Adversarial Machine Learning

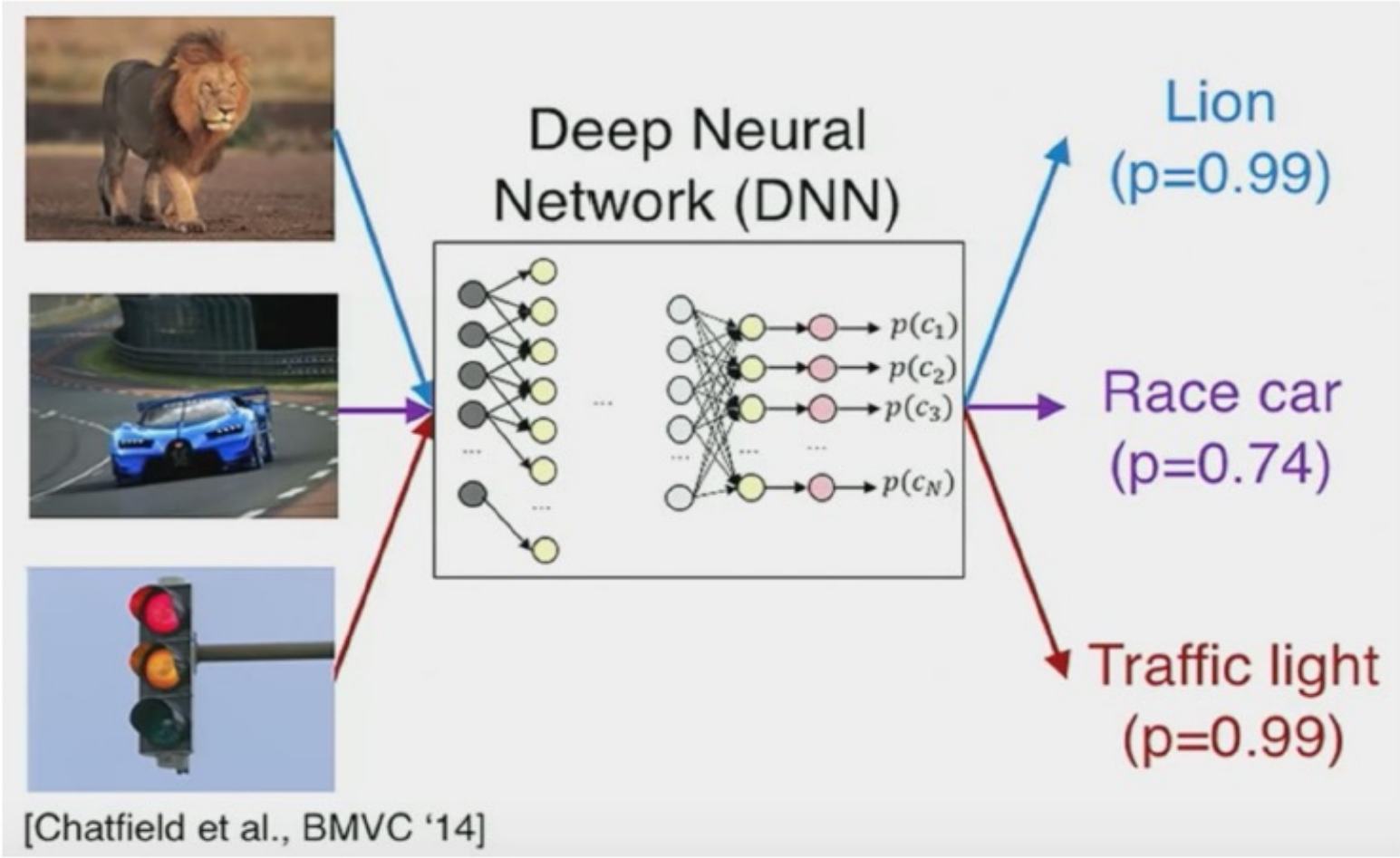
Adversarial ML

The classification accuracy of GoogLeNet on MNIST under adversarial attacks drops from 98% to 18% (for ProjGrad attack) or 1% (DeepFool attack)

