Black-box adversarial attacks using Evolution Strategies

Hao Qiu
University of Trento
Trento, Italy
qiuhaosai@gmail.com

Leonardo Lucio Custode
University of Trento
Trento, Italy
leonardo.custode@unitn.it

Giovanni Iacca
University of Trento
Trento, Italy
giovanni.iacca@unitn.it

ABSTRACT

In the last decade, deep neural networks have proven to be very powerful in computer vision tasks, starting a revolution in the computer vision and machine learning fields. However, deep neural networks, usually, are not robust to perturbations of the input data. In fact, several studies showed that slightly changing the content of the images can cause a dramatic decrease in the accuracy of the attacked neural network. Several methods able to generate adversarial samples make use of gradients, which usually are not available to an attacker in real-world scenarios. As opposed to this class of attacks, another class of adversarial attacks, called black-box adversarial attacks, emerged, which does not make use of information on the gradients, being more suitable for real-world attack scenarios. In this work, we compare three well-known evolution strategies on the generation of black-box adversarial attacks for image classification tasks. While our results show that the attacked neural networks can be, in most cases, easily fooled by all the algorithms under comparison, they also show that some black-box optimization algorithms may be better in "harder" setups, both in terms of attack success rate and efficiency (i.e., number of queries).

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; **Computer vision**; **Search methodologies**.

KEYWORDS

Adversarial attacks, evolution strategies, CMA-ES, neural networks

ACM Reference Format:
Hao Qiu, Leonardo Lucio Custode, and Giovanni Iacca. 2021. Black-box adversarial attacks using Evolution Strategies. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3449726.3463137

1 INTRODUCTION

The field of computer vision, in the last decade, had an impressive progress that enabled applications such as autonomous driving, medical applications and identification. All of these progresses are due to the capabilities of deep artificial neural networks in processing raw data such as images. While deep neural networks are able to recognize, with good accuracy, objects in an image, they usually suffer under adversarial attacks. An adversarial attack is an image δ crafted in such a way that, given a correctly-classified image x, $x + \delta$ is misclassified.

There are two main classes of adversarial attacks: *white-box* adversarial attacks and *black-box* adversarial attacks. White-box adversarial attacks can be seen as a simplified setting. In fact, in this case, the adversary has full access to the neural network, and thus she can compute gradients on the classified samples. This allows to find adversarial attacks by simply moving the image towards the direction that maximizes the gradient. On the other hand, black-box adversarial attacks are more similar to real settings. In fact, in this case the attacker has no access to the gradients. Instead, she can only access the prediction or the output probabilities given by the model. In this cases, gradient-based methods cannot be used. Evolutionary computation, since it does not rely on the computation of the gradients, is therefore an appropriate tool for this task.

While previous works on black-box adversarial attacks by means of evolutionary algorithms focused on solving the problem of generating adversarial attacks in black-box settings [2, 8], no work performed a comparison of various black-box optimization methods for these settings. In fact, since in general different evolutionary algorithms can perform very differently on the same problem, choosing which algorithm should be used for black-box adversarial attacks is difficult. Thus, comparing various algorithms on this task can be of great interest in practical applications.

In order to assess how different evolutionary algorithms perform on the generation of adversarial examples for deep neural networks in black-box settings, in this work we compare three different evolution strategies (ES), namely: (1+1)-ES [24], Natural Evolution Strategies [34] and the original version of CMA-ES [13, 14]. We decided to focus our investigation on these three variants of ES for three main reasons. First of all, as shown in [25] and further discussed in [29], ES can rival backpropagation-based algorithms in deep reinforcement learning (RL) problems and as such it has recently attracted research attention also in the deep learning community. Secondly, ES can be essentially considered a gradient-based algorithm, since it performs a stochastic gradient descent based on a finite-difference approximation of the gradient [29]. As such, it is worth investigating how ES can deal with a task such as the generation of adversarial examples that is typically tackled by (explicit) gradient-based methods. Lastly, we chose for our analysis two ES variants configured as population-less (i.e., handling one solution at a time), and CMA-ES, which is configured as a population-based algorithm and is considered nowadays the state-of-the-art in evolutionary optimization. Thus, we aim at finding if using a population rather than a single solution can provide a benefit on the task at hand.

