Theoretically Principled Trade-off between Robustness and Accuracy

Hongyang Zhang, TTIC

Yaodong Yu      Jiantao Jiao      Eric P. Xing      Laurent Ghaoui      Michael I. Jordan

Simons Institute for the Theory of Computing
Dec. 17th, 2019
Deep networks are unsafe

“Simons Institute”
87.7% confidence

+ .007 ×

“Simons Institute”
99.3% confidence

= 

TTIC
99.3% confidence

human thinks

machine thinks
Deep networks are unsafe

[BCZOCG’18] Unrestricted Adversarial Example, 2018
Why are there adversarial examples?

- We use a wrong loss function

**Linear Case**

- SVM decision boundary (largest margin)
- Cross-entropy min.
  decision boundary (poor margin)

**Non-Linear Case**
Trade-off between Robustness and Accuracy

\[ R_{\text{rob}}(f) := \mathbb{E}_{(X,Y) \sim D} 1\{\exists X' \in \mathbb{B}(X, \varepsilon) \text{ s.t. } f(X')Y \leq 0\} \]

\[ R_{\text{nat}}(f) := \mathbb{E}_{(X,Y) \sim D} 1\{f(X)Y \leq 0\} \]

- An example of trade-off:

An example of trade-off:

\[ \eta(x) = \text{Pr}(Y = +1 | X = x) \]

\[ X \sim U[0,1] \]

\[ \begin{array}{c|cc}
\hline
 & \text{Bayes Optimal Classifier} & \text{All-One Classifier} \\
\hline
\mathcal{R}_{\text{nat}} & 0 \text{ (optimal)} & 1/2 \\
\mathcal{R}_{\text{rob}} & 1 & 1/2 \text{ (optimal)} \\
\hline
\end{array} \]

Solution: minimize weighted average \( R_{\text{nat}}(f) + R_{\text{rob}}(f)/\lambda \! \)!

\[ \text{Computationally, weighted average } R_{\text{nat}}(f) + R_{\text{rob}}(f)/\lambda \text{ is non-differentiable.} \]
Surrogate Loss

- Classification-calibrated loss $\phi$:

$$H(\eta) := \min_{\alpha \in \mathbb{R}} (\eta \phi(\alpha) + (1 - \eta) \phi(-\alpha))$$

$$H^{-}(\eta) := \min_{\alpha: \alpha(2\eta - 1) \leq 0} (\eta \phi(\alpha) + (1 - \eta) \phi(-\alpha))$$

**Definition (classification-calibrated loss):**

$\phi$ is classification-calibrated loss, if for any $\eta \neq 1/2$, $H^{-}(\eta) > H(\eta)$.

**Intuitive explanation:**

- Think about $\eta$ as $\eta(x) = \Pr[Y = +1|X = x]$, and $\alpha$ as score of positive class by $f$
- Then $H(\eta) = \min_{f} R_{\phi}(f)$

$$H^{-}(\eta) = \min_{f} R_{\phi}(f) \text{ s.t. } f \text{ is inconsistent with Bayes optimal classifier}$$

- Classification-calibrated loss: wrong classifier leads to larger loss for all $\eta(x)$

[BJM’06] Convexity, Classification, and Risk Bounds, 2006
Surrogate Loss

[BJM’06] Convexity, Classification, and Risk Bounds, 2006
Main Results

Theorem 1 (Informal, upper bound, ZYJXGJ’19):

We have \( R_{ro} (f) - R_{nat}^* \leq R_{\phi} (f) - R_{\phi}^* + \sum_{x', \in B(x, \varepsilon)} \phi(f(x')) f(x)/\lambda \). 

Proof Sketch:

• An important decomposition: \( R_{ro} (f) = R_{nat} (f) + R_{bdy} (f) \)
  where \( R_{bdy} (f) = \sum_{x, y} \sum_{x' \in \varepsilon \text{ neighbour of } f} 1 \{ \exists X \in \varepsilon \text{ neighbour of } f \text{ s.t. } f(X)y > 0 \} \)

[ZYJXGJ’19] Theoretically Principled Trade-off between Robustness and Accuracy, ICML 2019
Main Results

Theorem 1 (Informal, upper bound, ZYJXGJ'19):

We have $R_{rob}(f) - R_{nat}^* \leq R_\phi(f) - R_\phi^* + \mathbb{E}_{\max_{X', \in \mathcal{B}(X, \varepsilon)}} \phi(f(X')f(X)/\lambda)$.

