Skip Connections Matter: On the transferability of adversarial examples generated with ResNets

Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma

Presented by:
Miles Crowe
Mauricio De Abreu
Outline:

• Introduction
• Background and Related Works
• Skip Gradient Method – SGM
• SGM Case Study
  • Experiment
  • Comparisons/Results
• Conclusion
• Discussion
Introduction

Why this matters...
Introduction:

• State of the art DNNs employ Skip Connections
• ResNets and DenseNets
• Backpropagation occurs both through residual block and skip connection
• Set preference to the skip connection gradients over residual modules gradients

Introduction:

- Trade-off between white box vs black box
  - White box: fools better / transfers poorly
  - Black box: inferior fooling / transfers better
- Previous work has treated entire network as single component
- Paper investigates elements of the network architecture, in particular skip connections
The question of whether or not the DNN architecture itself can expose more security weaknesses to adversarial attacks is an unexplored problem.”
- Wu et al.
Introduction:

- Paper highlights 3 contributions:
  1. Skip connections allow easy production of transferable AE’s
  2. SGM Method to craft AE’s
  3. Comprehensive Experimentation showing SGM improves transferability
Related Works

How will this be compared?
Related Works:

- White Box Attacks:
  - FGSM - Fast Gradient Sign Method
  - PGD – Projected Gradient Descent

\[ x_{adv} = x + \epsilon \cdot \text{sign}(\nabla_x \ell(f(x), y)) \]

\[ x_{adv}^{t+1} = \Pi_\epsilon (x_{adv}^t + \alpha \cdot \text{sign}(\nabla_x \ell(f(x_{adv}^t), y))) \]
Related Works:

- Black Box Attacks:
  - MI – Momentum Iterative boosting
  - DI – Diverse Input
Related Works:

- TI – Translation Invariant
  - W matrix \((2k+1) \times (2k+1)\)
  - Uses a set of translated images to compute gradients
  - Will be presented on 3/3

\[
\mathbf{x}_{adv}^{t+1} = \Pi_{\epsilon}(\mathbf{x}_{adv}^t + \alpha \cdot \text{sign}(W \ast \nabla \ell(f(\mathbf{x}_{adv}^t), y)))
\]
Skip Gradient Method

What is SGM?
SGM:

• Recall that Skip Connections are generally a repeated pattern in DNN architecture

• Assume 3 blocks, with input $z_0$ to output $z_3$

$$z_3 = z_2 + f_3(z_2) = [z_1 + f_2(z_1)] + f_3(z_1 + f_2(z_1))$$
$$= [z_0 + f_1(z_0) + f_2(z_0 + f_1(z_0))] + f_3((z_0 + f_1(z_0)) + f_2(z_0 + f_1(z_0)))$$

• Apply chain rule

$$\frac{\partial \ell}{\partial z_0} = \frac{\partial \ell}{\partial z_3} \frac{\partial z_3}{\partial z_2} \frac{\partial z_2}{\partial z_1} \frac{\partial z_1}{\partial z_0} = \frac{\partial \ell}{\partial z_3} (1 + \frac{\partial f_3}{\partial z_2})(1 + \frac{\partial f_2}{\partial z_1})(1 + \frac{\partial f_1}{\partial z_0})$$

Image Credit: Wu et. al.
SGM:

- Extending to L residual blocks

\[
\frac{\partial \ell}{\partial x} = \frac{\partial \ell}{\partial z_L} \prod_{i=l}^{L-1} \left( \frac{\partial f_{i+1}}{\partial z_i} + 1 \right) \frac{\partial z_L}{\partial x}
\]

- Introducing the decay factor

\[
\nabla_x \ell = \frac{\partial \ell}{\partial z_L} \prod_{i=0}^{L-1} \left( \gamma \frac{\partial f_{i+1}}{\partial z_i} + 1 \right) \frac{\partial z_0}{\partial x}
\]
SGM:

- Skip Gradient Method

\[ \mathbf{x}_{adv}^{t+1} = \Pi_\epsilon \left( \mathbf{x}_{adv}^t + \alpha \cdot \text{sign} \left( \frac{\partial \ell}{\partial z_L} \prod_{i=0}^{L-1} \left( \gamma \frac{\partial f_{i+1}}{\partial z_i} + 1 \right) \frac{\partial z_0}{\partial \mathbf{x}} \right) \right) \]
SGM:

