Fooling Convolutional Neural Network by using Evolution Algorithms

Viet-Nhat Nguyen, Hao Qiu
{vietnhat.nguyen, hao.qiu}@studenti.unitn.it

***Abstract*—Convolution Neural Networks (DNNs) have archive many successes in the on a variety of pattern-recognition tasks. However, these models have been widely known to be not robust with small changes in the input image. The most successful technique to attack models is white-box where the architecture of models is known. This project study and apply some Evolution Algorithms to attack DNNs in scenario of black-box where the only available information is the prediction of target model. We also propose method and suggestions for attacks on different models with different constraints.**

***Index Terms*—Adversarial machine learning, Evolution Algorithms, CMA-ES**

I. Introduction

Let $\mathcal{F}$ be the computational function of target model and x be an input image. A perturbed image $x'$of x is an image which satisfies these following properties: (1) the difference between x and x' are imperceptible or semi-imperceptible to human eyes, which means both x and x' represent the same object and (2) the predictions of target model on two images are different. Formally, x' is consider to be an adversarial image from x if

$$\begin{aligned} &\text{argmax}(\mathcal{F}(x)) \neq \text{argmax}(\mathcal{F}(x')) \\ &\text{such that } \phi(x, x') < \epsilon \end{aligned} \tag{1}$$

where ϕ is a function indicating the difference between two images and ϵ is the threshold of the difference. In the context of white box attack,the attacker has full access to the information of target model, including to structure as well as input normalization method of target model. This is usually easily accomplished by the gradient-based method. The input image will be iteratively amended in such a way that the confidence of the model on the original class decreases over iterations.

$$x'_{t+1} = x'_t + \alpha \frac{\partial \mathcal{F}}{\partial x'_t} \tag{2}$$

In practice, most of the case gradient information is not available since we do not have access to the interior architecture of the target model. One popular solution was proposed is attacking a substituted model which was created by trying to capture the decision function bounder of original target model and hopefully adversarial examples obtained on substituted model can also make original target model misclassify.

However, gradient-based methods usually produce real-number examples while the values of normal images could only be integers. Another drawback is if the data distribution of training set used to built substituted model is much different from the one used to trained the target model, created adversarial examples might have no transferability.

That is when black-box optimization gradient-free methods could be taken advantages to directly attack the target model. There are several works that employed Genetic Algorithm approach and got remarkable results. [1] has successfully generated adversarial handwritten images to fool various kinds of model, by using only classical GA. However their study has not conducted on natural images. [2] used GA to generate adversarial images on natural image dataset but unfortunately they did not take into account the constraint that each pixel value of image must be integer. The functions ϕ used in these works were typically the norm of different vector $\|x - x'\|_2$ or $\|x - x'\|_\infty$.

Recently, there is another work [3] applied Differential Evolution to create adversarial images by only modifying a certain small number of pixels. This work is interesting since it demonstrated that natural images can be misclassified even though these images just are changes one pixel. In this situation, the constraint function ϕ is $\|x - x'\|_0$ indicating number of different pixels between two images.

In our project, we will study about the abilities of Evolution Algorithms on generating adversarial images on convolution neural networks. Our work is divided into two parts. In the first part,we will demonstrate that EAs actually could generate adversarial images on high accuracy models, as suggested in previous literature. We also propose a novel method that allows us to use CMA-ES on this problem in which natural images have thousands of dimension. In this first part, attack methods change all pixel in the image with constraints on the strength of perturbation. In the second part, we will follow the works of [3] and replicate their ideas with other EA algorithms and compare their performances. In this situation, attack methods only modify several pixels of the image but these are no constraint on the strength of perturbation.

II. All Pixels Attack

The way to use EAs to create adversarial images, in this case, is quite straightforward, given that it is an optimization problem on space of pixel values. In this scenario, each candidate of EAs is a vector that has the size equal to the size of images. By defining a fitness function indicating how much a perturbation is good to fool the target model, we can evolve those perturbations to achieve our goal. Typically, when applying GA for this problem, mutation function used is Gaussion mutation.

