Neural Compression Restoration Against Black-box Adversarial Attacks

Lorenzo Agnolucci

Supervisor: Ing. Federico Becattini

Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Firenze

Index

Introduction

Introduction

- Due to the importance of neural networks classifiers in several real-world applications, in the last years the study of adversarial attacks and defense mechanisms against them has gained increasing attention

- In this work the defense strategy proposed in [Becattini et al.,], which achieved excellent results against white-box attacks, is evaluated against black-box attacks

- In particular, we consider Hop Skip Jump [Chen and Jordan, 2019] and Square [Andriushchenko et al., 2019] attacks

- The attacks are performed with a limited budget of queries and at various thresholds of l_2 perturbation error

Introduction
Adversarial attacks

- Adversarial examples are slightly perturbed versions of the original examples that manage to cause models to make wrong predictions but are almost identical to the original samples for the human eye

- A classifier can be denoted as a function $\mathcal{C}(x) : \mathcal{X} \rightarrow \mathcal{Y}$, where $x \in \mathcal{X}$ is the input image that is mapped to a class label $y \in \mathcal{Y} = \{1, 2, \ldots, C\}$, with C number of classes.

- Let y^* denote the ground-truth label of the clean input x, and x_a denote an adversarial example of x. An adversarial attack can be:

  1. untargeted: aims to cause a misclassification, $\mathcal{C}(x_a) \neq y^*$
  2. targeted: tries to achieve $\mathcal{C}(x_a) = \tilde{y}$, with $\tilde{y}$ given label and $\tilde{y} \neq y^*$

BLACK-BOX ATTACKS

Black-box attacks

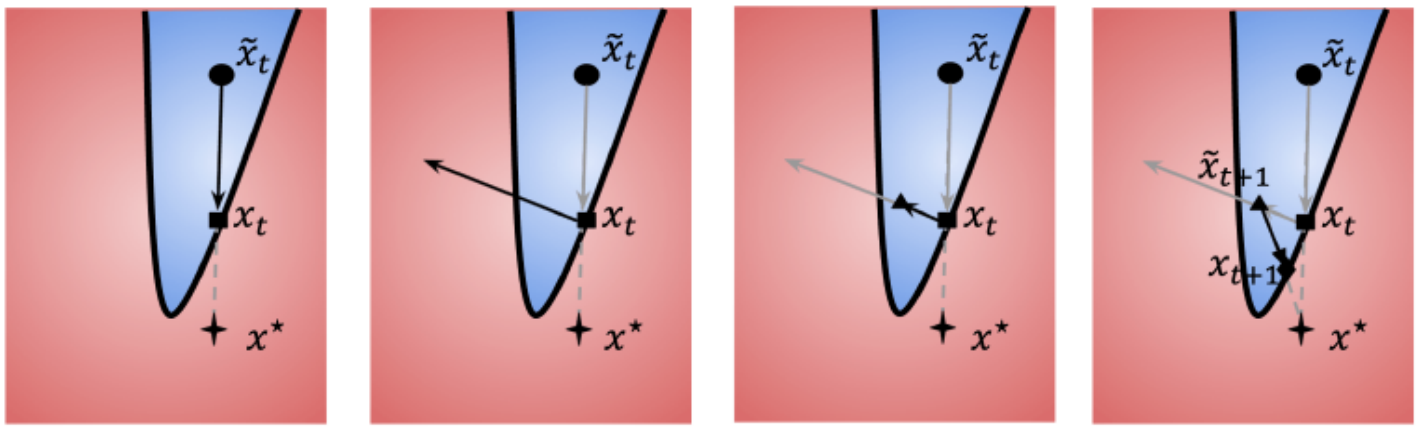
- Black-box attacks have not any knowledge on the model architecture or parameters and use various strategies:
  1. transfer-based attacks: generate adversarial examples on a previously trained surrogate model
  2. score-based attacks: can only acquire the output probabilities by querying the target model
  3. decision-based attacks: solely rely on the predicted classes of the queries
- Score-based and decision-based attacks usually try to approximate gradients to generate adversarial examples
- It is more realistic to evaluate the vulnerability of a machine learning system with a limited budget of model queries. Indeed online image classification platforms often set a limit on the allowed number of queries within a certain time period