- Method Summary
  - Multiple Step, untargeted, black-box attack
  - Exploits Skip Connections
  - Does not require any computational overhead
SGM Case Study / Results

Is it effective?
Experimental Setup:

• 8 Source Models
  • ResNet with regular residual blocks (RN18/34)
  • ResNet with bottleneck residual blocks (RN50/101/152)
  • DenseNet DN121, DN169, DN201

• Target Models
  • Unsecured (VGG19, RN152, DN201, SE154, IncV3, IncV4, IncResV2)
  • Secured (IncV3_{ens3}, IncV3_{ens4}, IncResV2_{ens3})
Experimental Setup:

- **ImageNet**
  - 5000 random images successfully classified for all source models

- **Attacks Black Box**
  - $L_{\infty}$ perturbation $\varepsilon=16$
  - Decay factor $\gamma=0.5$
  - 10 step
  - 5 runs
  - untargeted
Results:

- Comparing with PGD

<table>
<thead>
<tr>
<th></th>
<th>RN18</th>
<th>RN34</th>
<th>RN50</th>
<th>RN101</th>
<th>RN152</th>
<th>DN121</th>
<th>DN169</th>
<th>DN201</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGD</td>
<td>23.23±0.69</td>
<td>24.38±0.41</td>
<td>22.80±0.55</td>
<td>22.98±0.83</td>
<td>26.56±0.75</td>
<td>30.71±0.60</td>
<td>30.90±0.31</td>
<td>36.01±0.59</td>
</tr>
<tr>
<td>SGM</td>
<td>28.92±0.45</td>
<td>43.43±0.32</td>
<td>36.71±0.55</td>
<td>38.38±0.53</td>
<td>44.84±0.14</td>
<td>57.38±0.14</td>
<td>60.45±0.42</td>
<td>65.48±0.23</td>
</tr>
</tbody>
</table>
Results:

• Baselines
  • Now compared to FGSM, PGD, MI, DI, and TI
  • 10 step unsecured / 20 secured
  • Iterative methods $\alpha = 2$
  • $\gamma = 0.2 / 0.5$ ResNet/DenseNet PGD … +0.3/0.2 for FGSM respectively
  • SGM = FGSM+SGM one step or PGD+SGM for iterative
  • VGG, ResNet and DenseNet images cropped to 224x224
    • 299x299 otherwise
Results:

• Threat Model
  • AE’s generated on source
  • Used to attack target
  • Same test hyper params as PGD
Results:

- One step transferability against unsecured models

<table>
<thead>
<tr>
<th>Source</th>
<th>Attack</th>
<th>VGG19</th>
<th>RN152</th>
<th>DN201</th>
<th>SE154</th>
<th>IncV3</th>
<th>IncV4</th>
<th>IncResV2</th>
</tr>
</thead>
<tbody>
<tr>
<td>RN152</td>
<td>FGSM</td>
<td>41.96±0.52</td>
<td>71.53±0.34</td>
<td>37.49±0.10</td>
<td>30.00±0.56</td>
<td>25.66±0.07</td>
<td>21.55±0.16</td>
<td>19.90±0.49</td>
</tr>
<tr>
<td></td>
<td>TI</td>
<td>49.61±0.11</td>
<td>49.33±0.35</td>
<td>36.87±0.42</td>
<td>29.95±0.32</td>
<td>33.59±0.73</td>
<td>29.05±0.34</td>
<td>20.62±0.09</td>
</tr>
<tr>
<td></td>
<td>SGM</td>
<td>47.54±0.14</td>
<td>76.90±0.60</td>
<td>43.73±0.21</td>
<td>31.16±0.45</td>
<td>29.41±0.24</td>
<td>25.11±0.20</td>
<td>22.63±0.15</td>
</tr>
<tr>
<td>DN201</td>
<td>FGSM</td>
<td>49.87±0.17</td>
<td>38.89±0.29</td>
<td>81.51±0.33</td>
<td>34.94±0.53</td>
<td>31.21±0.47</td>
<td>27.08±0.23</td>
<td>23.87±0.45</td>
</tr>
<tr>
<td></td>
<td>TI</td>
<td>54.37±0.58</td>
<td>33.49±0.18</td>
<td>57.71±0.05</td>
<td>34.46±0.47</td>
<td>34.45±0.25</td>
<td>30.17±0.23</td>
<td>20.36±0.33</td>
</tr>
<tr>
<td></td>
<td>SGM</td>
<td>56.97±0.25</td>
<td>47.54±0.14</td>
<td>87.73±0.76</td>
<td>42.31±0.67</td>
<td>37.91±0.56</td>
<td>32.83±0.38</td>
<td>29.64±0.25</td>
</tr>
</tbody>
</table>