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07…$15.00
https://doi.org/10.1145/3449726.3463137

The rest of the paper is structured as follows. In the next section, we make a short overview of the recent developments in the field of adversarial attacks. In Section 3, we describe the details of the methods which are common to the three evolutionary algorithms. Then, in Section 4 we present the experimental results and discuss the performance of the three algorithms. Finally, in Section 5 we present the conclusions of this work.

2 RELATED WORK

In the last years, the field of adversarial attacks has seen a number of important developments.

Szegedy et al. [31] discovered that, while neural networks were able to reach good performance in image classification tasks, small perturbations of the input could cause them to misclassify samples that would be otherwise be classified correctly. Successively, Goodfellow et al. [11] proposed an efficient gradient-based method to generate adversarial attacks. In [22], the authors addressed the generation of adversarial attacks in limited L^0-"norm" scenarios, i.e., crafting adversarial samples by modifying only few pixels of the image. Carlini and Wagner [7] proposed a method to generate adversarial attacks that, by constraining the perturbation according to several metrics, were able to fool even neural networks that were trained on adversarial examples. In [19], the authors presented a method that generates adversarial examples by iteratively pushing the images outside the correct decision region. Baluja and Fischer [3] trained a neural network to generate adversarial examples for other neural networks.

All the approaches described before are designed for white-box settings, i.e., they require the attacker to have access to the model. Other approaches, on the other hand, focused on black-box settings, i.e., settings in which the attacker knows only the logits of the prediction.

Su et al. [28] proposed an approach based on the differential evolution algorithm [27] to evolve adversarial examples that involve a single pixel in an image. In [9], the authors defined a special-purpose loss function and used it to estimate the gradients. Ilyas et al. [16] proposed a variant of the natural evolution strategies [34] to generate adversarial attacks. In [10], the authors introduced a method to generate black-box adversarial attacks even in stricter cases, i.e., when the attacker has access only to the predicted class. Bhagoji et al. [4] proposed a loss function based on logits to estimate the gradients. Furthermore, they presented a query reduction technique to reduce the number of queries to obtain a successful attack. Narodytska et al. [20] proposed a greedy algorithm for the generation of black-box adversarial attacks. In [21], the authors made use of local search to generate black-box adversarial attacks. Brendel et al. [6] proposed an algorithm to produce adversarial attacks based on an initial random image that is moved near the image that has to be attacked, leading to an adversarial attack that lies near the original image while being in a different decision region. In [2, 8] the authors proposed a genetic algorithm to evolve adversarial perturbations. Li et al. [18] proposed a method to learn the probability density distribution of adversarial attacks. In [12] the authors generated adversarial samples by generating noise in a direction obtained by analyzing orthonormal vectors of the image vector space.

Finally, some works [17, 35] have shown that it is possible to perform adversarial attacks even in real-world scenarios with physical objects. For a more thorough review of the state of the art, the reader can refer to [1, 5, 23, 36].

3 METHODS

In this work, we compare three different evolution strategies to generate adversarial samples in black-box settings. The three algorithms we use are: (1+1)-ES [24], Natural Evolution Strategies [34] and CMA-ES [13, 14]. We omit, for brevity, the description of the three evolution strategies variants: for that, we refer the reader to the original papers.

The scheme of the process applied in our experiments to produce adversarial samples by means of these three evolutionary algorithms is shown in Figure 1.

In the following, we describe the method used to generate the samples with the three evolutionary algorithms.

3.1 Individual encoding

Each individual is encoded as a $H \times W \times 3$ tensor, where H and W are the height and the width of the adversarial sample that is then upsampled to match the size of the image to attack (see Section 3.3), and 3 is the number of channels.

We constrain the L^∞-norm of the generated adversarial perturbations. To do so, we clip the values of the perturbations in $[-\varepsilon, \varepsilon]$.

3.2 Fitness evaluation

In the fitness evaluation phase, we compute a perturbation of the current image $x' = x + \delta$ and we use it to compute the goodness of the perturbation. We refer to each time we assess the goodness of a perturbation of an adversarial attack as a *query*.

3.2.1 Untargeted attacks. The fitness function used to assess the quality of an adversarial attack corresponds to the cross-entropy loss related to that sample with respect to the true label (i):

$$F_u(x) = \mathcal{L}(x, y_i | y_i = 1) = -log(f_\theta^{(i)}(x))$$

where $f_\theta^{(i)}(x)$ is the i-th output of the neural network parametrized by the parameters θ, given the input x. In this case, instead of minimizing the cross-entropy loss (as done during training), we want to maximize it.