A. LS-CMA-ES

Covariance matrix adaptation evolution strategy (CMA-ES) is a powerful technique to find the optimal solution in continuous space [4]. In this algorithm, each generation is generated according to a multivariate Gaussian distribution determined by a mean value and a covariance matrix which represents the correlation between variables. It explores search space by iteratively updating the parameters of is Gaussian distribution, based on the fitness values of the previous generation. However, each step of CMA-ES does require the expensive computation of the covariance matrix in the search space. This is one of the biggest drawbacks of CMA-ES when applied to high dimension problems which are typically larger than 1000.

We propose a simple method to overcome this problem by using linear span vectors. Given a set of m vectors $S = \{v_1, v_2, ..., v_m\}$ in n-dimension space, the linear span of these vectors is defined as:

$$\text{span}(S) = \left\{ \sum_{i=1}^{m} \lambda_i v_i \middle| v_i \in S, w_i \in \mathbb{R} \right\} \quad (3)$$

This linear span is a subspace of $\mathbb{R}^n$. When $m = n$ and all vectors in S is linear independent, span(S) will become $\mathbb{R}^n$. If S is fixed, then the linear combination of S is a mapping $\mathbb{R}^m \to \mathbb{R}^n$. This suggests that we can use CMA-ES to search for weights λ in dimension m instead of directly searching for solutions in dimension n with typically $m \ll n$.

B. Attack on ImageNet dataset

1) Settings: ImageNet is one of the largest image datasets about natural images. In this work, we will focus on the classification task on this dataset, which requires to classify 1000 classes of objects appear on each image. The target model used in our experiment is Inception-v3 [5] developed by Google. This model achieves 3.46% top-5 error rate on the test set and is widely considered equal to human level. The input for this model is images resized to $224 \times 224 \times 3$.

Since there are many similar classes on ImageNet, we decided to perform the attack on the top-5 prediction of the target model. That means an attack is considered a success if and only if the prediction of perturbed image did not appear in top-5 predicted classes of the original image. Denote x be the original image and Δ is the added term to create the adversarial image. Let us firstly define the perturbation function:

$$\psi(x, \Delta) = \text{int}(\text{clip}(x + \Delta, 0, 255)) \quad (4)$$

This perturbation function ensures the pixel values of perturbed still are 8-bit integers. Similar to many other works, we use L_∞ constraint and set threshold to be 0.05 in scale of 1, that means $\max |\Delta| < 13$ in scale of 255.

Let $x\prime$ be the perturbed version of x, f is the computational function of the model and τ are classes of top-5 prediction of the target model on x. The fitness function is defined as below:

$$Fitness(x\prime) = \max_{o \notin \tau} \log f_o(x\prime) - \max_{t \in \tau} \log f_t(x\prime) \quad (5)$$

TABLE I
STATISTICS OF ATTACK RESULT ON 100 FIRST IMAGES

Attack Method	Success rate	Mean of query number
Genetic Algorithm	84	106.79
LS-CMA-ES	95	44.7

This fitness function aims to lower the confidences in top-5 prediction on the original image, and our goal is to maximize this fitness. As suggested in [2], we also found that adding log function helps to avoid instability and accelerate the search progress since the confidence of classes which are not in top-5 are usually very small.

2) Results: The baseline method we chose is a simple Genetic Algorithm using α plus λ and tournament selection with toursize = 3. For crossover, we use one point crossover and Gaussian with mean 0 and variance 0.05 for mutation operations. Values of candidates will be normalized to [0, 1] range. We select 100 first images from the test set of ImageNet 2012 challenge dataset to perform attacks. For both algorithms, each generation requires 18 times of fitness function computation. We set a limit of 300 generations for attacking one image, that means if our algorithm cannot find an adversarial image within 4800 queries, the attack will be considered unsuccessful.

Table I shows the success rate and average of query numbers after attacking 100 first images. We can see that although only using 128 base vectors, LS-CMA-ES method is still better than a normal GA algorithm.