Attack	Lenet				
Noise	Dataset	Acc@1 w/	Acc@5 w/	Acc@1 w/o	Acc@5 w/o
	MNIST	0.984	1.0	0.9858	1.0
	ILSVRC2012	NA	NA	NA	NA
Semantic	Dataset	Acc@1 w/	Acc@5 w/	Acc@1 w/o	Acc@5 w/o
	MNIST	0.233	0.645	0.986	1.0
	ILSVRC2012	NA	NA	NA	NA
Fast Gradient Sign Method	Dataset	Acc@1 w/	Acc@5 w/	Acc@1 w/o	Acc@5 w/o
	MNIST	0.509	0.993	0.986	1.0
	ILSVRC2012	NA	NA	NA	NA
Projected Gradient Descent	Dataset	Acc@1 w/	Acc@5 w/	Acc@1 w/o	Acc@5 w/o
	MNIST	0.187	0.982	0.986	1.0
	ILSVRC2012	NA	NA	NA	NA
DeepFool	Dataset	Acc@1 w/	Acc@5 w/	Acc@1 w/o	Acc@5 w/o
	MNIST	0.012	1.0	0.9858	1.0
	ILSVRC2012	NA	NA	NA	NA

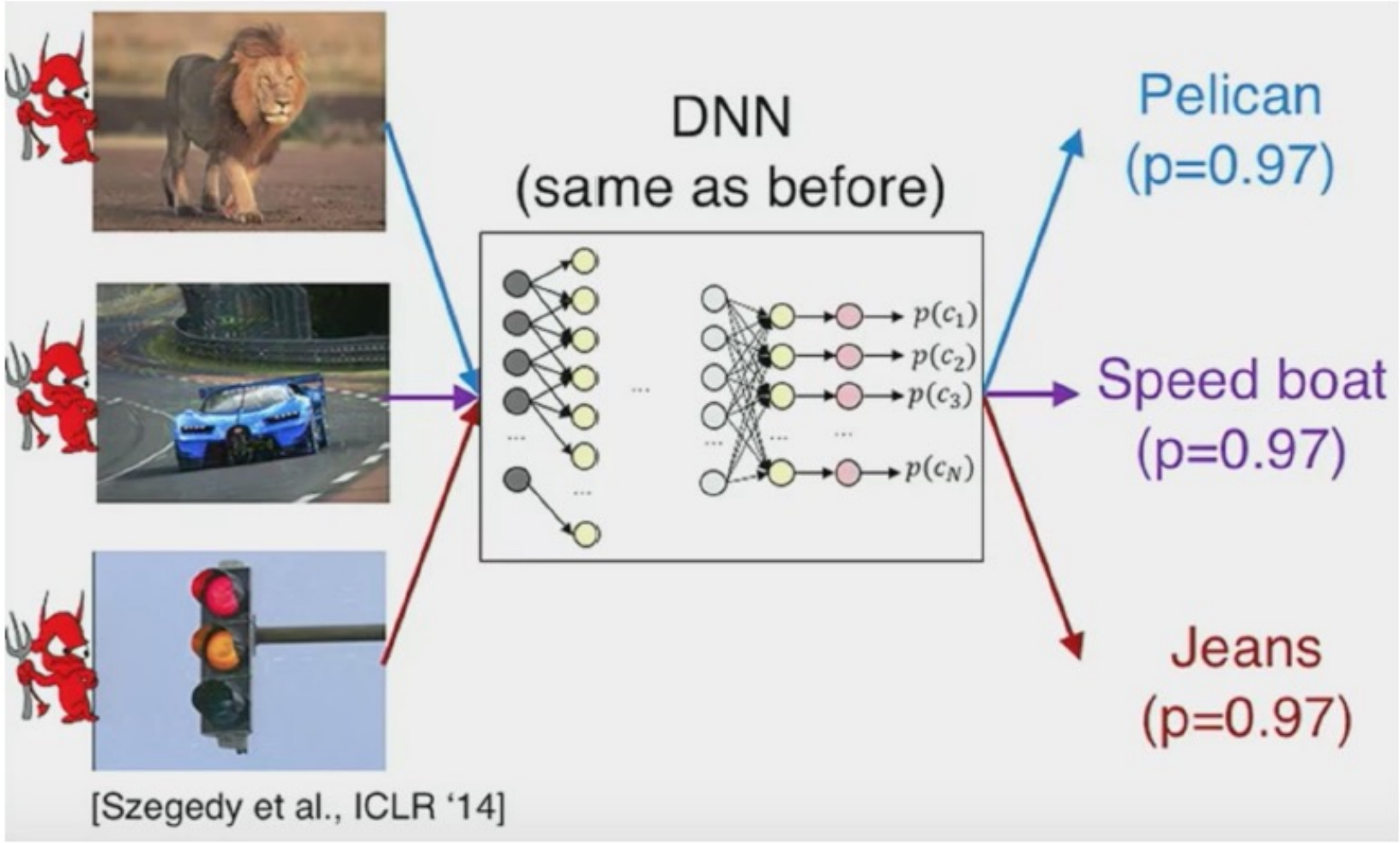
Adversarial Examples

What do you see?



