Evasion Attacks Against Various Machine Learning Models
Recall: Non-traditional Adversarial Attacks

• Leveraging generative adversarial networks --- diverse, realistic, efficient
• Spatially transformed adversarial examples/Wasserstein distance based adv --- diverse, realistic
• Effective physical world attack --- spatial constrained, robust under physical conditions
Adversarial examples for semantic segmentation and object detection

• Generating adv. is a critical step for evaluating and improving robustness of learning models.
• So far we introduced adv. against classifiers
• What about other learning tasks?
Adversarial examples for semantic segmentation and object detection

• Both segmentation and detection are based on classifying multiple targets on an image

• Dense adversary generation (DAG)
Adversarial examples for semantic segmentation and object detection

Problem statement

Untargeted attack
\[ \forall n, \arg\max_c f_c(X + r, t_n) \neq l_n \]

Targeted attack
\[ L(X, T, L, L') = \sum_{n=1}^{N} [f_{l_n}(X, t_n) - f'_{l_n}(X, t_n)] \]

Perturbation  targets  Ground truth

Algorithm 1: Dense Adversary Generation (DAG)

Input: input image \( X \);
the classifier \( f(\cdot, \cdot) \in \mathbb{R}^C \);
the target set \( T = \{t_1, t_2, \ldots, t_N\} \);
the original label set \( L = \{l_1, l_2, \ldots, l_N\} \);
the adversarial label set \( L' = \{l'_1, l'_2, \ldots, l'_N\} \);
the maximal iterations \( M_0 \);

Output: the adversarial perturbation \( r \);

1. \( X_0 \leftarrow X, r \leftarrow 0, m \leftarrow 0, T_0 \leftarrow T; \)
2. while \( m < M_0 \) and \( T_m \neq \emptyset \) do
3. \hspace{1em} \( T_m = \{t_n | \arg\max_c \{f_c(X_m, t_n)\} = l_n\}; \)
4. \hspace{1em} \( r_m \leftarrow \frac{\sum_{t_n \in T_m} [\nabla x_m f_{l_n}(X_m, t_n) - \nabla x_m f'_{l_n}(X_m, t_n)]}{\|r_m\|_\infty}; \)
5. \hspace{1em} \( r' \leftarrow r + r_m; \)
6. \hspace{1em} \( X_{m+1} \leftarrow X_m + r'_m; \)
7. \hspace{1em} \( m \leftarrow m + 1; \)
8. end
9. Return: \( r \)
Transferability analysis

• Cross training transfer
  • Models are trained with different subset of data

• Cross network transfer
  • Models are of different architecture

• Cross task transfer
  • Use the perturbation generated against detection to attack a segmentation network
## Cross training

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>58.70</td>
<td>61.07</td>
<td>69.14</td>
<td>72.07</td>
<td>76.40</td>
<td>78.06</td>
</tr>
<tr>
<td>FR-ZF-07 ($r_1$)</td>
<td>3.61</td>
<td>22.15</td>
<td>66.01</td>
<td>69.47</td>
<td>74.01</td>
<td>75.87</td>
</tr>
<tr>
<td>FR-ZF-0712 ($r_2$)</td>
<td>13.14</td>
<td>1.95</td>
<td>64.61</td>
<td>68.17</td>
<td>72.29</td>
<td>74.68</td>
</tr>
<tr>
<td>FR-VGG-07 ($r_3$)</td>
<td>56.41</td>
<td>59.31</td>
<td>5.92</td>
<td>48.05</td>
<td>72.84</td>
<td>74.79</td>
</tr>
<tr>
<td>FR-VGG-0712 ($r_4$)</td>
<td>56.09</td>
<td>58.58</td>
<td>31.84</td>
<td><strong>3.36</strong></td>
<td>70.55</td>
<td>72.78</td>
</tr>
<tr>
<td>$r_1 + r_3$</td>
<td><strong>3.98</strong></td>
<td>21.63</td>
<td><strong>7.00</strong></td>
<td>44.14</td>
<td>68.89</td>
<td>71.56</td>
</tr>
<tr>
<td>$r_1 + r_3$ (permute)</td>
<td>58.30</td>
<td>61.08</td>
<td>68.63</td>
<td><strong>71.82</strong></td>
<td>76.34</td>
<td>77.71</td>
</tr>
<tr>
<td>$r_2 + r_4$</td>
<td>13.15</td>
<td><strong>2.13</strong></td>
<td>28.92</td>
<td><strong>4.28</strong></td>
<td>63.93</td>
<td>67.25</td>
</tr>
<tr>
<td>$r_2 + r_4$ (permute)</td>
<td>58.51</td>
<td>61.09</td>
<td>68.68</td>
<td>71.78</td>
<td>76.23</td>
<td>77.71</td>
</tr>
</tbody>
</table>

