# Adversarial Examples

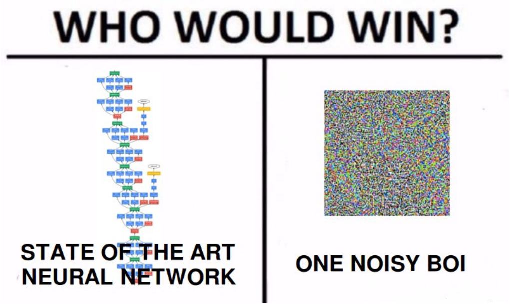
BRENOT, DARGENT, GUYONVARCH, LE POGAM

M2 AIDN /University of Bretagne Sud, Vannes

03 Décembre 2019

# Table of content

# Introduction



# What are adversarial examples ?

# Definition

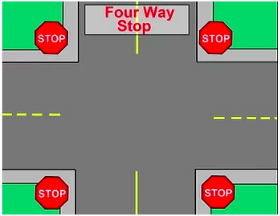
- Input that makes the model predict erroneously
- M(I) = ytrue
- Looking for A, such as M(A) != ytrue

# Application domain

- Create an audio
- Facial recognition
- Spam
- Autonomous car

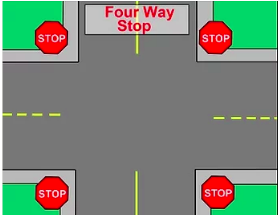
# Application domain

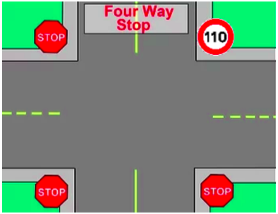
- Create an audio
- Facial recognition
- Spam
- Autonomous car



# Application domain

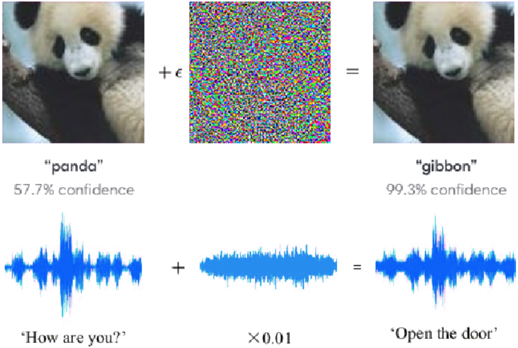
- Create an audio
- Facial recognition
- Spam
- Autonomous car



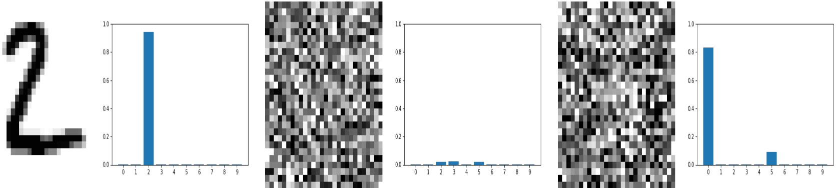
# Targeted examples

A constructed noise is added to the sample that seems similar for a human, but causes a misclassification for the neural network.



# Non targeted examples

We try to find any input that causes a misclassification for the neural network



# White and Black Box

- White Box
  - We know how the classification algorithm works
- Black Box
  - An attack without the knowledge of the classification algorithm
  - An adversarial examples which works for one model, can work for an other model

# Attack frequency

- One step attack : We optimize the adversarial example once
- Iterative attack : Take multiple times to improve the adversarial example
  - Adversarial examples more robust
  - Take more time to generate one example

# How can we generate adversarial examples ?

# Fast Gradient Sign Method

- Fool an already trained model
- Create an image that maximises the loss

$$adv_x = x + \epsilon * sign(\nabla_x J(\theta, x, y)) \tag{1}$$

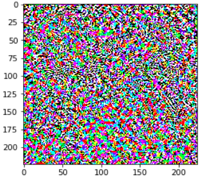
- $adv_x$ : adversarial example
- x : original input image
- y : original input label
- $\epsilon$ : coefficient of the attack
- $\nabla$ : gradient
- $\theta$ : model parameters