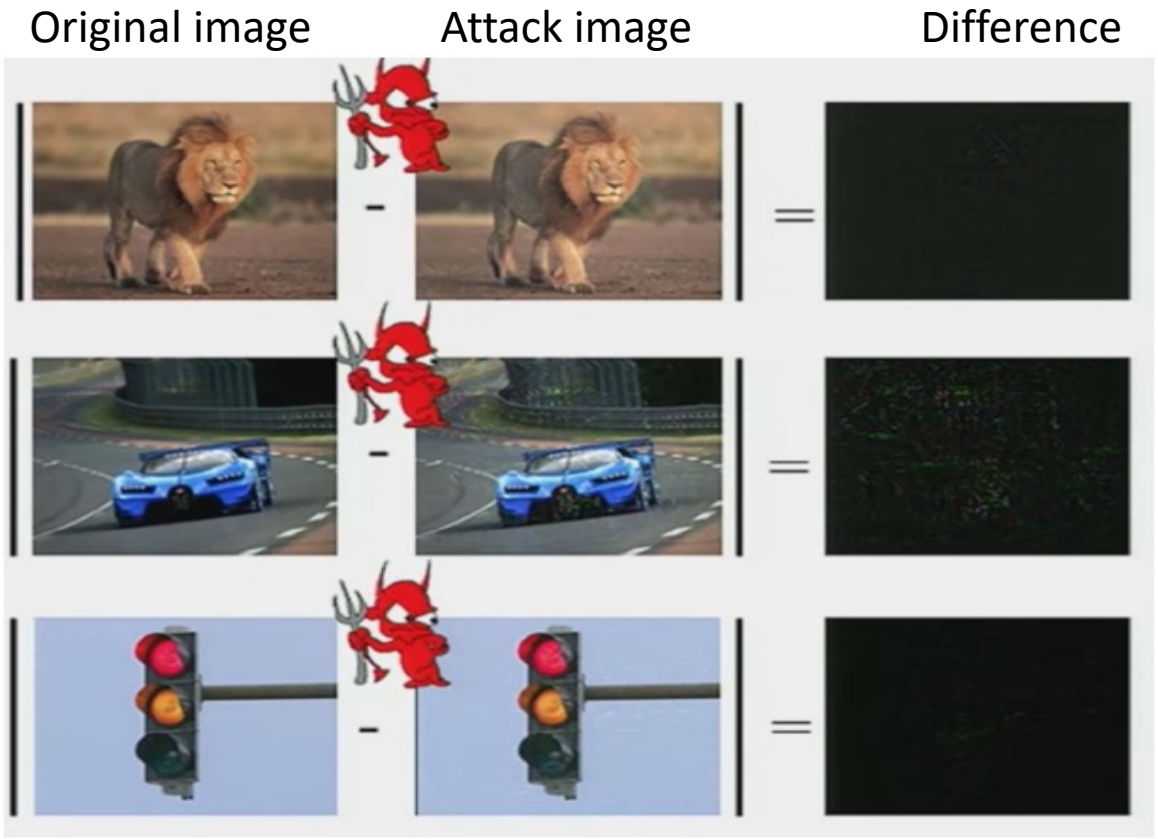
Adversarial Examples

The classifier misclassifies adversarially manipulated images



Adversarial Examples

The differences between the original and manipulated images are very small (hardly noticeable to the human eye)