Hop Skip Jump

- It is an iterative decision-based attack that tries to solve:

$$x_a = \underset{x' : x' \text{ is adversarial}}{\text{argmin}} \|x' - x\|_p$$

- It is initialized with a sample blended with uniform noise that is misclassified. Each iteration has three components:
  1. the result of last iteration is pushed towards the boundary between successful and unsuccessful perturbed images via a binary search
  2. the gradient direction is estimated via the Monte Carlo method
  3. the step size along the gradient direction is decreased via geometric progression until perturbation becomes successful

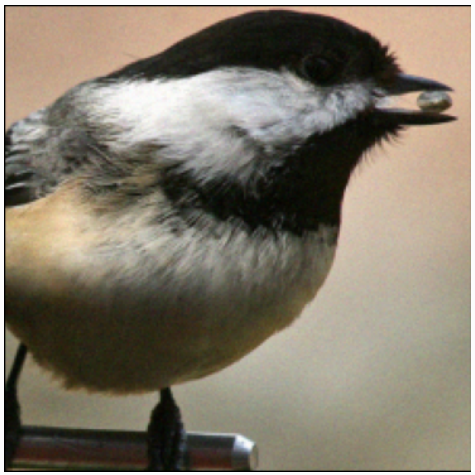


Hop Skip Jump

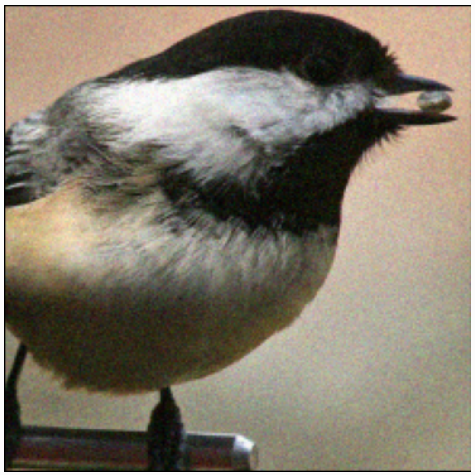
Examples of application

No attack

$l_2 = 0.01$



$l_2 = 0.03$



$l_2 = 0.07$

Square

- It is a score-based attack that tries to solve:

$$x_a = \underset{x' : \|x' - x\|_p \leq \epsilon}{\text{argmin}} \mathcal{J}(x', y)$$

  with $\mathcal{J}$ loss function

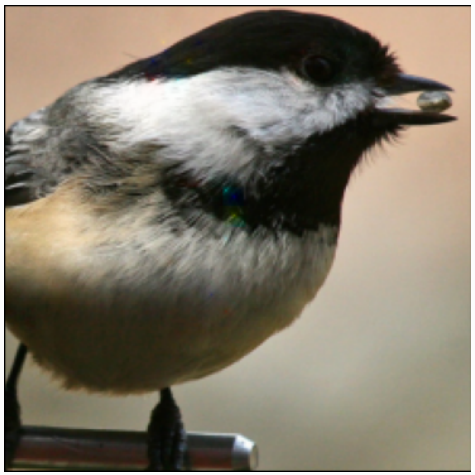
- It does not rely on local gradient information

- It is based on a randomized search scheme which selects localized square-shaped updates at random positions so that at each iteration the perturbation is situated approximately at the boundary of the feasible set (the l_p-ball $\{x_a : \|x_a - x\|_p \leq \epsilon\}$)

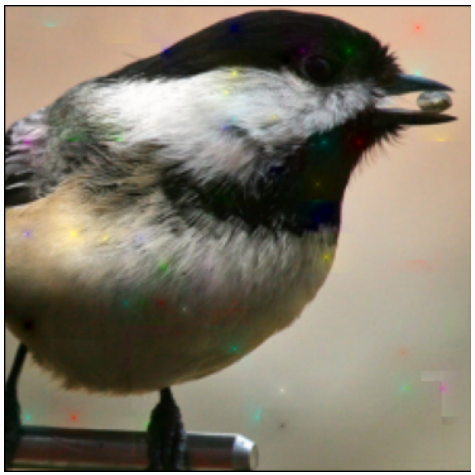
- The main idea is to sample a random update δ at each iteration according to a particular distribution, and to add this update to the current iterate x_a if it achieves an improvement in terms of $\mathcal{J}$

