# An Adversarial Approach to Image Classification

Hippolyte Gisserot, Guillaume Kunsch, Benjamin Sykes

# The problem, in short

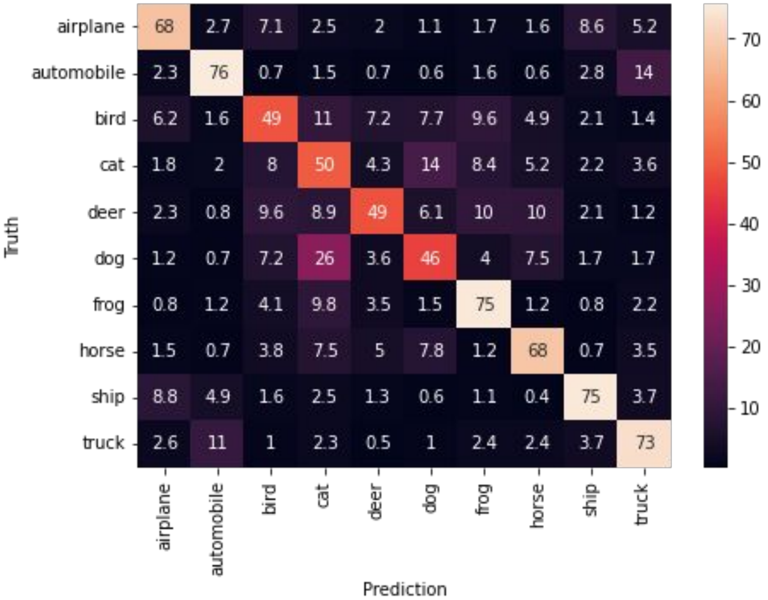
- Challenge:
  - Data set: CIFAR10, pictures belonging to 10 different classes
  - Goal: train a neural network robust to data attacks (notion to be developed in the next slides)
- Plan of attack:
  1. Train a basic model and evaluate its performance
  2. Implement attack mechanisms (FGSM, PGD) and evaluate model performance
  3. Implement defense mechanisms and compare performance on natural vs. attacked images

64 random CIFAR10 images

# Basic model training

- Architecture:
  - Conv2D + MaxPooling + Conv2D + 3*FC
  - Hidden layer activation: relu
  - Output layer activation: log_softmax
  - Loss: negative log-likelihood

- Performance:
  - **Network accuracy: 62.8%**
  - The model gets it right on more than 60% of the test data, but what if we intentionally try to fool it?

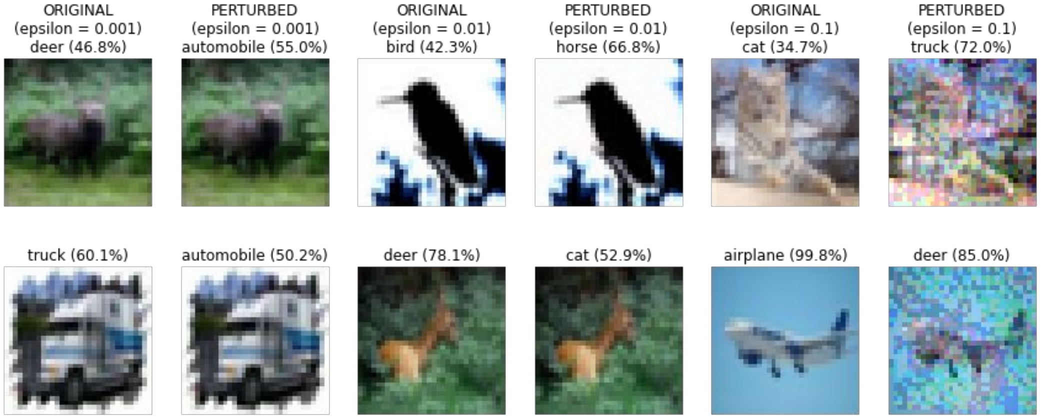



Confusion matrix on test data
(each row sums to 100%)

# FGSM attack: principle and implementation

- Principle:
  - For each image, perturb it as much as possible within a certain limit (epsilon-bound)
  - Mathematical formulation: $\arg\max_{||\delta||\leq\epsilon} l_f(x+\delta, y) \approx \arg\max_{||\delta||\leq\epsilon} \delta^T \nabla_x l_f(x,y)$

$$\approx \epsilon \mathrm{sign}(\nabla_x l_f(x,y)) \; (||\cdot|| = ||\cdot||_\infty)$$




Original vs. perturbed images for different values of epsilon

| **Epsilon** | **Accuracy** |
|---|---|
| 0 | 62.8% |
| 0.001 | 56.4% |
| 0.01 | 20.1% |
| 0.1 | 0.1% |