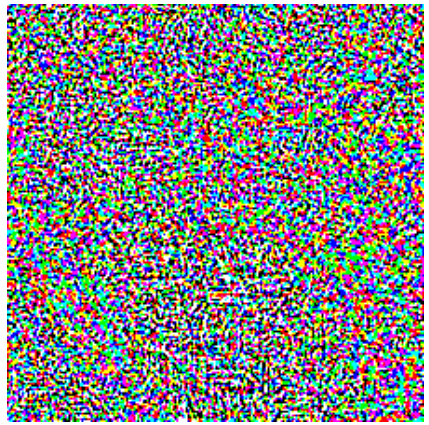


Adversarial Examples

- An adversarially perturbated image of a panda is misclassified as a gibbon
- The image with the perturbation to the human eye looks indistinguishable from the original image

Original image

Classified as panda
57.7% confidence



Small adversarial noise

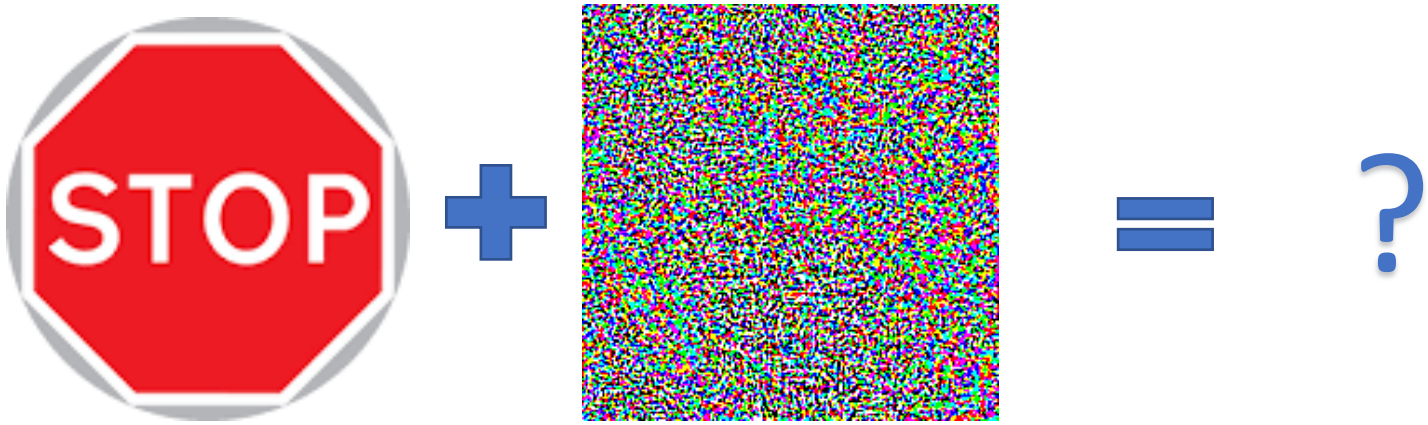
Adversarial image

Classified as gibbon
99.3% confidence

Gibbon

Adversarial Examples

If a stop sign is adversarially manipulated and it is not recognized by a self-driving car, it can result in an accident