3.2.2 Targeted attacks. In this case, the fitness function used is:

$$F_t(x) = -\mathcal{L}(x, y_t | y_t = 1) = log(f_\theta^{(t)}(x))$$

where t refers to the *target* class. Note that, in case of targeted attacks, we do not care about the correct class of the sample, but we only want to minimize the loss w.r.t. the target class.

When assessing the quality of an image, if the attack has success, the evolutionary process is terminated.

3.3 Dimensionality reduction

In order to reduce the computational burden of the optimization process, we evolve adversarial samples that are smaller than the image to attack, and then we increase their size by means of nearest neighbor interpolation. This means that, given an image I and a

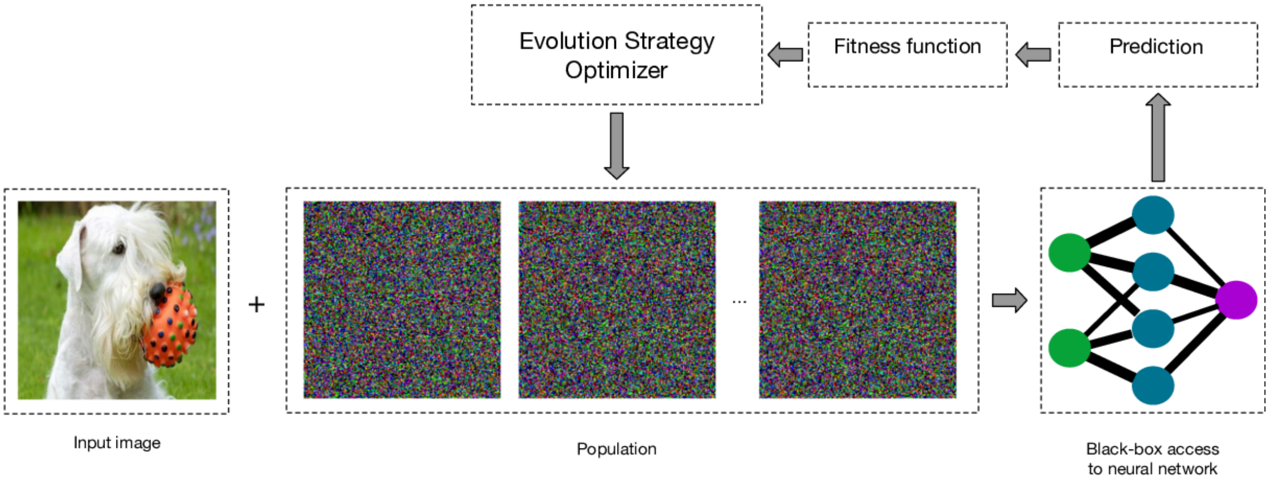
Evolution Strategy Optimizer
Fitness function
Prediction
+
...
Input image
Population
Black-box access to neural network

Figure 1: Optimization process used in our experiments to generate adversarial attacks.

scaling factor $s \geq 1$, we build another image I' such that:

$$I'(j \cdot s + k, i \cdot s + l) = I(j, i);\ \forall i, j;\ k, l \in [0, s[$$

An example of upsampling using the nearest neighbor interpolation is shown in Figure 2.

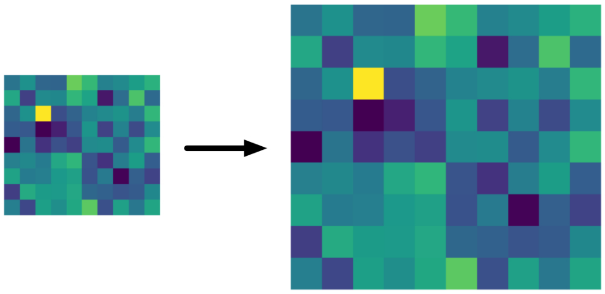

Figure 2: Example of upsampling using the nearest neighbor interpolation.

4 EXPERIMENTAL RESULTS

In this section, we present the experimental setting and the results obtained.

4.1 Experimental setting

4.1.1 Dataset. We test the three evolutionary algorithms on neural networks that have been trained on the ImageNet dataset, that is a huge database composed of more than 14 million images distributed over more than $2 \cdot 10^4$ classes. Each image is made of 224×224 pixels, encoded as a tensor in the RGB color space.

