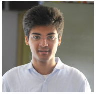
Efficient Defenses Against Adversarial Examples for Deep Neural Networks





Valentina Zantedeschi
@vzantedesc
Jean Monnet University

Irina Nicolae
@ririnicolae

Ambrish Rawat
@ambrishrawat

IBM Research AI

GreHack #5

November 17, 2017

Security and Machine Learning

So far...

- Machine learning for security
  - Intrusion detection[1]
  - Malware analysis[2]

This talk is about

- **Security for machine learning**

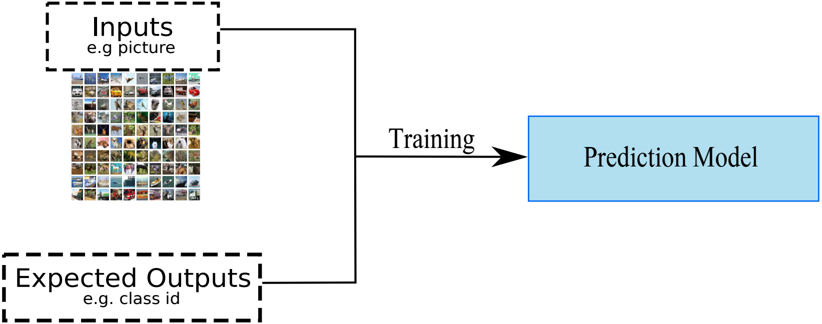
[1] Buczak & Guven, *A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection*. IEEE Comunications Surveys & Tutorials, 2015.

[2] Gandotra et al., *Malware Analysis and Classification: A Survey*, Journal of Information Security, 5, 56–64, 2014.

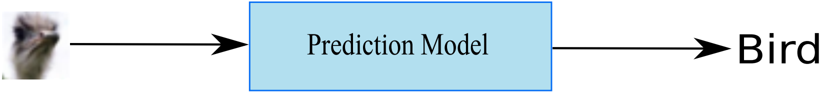
Machine Learning and Adversarial Examples

Machine Learning

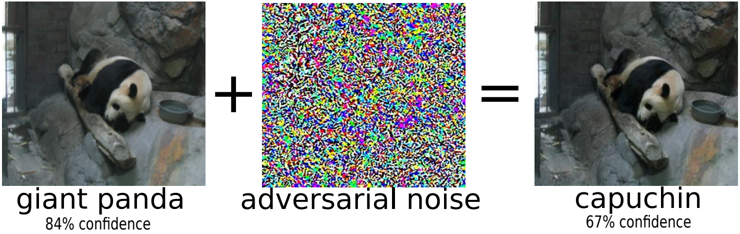
Training



Prediction



Adversarial Examples



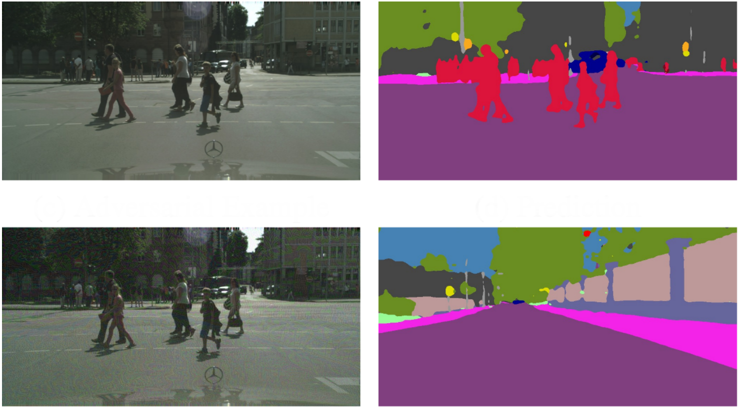
- Perturb model inputs with crafted noise
- Model fails to recognize input correctly
- Attack undetectable by humans
- Random noise does not work.

Practical Examples of Attacks

Self-Driving Cars

Image segmentation[3]

Attack noise hides pedestrians from the detection system.



[3] Metzen et al., *Universal Adversarial Perturbations Against Semantic Image Segmentation*. `https://arxiv.org/abs/1704.05712`.

Self-Driving Cars

Road signs[4]

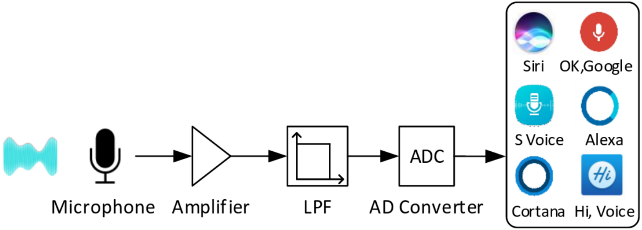
Car ends up ignoring the stop sign.