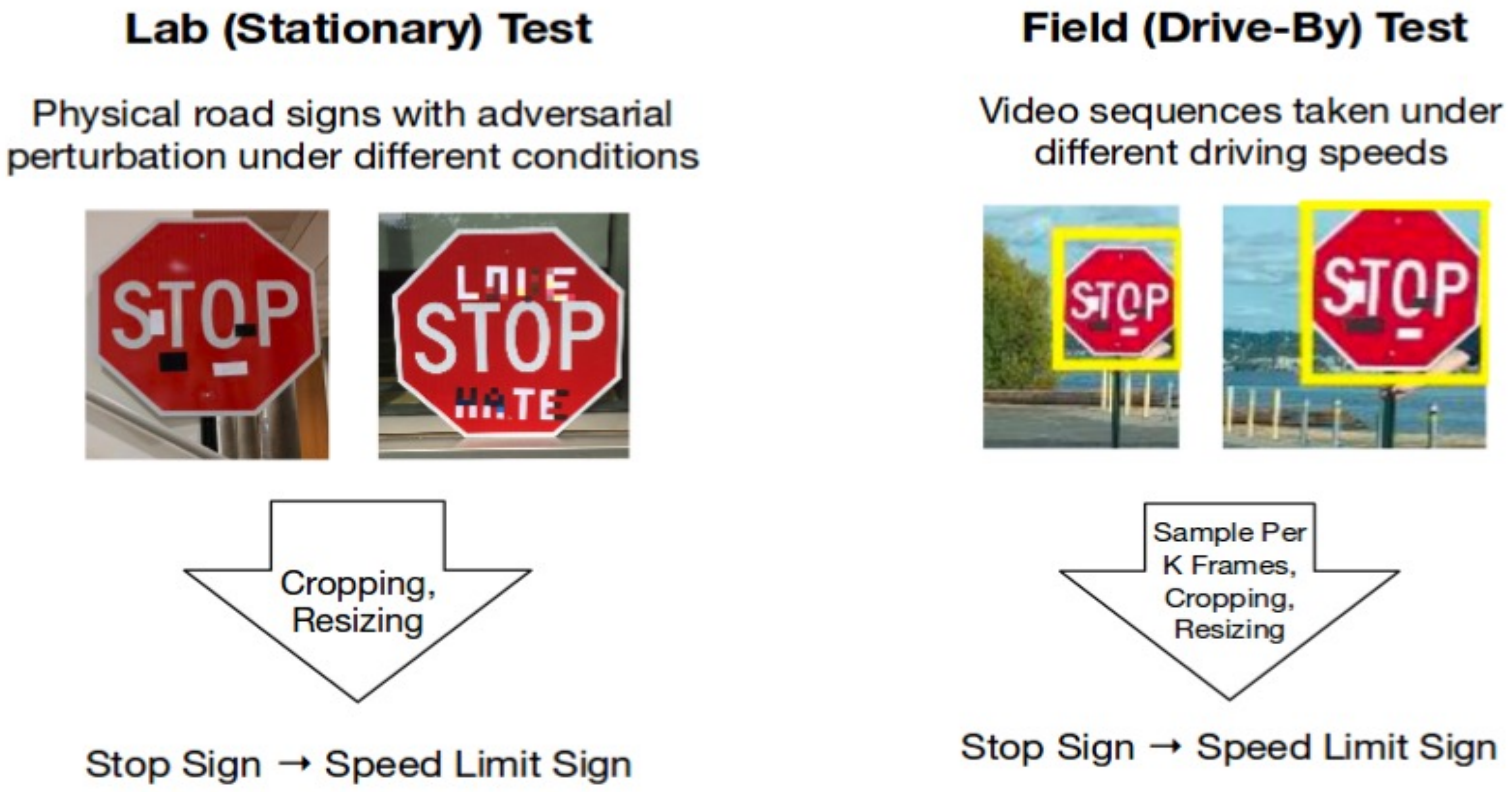


Small adversarial noise

Adversarial Examples

Some work manipulated a stop sign with adversarial patches

- Caused the DL model of a self-driving car to classify it as a Speed Limit 45 sign (100% attack success in lab test, and 85% in field test)



Adversarial Examples

- AML is a research field that lies at the intersection of ML and computer security
  - E.g., network intrusion detection, spam filtering, malware classification, biometric authentication (facial detection)
- ML algorithms in real-world applications mainly focus on increased accuracy
  - However, few techniques and design decisions focus on keeping the ML models secure and robust
- Adversarial ML: ML in adversarial settings
  - Attack is a major component of AML
  - Bad actors do bad things
    - Their main objective is not to get detected (change behavior to avoid detection)