4.1.2 Target neural networks. Besides testing the three evolutionary algorithms on the ImageNet dataset, we also test them on three different neural networks, to better understand their performance in different scenarios. The target neural networks are: VGG-16 [26], Inception-v3 [30] and ResNet-50 [15].

4.1.3 Individual encoding. As discussed earlier, the individuals are encoded as tensors of size $32 \times 32 \times 3$, that are then upsampled to $224 \times 224 \times 3$ tensors and added to the original image.

4.1.4 Algorithm parametrization. The parameters for the (1+1)-ES, NES and CMA-ES are shown in Table 1.

Algorithm	Parameter	Value
(1+1)-ES	Initialization strategy	$\mathcal{N}(0,1)$
	Adaptation rule	1/5 rule
	Generations	10000
NES	Initialization strategy	$\mathcal{N}(0,1)$
	Population size	1
	Step size	1
	Learning rate	0.05
	Generations	10000
CMA-ES	Initialization strategy	$\mathcal{N}(0,1)$
	Population size	25
	Generations	400

Table 1: Parameters used for the evolutionary algorithms employed in our experiments.

4.2 Experimental results

Table 2 shows the results obtained by using the three evolutionary algorithms in the untargeted setting on the three neural networks. The maximum number of queries (for all the experiments) is set to 10^4. We set the maximum perturbation strength to $\varepsilon = 0.05$.

Figure 3 graphically shows the attack success rate and the mean number of queries of the three algorithms on the three tested networks. It can be seen that CMA-ES "dominates" (as in Pareto domination) the other two algorithms in all the tested cases.

From the data shown in Table 2 and Figure 3, we can observe that Inception-v3 seems to be more robust to the type of adversarial

Neural Network	Algorithm	Success rate (%)	Mean queries	Median queries
VGG-16	(1+1)-ES	80.24	2006.00 (37.80)	3.00
	NES	94.86	828.90 (331.80)	3.00
	CMA-ES	100.00	155.80 (155.80)	3.00
Inception-v3	(1+1)-ES	69.52	3126.90 (114.00)	32.00
	NES	86.94	1765.50 (528.40)	71.00
	CMA-ES	98.91	451.20 (346.10)	53.00
ResNet-50	(1+1)-ES	70.68	2994.30 (88.50)	17.00
	NES	89.83	1496.10 (532.90)	19.00
	CMA-ES	99.87	294.20 (281.2)	21.00

Table 2: Comparison of the three evolutionary algorithms with a maximum perturbation strength $\varepsilon = 0.05$ on the three neural networks pretrained on ImageNet. The values in parentheses represent the mean number of queries computed only on the successful attacks.

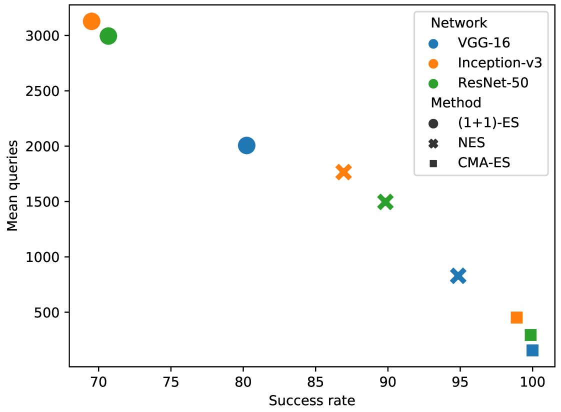

Figure 3: Performance of the three evolutionary algorithms in terms of success rate and mean number of queries required while attacking the three neural networks ($\varepsilon = 0.05$).

images produced in this work than VGG-16 and ResNet-50. Moreover, we can observe that CMA-ES largely outperforms the other two algorithms on all the three networks, both in terms of success rate and mean number of queries. This might be due to the use of an actual population in CMA-ES (as opposed to the single-solution approach of the other two methods), which might favor a better exploration of the fitness landscape. However, further investigations will be needed to confirm this hypothesis.