Accuracy vs. epsilon

# PGA attack: principle and implementation

- Principle:
  - Iterated version of FGSM, more precise
  - Mathematical formulation: $x_0 = x,\ x_{t+1} = \Pi_{B(x_0,\epsilon)}(x_t + \delta \mathrm{sign}(\nabla_x l_f(x,y)))$
  - Main parameters: 30 iterations, $\delta = \frac{\epsilon}{4}$

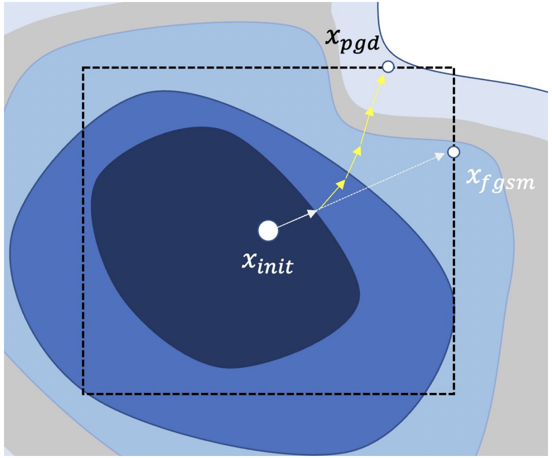


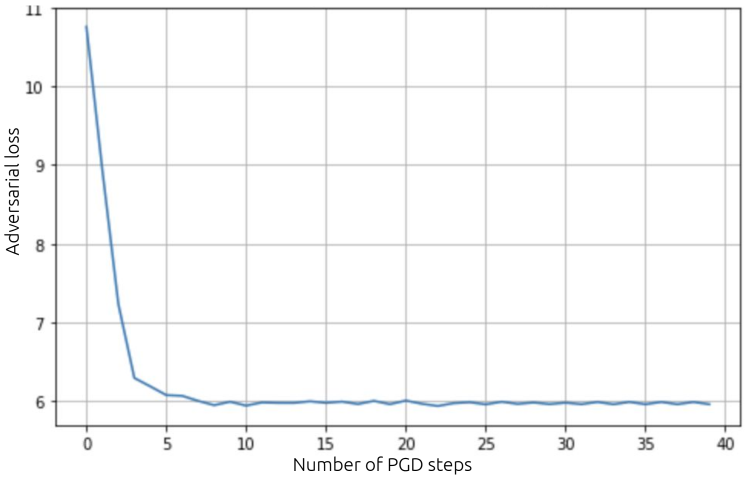

Illustration of the projected gradient process

| Epsilon | Accuracy |
|---|---|
| 0 | 62.8% |
| 0.001 | 56.4% |
| 0.01 | 12.6% |
| 0.1 | 0% |

Accuracy vs. epsilon

# Performances on targeted attacks

- Principle:
  - PGA becomes PGD $x_0 = x, x_{t+1} = \Pi_{B(x_o,\epsilon)}(x_t - \delta.\text{sign}(\nabla_x l_f(x, y_{\text{target}})))$
  - Main parameters: 30 iterations, $\delta = \frac{\epsilon}{4}$




| Epsilon | FGSM | PGD |
|---|---|---|
| 0.001 | 8% | 8% |
| 0.01 | 23% | 33% |
| 0.1 | 59% | 100% |

Rate of successful targeted attacks (target = deer)

# Defense implementation: FGSM case

- Principle:
  - Modify the loss function to incorporate perturbed data:
    $$l'_f(x, y) = \alpha l_f(x, y) + (1-\alpha) l_f(p(x), y)$$
  - In the FGSM case:
    $$l'_f(x, y) = \alpha l_f(x, y) + (1-\alpha) l_f(x + \epsilon \text{sign}(\nabla_x l_f(x, y)), y)$$
- In practice:
  - $\alpha = 0$
  - Equivalent to training the model on a fully perturbed data set

| **Epsilon** | **Natural images** | **Perturbed images** |
|---|---|---|
| 0 | 62.8% | 20.1% |
| 0.001 | 62.3% | 60.1% |
| 0.01 | 54.0% | 81.8% |
| 0.1 | 24.9% | 96.0% |

Accuracies of classical/robust models for natural/perturbed images

# Conclusion: next steps

- PGD defense mechanism
- Loss function:
  - Find the right balance between training on natural and perturbed images ($\alpha \neq 0$)
  - Incorporate different perturbation frameworks to the loss function:

$$l'_f(x,y) = \alpha_0 l_f(x,y) + \alpha_1 l_f(p_1(x),y) + \ \dots \ + \alpha_n l_f(p_n(x),y)$$

- Innovate on defense mechanisms
  - Enforce Lipschitz continuity to the classifier
  - Use a VAE to perform the classification task on a reconstructed image set

# Thank you for your attention!