Attack Taxonomy

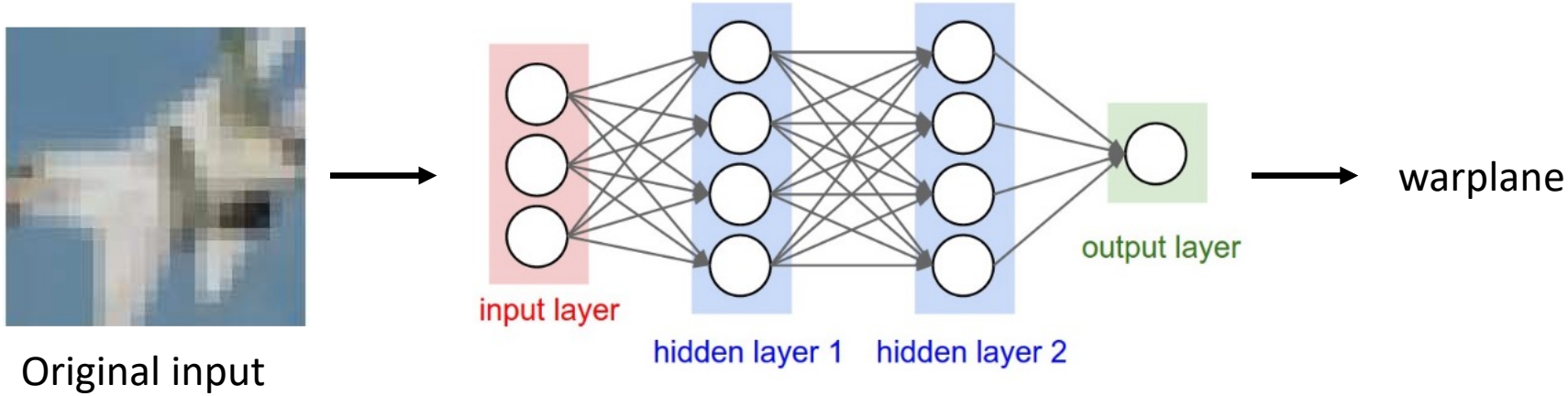
- Attack can be further classified into:
  - ***White-box attack***
    - Attackers have full knowledge about the ML model
    - I.e., they have access to parameters, hyperparameters, gradients, architecture, etc.
  - ***Black-box attack***
    - Attackers don't have access to the ML model parameters, gradients, architecture
    - Perhaps they have some knowledge about the used ML algorithm
      - E.g., attackers may know that a ResNet50 model is used for classification, but they don't have access to the model parameters
    - Attackers may query the model to obtain knowledge (can get examples)

Attack Taxonomy

- Each of the above attacks can further be:
  - ***Non-targeted*** attack
    - The goal is to mislead the classifier to predict any labels other than the ground truth label
    - Most existing work deals with this goal
    - E.g., perturb an image of a military tank, so that the model predicts it is any other class than a military tank
  - ***Targeted*** attack
    - The goal is to mislead the classifier to predict a target label for an image
    - More difficult
    - E.g., perturb an image of a turtle, so that the model predicts it is a riffle
    - E.g., perturb an image of a Stop sign, so that the model predicts it is a Speed Limit sign

Attack Taxonomy

- Find a new input (*similar* to original input) but classified as another class (untargeted or targeted)



Original input

- Adversarial attack image

Attack Taxonomy

- How to find adversarial images?
  - Given an image *x*, which is labeled by the classifier (e.g., LogReg, SVM, or NN) as class q, i.e., $C(x) = q$
  - Create an adversarial image x_{adv} by adding small perturbations δ to the original image, i.e., $x_{adv} = x + \delta$, such that the distance $D(x, x_{adv}) = D(x, x + \delta)$ is minimal
  - So that the classifier assigns a label to the adversarial image that is different than q, i.e., $C(x_{adv}) = C(x + \delta) = t \neq q$

$$\begin{aligned} \text{minimize} \quad & \mathcal{D}(x, x+\delta) \\ \text{such that} \quad & C(x+\delta) = t \\ & x + \delta \in [0,1]^n \end{aligned}$$

distance between *x* and *x*+δ

x+δ is classified as target class *t*

each element of *x*+δ is in [0,1] (to be a valid image)

Common Adversarial Attacks

- Noise attack
- Semantic attack
- Fast gradient sign method (FGSM) attack
- Basic iterative method (BIM) attack
- Projected gradient descent (PGD) attack
- DeepFool attack
- Carlini-Wagner (CW) attack

FGSM Attack

Whitebox attack methods

- ***Fast gradient sign method (FGSM) attack***

  Goodfellow (2015) - Explaining and Harnessing Adversarial Examples

- Classifier (e.g. ResNet50)

$$\hat{y} = f(w, x)$$

- Find adversarial image x_{adv} that maximizes the loss:

$$\mathcal{L}(x_{adv}, \mathrm{y}) = \mathcal{L}((f(w, x), y))$$

- Bounded perturbation:

$$|| x_{adv} - x || \leq \epsilon, \epsilon \text{ the attack strength}$$

FGSM Attack

- An adversarial image x_{adv} is created by adding perturbation noise to an image x

