=1}^{m}$, $\min_\theta \frac{1}{m} \sum_{i=1}^{m} l(h_\theta(x_i), y_i)$

# Notations

- Data: $(x, y) \in D(x \in \mathcal{X}, y \in \mathbb{Z})$
- Model: $h_\theta : \mathcal{X} \to \mathbb{R}^k$
- Loss function: $l : \mathbb{R}^k \times \mathbb{Z} \to \mathbb{R}_+$
  - Given $(x, y)$, $l(h_\theta(x), y)$
  - Cross entropy loss: $l(h_\theta(x), y) = \log(\sum_{j=1}^{k} \exp(h_\theta(x)_j)) - h_\theta(x)_y$
- Empirical risk: $R(h_\theta) = \mathbf{E}_{(x,y)\sim\mathcal{D}}[\ell(h_\theta(x)), y)]$
  - Given training set $\{x_i \in \mathcal{X}, y_i \in \mathbb{Z}\}_{i=1}^{m}$, $\min_\theta \frac{1}{m} \sum_{i=1}^{m} l(h_\theta(x_i), y_i)$

$$\mathbf{g}(\theta) = [g_1(\theta), g_2(\theta), \ldots, g_k(\theta)]^T \preccurlyeq \mathbf{0}$$
$$\mathbf{s}(\theta) = [s_1(\theta), s_2(\theta), \ldots, s_p(\theta)]^T = \mathbf{0}$$

# Attackers' goal

- The learned model VS the attackers' expected model ( $\theta^*$)
  - $R_A(\hat{\theta}) = ||\hat{\theta} - \theta^*||$
- The changes over the training set $D_0$
  - $C_A(D, D_0) = ||X - X_0||_F$
- With smallest changes over training set, the learned model should be as close as possible to the expected model

$$\min_{D,\hat{\theta}} R_A(\hat{\theta}) + C_A(D, D_0)$$

$$\text{subject to} \;\; \hat{\theta} = \text{argmin}_\theta R(h_\theta)$$

$$\text{subject to} \;\; \mathbf{g}(\theta) \preccurlyeq \mathbf{0}, \mathbf{s}(\theta) = \mathbf{0}$$

# Solve the optimization problem

$$\min_{D,\hat{\theta}} R_A(\hat{\theta}) + C_A(D, D_0)$$

$$\text{subject to} \quad \hat{\theta} = \text{argmin}_{\theta} R(h_\theta)$$

$$\text{subject to} \quad \mathbf{g}(\theta) \preccurlyeq \mathbf{0}, \mathbf{s}(\theta) = \mathbf{0}$$

# Solve the optimization problem

$$\min_{D,\hat{\theta}} R_A(\hat{\theta}) + C_A(D, D_0)$$

$$\text{subject to } \; \hat{\theta} = \text{argmin}_{\theta} R(h_\theta)$$

$$\text{subject to } \; \mathbf{g}(\theta) \preccurlyeq \mathbf{0}, \mathbf{s}(\theta) = \mathbf{0}$$

**KKT conditions**

$$\min_{D,\hat{\theta},\lambda,\mu} \; R_A(\hat{\theta}) + C_A(D, D_0)$$

$$\text{subject to } \; \partial_\theta(R(h_\theta) + \lambda^T \mathbf{g}(\theta) + \mu^T \mathbf{s}(\theta)) = \mathbf{0}$$

$$\mathbf{g}(\theta) \leq \mathbf{0}, \mathbf{h}(\theta) = \mathbf{0}, \lambda \geq \mathbf{0}, \lambda_i g_i(\theta) = 0$$

# Solve the optimization problem

$$\min_{D,\hat{\theta}} R_A(\hat{\theta}) + C_A(D, D_0)$$

$$\text{subject to } \ \hat{\theta} = \text{argmin}_{\theta} R(h_\theta)$$

$$\text{subject to } \ \mathbf{g}(\theta) \preccurlyeq \mathbf{0}, \mathbf{s}(\theta) = \mathbf{0}$$