True image

Adversarial image

[4] McDaniel et al., *Machine Learning in Adversarial Settings*. IEEE Security and Privacy, vol. 14, pp. 68-72, 2016.

Executing Voice Commands

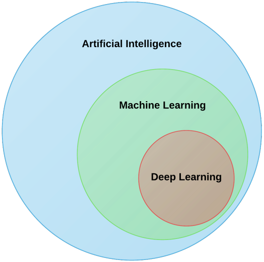


Okay Google, text John![5]

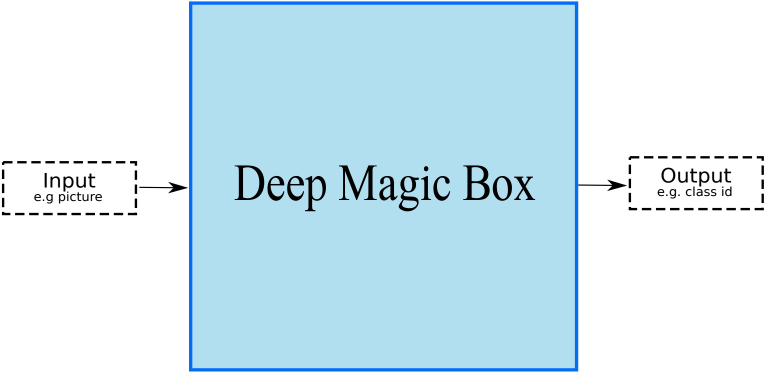
- Stealthy voice commands recognized by devices
- Humans cannot detect it.

[5] Zhang et al., *DolphinAttack: Inaudible Voice Commands*, ACM CCS 2017.