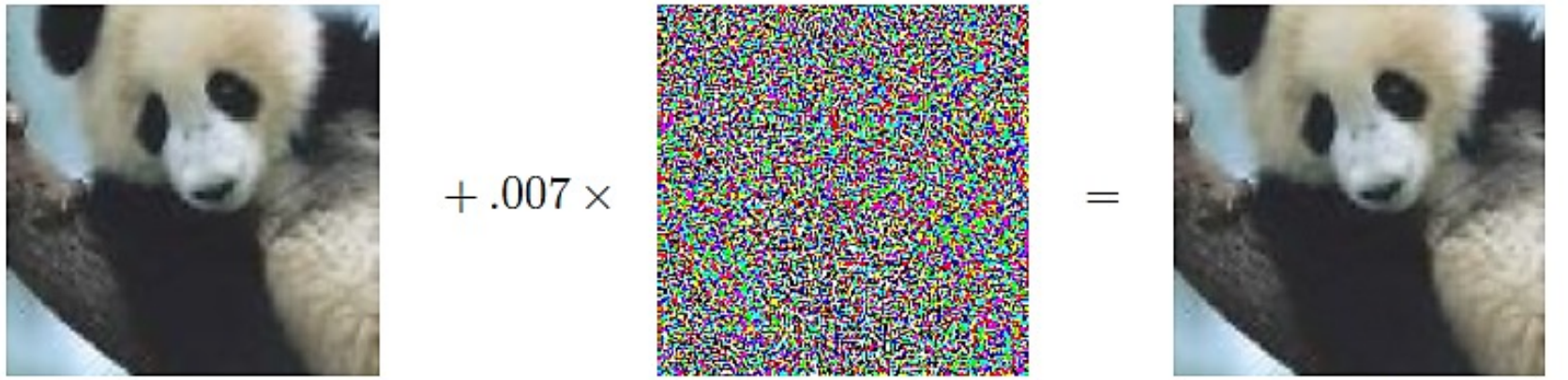
$$x_{adv} = x + \epsilon \cdot \text{sign}\left(\nabla_x \mathcal{L}(f(w, x), y)\right)$$

  - Notation: input image x, cost function $\mathcal{L}$, NN model f, NN weights (parameters) w, gradient ∇, noise magnitude ϵ
  - Perturbation noise is calculated as the gradient of the loss function $\mathcal{L}$ with respect to the input image x for the true class label y
  - This increases the loss for the true class $y \rightarrow$ the model misclassifies the image x_{adv}

$$\text{sgn}(x) := \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases}$$

FGSM Attack

- FGSM is a white-box non-targeted attack
  - White-box, since we need to know the gradients to create the adversarial image
  - The noise magnitude is ε = 0.007
    - Note: nematode is an insect referred to as roundworm



FGSM Attack

- FGSM attack example

Original image

Prediction: car mirror

Adversarial image

Prediction: sunglasses

Picture from: https://blog.floydhub.com/introduction-to-adversarial-machine-learning/

Defense Against Adversarial Attacks

- Adversarial samples can cause any ML algorithm to fail
  - However, they can be used to build more accurate and robust models
- AML is a two-player game:
  - Attackers aim to produce strong adversarial examples that deceive a model with high confidence while requiring only a small perturbation
  - Defenders aim to produce models that are robust to adversarial examples.
- Defense strategies against adversarial attacks include:
  - Adversarial training
  - Detecting adversarial examples
  - Gradient masking
  - Robust optimization (regularization, certified defenses)

Adversarial Training

- Learning the model parameters using adversarial samples is referred to as ***adversarial training*** (add adversarial examples to training set).
- The training dataset is augmented with adversarial examples produced by known types of attacks
- However, if a model is trained only on adversarial examples, the accuracy to classify regular examples will reduce significantly
- Possible strategies:
  - Train the model from scratch using regular and adversarial examples
  - Train the model on regular examples and afterward fine-tune with adversarial examples

Adversarial Training

- Found that training with an adversarial objective function based on the fast gradient sign method was an effective regularizer:

$$\tilde{J}(\boldsymbol{\theta}, \boldsymbol{x}, y) = \alpha J(\boldsymbol{\theta}, \boldsymbol{x}, y) + (1 - \alpha) J(\boldsymbol{\theta}, \boldsymbol{x} + \epsilon \text{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\right).$$

- Continually update our supply of adversarial examples, to make them resist the current version of the model
- Reduce the error rate from 0.94% without adversarial training to 0.84% with adversarial training.

Adversarial Training

Pros:

- simple to implement
- works well for the considered attack types

Cons:

- depends on specific attack type / strength
- less effective against blackbox attacks
- leads to accuracy drop of unperturbed images