On the other hand, it is interesting to note that a simple approach like (1+1)-ES consistently requires a smaller number of median queries. This, and the fact that the mean number of queries in case of success is smaller than the one of CMA-ES, suggest that (1+1)-ES may be a good approach in certain cases (e.g. when the needed success rate is not too high, or when one can use only a limited amount of computing resources). Figures 4, 5, 6 show the distribution of the queries performed while attacking the ResNet50 network for the (1+1)-ES, NES and CMA-ES, respectively.

To better understand how the maximum magnitude of the perturbation affects the adversarial attacks, we performed an additional test on the ResNet-50 using different perturbation strengths. We performed this test on ResNet-50 because (from the results shown above) in terms of robustness to the attacks this network can be seen as an average case between VGG-16 and Inception-v3. To do so, we tested the three evolutionary algorithms under varying perturbation strengths (L^{∞}-norm) from 0.01 to 0.09, with steps of 0.02. The results are shown in Table 3.

As we can see from the table, even with small perturbations CMA-ES is able to obtain a very good success rate. On the other hand, if the maximum perturbation is small, (1+1)-ES and NES are not able to achieve a satisfactory success rate.

Furthermore, higher perturbations ($\varepsilon \in [0.05, 0.09]$) show that even random noise is sufficient to obtain adversarial samples. In fact, the median number of queries is so low that it means that even individuals in the initial generations (i.e., randomly generated) are able to fool the target neural network. Nevertheless, also in this case CMA-ES proves to be superior to the other two evolutionary algorithms. In fact, while all the algorithms obtain satisfactory success rates in (almost) all the cases, CMA-ES is the only one able to achieve a success rate of 100%.

Finally, we tested the three algorithms on the generation of *targeted* attacks for the ResNet-50 network. The results of these experiments are presented in Table 4. The success rates under different L^{∞}-norms are shown graphically in Figure 7, while the mean number of queries is shown graphically in Figure 8.

Also in this case, we can observe that the performance of CMA-ES is significantly better than (1+1)-ES and NES. Moreover, we can observe that in this setting the performance of (1+1)-ES seems to be comparable to that of NES.

5 CONCLUSIONS

Despite being very powerful in computer vision tasks, deep neural networks may be brittle to adversarial attacks. Adversarial attacks may be easily carried out when the attacker has access to the gradients of the attacked neural network. However, in most real-world attack scenarios the attacker does not have access to the gradients, so it must rely on black-box methodologies, such as those offered by evolutionary computation.

In this paper, we compared three different evolution strategies for the generation of black-box untargeted and targeted adversarial attacks: (1+1)-ES, Natural Evolution Strategies and CMA-ES.

L^∞-norm	Algorithm	Success rate (%)	Mean queries	Median queries
0.01	(1+1)-ES	22.13	8009.60 (1007.80)	10^4
	NES	35.70	6935.50 (1417.20)	10^4
	CMA-ES	83.53	3218.80 (1881.80)	1276.00
0.03	(1+1)-ES	48.61	5245.10 (219.60)	10^4
	NES	71.01	3522.50 (878.20)	687.00
	CMA-ES	99.60	701.00 (664.10)	335.00
0.05	(1+1)-ES	70.68	2994.30 (88.50)	17.00
	NES	89.83	1496.10 (532.90)	19.00
	CMA-ES	99.87	294.20 (281.20)	21.00
0.07	(1+1)-ES	86.42	1416.60 (68.90)	4.00
	NES	96.57	614.40 (281.40)	3.00
	CMA-ES	100.00	125.40 (125.40)	3.00
0.09	(1+1)-ES	95.25	494.90 (21.60)	2.00
	NES	98.94	250.40 (146.50)	1.00
	CMA-ES	100.00	59.70 (59.70)	1.00

Table 3: Comparison of the three evolutionary algorithms on the generation of adversarial attacks for ResNet-50 under different maximum perturbation strengths. The values in parentheses represent the mean number of queries computed only on the successful attacks.