**KKT conditions**

$$\min_{D,\hat{\theta},\lambda,\mu} \ R_A(\hat{\theta}) + C_A(D, D_0)$$

$$\text{subject to } \ \partial_\theta (R(h_\theta) + \lambda^T \mathbf{g}(\theta) + \mu^T \mathbf{s}(\theta)) = \mathbf{0}$$

$$\mathbf{g}(\theta) \leq \mathbf{0}, \mathbf{h}(\theta) = \mathbf{0}, \lambda \geq \mathbf{0}, \lambda_i g_i(\theta) = 0$$

$$D^{(t+1)} = \mathcal{P}(D^{(t)} + \alpha_t \nabla_D (R_A(\theta) + C_A(D, D_0))|_{D=D^{(t)}})$$

# Solve the optimization problem

$$\min_{D,\hat{\theta}} R_A(\hat{\theta}) + C_A(D, D_0)$$

$$\text{subject to } \hat{\theta} = \text{argmin}_\theta R(h_\theta)$$

$$\text{subject to } \mathbf{g}(\theta) \preccurlyeq \mathbf{0}, \mathbf{s}(\theta) = \mathbf{0}$$

**KKT conditions**

$$\min_{D,\hat{\theta},\lambda,\mu} R_A(\hat{\theta}) + C_A(D, D_0)$$

$$\text{subject to } \partial_\theta(R(h_\theta) + \lambda^T \mathbf{g}(\theta) + \mu^T \mathbf{s}(\theta)) = \mathbf{0}$$

$$\mathbf{g}(\theta) \le \mathbf{0}, \mathbf{h}(\theta) = \mathbf{0}, \lambda \ge \mathbf{0}, \lambda_i g_i(\theta) = 0$$

$$D^{(t+1)} = \mathcal{P}(D^{(t)} + \alpha_t \nabla_D (R_A(\theta) + C_A(D, D_0))|_{D=D^{(t)}})$$

$$\nabla_D (R_A(\theta) + C_A(D, D_0))$$
$$= \nabla_\theta (R_A(\theta) + C_A(D, D_0)\frac{\partial \theta}{\partial D}$$

# Solve the optimization problem

$$\min_{D,\hat{\theta}} R_A(\hat{\theta}) + C_A(D, D_0)$$

$$\text{subject to } \hat{\theta} = \text{argmin}_\theta R(h_\theta)$$

$$\text{subject to } \mathbf{g}(\theta) \preccurlyeq \mathbf{0}, \mathbf{s}(\theta) = \mathbf{0}$$

**KKT conditions**

$$\min_{D,\hat{\theta},\lambda,\mu} R_A(\hat{\theta}) + C_A(D, D_0)$$

$$\text{subject to } \partial_\theta(R(h_\theta) + \lambda^T \mathbf{g}(\theta) + \mu^T \mathbf{s}(\theta)) = \mathbf{0}$$

$$\mathbf{g}(\theta) \leq \mathbf{0}, \mathbf{h}(\theta) = \mathbf{0}, \lambda \geq \mathbf{0}, \lambda_i g_i(\theta) = 0$$

$$D^{(t+1)} = \mathcal{P}(D^{(t)} + \alpha_t \nabla_D (R_A(\theta) + C_A(D, D_0))|_{D=D^{(t)}})$$

$$\nabla_D (R_A(\theta) + C_A(D, D_0))$$
$$= \nabla_\theta (R_A(\theta) + C_A(D, D_0) \frac{\partial \theta}{\partial D}$$

$$\mathbf{f}(D, \theta, \lambda, \mu) = \begin{bmatrix} \partial_\theta(R(h_\theta) + \lambda^T \mathbf{g}(\theta) + \mu^T \mathbf{s}(\theta)) \\ \lambda_i g_i(\theta), i = 1, 2, \ldots \\ \mathbf{h}(\theta) \end{bmatrix} = \mathbf{0}$$

# Solve the optimization problem

$$\min_{D,\hat{\theta}} R_A(\hat{\theta}) + C_A(D, D_0)$$

$$\text{subject to } \hat{\theta} = \text{argmin}_{\theta} R(h_\theta)$$

$$\text{subject to } \mathbf{g}(\theta) \preccurlyeq \mathbf{0}, \mathbf{s}(\theta) = \mathbf{0}$$

