Adversarial Attacks and Defences

Yingzhen Li

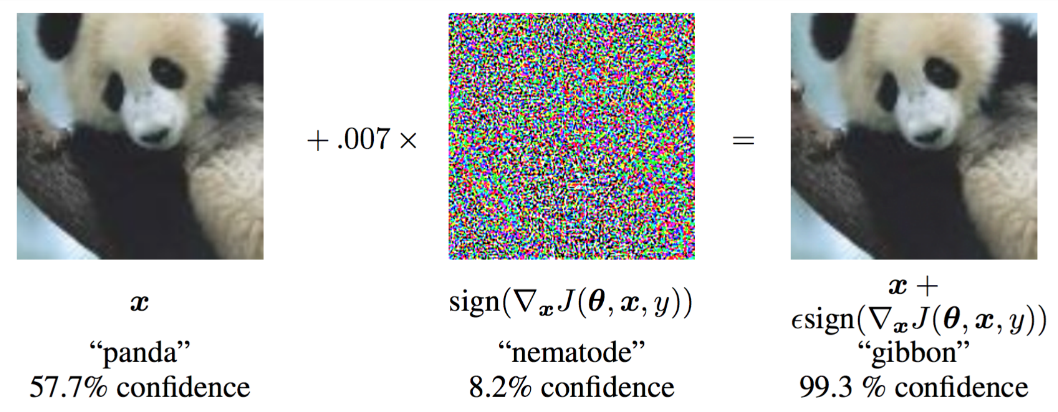
Machine Learning Group, University of Cambridge
Part of this talk is based a joint work with Yarin Gal (Oxford)

AI safety is not only restricted to RL

Long-standing practice: Spam emails and filters

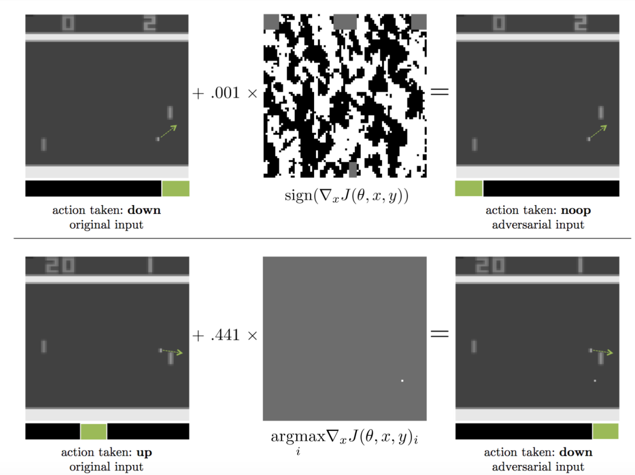
- Defender: building better spam filters
- Attacker: figure out how the spam filters work and then cheat

Adversarial attack to image classifiers



Goodfellow et al. ICLR 2015

Adversarial attack to NN policies



Huang et al. arXiv 2017

CV-based autonomous driving system can get fooled!

Evtimov et al. arXiv 2017

Closely related research topics

- computer security
- machine learning
    - differential privacy
    - interpretability

Attacks

Threat Model

- Adversary's goal: maximise some utility function
- Adversary's capability, for example:
  - modifying input data (limited or unlimited)
  - modifying feature vectors
  - facing constraints: amount of modifications, computation time

Biggio et al. ECML PKDD 2013

Threat Model

- Adversary's knowledge:
  - zero-knowledge (or black-box attack)
  - perfect-knowledge (white-box attack)
  - limited-knowledge

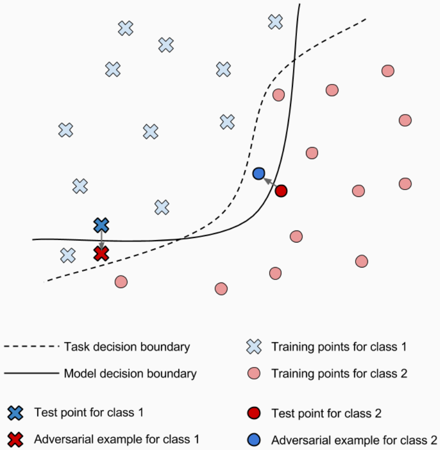
Biggio et al. ECML PKDD 2013

Carlini and Wagner. ACM 2017

Attacks to classifiers

- clean input: $\mathbf{x}$ with class label y
- "victim": a classifier $y_{\text{pred}} = C(\mathbf{x})$ (or $p(y|\mathbf{x})$) than can output the correct label given a clean input
- corrupted input: $\mathbf{x}_{\text{adv}} = \mathbf{x} + \epsilon$ (adversarial example)
- goal: make the classifier predict wrong labels (untargeted attack) or a given label other than the true one (targeted attack)