Proof Sketch:

- An important decomposition: $R_{rob}(f) = R_{nat}(f) + R_{bdy}(f)$
  where $R_{bdy}(f) = \mathbb{E}_{(X,Y) \sim D} 1\{\exists X \in \varepsilon \text{ neighbour of } f \text{ s.t. } f(X)Y > 0\}$
- $R_{rob}(f) - R_{nat}^* = R_{nat}(f) - R_{nat}^* + R_{bdy}(f)$
- $R_{nat}(f) - R_{nat}^* \leq R_\phi(f) - R_\phi^*$ by [BJM’06]
- $R_{bdy}(f) \leq \mathbb{E}_{\max_{X', \in \mathcal{B}(X, \varepsilon)}} 1(f(X')f(X) < 0) \leq \mathbb{E}_{\max_{X', \in \mathcal{B}(X, \varepsilon)}} \phi(f(X')f(X)/\lambda)$

[BJM'06] Convexity, Classification, and Risk Bounds, 2006
[ZYJXGJ’19] Theoretically Principled Trade-off between Robustness and Accuracy, ICML 2019
Main Results

Theorem 1 (Informal, upper bound, ZYJXGJ’19):
We have $R_{rob}(f) - R^*_{nat} \leq R_{\phi}(f) - R^*_{\phi} + \mathbb{E} \max_{X' \in B(X, \varepsilon)} \phi(f(X'))f(X)/\lambda$.

Theorem 2 (Informal, lower bound, ZYJXGJ’19):
There exist a data distribution, a classifier $f$, and an $\lambda > 0$ such that $R_{rob}(f) - R^*_{nat} \geq R_{\phi}(f) - R^*_{\phi} + \mathbb{E} \max_{X' \in B(X, \varepsilon)} \phi(f(X'))f(X)/\lambda$.

[ZYJXGJ’19] Theoretically Principled Trade-off between Robustness and Accuracy, ICML 2019
Main Results

Theorem 1 (Informal, upper bound, ZYJXGJ’19):
We have $R_{rob}(f) - R_{nat}^* \leq R_{\phi}(f) - R_{\phi}^* + \mathbb{E} \max_{X', \in B(X, \varepsilon)} \phi(f(X')f(X)/\lambda)$.

- New Surrogate Loss (TRADES):
  \[
  \min_f [\mathbb{E} \phi(Yf(X)) + \mathbb{E} \max_{X', \in B(X)} \phi(f(X)f(X')/\lambda)]
  \]
Significant Experimental Results
Experiments --- CIFAR10

<table>
<thead>
<tr>
<th>Defense</th>
<th>Defense type</th>
<th>Under which attack</th>
<th>Dataset</th>
<th>Distance</th>
<th>$A_{nat}(f)$</th>
<th>$A_{rob}(f)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>[BRRG18]</td>
<td>gradient mask</td>
<td>[ACW18]</td>
<td>CIFAR10</td>
<td>0.031 ($\ell_\infty$)</td>
<td>-</td>
<td>0%</td>
</tr>
<tr>
<td>[MLW+18]</td>
<td>gradient mask</td>
<td>[ACW18]</td>
<td>CIFAR10</td>
<td>0.031 ($\ell_\infty$)</td>
<td>-</td>
<td>5%</td>
</tr>
<tr>
<td>[DAL+18]</td>
<td>gradient mask</td>
<td>[ACW18]</td>
<td>CIFAR10</td>
<td>0.031 ($\ell_\infty$)</td>
<td>-</td>
<td>0%</td>
</tr>
<tr>
<td>[SKN+18]</td>
<td>gradient mask</td>
<td>[ACW18]</td>
<td>CIFAR10</td>
<td>0.031 ($\ell_\infty$)</td>
<td>-</td>
<td>9%</td>
</tr>
<tr>
<td>[NKM17]</td>
<td>gradient mask</td>
<td>[ACW18]</td>
<td>CIFAR10</td>
<td>0.015 ($\ell_\infty$)</td>
<td>-</td>
<td>15%</td>
</tr>
<tr>
<td>[WSMK18]</td>
<td>robust opt.</td>
<td>FGSM$_{20}$ (PGD)</td>
<td>CIFAR10</td>
<td>0.031 ($\ell_\infty$)</td>
<td>27.07%</td>
<td>23.54%</td>
</tr>
<tr>
<td>[MMS+18]</td>
<td>robust opt.</td>
<td>FGSM$_{20}$ (PGD)</td>
<td>CIFAR10</td>
<td>0.031 ($\ell_\infty$)</td>
<td>87.30%</td>
<td>47.04%</td>
</tr>
</tbody>
</table>

$$\min_f \max_{X \in B_\epsilon(X)} \phi(Yf(X'))$$ (by Madry et al.)