## Cross Network

<table>
<thead>
<tr>
<th>Adversarial Perturbations from</th>
<th>FCN-Alex</th>
<th>FCN-Alex*</th>
<th>FCN-VGG</th>
<th>FCN-VGG*</th>
<th>DL-VGG</th>
<th>DL-RN101</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>48.04</td>
<td>48.92</td>
<td>65.49</td>
<td>67.09</td>
<td>70.72</td>
<td>76.11</td>
</tr>
<tr>
<td>FCN-Alex ($r_5$)</td>
<td>3.98</td>
<td>7.94</td>
<td>64.82</td>
<td>66.54</td>
<td>70.18</td>
<td>75.45</td>
</tr>
<tr>
<td>FCN-Alex* ($r_6$)</td>
<td>5.10</td>
<td><strong>3.98</strong></td>
<td>64.60</td>
<td>66.36</td>
<td>69.98</td>
<td>75.52</td>
</tr>
<tr>
<td>FCN-VGG ($r_7$)</td>
<td>46.21</td>
<td>47.38</td>
<td><strong>4.09</strong></td>
<td>16.36</td>
<td>45.16</td>
<td>73.98</td>
</tr>
<tr>
<td>FCN-VGG* ($r_8$)</td>
<td>46.10</td>
<td>47.21</td>
<td>12.72</td>
<td><strong>4.18</strong></td>
<td>46.33</td>
<td>73.76</td>
</tr>
<tr>
<td>$r_5 + r_7$</td>
<td><strong>4.83</strong></td>
<td>8.55</td>
<td><strong>4.23</strong></td>
<td>17.59</td>
<td>43.95</td>
<td>73.26</td>
</tr>
<tr>
<td>$r_5 + r_7$ (permute)</td>
<td>48.03</td>
<td>48.90</td>
<td>65.47</td>
<td>67.09</td>
<td>70.69</td>
<td>76.04</td>
</tr>
<tr>
<td>$r_6 + r_8$</td>
<td>5.52</td>
<td><strong>4.23</strong></td>
<td>13.89</td>
<td><strong>4.98</strong></td>
<td>44.18</td>
<td>73.01</td>
</tr>
<tr>
<td>$r_6 + r_8$ (permute)</td>
<td>48.03</td>
<td>48.90</td>
<td>65.47</td>
<td>67.05</td>
<td>70.69</td>
<td>76.05</td>
</tr>
</tbody>
</table>
Takeaways

• Heuristically generate perturbation to move each target towards the adversarial goal
• Transferability exists for adversarial examples for segmentation/detection
• Adding multiple adversarial perturbations often works better than adding a single source of perturbation in terms of transferability
Similar work

• Delving into transferable adversarial examples and black-box attacks
  • Apply ensemble attack to attack multiple models to increase targeted transferability
  • Multi-source perturbation helps?
Ground truth: running shoe

<table>
<thead>
<tr>
<th>Model</th>
<th>Ground Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG16</td>
<td>Military uniform</td>
</tr>
<tr>
<td>ResNet50</td>
<td>Jigsaw puzzle</td>
</tr>
<tr>
<td>ResNet101</td>
<td>Motor scooter</td>
</tr>
<tr>
<td>ResNet152</td>
<td>Mask</td>
</tr>
<tr>
<td>GoogLeNet</td>
<td>Chainsaw</td>
</tr>
</tbody>
</table>
Targeted Adversarial Example’s Transferability Among **Two Models** is **Poor!**

<table>
<thead>
<tr>
<th></th>
<th>ResNet152</th>
<th>ResNet101</th>
<th>ResNet50</th>
<th>VGG16</th>
<th>GoogLeNet</th>
<th>Incept-v3</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet152</td>
<td>100%</td>
<td>2%</td>
<td>1%</td>
<td>1%</td>
<td>1%</td>
<td>0%</td>
</tr>
<tr>
<td>ResNet101</td>
<td>3%</td>
<td>100%</td>
<td>3%</td>
<td>2%</td>
<td>1%</td>
<td>1%</td>
</tr>
<tr>
<td>ResNet50</td>
<td>4%</td>
<td>2%</td>
<td>100%</td>
<td>1%</td>
<td>1%</td>
<td>0%</td>
</tr>
<tr>
<td>VGG16</td>
<td>2%</td>
<td>1%</td>
<td>2%</td>
<td>100%</td>
<td>1%</td>
<td>0%</td>
</tr>
<tr>
<td>GoogLeNet</td>
<td>1%</td>
<td>1%</td>
<td>0%</td>
<td>1%</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>Incept-v3</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>100%</td>
</tr>
</tbody>
</table>

*Only 2% of the adversarial images generated for VGG16 (row) can be predicted as the targeted label by ResNet50 (column)*
Black-box Attacks Based On Transferability

Adversary

White-Box Model

Adversarial Examples

Transfer to

Black-Box System
Ensemble Targeted Black-box Attacks Based On Transferability
Clarifai.com

Ground truth from ImageNet: broom

jacamar
Adversarial Example on Clarifai.com

- Ground truth: **broom**
- Target label: **jacamar**
Similar work

• Physical Adversarial Examples for Object Detectors

\[ J_d(x,y) = \max_{s \in S^2, b \in B} P(s, b, y, f_\theta(x)) \]

Difference: instead of ensemble over models, here it ensembles over object regions
Houdini: Fooling Deep Structured Prediction Models

• Other deterministic objective function for attacking different learning models?