Figure 1 demonstrates some successfully attacked cases by the two algorithms. We can see that the modified images are semi-imperceptible to human. Due to the stochasticity and different approaches, the new predictions on the modified image could be dissimilar between algorithms and even between different runs.

A natural question comes out is that what is the role of the number of base vectors? Figure 2 shows number of successful cases and expected number of queries on the different number of base vectors. It suggests a good choice in practice is using around 128 to 256 base vectors.

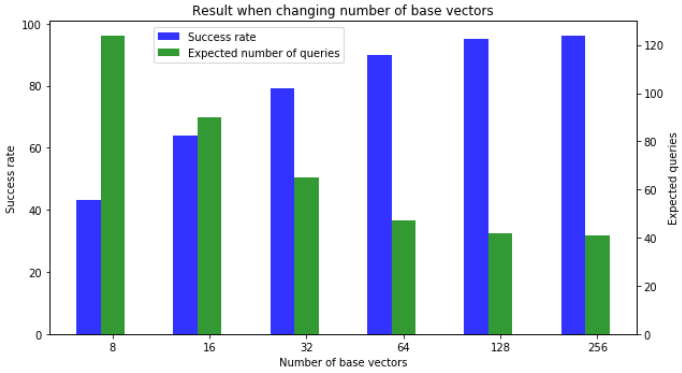

Fig. 2. The effects of number of base vectors on attacking performance.

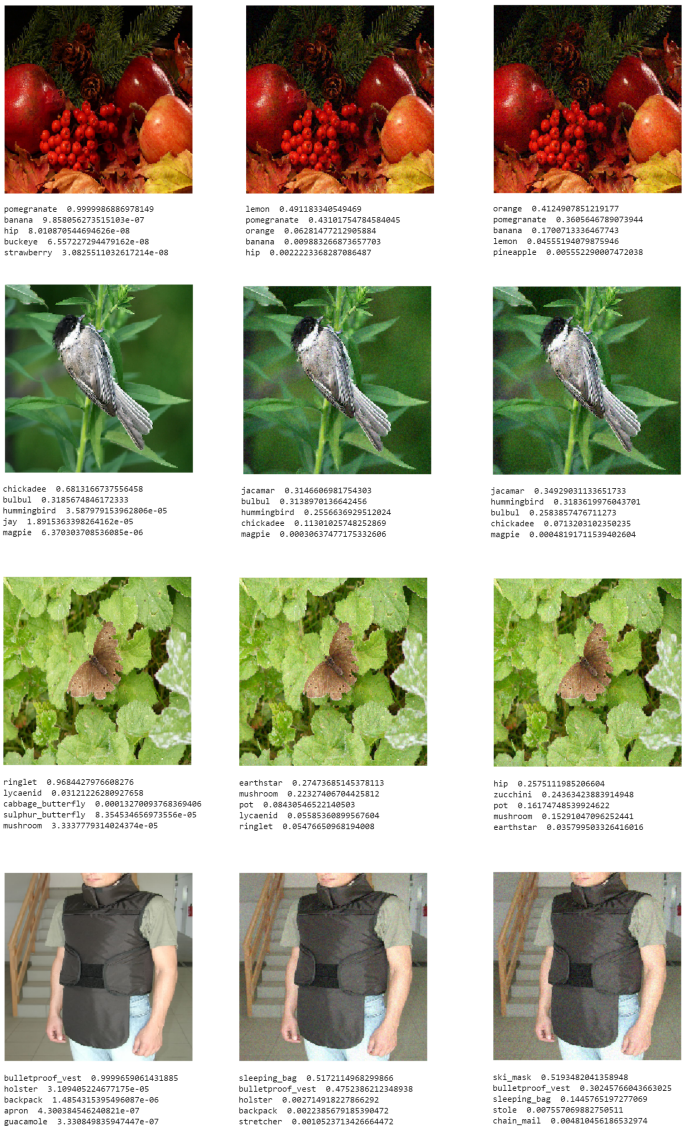

Fig. 1. Top-5 prediction on some random successful cases, from left to right: original images, perturbed images by GA and perturbed images by cma-es

III. One Pixel Attack

A. Method

In this section, we will present the one-pixel attack method, by following the works of [3]. Unlike the context of all pixels attack, gradient-based methods cannot work efficiently in this case where we do not know which pixel needed to modify. On the other hand, the information about position of the modified pixel is not obtained differential according to the fitness function. For this reason, EAs are naturally suitable for the context of this setting.