Attacks to classifiers



cleverhans.io. Blogpost 2016

White-box attacks

Fast gradient sign method (FGSM):

untargeted attack:

$$\mathbf{x}_{\text{adv}} = \mathbf{x} - \eta \cdot \operatorname{sgn}(\nabla_{\mathbf{x}} \max_{y} \log p(y|\mathbf{x}))$$

targeted attack:

$$\mathbf{x}_{\text{adv}} = \mathbf{x} + \eta \cdot \operatorname{sgn}(\nabla_{\mathbf{x}} \log p(y_{\text{target}}|\mathbf{x}))$$

Goodfellow et al. ICLR 2015

White-box attacks

Iterative attacks:

keep updating $\mathbf{x}_{\text{adv}}$ until successful/running out of time

example: iterative FGSM (targeted), or called BIM:

$$\mathbf{x}_{\text{adv}}^{t} = \mathbf{x}_{\text{adv}}^{t-1} + \eta \cdot \text{sgn}(\nabla_{\mathbf{x}} \log p(y_{\text{target}}|\mathbf{x}_{\text{adv}}^{t-1}))$$

example: Jacobian-based saliency map (JSMA): iteratively,

- compute $\nabla_{\mathbf{x}} p(y|\mathbf{x})$ for all possible classes $y = 1, ..., N_{\text{class}}$
- compute the saliency map for each element of the input $\mathbf{x}_i$:

$$\boldsymbol{S}(\mathbf{x}, y_{\text{target}})_i = \begin{cases} 0 & \text{if } \nabla_{\mathbf{x}_i} p(y_{\text{target}}|\mathbf{x}_i) < 0 \\ 0 & \text{if } \sum_{y \neq y_{\text{target}}} \nabla_{\mathbf{x}_i} p(y|\mathbf{x}_i) > 0 \\ \nabla_{\mathbf{x}_i} p(y_{\text{target}}|\mathbf{x}_i) |\sum_{y \neq y_{\text{target}}} \nabla_{\mathbf{x}_i} p(y|\mathbf{x}_i)| & \text{otherwise} \end{cases}$$

- pick the pixel with maximum entry to the map
- modify that pixel

Papernot et al. IEEE 2016

White-box attacks

Biggio et al. framework:

$$\min_{\epsilon} dist(\mathbf{x}, \mathbf{x}+\epsilon) \quad \text{s.t. } C(\mathbf{x}+\epsilon) = y_{\text{target}}, \ \mathbf{x}+\epsilon \text{ is a valid image}$$

C&W framework:

$$\min_{\epsilon} dist(\mathbf{x}, \mathbf{x}+\epsilon) + \lambda f(\mathbf{x}+\epsilon) \quad \text{s.t. } \mathbf{x}+\epsilon \text{ is a valid image}$$

$-f(\mathbf{x}+\epsilon)$ is some utility function that needs to be specified.

Biggio et al. ECML PKDD 2013

Carlini and Wagner. IEEE 2017

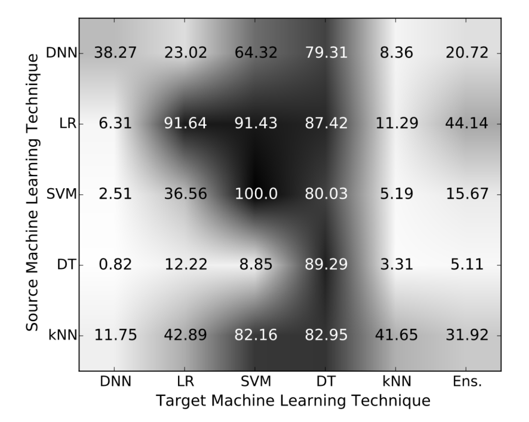
Black-box attacks

- intra-task transfer: assume dataset A and dataset B has very similar underlying data distributions. Then attacks that can fool models trained on A is also likely to fool models trained on B
- cross-technique transfer: assume models C and D are trained on the same dataset. Then attacks that can fool C is also likely to fool D.