| TRADES (1/\lambda = 1) | regularization | FGSM$_{20}$ (PGD) | CIFAR10 | 0.031 ($\ell_\infty$) | 88.64% | 49.14% |
| TRADES (1/\lambda = 6) | regularization | FGSM$_{20}$ (PGD) | CIFAR10 | 0.031 ($\ell_\infty$) | 84.92% | 56.61% |

$$\min_f [\mathbb{E} \phi(Yf(X)) + \mathbb{E} \max_{X \in B_\epsilon(X)} \phi(f(X)f(X'))/\lambda]$$ (ours)

| TRADES (1/\lambda = 6) | regularization | LBFGSAAttacK     | CIFAR10 | 0.031 ($\ell_\infty$) | 84.92% | 81.58% |
| TRADES (1/\lambda = 1) | regularization | MI-FGSM          | CIFAR10 | 0.031 ($\ell_\infty$) | 88.64% | 51.26% |
| TRADES (1/\lambda = 6) | regularization | MI-FGSM          | CIFAR10 | 0.031 ($\ell_\infty$) | 84.92% | 57.95% |
| TRADES (1/\lambda = 1) | regularization | C&W              | CIFAR10 | 0.031 ($\ell_\infty$) | 88.64% | 84.03% |
| TRADES (1/\lambda = 6) | regularization | C&W              | CIFAR10 | 0.031 ($\ell_\infty$) | 84.92% | 81.24% |
| [SKC18]   | gradient mask | [ACW18]           | MNIST   | 0.005 ($\ell_2$)     | -      | 55%     |
| [MMS+18]  | robust opt.  | FGSM$_{40}$ (PGD) | MNIST   | 0.3 ($\ell_\infty$)  | 99.36% | 96.01% |
| TRADES (1/\lambda = 6) | regularization | FGSM$_{40}$ (PGD) | MNIST   | 0.3 ($\ell_\infty$)  | 99.48% | 96.07% |
| TRADES (1/\lambda = 6) | regularization | C&W              | MNIST   | 0.005 ($\ell_2$)     | 99.48% | 99.46% |
Interpretability

the class of bicycle

(a) clean example  (b) adversarial example by boundary attack with random spatial transformation

(c) clean example  (d) adversarial example by boundary attack with random spatial transformation

(e) clean example  (f) adversarial example by boundary attack with random spatial transformation

the class of bird

(a) clean example  (b) adversarial example by boundary attack with random spatial transformation

(c) clean example  (d) adversarial example by boundary attack with random spatial transformation

(e) clean example  (f) adversarial example by boundary attack with random spatial transformation
Competition: NeurIPS 2018 Adversarial Vision Challenge

- Evaluation criterion
  - 400+ teams, ~2,000 submissions
  - Tiny ImageNet dataset
  - Model Track and Attack Track
  - Participants in the two tracks play against each other
Competition: NeurIPS 2018 Adversarial Vision Challenge

Final Result

<table>
<thead>
<tr>
<th>Rank</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>2.256</td>
</tr>
<tr>
<td>2nd</td>
<td>2.025</td>
</tr>
<tr>
<td>3rd</td>
<td>1.637</td>
</tr>
<tr>
<td>4th</td>
<td>1.585</td>
</tr>
<tr>
<td>5th</td>
<td>1.476</td>
</tr>
<tr>
<td>6th</td>
<td>1.401</td>
</tr>
</tbody>
</table>
Recent Developments of TRADES

- **Acceleration** [SNG+19, ZZL+19]:
  - Achieve 30x speed-up, almost as fast as natural training

- **Semi-supervised learning/unlabel data** [CRS+19, SFK+19]:
  - TRADES + self-training (500K) improves robustness by +5% on CIFAR10

- **Applications** [JHC+19]:
  - 1st place in Glue leaderboard (up until Dec. 9th) in NLP --- SMART

- **Theoretical understanding** (upcoming):
  - Benefits of local Lipschitzness
  - Provable certification of TRADES by random smoothing

---

Conclusions

- Adversarial Robustness
  - Trade-off matters in the adversarial defense
  - Matching upper and lower bounds on $R_{rob}(f) - R^*_{nat}$
  - New surrogate loss for adversarial defense
  - Winners of NeurIPS 2018 Adversarial Vision Challenge
  - Some recent developments
Thank You
Trade-off between Robustness and Accuracy