Each perturbation corresponding to 1 pixel is encoded into an array which consists of 5 numbers: 2 for the coordinate of the pixel and 3 for new RGB values of that pixel. All these numbers are clipped to make sure that the coordinate of that pixel is still inside the image and the adversarial image is still valid.

TABLE II
Statistics of attack result on 100 first images

Attack Method	1 pixel	3 pixels	5 pixels	7 pixels
Genetic Algorithm	32	76	86	92
CMA-ES	29	73	88	93
Differential Evolution	37	77	85	87

Su et al suggest using Differential Evolution to evolve perturbation. At each iteration, each solution candidate x will produce a child y by using the usual DE formula:

$$y = \overline{CR} \times x + CR \times (a + F \times (b - c))$$
$$\text{such that } x \neq a \neq b \neq c \tag{6}$$

In that $CR \sim \mathrm{B}(p)$ is a multivariate Bernoulli distribution with p is called *crossover probability*. $\overline{CR}$ is the complement of CR, which means $CR + \overline{CR} = \mathbf{1}$. F is called *differential weight*. Each child y directly compete with its parent x to determine will survive to next generation.

However, since the problem has been well defined, we can apparently use other EAs such as GA and CMA-ES.

B. Experiment

In the situation using this strategy, we will perform attacks on CIFAR-10, a smaller version of ImageNet. CIFAR-10 contains images from 10 classes, and each image has a size of $32 \times 32 \times 3$. The target model used in our experiment achieve the accuracy of 92.3 on the test set, by using ResNet architecture [6]. Each attack will run within 200 generations to decide whether the attack is successful or not. The number of pixel in this problem is restricted therefore we only care about success rate.

Table II shows number success when the attack on first 100 images from the test set. Although the method presented above is for the 1-pixel attack, it could be easily extended to k-pixel. DE is better than GA and ES when k is small, but CMA-ES become much better when increasing k. We argue that because CMA-ES is a local search algorithm, it tends to be stuck at local minima when the number of dimensions is small, especially if initial points is far from global optima. When we increase k, each initial point only needs to search in a smaller area, that is more suitable to CMA-ES. On the other hand, DE maintains the diversity of the population. Therefore it can work well in case 1-pixel attack.

Figure 3 demonstrates some successful cases. It is quite interesting that some perturbations can fool the target network even they only alter pixels in the background. This is an intriguing property of neural networks.

IV. Conclusion and Further Discussion

A. Conclusion

In this project, we studied and conducted experiments to fool convolutional neural networks by using EAs. In particular, we have examined 2 different types of black-box attack: all pixels attack and one (or several) pixel(s) attack, on 2 popular datasets: ImageNet and CIFAR-10. We showed that EAs could