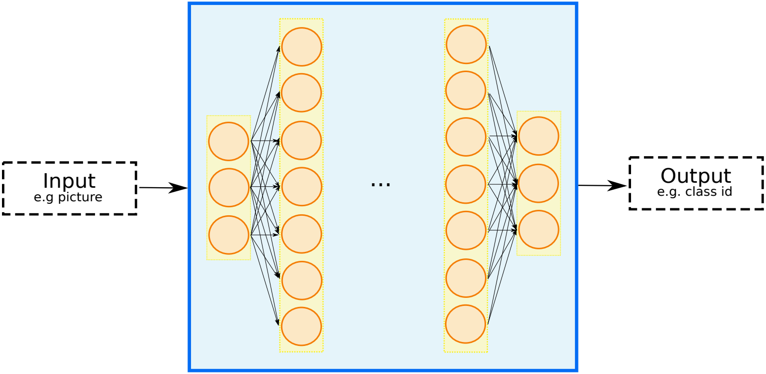
Deep Learning and Adversarial Samples



Deep Neural Networks

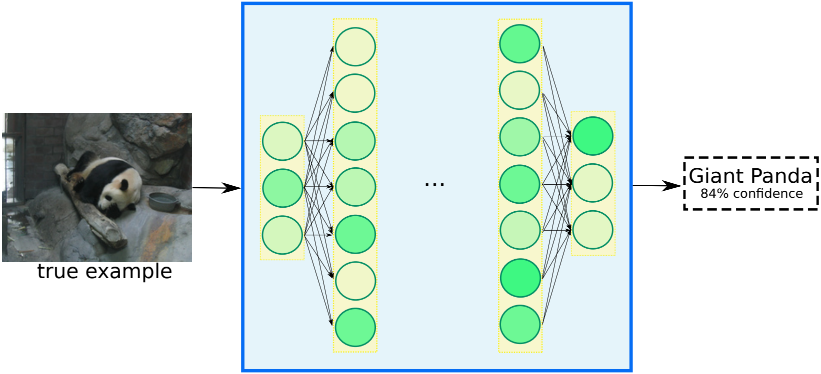


Deep Neural Networks



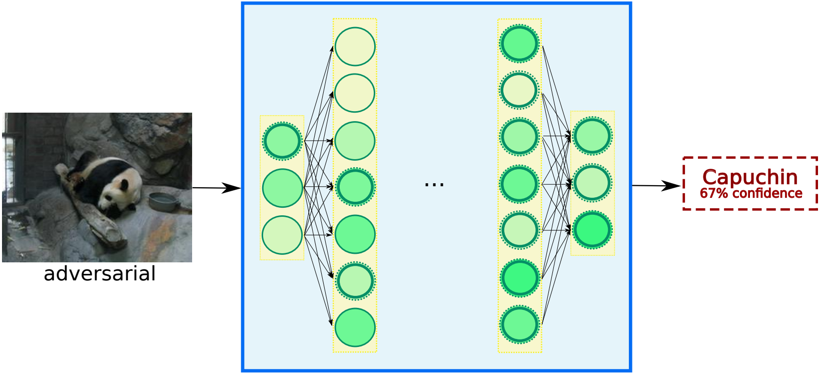
- Interconnected layers propagate the information forward.
- Model learns weights for each neuron.

Deep Neural Networks



- Specific neurons light-up depending on the input.
- Cumulative effect of activation moves forward in the layers.

Deep Neural Networks

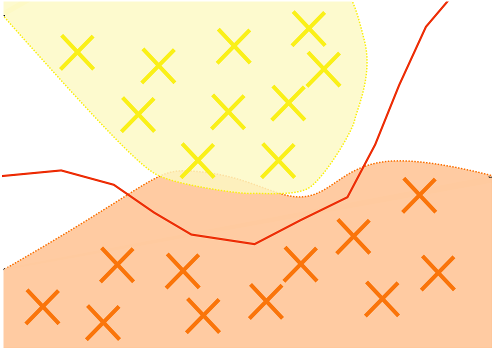


Small variations in the input → important changes in the output.

+ Enhanced discriminative capacities

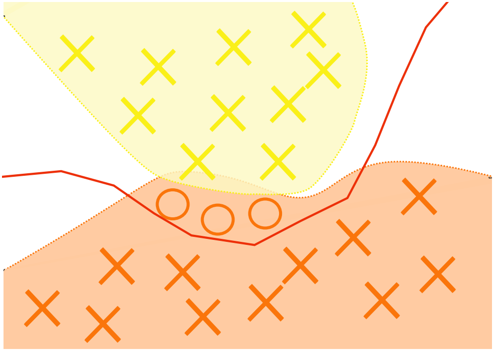
– Opens the door to adversarial examples

Decision Boundary of the Model



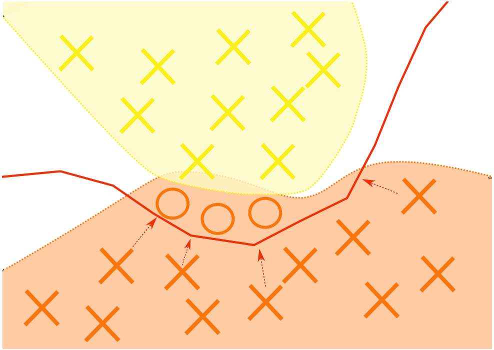
The **learned model** slightly differs from the **true** data distribution...

The Space of Adversarial Examples



... which makes room for **adversarial examples**.

Attack: Use the Adversarial Directions



- Most attacks try to move inputs across the boundary.
- Attacking with a random distortion doesn't work well in practice.

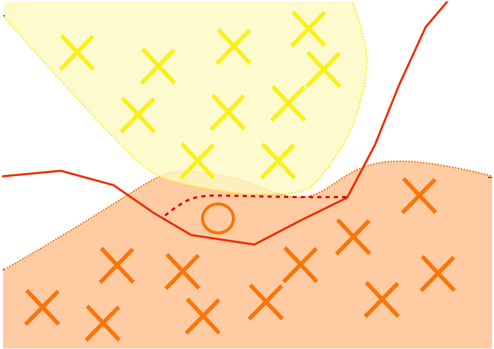
Finding Adversarial Examples

Given x, find x' where

- x and x' are close
- $\text{output}(x) \neq \text{output}(x')$

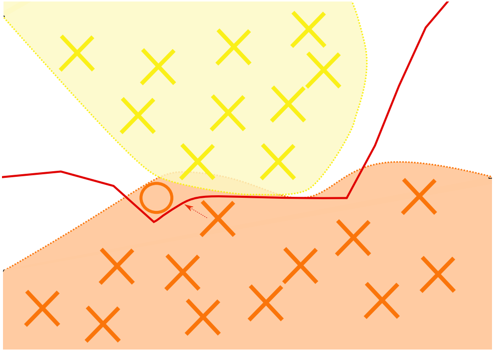
Approximations of the original problem	
FGSM [1]	quick, rough, fixed budget
Random + FGSM [2]	random step, then FGSM
DeepFool [3]	find minimal perturbations
JSMA [4]	modify most salient pixels
C&W [5]	strongest to date

Defense: Adversarial Training



- Adapt the classifier to attack directions by including adversarial data at training.