- Our goal: Find a classifier \( \hat{f} \) such that \( R_{rob}(\hat{f}) \leq \text{OPT} + \delta \)

\[
\text{OPT: } = \min_f R_{rob}(f), \quad \text{s.t.} \quad R_{nat}(f) \leq R^*_{nat} + \delta
\]

suffice to show \( R_{rob}(f) - R^*_{nat} \leq \delta \)
New Surrogate Loss:

\[
\min_f \left[ \mathbb{E} \phi(Yf(X)) + \mathbb{E} \max_{X' \in B_\varepsilon(X)} \phi(f(X)f(X')/\lambda) \right]
\]

Natural training:

```python
def train(args, model, device, train_loader, optimizer, epoch):
    model.train()
    for batch_idx, (data, target) in enumerate(train_loader):
        data, target = data.to(device), target.to(device)
        optimizer.zero_grad()
        loss = F.cross_entropy(model(data), target)
        loss.backward()
        optimizer.step()
```

Adversarial training by TRADES:

To apply TRADES, cd into the directory, put 'trades.py' to the directory.

```python
from trades import trades_loss

def train(args, model, device, train_loader, optimizer, epoch):
    model.train()
    for batch_idx, (data, target) in enumerate(train_loader):
        data, target = data.to(device), target.to(device)
        optimizer.zero_grad()
        # calculate robust loss - TRADES loss
        loss = trades_loss(model=model,
                           x_natural=data,
                           y=target,
                           optimizer=optimizer,
                           step_size=args.step_size,
                           epsilon=args.epsilon,
                           perturb_steps=args.num_steps,
                           batch_size=args.batch_size,
                           beta=args.beta,
                           distance='l_inf')
        loss.backward()
        optimizer.step()
```

- Link: https://github.com/yaodongyu/TRADES
Competition II: Unrestricted Adversarial Example

Unrestricted Adversarial Examples Challenge

In the Unrestricted Adversarial Examples Challenge, attackers submit arbitrary adversarial inputs, and defenders are expected to assign low confidence to difficult inputs while retaining high confidence and accuracy on a clean, unambiguous test set. You can learn more about the motivation and structure of the contest in our recent paper.

This repository contains code for the warm-up to the challenge, as well as the public proposal for the contest. We are currently accepting defenses for the warm-up.

Warm-up & Contest Timeline
## Competition II: Unrestricted Adversarial Example

<table>
<thead>
<tr>
<th>Defense</th>
<th>Submitted by</th>
<th>Clean data</th>
<th>Common corruptions</th>
<th>Spatial grid attack</th>
<th>SPSA attack</th>
<th>Boundary attack</th>
<th>Submission Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pytorch ResNet50 (trained on bird-or-bicycle extras)</td>
<td>TRADESv2</td>
<td>100.0%</td>
<td>100.0%</td>
<td>99.5%</td>
<td>100.0%</td>
<td>95.0%</td>
<td>Jan 17th, 2019 (EST)</td>
</tr>
<tr>
<td>Keras ResNet (trained on ImageNet)</td>
<td>Google Brain</td>
<td>100.0%</td>
<td>99.2%</td>
<td>92.2%</td>
<td>1.6%</td>
<td>4.0%</td>
<td>Sept 29th, 2018</td>
</tr>
<tr>
<td>Pytorch ResNet (trained on bird-or-bicycle extras)</td>
<td>Google Brain</td>
<td>98.8%</td>
<td>74.6%</td>
<td>49.5%</td>
<td>2.5%</td>
<td>8.0%</td>
<td>Oct 1st, 2018</td>
</tr>
</tbody>
</table>
### Competition II: Unrestricted Adversarial Example

<table>
<thead>
<tr>
<th>Defense</th>
<th>Submitted by</th>
<th>Clean data</th>
<th>Common corruptions</th>
<th>Spatial grid attack</th>
<th>SPSA attack</th>
<th>Boundary attack</th>
<th>Submission Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>PyTorch ResNet50 (trained on bird-or-bicycle extras)</td>
<td>TRADESv2</td>
<td>100.0%</td>
<td>100.0%</td>
<td>99.5%</td>
<td>100.0%</td>
<td>95.0%</td>
<td>Jan 17th, 2019 (EST)</td>
</tr>
<tr>
<td>Keras ResNet (trained on ImageNet)</td>
<td>Google Brain</td>
<td>100.0%</td>
<td>99.2%</td>
<td>92.2%</td>
<td>1.6%</td>
<td>4.0%</td>
<td>Sept 29th, 2018</td>
</tr>
<tr>
<td>PyTorch ResNet (trained on bird-or-bicycle extras)</td>
<td>Google Brain</td>
<td>98.8%</td>
<td>74.6%</td>
<td>49.5%</td>
<td>2.5%</td>
<td>8.0%</td>
<td>Oct 1st, 2018</td>
</tr>
</tbody>
</table>
## Competition II: Unrestricted Adversarial Example