**KKT conditions**

$$\min_{D,\hat{\theta},\lambda,\mu} R_A(\hat{\theta}) + C_A(D, D_0)$$

$$\text{subject to } \partial_\theta(R(h_\theta) + \lambda^T \mathbf{g}(\theta) + \mu^T \mathbf{s}(\theta)) = \mathbf{0}$$

$$\mathbf{g}(\theta) \leq \mathbf{0}, \mathbf{h}(\theta) = \mathbf{0}, \lambda \geq \mathbf{0}, \lambda_i g_i(\theta) = 0$$

$$D^{(t+1)} = \mathcal{P}(D^{(t)} + \alpha_t \nabla_D (R_A(\theta) + C_A(D, D_0))|_{D=D^{(t)}})$$

$$\nabla_D (R_A(\theta) + C_A(D, D_0))$$
$$= \nabla_\theta (R_A(\theta) + C_A(D, D_0) \frac{\partial \theta}{\partial D}$$

$$\mathbf{f}(D, \theta, \lambda, \mu) = \begin{bmatrix} \partial_\theta(R(h_\theta) + \lambda^T \mathbf{g}(\theta) + \mu^T \mathbf{s}(\theta)) \\ \lambda_i g_i(\theta), i = 1, 2, \ldots \\ \mathbf{h}(\theta) \end{bmatrix} = \mathbf{0}$$

$$vec(\theta, \lambda, \mu) \triangleq \mathcal{F}(D)$$

# Solve the optimization problem

$$\min_{D,\hat{\theta}} R_A(\hat{\theta}) + C_A(D, D_0)$$

$$\text{subject to } \hat{\theta} = \text{argmin}_\theta R(h_\theta)$$

$$\text{subject to } \mathbf{g}(\theta) \preccurlyeq \mathbf{0}, \mathbf{s}(\theta) = \mathbf{0}$$

**KKT conditions**

$$\min_{D,\hat{\theta},\lambda,\mu} R_A(\hat{\theta}) + C_A(D, D_0)$$

$$\text{subject to } \partial_\theta(R(h_\theta) + \lambda^T \mathbf{g}(\theta) + \mu^T \mathbf{s}(\theta)) = \mathbf{0}$$

$$\mathbf{g}(\theta) \leq \mathbf{0}, \mathbf{h}(\theta) = \mathbf{0}, \lambda \geq \mathbf{0}, \lambda_i g_i(\theta) = 0$$

$$D^{(t+1)} = \mathcal{P}(D^{(t)} + \alpha_t \nabla_D (R_A(\theta) + C_A(D, D_0))|_{D=D^{(t)}})$$

$$\nabla_D (R_A(\theta) + C_A(D, D_0))$$
$$= \nabla_\theta (R_A(\theta) + C_A(D, D_0) \frac{\partial \theta}{\partial D}$$

$$\mathbf{f}(D, \theta, \lambda, \mu) = \begin{bmatrix} \partial_\theta(R(h_\theta) + \lambda^T \mathbf{g}(\theta) + \mu^T \mathbf{s}(\theta)) \\ \lambda_i g_i(\theta), i = 1, 2, \ldots \\ \mathbf{h}(\theta) \end{bmatrix} = \mathbf{0}$$

$$vec(\theta, \lambda, \mu) \triangleq \mathcal{F}(D)$$

$$\frac{\partial \mathcal{F}}{\partial D} = -\left(\frac{\partial \mathbf{f}}{\partial vec[\theta, \lambda, \mu]}\right)^{-1} \frac{\partial \mathbf{f}}{\partial D}$$

# What’s next

- How to attack ML model via poisoning existing training set [Mei et al. 2015]
- How to attack ML model via adding poisoned training samples [Steinhardt et al. 2017]