• Houdini: tailored for the final performance measure
  • Speech recognition
  • Pose estimation
  • Semantic segmentation
Houdini: Fooling Deep Structured Prediction Models

• Optimization based method

\[ \hat{x} = \arg\max_{\tilde{x} : \|\tilde{x} - x\|_p \leq \epsilon} \ell(y_\theta(\tilde{x}), y) \]
\[ f_2(x') = (\max_{i \neq t} (F(x')_i) - F(x')_t)^+ \]

• Houdini

\[ \bar{\ell}_H(\theta, x, y) = \mathbb{P}_{\gamma \sim \mathcal{N}(0,1)} \left[ g_\theta(x, y) - g_\theta(x, \hat{y}) < \gamma \right] \cdot \ell(\hat{y}, y) \]

Stochastic margin
Confidence of the model
Task loss
<table>
<thead>
<tr>
<th>$\epsilon$</th>
<th>WER</th>
<th>CER</th>
<th>WER</th>
<th>CER</th>
<th>WER</th>
<th>CER</th>
<th>WER</th>
<th>CER</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon = 0.3$</td>
<td>68</td>
<td>9.3</td>
<td>51</td>
<td>6.9</td>
<td>29.8</td>
<td>4</td>
<td>20</td>
<td>2.5</td>
</tr>
<tr>
<td>$\epsilon = 0.2$</td>
<td>96.1</td>
<td>12</td>
<td>85.4</td>
<td>9.2</td>
<td>66.5</td>
<td>6.5</td>
<td>46.5</td>
<td>4.5</td>
</tr>
<tr>
<td>$\epsilon = 0.1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\epsilon = 0.05$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Groundtruth Transcription:
The fact that a man can recite a poem does not show he remembers any previous occasion on which he has recited it or read it.

G-Voice transcription of the original example:
The fact that a man can decide a poem does not show he remembers any previous occasion on which he has *work cited* or read it.

G-Voice transcription of the adversarial example:
The fact that I can rest I'm just not sure that you heard there is any previous occasion I am at he has your side it or read it.

Groundtruth Transcription:
Her bearing was graceful and animated she led her son by the hand and before her walked two maids with wax lights and silver candlesticks.

G-Voice transcription of the original example:
The bearing was graceful an animated she let her son by the hand and before he walks two maids with wax lights and silver candlesticks.

G-Voice transcription of the adversarial example:
Mary was grateful then admitted she let her son before the walks to Mays would like slice furnace filter count six.
Takeaways

• By adding margin based constraint together with the task loss, the attack can be generated against a range of tasks with high confidence.

• Targeted attacks seem to be more challenging when dealing with speech recognition systems than when we consider artificial visual systems such as pose estimators or semantic segmentation systems.

• Adversarial audios also transfer among models.
Adversarial Examples for Generative Models

• Idea: Create adversarial inputs that can control the latent space of a generative model.

• Generate based on adversarial target
Adversarial Examples for Generative Models

• **Generative Models.**
  • An **encoder** maps a high-dimensional input into lower-dimensional latent representation.
  • A **decoder** maps the latent representation back to a high-dimensional reconstruction.
  • A **latent space** is an internal representation of the data.
Adversarial Examples for Generative Models

- An example attack scenario:
  - Generative model used as a compression scheme

- Attacker’s goal: for the decompressor to reconstruct a different image from the one that the compressor sees.
Adversarial Examples for Generative Models
Adversarial Examples for Generative Models
Adversarial Examples for Generative Models

Adversarial Input

Latent Space

Generated Output

Encoder $f_{enc}$

Decoder $f_{dec}$

Discriminator $f_{disc}$

Classifier $f_{class}$
Adversarial Examples for Generative Models

Target VAE-GAN (or other latent generative model)
Adversarial Examples for Generative Models

Latent Space

Target VAE - GAN (or other latent generative model)

Optional attacker-trained classifier to leverage attacks like FGS

\[ \arg \min_{x^*} L(x, x^*) \quad s.t. \text{Oracle}(G_{\text{targ}}(x^*)) = y^t \]
Adversarial Examples for Generative Models

Original Inputs

Reconstructions
Adversarial Examples for Generative Models
Attacking Deep Reinforcement Learning
Attacking Deep Reinforcement Learning
Adversarial Attacks on Neural Network Policies
A3C: A Deep Policy on Pong

Reinforcement learning algorithms:

- Actor – policy network to predict the action based on each frame
- Critics – value function to predict the value of each frame, and the action is chosen to maximize the expected value
- Actor-critics (A3C) – combine value function into the policy network to make prediction
Agent in Action: attack the policy network

Original Frames

Adversarial perturbation injected into every frame
Attacking Deep Reinforcement Learning
Attacking Deep Reinforcement Learning
Attacks on dynamic environments