Papernot et al. arXiv 2016

Black-box attacks

Example: cross technique adversary transfer



Papernot et al. arXiv 2016

Attacks with limited knowledge

- Assume you cannot compute gradient/saliency maps using the victim model
- However, given an image you can query the victim for its prediction
- Idea: distillation + crafting adversarial samples on the substitute
  - distillation: train a student/substitute model to mimic the behaviour of the victim model
  - importantly the substitute model is used to approximate the decision boundary of the victim model
  - adversarial examples that fool the substitute model is also likely to fool the victim model

Papernot et al. arXiv 2016

Defences

Adversarial training

Idea: add the adversarial images to the training set

- $(\mathbf{x}, y)$ – clean data input and output pair
- $\mathbf{x}_{\text{adv}}$ – adversarial example
- then we can add $(\mathbf{x}_{\text{adv}}, y)$ to the training data

Szegedy et al. ICLR 2014

Goodfellow et al. ICLR 2015

Defence distillation/label smoothing

Defence distillation:

- train a big teacher network on data
- train a small student network using softmax output from the teacher
- need to divide the pre-softmax values by T

Label smoothing: convert one hot labels:

$$(0, 1, 0, ..., 0) \rightarrow \left(\frac{\epsilon}{N_{\text{class}} - 1}, 1 - \epsilon, \frac{\epsilon}{N_{\text{class}} - 1}, ..., \frac{\epsilon}{N_{\text{class}} - 1}\right)$$

Papernot et al. arXiv 2015

Warde-Farley and Goodfellow. Perturbations, Optimization, and Statistics, 2016

Augmenting neural networks for detection/recovery

Train another network to detect/recover from attack:

- let's say the original class labels $y = 1, ..., N_{\text{class}}$
- add a new label $y = N_{\text{class}} + 1$ to represent adversarial examples
- augment the neural network to include class $N_{\text{class}} + 1$
- add $(\mathbf{x}_{\text{adv}}, N_{\text{class}} + 1)$ to the training data

Grosse et al. arXiv 2017

Gong et al. arXiv 2017

Augmenting neural networks for detection/recovery

Train an auto-encoder to "de-noise" the adversarial images:

- train an AE to map both $\mathbf{x}$ and $\mathbf{x}_{\text{adv}}$ back to $\mathbf{x}$
- in test time: given any input $\mathbf{x}^*$, compute label on AE($\mathbf{x}^*$) as the prediction

Gu and Rigazio. arXiv 2014

Two sample test

Statistical testing to find adversarial examples

- assume we have two sets A and B, we know A contains clean data, B contains either clean data or adversarial examples
- do statistical testing to determine whether A and B are identically distributed
- test statistic selection is key here

Grosse et al.. arXiv 2017

Examining uncertainty using Bayesian neural networks

Best defence technique so far!
(caveat: I haven't check all ICLR 2018 submissions)

Concurrently considered by us (joint work with Yarin Gal) and a few other groups.

Li and Gal ICML 2017

Feinman et al. arXiv 2017

Louizos and Welling. ICML 2017

Why BNNs could be more robust to adversarial attacks?

A simple reasoning for improved robustness:

- Let's say you have an ensemble of neural nets
- In most cases the attacker can access the majority vote of the ensemble
- i.e. the attacker needs to fool more than a half of them

Why BNNs could be more robust to adversarial attacks?

A simple reasoning for improved robustness:

- Let's say you have an ensemble of neural nets
- In most cases the attacker can access the majority vote of the ensemble
- i.e. the attacker needs to fool more than a half of them

BNN is better than naive ensembling!

- Bayesian prediction $\Leftrightarrow$ constructing an infinite ensemble in a principled way
- MC sampling returns a random set of ensembles

Being robust $\neq$ being able to detect!

- Adversarial training: more robust, but still provide point estimates
- Ensembles: even when majority vote is fooled, disagreement can still exist!

  (describes uncertainty in some sense)

[1] other possible idea: bootstrapping and bagging

Being robust $\neq$ being able to detect!