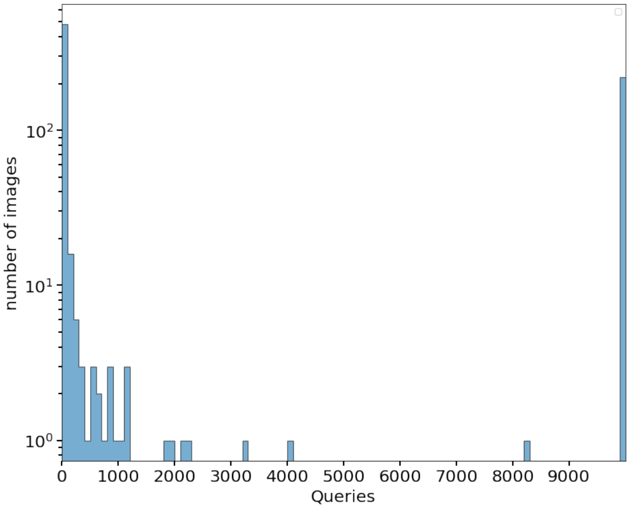

Figure 4: Distribution of the number of queries required to successfully attack the ResNet50 model by using (1+1)-ES.

Algorithm	SR (%)	Mean queries	Median queries
(1+1)-ES	6.22	9640.20 (4220.80)	10^4
NES	4.24	9701.60 (2960.40)	10^4
CMA-ES	77.09	5484.50 (4142.30)	4876.00

Table 4: Comparison of the three evolutionary algorithms on the generation of targeted adversarial attacks for ResNet-50 pretrained on ImageNet. The values in parentheses represent the mean number of queries computed only on the successful attacks.

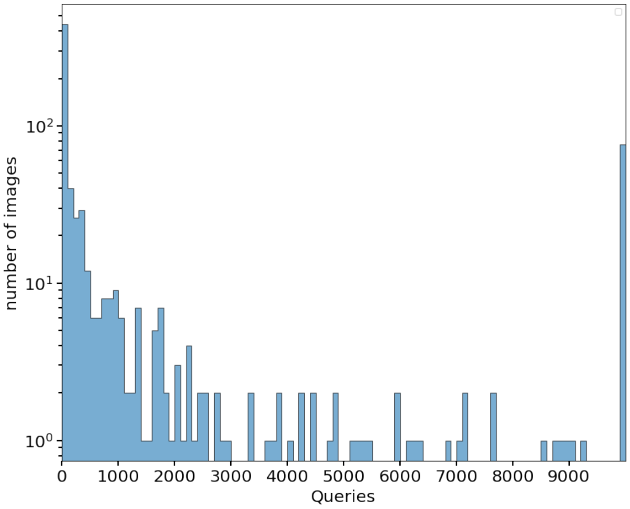

Figure 5: Distribution of the number of queries required to successfully attack the ResNet50 model by using NES.

We tested these algorithms on three well-known neural networks (VGG16, Inception-v3 and ResNet50) on a widely known computer vision benchmark, the ImageNet dataset. The results show that all the algorithms were able to find untargeted adversarial attacks for the high majority of the samples in ImageNet. Moreover, a more focused analysis on ResNet-50 revealed that that only CMA-ES was able to find good adversarial attacks with very small perturbations (intended as L^∞-norm) of the input. Another advantage of CMA-ES is that it required a lower number of mean queries to successfully attack the neural network in all the test cases. These results suggest that CMA-ES is better than the other two algorithms at both exploring and exploiting the landscape of adversarial perturbation under the observed setup. Finally, the experiments on the targeted attacks

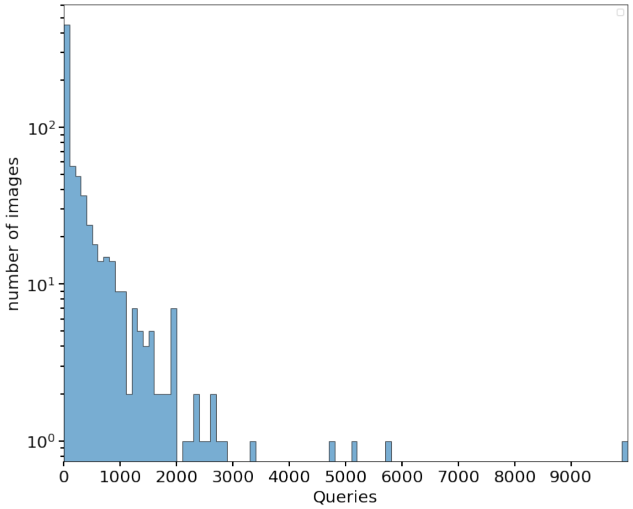

Figure 6: Distribution of the number of queries required to successfully attack the ResNet50 model by using CMA-ES.

for the ResNet50 confirmed the superiority of CMA-ES as a generator of adversarial samples, since it showed a success rate more than an order of magnitude higher than the other two algorithms. In fact, CMA-ES was able to successfully attack more than 3/4 of the images also in this setting, where the other two algorithms were extremely ineffective.

