Adversarial Examples and Adversarial Training

Ian Goodfellow, Staff Research Scientist, Google Brain
CS 231n, Stanford University, 2017-05-30
Overview

• What are adversarial examples?

• Why do they happen?

• How can they be used to compromise machine learning systems?

• What are the defenses?

• How to use adversarial examples to improve machine learning, even when there is no adversary

(Goodfellow 2016)
Since 2013, deep neural networks have matched human performance at...

...recognizing objects and faces....

(Szegedy et al, 2014)

(Taigmen et al, 2013)

...solving CAPTCHAs and reading addresses...

(Goodfellow et al, 2013)

(Goodfellow et al, 2013)

and other tasks...
Adversarial Examples

Timeline:
“Adversarial Classification” Dalvi et al 2004: fool spam filter
“Evasion Attacks Against Machine Learning at Test Time”
Biggio 2013: fool neural nets
Szegedy et al 2013: fool ImageNet classifiers imperceptibly
Goodfellow et al 2014: cheap, closed form attack
Turning Objects into “Airplanes”
Attacking a Linear Model
Not just for neural nets

• Linear models
  • Logistic regression
  • Softmax regression
  • SVMs
• Decision trees
• Nearest neighbors
Adversarial Examples from Overfitting

(Goodfellow 2016)
Adversarial Examples from Excessive Linearity
Modern deep nets are very piecewise linear

Rectified linear unit

Maxout

Carefully tuned sigmoid

LSTM

(Goodfellow 2016)
Nearly Linear Responses in Practice

[Graph showing nearly linear responses with argument to softmax on the y-axis and \( \epsilon \) on the x-axis. The graph includes labels for various categories like airplane, automobile, bird, cat, etc., each represented by a line with a legend box on the right.]
Small inter-class distances

All three perturbations have L2 norm 3.96
This is actually small. We typically use 7!

Perturbation changes the true class
Random perturbation does not change the class
Perturbation changes the input to “rubbish class”
The Fast Gradient Sign Method

\[ J(\tilde{x}, \theta) \approx J(x, \theta) + (\tilde{x} - x)^\top \nabla_x J(x). \]

Maximize

\[ J(x, \theta) + (\tilde{x} - x)^\top \nabla_x J(x) \]

subject to

\[ ||\tilde{x} - x||_\infty \leq \epsilon \]

\[ \Rightarrow \tilde{x} = x + \epsilon \text{sign} (\nabla_x J(x)). \]
Maps of Adversarial and Random Cross-Sections

(collaboration with David Warde-Farley and Nicolas Papernot)
Maps of Adversarial Cross-Sections
Maps of Random Cross-Sections

Adversarial examples are not noise

(collaboration with David Warde-Farley and Nicolas Papernot)
Estimating the Subspace Dimensionality

(Tramèr et al, 2017)
Clever Hans

(“Clever Hans, Clever Algorithms,” Bob Sturm)
Wrong almost everywhere
Adversarial Examples for RL

(Huang et al., 2017)
High-Dimensional Linear Models

Weights

Signs of weights

Clean examples

Adversarial

(Goodfellow 2016)
Linear Models of ImageNet

(Andrej Karpathy, “Breaking Linear Classifiers on ImageNet”)

(Goodfellow 2016)
RBFs behave more intuitively
Cross-model, cross-dataset generalization
Cross-technique transferability

(Papernot 2016)
Transferability Attack

Target model with unknown weights, machine learning algorithm, training set; maybe non-differentiable

Train your own model

Substitute model mimicking target model with known, differentiable function

Deploy adversarial examples against the target; transferability property results in them succeeding

Adversarial examples

Adversarial crafting against substitute

(Goodfellow 2016)
Cross-Training Data Transferability

(Papernot 2016)
Enhancing Transfer With Ensembles

Table 4: Accuracy of non-targeted adversarial images generated using the optimization-based approach. The first column indicates the average RMSD of the generated adversarial images. Cell \((i, j)\) corresponds to the accuracy of the attack generated using four models except model \(i\) (row) when evaluated over model \(j\) (column). In each row, the minus sign “−” indicates that the model of the row is not used when generating the attacks. Results of top-5 accuracy can be found in the appendix (Table 14).

(Liu et al, 2016)
Adversarial Examples in the Human Brain

These are concentric circles, not intertwined spirals.

(Pinna and Gregory, 2002)
Practical Attacks

- Fool real classifiers trained by remotely hosted API (MetaMind, Amazon, Google)
- Fool malware detector networks
- Display adversarial examples in the physical world and fool machine learning systems that perceive them through a camera
Adversarial Examples in the Physical World

(Kurakin et al, 2016)
Failed defenses

Generative pretraining

Adding noise at test time

Confidence-reducing perturbation at test time

Weight decay

Various non-linear units

Removing perturbation with an autoencoder

Ensembles

Multiple glimpses

Double backprop

Adding noise at train time

Dropout

Error correcting codes

(Goodfellow 2016)
Generative Modeling is not Sufficient to Solve the Problem.
Universal approximator theorem

Neural nets can represent either function:

Maximum likelihood doesn’t cause them to learn the right function. But we can fix that...
Training on Adversarial Examples

Test misclassification rate vs. Training time (epochs)

- Blue line: Train=Clean, Test=Clean
- Green line: Train=Clean, Test=Adv
- Red line: Train=Adv, Test=Clean
- Cyan line: Train=Adv, Test=Adv

Logarithmic scale on the y-axis from $10^{-2}$ to $10^0$.
Adversarial Training of other Models

- Linear models: SVM / linear regression cannot learn a step function, so adversarial training is less useful, very similar to weight decay

- $k$-NN: adversarial training is prone to overfitting.

- Takeway: neural nets can actually become more secure than other models. *Adversarially trained neural nets have the best empirical success rate on adversarial examples of any machine learning model.*
Weaknesses Persist
Adversarial Training

Labeled as bird

Decrease probability of bird class

Still has same label (bird)
Virtual Adversarial Training

Unlabeled; model guesses it’s probably a bird, maybe a plane

Adversarial perturbation intended to change the guess

New guess should match old guess (probably bird, maybe plane)
Text Classification with VAT

RCV1 Misclassification Rate

<table>
<thead>
<tr>
<th>Method</th>
<th>Misclassification Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earlier SOTA</td>
<td>7.70</td>
</tr>
<tr>
<td>SOTA</td>
<td>7.20</td>
</tr>
<tr>
<td>Our baseline</td>
<td>7.40</td>
</tr>
<tr>
<td>Adversarial</td>
<td>7.12</td>
</tr>
<tr>
<td>Virtual Adversarial</td>
<td>7.05</td>
</tr>
<tr>
<td>Both</td>
<td>6.97</td>
</tr>
<tr>
<td>Both + bidirectional model</td>
<td>6.68</td>
</tr>
</tbody>
</table>

(Zoomed in for legibility)
Universal engineering machine (model-based optimization)

Make new inventions by finding input that maximizes model’s predicted performance

Training data

Extrapolation

(Goodfellow 2016)
Conclusion

• Attacking is easy

• Defending is difficult

• Adversarial training provides regularization and semi-supervised learning

• The out-of-domain input problem is a bottleneck for model-based optimization generally
cleverhans

Open-source library available at:
https://github.com/openai/cleverhans

Built on top of TensorFlow (Theano support anticipated)
Standard implementation of attacks, for adversarial training and reproducible benchmarks