Defense: Adversarial Training



- Adapt the classifier to attack directions by including adversarial data at training.
- But there are always new adversarial samples to be crafted.

Defenses

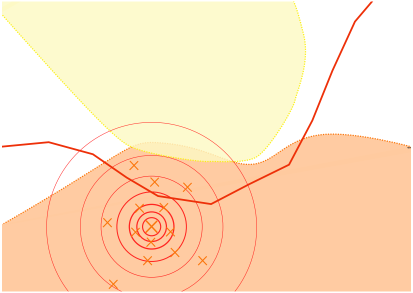
	Type	Description
AT	data augmentation	train also with adv. examples
VAT	data augmentation	train also with virtual adv. examples
FS	preprocessing	squeeze input domain
LS	preprocessing	smooth target outputs

- Adversarial Training (AT) [1]
- Virtual Adversarial Training (VAT) [6]
- Feature Squeezing (FS) [7]
- Label Smoothing (LS) [8]

Contribution: Effective Defenses Against Adversarial Samples

Gaussian Data Augmentation (GDA)

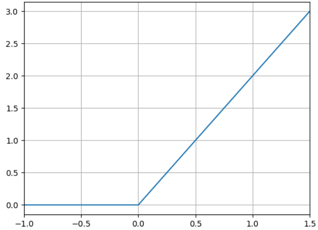
Gaussian noise does not work for attacks, but does it work as a defense?



- Reinforce neighborhoods around points using random noise.
- For each input image, generate N versions by adding Gaussian noise to the pixels.
- Train the model on the original data and the noisy inputs.

Bounding the Activation Function

Objective Limit the cumulative effect of errors in the layers.

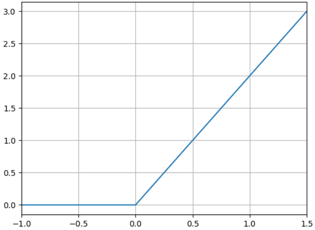


RELU

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0. \end{cases}$$

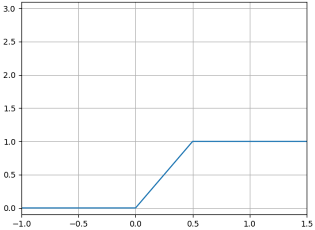
Bounding the Activation Function

Objective Limit the cumulative effect of errors in the layers.



RELU

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0. \end{cases}$$



Bounded RELU

$$f_t(x) = \begin{cases} 0, & x < 0 \\ x, & 0 \leq x < t \\ t, & x \geq t. \end{cases}$$

Comparison with Other Defenses

Defense	Training	Prediction
Feature Squeezing	preproc. input	preproc. input, perf. loss
Label Smoothing	preproc. output	-
Adversarial Training	train + attack + retrain	-
GDA + BRELU	add noise	-

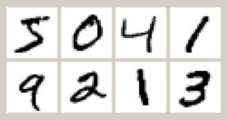
Advantages of GDA + BRELU

- Defense agnostic to attack strategy
- Model performance for original inputs is conserved
- Performs better than other defenses on adversarial samples
- Almost no overhead for training and prediction.

Experiments

Setup

- MNIST dataset of handwritten digits
  - 60,000 training + 10,000 test images
- CIFAR-10 dataset of 32×32 RGB images
  - 50,000 training + 10,000 test images
  - 10 categories
- Convolutional neural net (CNN) architecture



Setup

Threat model

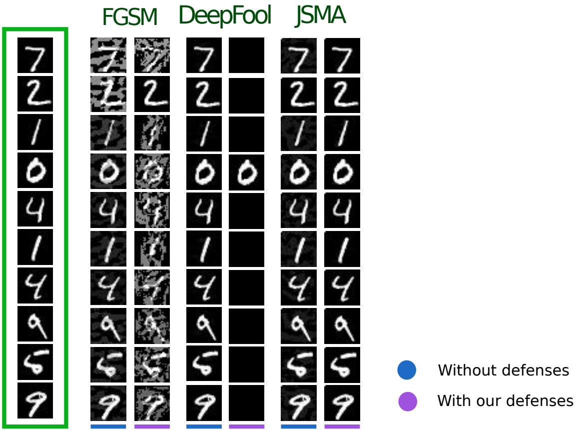
- **Black-box**: attacker has access to inputs and outputs
- **White-box**: attacker also has access to model parameters