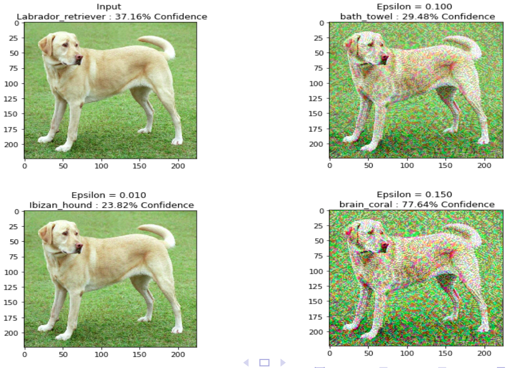
# Fast Gradient Sign Method

- Use the gradient of the neural network
- gradients are taken from the input image not from the model parameters
- for each pixel we want to know how much it contributes to the loss
- generation of perturbations as an image (noise)

# Fast Gradient Sign Method - Example

- in a way it is a gradient ascent of the cost function
- hence it increases the model error



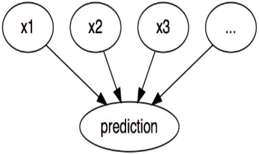


# Fast Gradient Sign Method

- the architecture must be known to compute de classifier's gradient (White box type)

# 1-pixel attack

- Similar to Counterfactual explanation




Describes the smallest change to the feature values that changes the prediction to a predefined output.

# 1-pixel attack

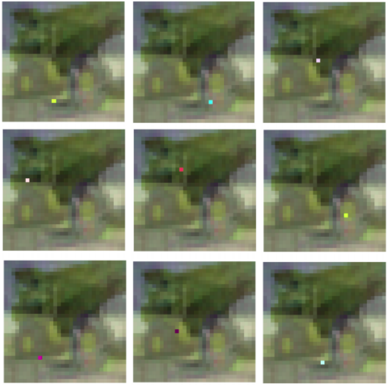
Aim : Modify one pixel and analyse what changes happend. We want to find the best adversarial example but also stay close from the original image.

# 1-pixel attack method

- Differencial Evolution
  - A group of candidats, each candidat is an evolution of the parent generation which represents a potential solution.
  - each candidat is a modification of one pixel of the image
  - Candidat representation is a vector containing the X and Y coordinates and the RGB value of the pixel.
  - Each child is generate following the formula :

$$x_i(g+1) = x_{r1}(g) + F.(x_{r2}(g) + x_{r3}(g)) \tag{2}$$

# 1-pixel example



| | | |
|---|---|---|
| Airplane | Automobile | Bird |
| Cat | Deer | Frog |
| Horse | Ship | Truck |

Target classes

# Label attack

Aim : Creation of one label which create an adversarial examples when it's put on an image. The label can be anything.

# Label attack method

- Method
  - We select an image as a patch and apply random transformations.
  - We test on different images.
  - Patch Application Operator
    - Apply transformations on the patch and put it in the image.
    - We train the patch to optimize the probability to obtain the target class
  - Universal transformation

# Label attack examples

| | |
|---|---|
| toaster | 0.96 |
| cellular telephone, cellula... | 0.03 |
| mouse, computer mouse | 0.00 |
| printer | 0.00 |
| iPod | 0.00 |

# Examples

https://www.youtube.com/watch?v=piYnd_wYlT8 https://www.youtube.com/watch?v=i1sp4X57TL4&feature=youtu.be

# How to defend against adversarial examples ?

# Types of Defenses

- Reactive : We detect adversarial examples once the neural network build.
- Proactive : We create a more robust neural network againt adversarial examples.

# Statistical Sequence Irregularity Detection

- Adversarial examples are out-of-distribution samples
- This irregularity can be used to detect AE
  - Analyse the conditionnal probability between elements
  - Compute the maximum mean divergence
- Example :
  - In the case of network development, CreateSocket() always appears before CloseSocket()
  - In the case of a 1-pixel attack, one pixel will stand out, thus creating a high divergence

# Sequence Squeezing

- Lower the dimensions of the input data
- If the distance between the squeezed, and non-squeezed output is above a given threshold, then it's an adversarial example
- Example :
  - In the case of an image, reduce the color data from 24 bits (RGB) to 8 bits (R, G, B or black  white)
  - Blur the image

# Nearest Neighbor

- Return the class of the training set's most similar sample
- Use Euclidean distance (other distances might result in a worse score than without defense)
- The larger your training set is, the longer it will take to compute