Square
Examples of application

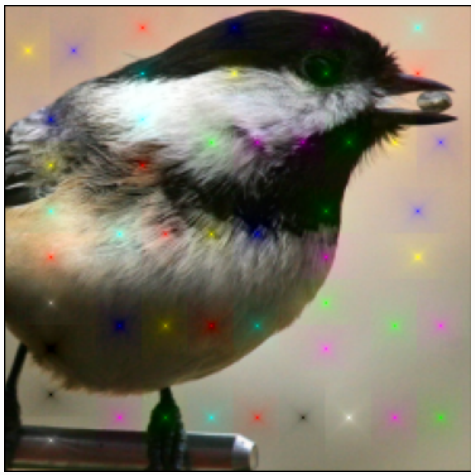
No attack



$l_2 = 0.01$



$l_2 = 0.03$



$l_2 = 0.07$

Defense strategy

Defense strategy

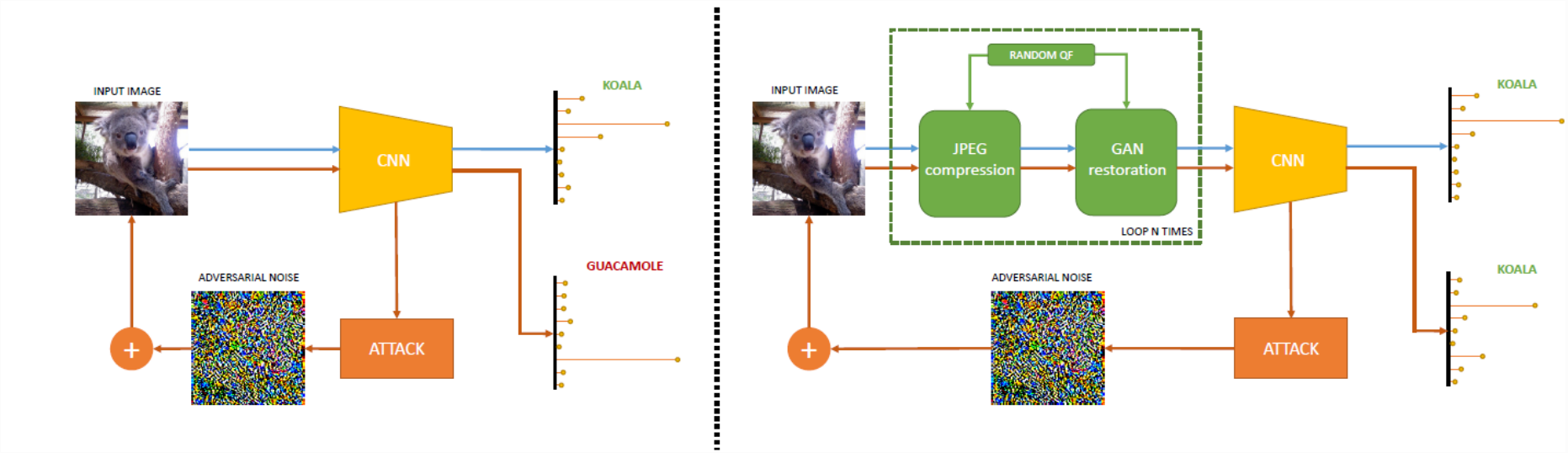
- The approach proposed in [Becattini et al.,] is based on JPEG compression and generative restoration models
- JPEG compression is a non-differentiable input transformation that reduces adversarial perturbations but degrades the image and makes the classification worse
- To counter this effect, restoration GAN models previously trained at different quality factors are used to restore image quality
- This solution has two positive sides:
  1. the quality factor can be chosen randomly
  2. the probability of correctly classifying compressed and restored (non attacked) images is higher
- Also it is possible to iterate the compression and restoration steps N times without degrading the image but increasing the complexity for the attacker

Defense strategy

Pipeline

The pipeline is the following:

1. Apply JPEG compression to the input image with a random quality factor (QF)
2. Choose the GAN restoration model trained with the QF closest to the one chosen randomly at the previous step
3. Repeat the process for a given number of iterations



Experimental results

Dataset

- The dataset is based on the ILSVRC validation set, which has 1000 classes and $224 \times 224 \times 3$ images.
- One image for each class is selected such that the chosen classifier achieves 100% accuracy on non-attacked images
- Using already misclassified images to evaluate the defense would be useless, because the attack would be successful by definition.