Steps

- Train model with different defenses
- Generate attack images
- Compute defense performance on attack images

Minimal Perturbation

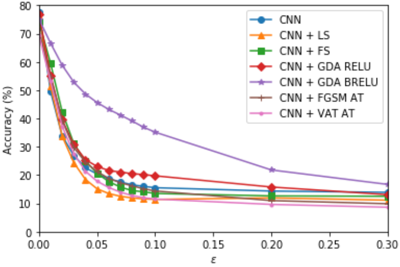
Amount of perturbation necessary to fool the model



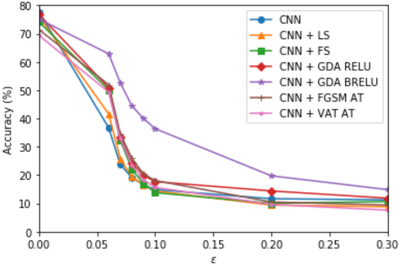
With GDA + BRELU, the perturbation necessary for an attack becomes **visually detectable**.

White-Box Attacks

Comparison of different defenses against white-box attacks



(a) FGSM attack



(b) Random + FGSM attack

CIFAR-10

Accuracy = % of correct predictions = TP + TN

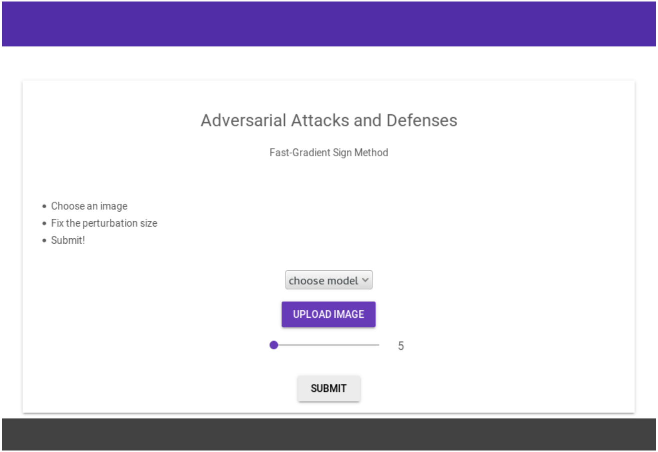
Comparison of different defenses against black-box attacks

Defense \ Attack	FGSM	Rand+FGSM	DeepFool	JSMA	C&W
CNN	94.46	40.70	92.95	97.95	93.10
Feature squeezing	96.31	**91.09**	96.68	97.48	96.75
Label smoothing	86.79	20.28	84.58	95.86	84.81
FGSM adv. training	91.86	49.77	85.91	98.62	97.71
VAT	97.53	74.35	96.03	98.26	96.11
GDA + RELU	**98.47**	**80.25**	**97.84**	**98.96**	**97.87**
GDA + BRELU	**98.08**	75.50	**98.00**	**98.88**	**98.03**

Attacks transferred from ResNet to CNN on MNIST

Accuracy = % of correct predictions = TP + TN

Demo


Adversarial Attacks and Defenses
Fast-Gradient Sign Method
Choose an image
Fix the perturbation size
Submit!
choose model
UPLOAD IMAGE
5
SUBMIT

Conclusion

Conclusion

Our contribution

- Improved defense against multiple types of attacks
- Model performance for clean inputs is preserved
- No retraining, no overhead for prediction
- Easy to integrate into models.

Takeaway

- The problem of adversarial examples needs to be solved before applying machine learning.

`nemesis`

- Our library of attacks and defenses
- Soon to be open source.

Full paper at `https://arxiv.org/pdf/1707.06728.pdf`

References I

[1] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014. URL `http://arxiv.org/abs/1412.6572`.

[2] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[3] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. *CoRR*, abs/1511.04599, 2015. URL `http://arxiv.org/abs/1511.04599`.

[4] Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. *CoRR*, abs/1511.07528, 2015. URL `http://arxiv.org/abs/1511.07528`.

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017. URL `https://arxiv.org/abs/1608.04644`.

References II

[6] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *arXiv preprint arXiv:1704.03976*, 2017.

[7] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *CoRR*, abs/1704.01155, 2017. URL `http://arxiv.org/abs/1704.01155`.

[8] David Warde-Farley and Ian Goodfellow. Adversarial perturbations of deep neural networks. In Tamir Hazan, George Papandreou, and Daniel Tarlow, editors, *Perturbation, Optimization, and Statistics*. 2016.