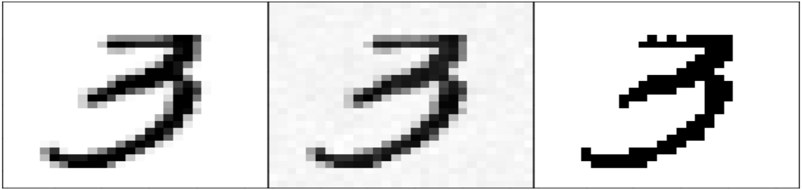
# Adding adversarial examples to the train set

- For each class, consider adding multiple adversarial examples to the training set
- You will need (at least) twice as much data
- Risk of overtraining

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

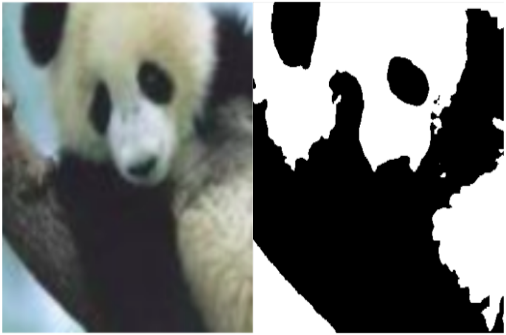
# Use a binary threshold

Aim : Remove the perturbations using a threshold



# Use a binary threshold

Aim : Remove the perturbations using a threshold



# Input reconstruction

Aim : Cleaning the data with an autoencoder

- Deep contractive autoencoder
  - Remove adversary perturbations
  - We use regularization to show what part of the image are important

# RNN-Ensemble

- Use multiples classification models
- Each model can be trained individually
- The output is a combination of the output of each model

# RNN-Ensemble - Model methods

- Regular method
  - Each model is trained with different initial weights
- Adversarial method
  - The models are trained on the same training set
  - We add to this training set, a set of adversarial examples
  - Each set of adversarial example is different from the others
- Subsequence method
  - For a given training set, each model is train on a sequence of this training set

# RNN-Ensemble - Output methods

- For each model, the output is a probability of a class, or a set of probabilities of multiple classes
- Hard voting
  - The global output is the class that has the majority (just like the president elections)
- Soft voting
  - Sum up all probabilities for each class, and the golbal output will be the class with the higher sum

# Why so many methods ?

- Hard to defend against every type of attacks
- Each method has its pros and cons
- At the moment, there's no technique that has over 90-95% succes rate
- Defending against an adaptative attacker is a research field

# Conclusion


Les gens alarmistes qui connaissent rien à l'IA et disent que les robots vont prendre le contrôle de la planète
Mon réseau de neurones artificiels :
Chien
lesjoiesducode.fr

# References


- Generating Natural Adversarial Hyperspectral examples with a modified Wasserstein GAN - *Jean-Christophe Burnel, Kilian Fatras, Nicolas Courty*
- Adversarial example using FGSM - https://www.tensorflow.org/tutorials/generative/adversarial_fgsm
- Adversarial Patch - https://arxiv.org/pdf/1712.09665.pdf
- Adversarial Examples - *Christoph Molnar* -https://christophm.github.io/interpretable-ml-book/adversarial.html
- One Pixel Attack for Fooling Deep Neural Networks - https://arxiv.org/pdf/1710.08864.pdf
- Counterfactual Explanations - *Christoph Molnar* - https://christophm.github.io/interpretable-ml-book/counterfactual.html
- Deep Learning Papers review : Universal Adversarial Patch - https://medium.com/deep-dimension/deep-learning-papers-review-universal-adversarial-patch-a5ad222a62d2
- Adversarial Examples : Attacks and Defenses for Deep Learning - *Xiaoyong Yuan, Pan He, Qile Zhu, Xiaolin Li*
- Tricking Neural Networks : Create your own Adversarial Examples - https://medium.com/@ml.at.berkeley/tricking-neural-networks-create-your-own-adversarial-examples-a61eb7620fd8

# References


- Detecting Adversarial Samples from Artifacts - https://arxiv.org/pdf/1703.00410.pdf
- Defense Methods Against Adversarial Examples for Recurrent Neural Networks - https://arxiv.org/pdf/1901.09963.pdf
- Attacking Machine Learning with Adversarial Examples - https://openai.com/blog/adversarial-example-research/
- Explaining And Harnessing Adversarial Examples - *Ian J. Goodfellow, Jonathon Shlens  Christian Szegedy*