# Notations

- Clean training set (size of $n$): $D_c$
- Poisoned data set (size of $\epsilon n$): $D_p$
- The defender trains a model over $D_c \bigcup D_p$
- Test Loss: $\mathbf{L}(\theta) = \mathbf{E}_{(x,y)\sim p^*}[\ell(h_\theta(x), y)]$
- The defender aims at minimizing the test loss while the attacker aims at maximizing the test loss

# Notations

- Clean training set (size of $n$): $D_c$
- Poisoned data set (size of $\epsilon n$): $D_p$
- The defender trains a model over $D_c \bigcup D_p$
- Test Loss: $\mathbf{L}(\theta) = \mathbf{E}_{(x,y)\sim p^*} \boxed{[\ell(h_\theta(x), y)]}$

  Convex loss function
- The defender aims at minimizing the test loss while the attacker aims at maximizing the test loss

# Defense strategies

- Data sanitization defense:

$$\hat{\theta} = \text{argmin}_\theta \sum_{(x,y)\in(D_c \bigcup D_p)\bigcap \mathbb{F}} l(h_\theta(x), y)$$

- Typical defense (**sphere defense**, the radius is $r_y$ and the centroid is $\mu_y$):
  - $\mathbb{F}_{\text{sphere}} = \{(x, y) | ||x - \mu_y|| \leq r_y\}$
- Fixed defense:
  - $\mathbb{F}$ not dependent on $D_p$
- Data-dependent defense:
  - $\mathbb{F}$ dependent on $D_p$

# Approximation for defense and attack

- Upper bounding test loss:

$$\mathbf{L}(\theta) = \mathbf{E}_{(x,y)\sim p^*}[\ell(h_\theta(x), y)] \approx \frac{1}{n}\sum_{(x,y)\in D_c} \ell(h_\theta(x), y) \leq \frac{1}{n}\sum_{(x,y)\in D_c \bigcup D_p} \ell(h_\theta(x), y)$$

- Trained model parameters approx:

$$\hat{\theta} = \text{argmin}_\theta \sum_{(x,y)\in(D_c \bigcup D_p)\bigcap \mathbb{F}} l(h_\theta(x), y)$$
$$\approx \tilde{\theta} = \text{argmin}_\theta \sum_{(x,y)\in D_c \bigcup (D_p \bigcap \mathbb{F})} l(h_\theta(x), y)$$

- Upper bounding attack:

$$\max_{D_p} \mathbf{L}(\hat{\theta}) \approx \max_{D_p} \frac{1}{n}\sum_{(x,y)\in D_c} \ell(h_{\hat{\theta}}(x), y) \leq \max_{D_p} \frac{1}{n}\sum_{(x,y)\in D_c \bigcup (D_p \bigcap \mathbb{F})} \ell(h_{\hat{\theta}}(x), y)$$
$$\approx \max_{D_p} \frac{1}{n}\sum_{(x,y)\in D_c \bigcup (D_p \bigcap \mathbb{F})} \ell(h_{\tilde{\theta}}(x), y) = \max_{D_p \subseteq \mathbb{F}} \min_\theta \frac{1}{n}\sum_{(x,y)\in D_c \bigcup D_p} \ell(h_\theta(x), y)$$

# Approximation for defense and attack

- Upper bounding test loss:

$$\mathbf{L}(\theta) = \mathbf{E}_{(x,y)\sim p^*}[\ell(h_\theta(x), y)] \approx \frac{1}{n}\sum_{(x,y)\in D_c} \ell(h_\theta(x), y) \leq \frac{1}{n}\sum_{(x,y)\in D_c \bigcup D_p} \ell(h_\theta(x), y)$$

- Trained model parameters approx:

$$\hat{\theta} = \mathrm{argmin}_\theta \sum_{(x,y)\in(D_c \bigcup D_p)\bigcap \mathbb{F}} l(h_\theta(x), y)$$

$$\approx \tilde{\theta} = \mathrm{argmin}_\theta \sum_{(x,y)\in D_c \bigcup (D_p \bigcap \mathbb{F})} l(h_\theta(x), y)$$

- Upper bounding attack:

Defined as $\mathbf{M}$

$$\max_{D_p} \mathbf{L}(\hat{\theta}) \approx \max_{D_p} \frac{1}{n}\sum_{(x,y)\in D_c} \ell(h_{\hat{\theta}}(x), y) \leq \max_{D_p} \frac{1}{n}\sum_{(x,y)\in D_c \bigcup (D_p \bigcap \mathbb{F})} \ell(h_{\hat{\theta}}(x), y)$$

$$\approx \max_{D_p} \frac{1}{n}\sum_{(x,y)\in D_c \bigcup (D_p \bigcap \mathbb{F})} \ell(h_{\tilde{\theta}}(x), y) = \boxed{\max_{D_p \subseteq \mathbb{F}} \min_\theta \frac{1}{n}\sum_{(x,y)\in D_c \bigcup D_p} \ell(h_\theta(x), y)}$$

# Attack under fixed defense

- Further bounding $\mathbf{M}$

$$\mathbf{M} = \max_{D_p \subseteq \mathbb{F}} \min_\theta \frac{1}{n} \sum_{(x,y) \in D_c \bigcup D_p} \ell(h_\theta(x), y)$$
$$\leq \min_\theta \max_{D_p \subseteq \mathbb{F}} \frac{1}{n} \sum_{(x,y) \in D_c \bigcup D_p} \ell(h_\theta(x), y)$$
$$= \min_\theta \frac{1}{n} \sum_{(x,y) \in D_c} \ell(h_\theta(x), y) + \min_\theta \max_{D_p \subseteq \mathbb{F}} \frac{1}{n} \sum_{(x,y) \in D_p} \ell(h_\theta(x), y)$$
$$\leq \min_\theta \{\frac{1}{n} \sum_{(x,y) \in D_c} \ell(h_\theta(x), y) + \epsilon \max_{(x,y) \in \mathbb{F}} \ell(h_\theta(x), y)\}$$

# Attack under fixed defense

- Further bounding $\mathbf{M}$

$$\mathbf{M} = \max_{D_p \subseteq \mathbb{F}} \min_\theta \frac{1}{n} \sum_{(x,y)\in D_c \bigcup D_p} \ell(h_\theta(x), y)$$
$$\leq \min_\theta \max_{D_p \subseteq \mathbb{F}} \frac{1}{n} \sum_{(x,y)\in D_c \bigcup D_p} \ell(h_\theta(x), y)$$
$$= \min_\theta \frac{1}{n} \sum_{(x,y)\in D_c} \ell(h_\theta(x), y) + \min_\theta \max_{D_p \subseteq \mathbb{F}} \frac{1}{n} \sum_{(x,y)\in D_p} \ell(h_\theta(x), y)$$
$$\leq \min_\theta \boxed{\{\frac{1}{n} \sum_{(x,y)\in D_c} \ell(h_\theta(x), y) + \epsilon \max_{(x,y)\in \mathbb{F}} \ell(h_\theta(x), y)\}}$$

Defined as $U(\theta)$

# Attack under fixed defense – cont.

**Algorithm 1:** Online learning algorithm for generating an upper bound and candidate attack.

**Input** : clean dataset $D_c$ of size $n$, feasible set $\mathcal{F}$, $\epsilon, \eta$

**Output:** The upper bound of loss $U^*$ and candidate attack $\mathrm{D}_p = \{(x^{(t)}, y^{(t)})\}_{t=1}^{\epsilon n}$

Initialize $\theta^{(0)} = \mathbf{0}$

**for** $t = 1, 2, \ldots, \epsilon n$ **do**

  compute $(x^{(t)}, y^{(t)}) = \text{argmax}_{(x,y)\in\mathbb{F}} \ell(h_{\theta^{(t-1)}}(x), y)$

  $U(\theta^{(t-1)}) = \frac{1}{n} \sum_{(x,y)\in D_c} \ell(h_{\theta^{(t-1)}}(x), y) + \epsilon \ell(h_{\theta^{(t-1)}}(x^{(t)}), y^{(t)})$

  update $\theta^{(t)}$ by gradient descent

**end**

$U^* = \min_{t=1}^{\epsilon n} U(\theta^{(t)})$

# Analysis over the algorithm

- Upper bound:

$$\mathbf{M} \le \min_\theta \{\frac{1}{n}\sum_{(x,y)\in D_c} \ell(h_\theta(x),y) + \epsilon \max_{(x,y)\in\mathbb{F}} \ell(h_\theta(x),y)\}$$
$$\le \min_{t=1}^{\epsilon n} U(\theta^{(t)}) = U^*$$

- Lower bound:

$$\mathbf{M} = \max_{D_p\subseteq\mathbb{F}} \min_\theta \frac{1}{n}\sum_{(x,y)\in D_c\bigcup D_p} \ell(h_\theta(x),y)$$
$$\ge \min_\theta \frac{1}{n}\sum_{(x,y)\in D_c\bigcup D_p} \ell(h_\theta(x),y) = \frac{1}{n}\sum_{(x,y)\in D_c\bigcup D_p} \ell(h_{\tilde\theta}(x),y)$$

- Gap between the two bounds vanishes:

$$U^* - \frac{1}{n}\sum_{(x,y)\in D_c\bigcup D_p} \ell(h_{\tilde\theta}(x),y) \le \frac{\mathrm{Regret}(\epsilon n)}{\epsilon n}$$

# Analysis over the algorithm

- Upper bound:

$$\mathbf{M} \le \min_\theta\{\frac{1}{n}\sum_{(x,y)\in D_c} \ell(h_\theta(x),y) + \epsilon \max_{(x,y)\in\mathbb{F}} \ell(h_\theta(x),y)\}$$
$$\le \min_{t=1}^{\epsilon n} U(\theta^{(t)}) = U^*$$

- Lower bound:

$$\mathbf{M} = \max_{D_p\subseteq\mathbb{F}} \min_\theta \frac{1}{n}\sum_{(x,y)\in D_c\bigcup D_p} \ell(h_\theta(x),y)$$
$$\ge \min_\theta \frac{1}{n}\sum_{(x,y)\in D_c\bigcup D_p} \ell(h_\theta(x),y) = \frac{1}{n}\sum_{(x,y)\in D_c\bigcup D_p} \ell(h_{\tilde{\theta}}(x),y)$$

- Gap between the two bounds vanishes:

$$U^* - \frac{1}{n}\sum_{(x,y)\in D_c\bigcup D_p} \ell(h_{\tilde{\theta}}(x),y) \le \boxed{\frac{\mathrm{Regret}(\epsilon n)}{\epsilon n}}$$

Small enough

# Attack under Data-dependent defense

- Upper bounding $\mathbf{M}$:

$$
\begin{aligned}
\mathbf{M} &= \max_{D_p \subseteq \mathbb{F}(D_p)} \min_\theta \tfrac{1}{n} \sum_{(x,y)\in D_c \bigcup D_p} \ell(h_\theta(x), y) \\
&= \min_\theta \tfrac{1}{n} \sum_{(x,y)\in D_c} \ell(h_\theta(x), y) + \min_\theta \max_{D_p \subseteq \mathbb{F}(D_p)} \tfrac{1}{n} \sum_{(x,y)\in D_p} \ell(h_\theta(x), y) \\
&\leq \min_\theta \{ \tfrac{1}{n} \sum_{(x,y)\in D_c} \ell(h_\theta(x), y) + \epsilon \max_{(x,y)\in \mathbb{F}(D_p)} \ell(h_\theta(x), y) \}
\end{aligned}
$$

# Attack under Data-dependent defense

- Upper bounding $\mathbf{M}$:

$$\begin{aligned}
\mathbf{M} &= \max_{D_p \subseteq \mathbb{F}(D_p)} \min_\theta \frac{1}{n} \sum_{(x,y)\in D_c \bigcup D_p} \ell(h_\theta(x), y) \\
&= \min_\theta \frac{1}{n} \sum_{(x,y)\in D_c} \ell(h_\theta(x), y) + \min_\theta \max_{D_p \subseteq \mathbb{F}(D_p)} \frac{1}{n} \sum_{(x,y)\in D_p} \ell(h_\theta(x), y) \\
&\cancel{\leq \min_\theta \{ \frac{1}{n} \sum_{(x,y)\in D_c} \ell(h_\theta(x), y) + \epsilon \max_{(x,y)\in \mathbb{F}(D_p)} \ell(h_\theta(x), y) \}}
\end{aligned}$$

- Consider a probability distribution over $D_p$

# Putting all together

- Objective function:
  - Adversarial robustness for prediction:

$$\min_\theta \frac{1}{|D_{\text{train}}|} \sum_{(x,y)\in D_{\text{train}}} \max_{\delta\in\Delta(x)} \ell(h_\theta(x+\delta)), y).$$

  - Adversarial robustness for training (data poisoning):
    - Poisoning existing training samples:

$$\min_{D,\hat{\theta}} ||\hat{\theta} - \theta^*|| + ||X - X_0|| \ s.t. \ \hat{\theta} = \text{argmin}_\theta R(h_\theta) \ s.t. \ \mathbf{g}(\theta) \preccurlyeq \mathbf{0}, \mathbf{s}(\theta) = \mathbf{0}$$

    - Adding poisoning data:

$$\mathbf{M} = \max_{D_p\subseteq\mathbb{F}} \min_\theta \frac{1}{n} \sum_{(x,y)\in D_c \bigcup D_p} \ell(h_\theta(x), y)$$
$$\leq \min_\theta \max_{D_p\subseteq\mathbb{F}} \frac{1}{n} \sum_{(x,y)\in D_c \bigcup D_p} \ell(h_\theta(x), y)$$

# More about adversarial attack and defense

- Adversarial training vs generalization/overfitting?
  - **Theorem:** Sample complexity of adv. robust generalization can be **significantly larger** than that of “standard” generalization [Schmidt Santurkar Tsipras Talwar M 2018]
  - **Theorem:** No free lunch: can exist a tradeoff between accuracy and robustness [Tsipras Santukar Engstrom Turner Madry 2018]

# References


- [Goodfellow et al. (2014b)] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).
- [Na et al., 2018] Na, Taesik, Jong Hwan Ko, and Saibal Mukhopadhyay. "Cascade adversarial machine learning regularized with a unified embedding." arXiv preprint arXiv:1708.02582 (2017).
- [Goodfellow et al. (2018)] Buckman, Jacob, Aurko Roy, Colin Raffel, and Ian Goodfellow. "Thermometer encoding: One hot way to resist adversarial examples." (2018).

# References


- [Athalye et al. (2018)] Athalye, Anish, Nicholas Carlini, and David Wagner. "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples." *arXiv preprint arXiv:1802.00420* (2018).
- [Dhillon et al., 2018] Dhillon, Guneet S., Kamyar Azizzadenesheli, Zachary C. Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. "Stochastic activation pruning for robust adversarial defense." arXiv preprint arXiv:1803.01442 (2018).
- [adml] https://adversarial-ml-tutorial.org/

# References


- [Mei et al. 2015] Mei, Shike, and Xiaojin Zhu. "Using machine teaching to identify optimal training-set attacks on machine learners." In *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015.
- [Steinhardt et al. 2017] Steinhardt, Jacob, Pang Wei W. Koh, and Percy S. Liang. "Certified defenses for data poisoning attacks." In *Advances in neural information processing systems*, pp. 3517-3529. 2017.
- [Xie et al. 2018] Xie, C., Wang, J., Zhang, Z., Ren, Z., and Yuille, A. Mitigating adversarial effects through randomization. International Conference on Learning Representations, 2018.

# References


- [Schmidt Santurkar Tsipras Talwar M 2018] Schmidt, Ludwig, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. "Adversarially robust generalization requires more data." In *Advances in Neural Information Processing Systems*, pp. 5014-5026. 2018.
- [Tsipras Santukar Engstrom Turner Madry 2018] Tsipras, Dimitris, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. "Robustness may be at odds with accuracy." *arXiv preprint arXiv:1805.12152*(2018).
- [Tjeng et al. 2018] Tjeng, Vincent, Kai Y. Xiao, and Russ Tedrake. "Evaluating robustness of neural networks with mixed integer programming." (2018).