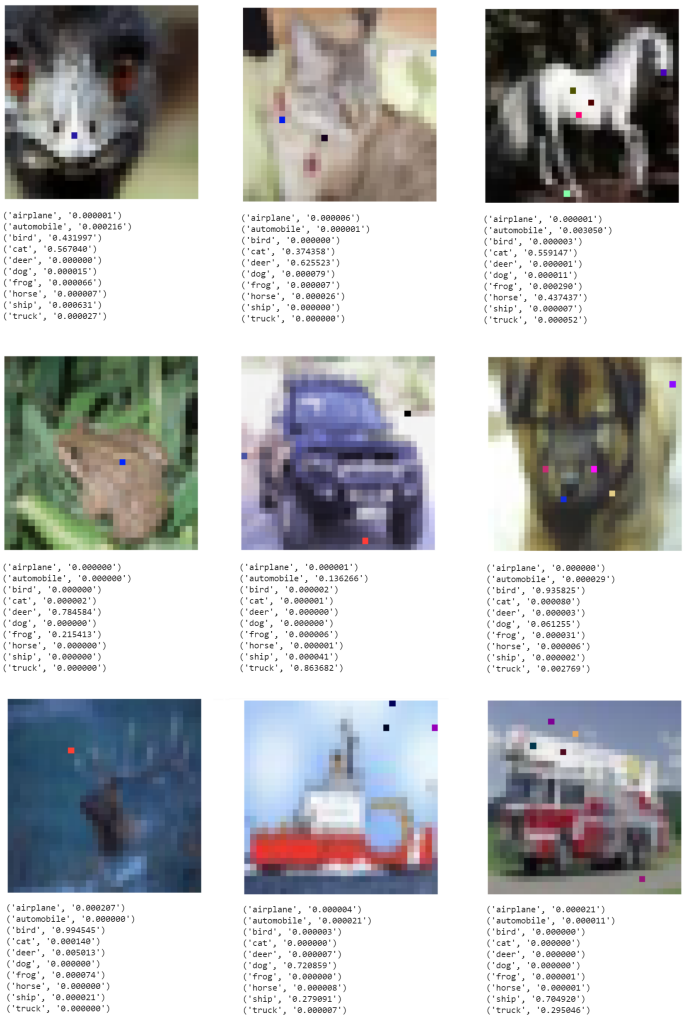

Fig. 3. Prediction on some random successful cases, from top to bottom: GA, CMA-ES and DE.

absolutely be used to create adversarial examples making high-accurate models misclassify. The models we attacked in our experiments are Inception and Resnet which are architectures widely used in state-of-the-art models.

We proposed a simple method to enable using CMA-ES in high dimensional spaces. By using linear span, we can let algorithm search in a smaller dimension comparing with original one. Our result suggested that adversarial perturbations spread over the image space and could be easily captured by a small random sub space.

Following the work of Su et al [3], we studied the ability of different algorithms in creating adversarial examples when number of pixels is limited. Our result suggested that when number of pixels is large enough comparing with size of images, we can use CMA-ES besides other EAs.

B. Discussion

One of the obstacles when doing this project is that we do not know whether the input image actually could be perturbed to create adversarial example which satisfies given constraints. We have to run some attacks on training set to choose hyper-parameters and set maximum number of generations heuristically.

Running time is also a factor we have to take into account. We choose DEAP [1] as it is quite straightforward to be applied. Unfortunately, the usage of DEAP runs evaluations of individuals separately. Since we do not have distributed machines, we have to use to native implementation of DEAP and modify it to run parallel on GPU.

The idea of using linear span is quite easily to came up with. However, how to choose base vectors is still an open question for future works. In the early attempts we also tried to add some mutations to adjust base vectors while running our algorithm but it did not work well.

We cannot compare our result with other works since in the first place they do not public their code and their target models and in the second place we also use different settings. For instance, most of other work create adversarial examples having float values of pixels while we keep the values of pixels integer numbers. However, our results strongly suggest that we can also use EAs to create various other different kinds of attacks related to image transforms such as blurring, color transfer, geometrical transformation, etc.

ACKNOWLEDGMENT

This report is a part of final project in Bio-Inspired Computational Intelligence at University of Trento, 2018.

REFERENCES

[1] P. Vidnerová and R. Neruda, "Vulnerability of machine learning models to adversarial examples," in *ITAT*, 2016.

[2] M. Alzantot, Y. Sharma, S. Chakraborty, and M. B. Srivastava, "Genattack: Practical black-box attacks with gradient-free optimization," *CoRR*, vol. abs/1805.11090, 2018. [Online]. Available: http://arxiv.org/abs/1805.11090

[3] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *CoRR*, vol. abs/1710.08864, 2017. [Online]. Available: http://arxiv.org/abs/1710.08864

[4] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001. [Online]. Available: https://doi.org/10.1162/106365601750190398

[5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[1] https://github.com/DEAP/deap