This study did not take into account the generation of adversarial samples for neural networks that included adversarial examples in the training set. Therefore, in future work we plan to benchmark the three evolutionary algorithms on neural networks trained to either recognize or be robust to adversarial examples. Furthermore, we will extend the comparison to other optimization algorithms, in particular more recent variants of CMA-ES such as those presented in [32, 33]. Finally, we will consider the generation of adversarial examples with constraints on other norms (single-objective optimization) or a combination of different norms (multi-objective optimization).

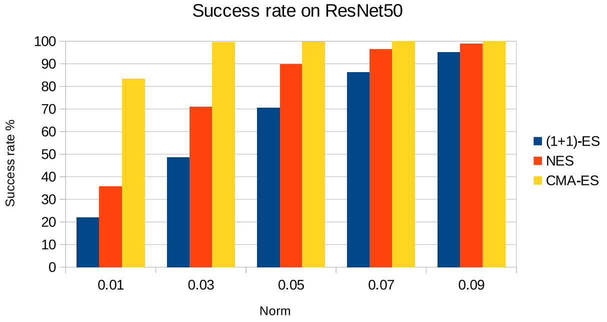

Figure 7: Success rate of the targeted attacks while using the three evolutionary algorithms to attack ResNet50 under different L^∞-norms.

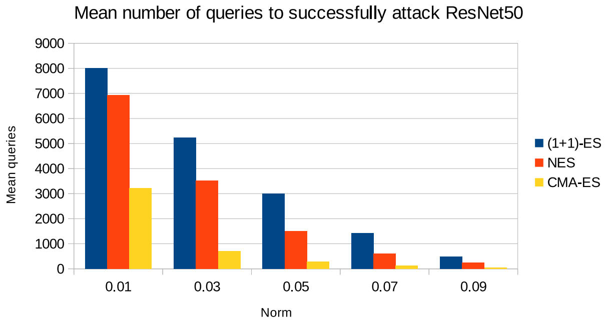

Figure 8: Mean number of queries needed to generate a successful targeted attack while using the three evolutionary algorithms to attack ResNet50 under different L^∞-norms.

REFERENCES

[1] N. Akhtar and A. Mian. 2018. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey. *IEEE Access* 6 (2018), 14410–14430.

[2] Moustafa Alzantot, Yash Sharma, Supriyo Chakraborty, Huan Zhang, Cho-Jui Hsieh, and Mani B. Srivastava. 2019. GenAttack: Practical Black-Box Attacks with Gradient-Free Optimization. In *Genetic and Evolutionary Computation Conference* (Prague, Czech Republic). ACM, New York, NY, USA, 1111–1119.

[3] Shumeet Baluja and Ian Fischer. 2017. Adversarial Transformation Networks: Learning to Generate Adversarial Examples. arXiv:1703.09387 [cs.NE]

[4] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. 2018. Practical Black-Box Attacks on Deep Neural Networks Using Efficient Query Mechanisms. In *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Vol. 11216. Springer International Publishing, Cham, Switzerland, 158–174. Series Title: Lecture Notes in Computer Science.

[5] Siddhant Bhambri, Sumanyu Muku, Avinash Tulasi, and Arun Balaji Buduru. 2020. A Survey of Black-Box Adversarial Attacks on Computer Vision Models. arXiv: 1912.01667.

[6] Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2018. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. arXiv:1712.04248 [stat.ML]

[7] Nicholas Carlini and David Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *Symposium on Security and Privacy*. IEEE, Los Alamitos, CA, USA, 39–57.

[8] Jinyin Chen, Mengmeng Su, Shijing Shen, Hui Xiong, and Haibin Zheng. 2019. POBA-GA: Perturbation optimized black-box adversarial attacks via genetic algorithm. *Computers & Security* 85 (2019), 89–106.

[9] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017. ZOO: Zeroth Order Optimization Based Black-Box Attacks to Deep Neural Networks without Training Substitute Models. In *Workshop on Artificial Intelligence and Security* (Dallas, Texas, USA) *(AISec '17)*. ACM, New York, NY, USA, 15–26.