- Adversarial training: more robust, but still provide point estimates
- Ensembles: even when majority vote is fooled, disagreement can still exist!
  (describes uncertainty in some sense)

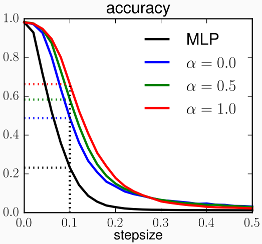
However, we need reliable "uncertainty" here:

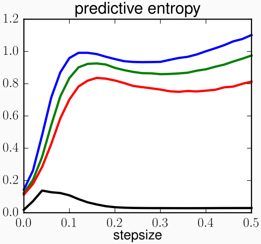
- ideal case: uncertainty level grows as we move away from the data manifold
- meaning we need calibrated uncertainty estimates
- Bayesian method is one of the natural choice [1]

[1] other possible idea: bootstrapping and bagging

Adversarial attack detection: being Bayesian helps!

- Attack: FGSM
- Detection metric: predictive entropy $\mathbb{H}(p(\mathbf{y}|\mathbf{x}_{\text{adv}}, \mathbf{X}, \mathbf{Y}))$ with the predictive distribution approximated by MC-dropout
- All 3 BNNs are more robust!
- ... and indeed very uncertain at $\mathbf{x}_{\text{adv}}$





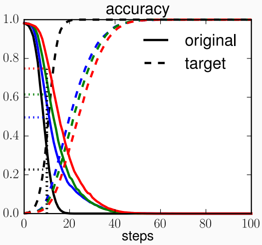
MLP	$\alpha = 0.0$	$\alpha = 0.5$	$\alpha = 1.0$
0.2318	0.4879	0.5832	0.663

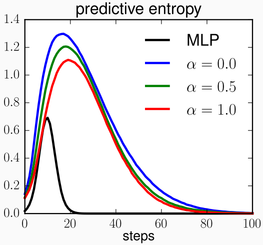
accuracy for stepsize=0.1

Li and Gal. ICML 2017

Adversarial attack detection: being Bayesian helps!

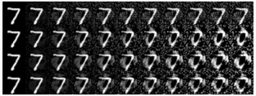
- Attack: Iterative Targeted FGSM
- All 3 BNNs are agian more robust!
- This attack on BNNs produces trajectories on the manifold!





MLP	$\alpha = 0.0$	$\alpha = 0.5$	$\alpha = 1.0$
0.2271	0.4960	0.6143	0.7480

original class acc. for #steps=10



Li and Gal. ICML 2017

Carlini and Wagner's comments

Carlini and Wagner claimed that all the above defences can be bypassed!

However, they did not conclude the (dropout) Bayesian NN technique to be completely broken:

"...At this time, we believe this is the most promising direction of future work."

Carlini and Wagner. ACM 2017

References

Goodfellow et al. Explaining and Harnessing Adversarial Examples. ICLR 2015

Huang et al. Adversarial Attacks on Neural Network Policies. arXiv 1702.02284

Evtimov et al. Robust Physical-World Attacks on Deep Learning Models. arXiv 1707.08945

Biggio et al. Evasion Attacks against Machine Learning at Test Time. ECML PKDD 2013

Carlini and Wagner. Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods. ACM Workshop on Artificial Intelligence and Security, 2017.

Papernot et al. The Limitations of Deep Learning in Adversarial Settings. 1st IEEE European Symposium on Security and Privacy, IEEE 2016

References

Carlini and Wagner. Towards Evaluating the Robustness of Neural Networks. IEEE Symposium on Security and Privacy, 2017.

Papernot et al. Practical Black-Box Attacks against Machine Learning. arXiv 1602.02697

Papernot et al. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. arXiv 1605.07277

Szegedy et al. Intriguing properties of neural networks. ICLR 2014

Papernot et al. Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks. arXiv 1511.04508

Warde-Farley and Goodfellow. Adversarial Perturbations of Deep Neural Networks. Perturbations, Optimization, and Statistics. 2016

References

Grosse et al. On the (Statistical) Detection of Adversarial Examples. arXiv 1702.06280

Gong et al. Adversarial and Clean Data Are Not Twins. arXiv 1704.04960

Gu and Rigazio. Towards Deep Neural Network Architectures Robust to Adversarial Examples. arXiv 1412.5068

Li and Gal. Dropout Inference in Bayesian Neural Networks with Alpha-divergences. ICML 2017

Feinman et al. Detecting Adversarial Samples from Artifacts. arXiv 1703.00410

Louizos and Welling. Multiplicative Normalizing Flows for Variational Bayesian Neural Networks. ICML 2017