<table>
<thead>
<tr>
<th>Defense</th>
<th>Submitted by</th>
<th>Clean data</th>
<th>Common corruptions</th>
<th>Spatial grid attack</th>
<th>SPSA attack</th>
<th>Boundary attack</th>
<th>Submission Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pytorch ResNet50 (trained on bird-or-bicycle extras)</td>
<td>TRADESv2</td>
<td>100.0%</td>
<td>100.0%</td>
<td>99.5%</td>
<td>100.0%</td>
<td>95.0%</td>
<td>Jan 17th, 2019 (EST)</td>
</tr>
<tr>
<td>Keras ResNet (trained on ImageNet)</td>
<td>Google Brain</td>
<td>100.0%</td>
<td>99.2%</td>
<td>92.2%</td>
<td>1.6%</td>
<td>4.0%</td>
<td>Sept 29th, 2018</td>
</tr>
<tr>
<td>Pytorch ResNet (trained on bird-or-bicycle extras)</td>
<td>Google Brain</td>
<td>98.8%</td>
<td>74.6%</td>
<td>49.5%</td>
<td>2.5%</td>
<td>8.0%</td>
<td>Oct 1st, 2018</td>
</tr>
</tbody>
</table>
# Competition II: Unrestricted Adversarial Example

<table>
<thead>
<tr>
<th>Defense</th>
<th>Submitted by</th>
<th>Clean data</th>
<th>Common corruptions</th>
<th>Spatial grid attack</th>
<th>Boundary attack</th>
<th>Submission Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pytorch ResNet50 (trained on bird-or-bicycle extras)</td>
<td>TRADESv2</td>
<td>100.0%</td>
<td>100.0%</td>
<td>99.5%</td>
<td>100.0%</td>
<td>Jan 17th, 2019 (EST)</td>
</tr>
<tr>
<td>Keras ResNet (trained on ImageNet)</td>
<td>Google Brain</td>
<td>100.0%</td>
<td>99.2%</td>
<td>92.2%</td>
<td>1.6%</td>
<td>Sept 29th, 2018</td>
</tr>
<tr>
<td>Pytorch ResNet (trained on bird-or-bicycle extras)</td>
<td>Google Brain</td>
<td>98.8%</td>
<td>74.6%</td>
<td>49.5%</td>
<td>2.5%</td>
<td>Oct 1st, 2018</td>
</tr>
</tbody>
</table>
## Competition II: Unrestricted Adversarial Example

<table>
<thead>
<tr>
<th>Defense</th>
<th>Submitted by</th>
<th>Clean data</th>
<th>Common corruptions</th>
<th>Spatial grid attack</th>
<th>SPSA attack</th>
<th>Boundary attack</th>
<th>Submission Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pytorch ResNet50 (trained on bird-or-        TRADESv2</td>
<td>100.0%</td>
<td>100.0%</td>
<td>99.5%</td>
<td>100.0%</td>
<td>95.0%</td>
<td>Jan 17th, 2019 (EST)</td>
<td></td>
</tr>
<tr>
<td>bicycle extras)</td>
<td>Google Brain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Keras ResNet (trained on ImageNet)           Google Brain</td>
<td>100.0%</td>
<td>99.2%</td>
<td>92.2%</td>
<td>1.6%</td>
<td>4.0%</td>
<td>Sept 29th, 2018</td>
<td></td>
</tr>
<tr>
<td>Pytorch ResNet (trained on bird-or-          Google Brain</td>
<td>98.8%</td>
<td>74.6%</td>
<td>49.5%</td>
<td>2.5%</td>
<td>8.0%</td>
<td>Oct 1st, 2018</td>
<td></td>
</tr>
<tr>
<td>bicycle extras)</td>
<td>Google Brain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Future Directions about Robustness

- Computational and Statistical Theory
  - Understand the optimization principal of new surrogate loss
  - (Tight) sample complexity of adversarial learning
- Applications of AI Security
  - Robotics, autonomous cars
  - Medical diagnose
- Extensions with other frameworks
  - Self-supervised/semi-supervised learning
  - Neural ODE