[10] Minhao Cheng, Simranjit Singh, Patrick Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. 2020. Sign-OPT: A Query-Efficient Hard-label Adversarial Attack. arXiv:1909.10773 [cs.LG]

[11] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. arXiv:1412.6572 [stat.ML]

[12] Chuan Guo, Jacob R. Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Q. Weinberger. 2019. Simple Black-box Adversarial Attacks. arXiv:1905.07121 [cs.LG]

[13] Nikolaus Hansen. 2006. The CMA Evolution Strategy: A Comparing Review.

[14] Nikolaus Hansen and Andreas Ostermeier. 1996. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In *International Conference on Evolutionary Computation*. IEEE, Piscataway, NJ, USA, 312–317.

[15] Kaiming He, Xiangy Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition*. IEEE, Piscataway, NJ, USA, 770–778.

[16] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. 2018. Black-box adversarial attacks with limited queries and information.

[17] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2017. Adversarial examples in the physical world. arXiv:1607.02533 [cs.CV]

[18] Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. 2019. NATTACK: Learning the Distributions of Adversarial Examples for an Improved Black-Box Attack on Deep Neural Networks. arXiv:1905.00441 [cs.LG]

[19] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. DeepFool: a simple and accurate method to fool deep neural networks. , 2574–2582 pages.

[20] Nina Narodytska and Shiva Prasad Kasiviswanathan. 2016. Simple Black-Box Adversarial Perturbations for Deep Networks. arXiv:1612.06299 [cs.LG]

[21] Nina Narodytska and Shiva Prasad Kasiviswanathan. 2017. Simple Black-Box Adversarial Attacks on Deep Neural Networks. In *Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, Piscataway, NJ, USA, 1310–1318.

[22] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. 2016. The limitations of deep learning in adversarial settings. In *Symposium on Security and Privacy*. IEEE, Piscataway, NJ, USA, 372–387.

[23] Shilin Qiu, Qihe Liu, Shijie Zhou, and Chunjiang Wu. 2019. Review of Artificial Intelligence Adversarial Attack and Defense Technologies. *Applied Sciences* 9, 5 (2019), 909.

[24] Ingo Rechenberg. 1994. *Evolutionsstrategie'94*. frommann-holzboog, Stuttgart-Bad Cannstatt, Germany.

[25] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. 2017. Evolution strategies as a scalable alternative to reinforcement learning. arXiv:1703.03864 [stat.ML]

[26] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition.

[27] Rainer Storn and Kenneth Price. 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization* 11, 4 (1997), 341–359.

[28] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. 2019. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation* 23, 5 (2019), 828–841.

[29] Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O Stanley, and Jeff Clune. 2017. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. arXiv:1712.06567 [cs.NE]

[30] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *Conference on Computer Vision and Pattern Recognition*. IEEE, Los Alamitos, CA, USA, 2818–2826.

[31] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. arXiv:1312.6199 [cs.CV]

[32] Konstantinos Varelas, Anne Auger, Dimo Brockhoff, Nikolaus Hansen, Ouassim Ait ElHara, Yann Semet, Rami Kassab, and Frédéric Barbaresco. 2018. A Comparative Study of Large-Scale Variants of CMA-ES. In *Parallel Problem Solving from Nature – PPSN XV*, Anne Auger, Carlos M. Fonseca, Nuno Lourenço, Penousal Machado, Luís Paquete, and Darrell Whitley (Eds.). Springer International Publishing, Cham, 3–15.

[33] Diederick Vermetten, Sander van Rijn, Thomas Bäck, and Carola Doerr. 2019. Online Selection of CMA-ES Variants. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Prague, Czech Republic) *(GECCO '19)*. Association for Computing Machinery, New York, NY, USA, 951–959. https://doi.org/10.1145/3321707.3321803

[34] Daan Wierstra, Tom Schaul, Jan Peters, and Juergen Schmidhuber. 2008. Natural evolution strategies. In *Congress on Evolutionary Computation*. IEEE, Piscataway, NJ, USA, 3381–3387.

[35] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. 2020. Adversarial T-shirt! Evading Person Detectors in A Physical World. arXiv:1910.11099 [cs.CV]

[36] Chaoning Zhang, Philipp Benz, Chenguo Lin, Adil Karjauv, Jing Wu, and In So Kweon. 2021. A Survey On Universal Adversarial Attack. arXiv: 2103.01498.