Evaluation metrics

- The performance of the defense is evaluated with two metrics:
  1. degradation: quantifies the effect of the defense on the classification accuracy of non-attacked images
  2. accuracy vs perturbation budget: measures the robustness of the defense strategy increasing the degradation applied to the image by the adversarial attack
- The amplitude of the perturbation is measured with the normalized dissimilarity $l_2(x, x_a) = \frac{\|x - x_a\|_2}{\|x\|_2}$, with x and x_a respectively the original and the adversarial image.
- The common alternative l_∞ is considered more strict for the attacker on large images such as ImageNet ones
- Indeed l_2 represents an average measurement of the error over the whole image, while the l_∞ provides the maximum error across all the pixels

Test details

- The attacks were executed with the implementations of the *Adversarial Robustness Toolbox (ART)* library
- The metrics are evaluated for three defense strategies: no defense, simple JPEG compression with fixed 40 QF and the approach proposed in [Becattini et al.,]
- All the attacks performed in this work were untargeted and had a limited budget of 5k queries, chosen considering the limited computational resources available
- Regarding the defense strategy, the tests were carried out with ResNet50 as the architecture, random QF in the range [20, 60], GANs trained with QF of 20, 40 and 60 and 3 iterations of the defense pipeline

Results
Hop Skip Jump

- The original implementation of the algorithm did not include a perturbation budget, so it was modified to stop after reaching the maximum number of queries or after finding an adversarial example with a l_2 perturbation smaller than the budget
- The attack is considered successful if it achieves a perturbation error lower than the considered perturbation threshold
- The chosen budget parameter was *0.01* because in this way for each image we can obtain the lowest possible error with the given query limit

Defense	No attack	$l_2 = 0.01$	$l_2 = 0.03$	$l_2 = 0.05$	$l_2 = 0.06$	$l_2 = 0.07$	$l_2 = 0.08$
No defense	**100**	20.9	0.4	0.1	0.1	0.1	0.1
JPEG 40	99.4	97.8	94.2	86.2	80.9	74.4	68.7
[Becattini et al.,] approach	98.8	**98.6**	**96.5**	**94.4**	**93.5**	**92.6**	**91.1**

Results
Square

- Since Square attack outputs an adversarial example with constrained perturbation, the original implementation did not need to be modified to include a perturbation budget
- The attack is considered successful if it finds an adversarial example before reaching the query limit. By definition of the algorithm this example will have a perturbation error lower than the budget
- Starting from a perturbation threshold equal to *0.08*, we lowered it only for the images for which the attack was successful

Defense	No attack	$l_2 = 0.01$	$l_2 = 0.03$	$l_2 = 0.05$	$l_2 = 0.06$	$l_2 = 0.07$	$l_2 = 0.08$
No defense	**100**	96.3	63.6	19.8	10.0	5.3	2.1
JPEG 40	99.4	97.0	73.6	38.6	25.8	17.0	8.9
[Becattini et al.,] approach	98.8	**98.0**	**95.8**	**91.2**	**87.6**	**84.3**	**78.6**

Conclusions

Conclusions

- In this work the defense strategy towards adversarial examples proposed in [Becattini et al.,] was evaluated against black-box attacks
- In particular we considered the Hop Skip Jump and the Square attack
- The defense mechanism proved to be robust and effective even for high perturbation budgets that make the distortion easily recognizable
- This fact make the studied approach a strong candidate for security-sensitive real-world applications
- As a future work, the defense strategy could be evaluated against other black-box attacks

References

Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. (2019).
Square attack: a query-efficient black-box adversarial attack via random search.
CoRR, abs/1912.00049.

Becattini, F., Ferrari, C., and Galteri, L.
Neural compression restoration against gradient-based adversarial attacks.

Chen, J. and Jordan, M. (2019).
HopSkipJumpAttack: A Query-Efficient Decision-Based Adversarial Attack.
CoRR, abs/1904.02144.