- Does not transfer as well as TI to a different base architecture if not enough skip connections in source model
- More effective for more skip connections
Results:

- Multi-step transferability
- Virtually tied on RN152
- Results are nearly insignificant on DN201
- Modest improvement against Inception based networks
- SGM outperforms all other methods given sufficient skip connections in source model
Results:

- Combined Attacks
- SGM shows a solid boost to existing methods
Results:

- Transferability against Robustly Trained Models
- SGM does not transfer well to ensembles alone

<table>
<thead>
<tr>
<th>Source</th>
<th>Attack</th>
<th>IncV3_{ens3}</th>
<th>IncV_{ens4}</th>
<th>IncRes_{ens3}</th>
</tr>
</thead>
<tbody>
<tr>
<td>RN152</td>
<td>PGD</td>
<td>12.47±1.27</td>
<td>10.72±1.37</td>
<td>6.97±0.71</td>
</tr>
<tr>
<td></td>
<td>TI</td>
<td>45.36±0.97</td>
<td>45.81±0.93</td>
<td>38.19±0.81</td>
</tr>
<tr>
<td></td>
<td>MI</td>
<td>24.20±1.15</td>
<td>22.04±0.98</td>
<td>16.10±0.56</td>
</tr>
<tr>
<td></td>
<td>DI</td>
<td>28.48±1.21</td>
<td>24.19±1.22</td>
<td>17.31±0.77</td>
</tr>
<tr>
<td></td>
<td>SGM</td>
<td>31.57±0.55</td>
<td>27.77±0.47</td>
<td>20.02±0.66</td>
</tr>
<tr>
<td></td>
<td>TI+SGM</td>
<td>52.62±0.40</td>
<td>52.80±0.79</td>
<td>43.96±0.62</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>Attack</th>
<th>IncV3_{ens3}</th>
<th>IncV_{ens4}</th>
<th>IncRes_{ens3}</th>
</tr>
</thead>
<tbody>
<tr>
<td>DN201</td>
<td>PGD</td>
<td>18.16±0.56</td>
<td>15.30±0.62</td>
<td>10.40±0.49</td>
</tr>
<tr>
<td></td>
<td>TI</td>
<td>42.76±0.91</td>
<td>42.01±0.79</td>
<td>34.28±0.88</td>
</tr>
<tr>
<td></td>
<td>MI</td>
<td>31.79±0.83</td>
<td>28.21±0.15</td>
<td>20.60±0.38</td>
</tr>
<tr>
<td></td>
<td>DI</td>
<td>34.84±1.35</td>
<td>29.23±0.83</td>
<td>21.64±0.80</td>
</tr>
<tr>
<td></td>
<td>SGM</td>
<td>41.45±0.30</td>
<td>37.85±0.22</td>
<td>29.41±0.02</td>
</tr>
<tr>
<td></td>
<td>TI+SGM</td>
<td>46.11±1.23</td>
<td>47.38±0.89</td>
<td>39.32±0.80</td>
</tr>
</tbody>
</table>
Results - closer look:

- Effects of variation in $\gamma$
- High decay in residuals leads to poor performance in DenseNets
  - Peaks at $\gamma=0.5$
Results - closer look:

• AEs crafted on Ensembles against unsecured/secured models

<table>
<thead>
<tr>
<th>Source</th>
<th>Attack</th>
<th>VGG19</th>
<th>RN152</th>
<th>DN201</th>
<th>SE154</th>
<th>IncV3</th>
<th>IncV4</th>
<th>IncRes</th>
</tr>
</thead>
<tbody>
<tr>
<td>RN34 + RN152 + DN201</td>
<td>PGD</td>
<td>86.69±0.20</td>
<td>99.99±0.01</td>
<td>99.99±0.02</td>
<td>69.65±0.71</td>
<td>65.95±0.35</td>
<td>59.30±0.32</td>
<td>53.91±0.40</td>
</tr>
<tr>
<td></td>
<td>TI</td>
<td>84.35±0.21</td>
<td>99.59±0.11</td>
<td>99.77±0.06</td>
<td>71.67±0.19</td>
<td>67.22±0.82</td>
<td>66.02±0.66</td>
<td>56.83±0.89</td>
</tr>
<tr>
<td></td>
<td>MI</td>
<td>92.86±0.19</td>
<td>99.91±0.04</td>
<td>99.91±0.06</td>
<td>86.11±0.38</td>
<td>83.25±0.35</td>
<td>79.25±1.16</td>
<td>76.53±0.66</td>
</tr>
<tr>
<td></td>
<td>DI</td>
<td>96.34±0.23</td>
<td>99.84±0.20</td>
<td>99.84±0.20</td>
<td>89.72±0.52</td>
<td>87.53±0.29</td>
<td>85.04±0.75</td>
<td>81.11±0.44</td>
</tr>
<tr>
<td></td>
<td>SGM</td>
<td>97.36±0.17</td>
<td>99.87±0.07</td>
<td>99.86±0.09</td>
<td>90.40±0.26</td>
<td>87.86±0.51</td>
<td>82.97±0.71</td>
<td>80.93±0.55</td>
</tr>
<tr>
<td>DI+SGM</td>
<td></td>
<td>98.65±0.08</td>
<td>99.84±0.04</td>
<td>99.86±0.04</td>
<td>94.36±0.19</td>
<td>93.08±0.41</td>
<td>89.56±0.07</td>
<td>88.27±0.43</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>Attack</th>
<th>IncV3_{ens3}</th>
<th>IncV_{ens4}</th>
<th>IncRes_{ens3}</th>
</tr>
</thead>
<tbody>
<tr>
<td>RN34 + RN152 + DN201</td>
<td>PGD</td>
<td>37.63±0.37</td>
<td>32.69±0.62</td>
<td>23.49±0.55</td>
</tr>
<tr>
<td></td>
<td>TI</td>
<td>75.04±0.50</td>
<td>75.94±0.63</td>
<td>66.24±0.45</td>
</tr>
<tr>
<td></td>
<td>MI</td>
<td>54.68±0.27</td>
<td>50.24±0.48</td>
<td>39.27±0.33</td>
</tr>
<tr>
<td></td>
<td>DI</td>
<td>65.29±0.31</td>
<td>57.48±0.45</td>
<td>46.41±0.42</td>
</tr>
<tr>
<td></td>
<td>SGM</td>
<td>66.08±0.42</td>
<td>62.22±0.73</td>
<td>51.16±0.12</td>
</tr>
<tr>
<td>TI+SGM</td>
<td></td>
<td>87.65±1.00</td>
<td>85.11±0.27</td>
<td>77.75±0.41</td>
</tr>
</tbody>
</table>
Results:

• Improving for Weak White-box attacks
Conclusion

Then a little For/Against
Conclusion:

- Skip connections are very useful in crafting AE’s
- Extensive results show SGM to improve attack strength, at least marginally
- Reminder to be mindful of new vulnerabilities with new architectures
For

- SGM is a very simple concept.
- The SGM method is a very efficient and flexible boost to existing gradient attacks.
- The authors tested extensively with varying models and methods.
- For the stated purpose, networks with large number of skip connections validate the claim well
- Shows improvement in transfer attacks.
• Fooling rate boost only scales with network depth.
• DenseNet provides more skip connections, yet AE's crafted with ResNet are more effective against secured models.
• SGM alone is not sufficient to attack secured models
• Not as strong for white box setup
• Small discrepancies noted between results commented in the text with values shown in table
• Only tested in ImageNet
Thank You
Questions?
All image and material contribution credit: