GenAttack: Practical Black-box Attacks with Gradient-Free Optimization

Moustafa Alzantot
Computer Science Dept., UCLA
malzantot@ucla.edu

Yash Sharma
The Cooper Union
sharma2@cooper.edu

Supriyo Chakraborty
IBM T. J. Watson Research Center
supriyo@us.ibm.com

Mani Srivastava
Computer Science Dept., UCLA
mbs@ucla.edu

Abstract

Deep neural networks (DNNs) are vulnerable to adversarial examples, even in the black-box case, where the attacker is limited to solely query access. Existing black-box approaches to generating adversarial examples typically require a significant amount of queries, either for training a substitute network or estimating gradients from the output scores. We introduce `GenAttack`, a gradient-free optimization technique which uses genetic algorithms for synthesizing adversarial examples in the black-box setting. Our experiments on the MNIST, CIFAR-10, and ImageNet datasets show that `GenAttack` can successfully generate visually imperceptible adversarial examples against state-of-the-art image recognition models with orders of magnitude fewer queries than existing approaches. For example, in our CIFAR-10 experiments, `GenAttack` required roughly 2,568 times less queries than the current state-of-the-art black-box attack. Furthermore, we show that `GenAttack` can successfully attack both the state-of-the-art ImageNet defense, ensemble adversarial training, and non-differentiable, randomized input transformation defenses. `GenAttack`'s success against ensemble adversarial training demonstrates that its *query efficiency* enables it to exploit the defense's weakness to direct black-box attacks. `GenAttack`'s success against non-differentiable input transformations indicates that its *gradient-free nature* enables it to be applicable against defenses which perform gradient masking/obfuscation to confuse the attacker. Our results suggest that population-based optimization opens up a promising area of research into effective gradient-free black-box attacks.

1 Introduction

Deep neural networks (DNNs) have achieved state-of-the-art performance in various tasks in machine learning and artificial intelligence, such as image classification, speech recognition, machine translation and game-playing. Despite their effectiveness, recent studies have illustrated the vulnerability of DNNs to adversarial examples [23, 7]. For instance, a carefully designed perturbation to an image can lead a well-trained DNN to misclassify. Targeted adversarial examples can even cause misclassification to a chosen class. Even worse, effective adversarial examples can also be made virtually indistinguishable to human perception. Moreover, researchers have shown that these adversarial examples are still effective even in the physical world [12, 2]. The lack of robustness exhibited by DNNs to adversarial examples has raised serious concerns for security-critical applications.

Prior work has shown that adversarial examples can be generated in the *white-box* setting, where an attacker is given full access and control over a targeted DNN. However, when attacking real-

Preprint. Work in progress.

world systems, one needs to consider the problem of performing adversarial attacks against *black-box* machine learning models. Black-box models reveal nothing about the network architecture, parameters, or training data. In such a case, the attacker only has access to the input-output pairs of the classifier. Dominant approaches in this setting have relied on attacking trained substitute networks, and hoping the generated examples transfer to the target model [18]. This approach suffers from imperfect transferability and the computational cost of training a substitute network. Recent work has used coordinate-based finite difference methods in order to directly estimate the gradients from the confidence scores, however the attacks are still computationally expensive, relying on optimization tricks to remain tractable [5]. Both approaches are very query-intensive, thus limiting their practicality in real-world scenarios.

Nearly all previous work on adversarial attacks, [7, 17, 8, 12, 16, 5, 4] has used gradient-based optimization in order to find successful adversarial examples. However, we note that the objective function used for attacks is typically vector-valued, with multiple objectives (i.e., perform targeted misclassification attack while minimizing output distortion). Therefore, a total ordering of the feasible solutions is often not possible. Meta-heuristic approaches that pursue a single solution (or hypothesis) in each round are thus vulnerable to local minima, and inefficient exploration of the state space [21].

Motivated by the above, in this paper, we present `GenAttack`, a powerful and efficient black-box attack using a *gradient-free optimization* scheme. By adopting a population-based approach using genetic algorithms, `GenAttack` intuitively is better suited to handle multi-objective optimization problems. `GenAttack` maintains a population of feasible solutions in each round, and bypasses the requirement to compute, *or even approximate*, the network gradients. By simultaneously pursuing multiple hypotheses, `GenAttack` is more resilient to poor local minima, which goes a long way in reducing the number of queries needed to find successful adversarial examples. Furthermore, by not attempting to use the gradient, `GenAttack` is robust to defenses which perform gradient masking/obfuscation [1].

We evaluate `GenAttack` against state-of-the-art models for the three most popular image recognition datasets: MNIST [14], CIFAR-10 [11], and ImageNet [6]. Our results show that `GenAttack` is successful at performing *targeted* black-box attacks against these models, with significantly less queries than previous approaches. For example, in our CIFAR-10 experiments, `GenAttack` required roughly 2,568 times less queries than the current state-of-the-art black-box attack. Additionally, we also demonstrate the success of `GenAttack` against ensemble adversarial training [24], the state-of-the-art ImageNet defense, and randomized, non-differentiable input transformation defenses [9]. These results illustrate the power of `GenAttack`'s query efficiency and gradient-free nature.

2 Related Work

In what follows, we summarize recent approaches for generating adversarial examples, in both the white-box and black-box cases, as well as defending against adversarial examples. Please refer to the cited works for further detail.

2.1 White-box attacks & Transferability

In the *white-box* case, attackers have complete knowledge of and full access to the targeted DNN. In this scenario, the adversary is able to use backpropagation for gradient computation, which obviously increases the strength of gradient-based attacks.

White-box attacks can also be used in black-box cases by taking advantage of *transferability*. Transferability refers to the property that adversarial examples generated using one model are often misclassified by another model. The substitute model approach to black-box attacks takes advantage of this property to generate successful adversarial examples.

FGSM & I-FGSM

In [7], the authors proposed the Fast Gradient Sign Method (FGSM), a quick and reliable approach for generating adversarial examples. Let $\mathbf{x}_0$ and $\mathbf{x}$ denote the original and adversarial examples, respectively, and let t denote the target class to attack. FGSM uses the gradient ∇J of the training

loss J with respect to $\mathbf{x}_0$ for crafting adversarial examples. An L_∞ attack, $\mathbf{x}$ is crafted by

$$\mathbf{x} = \mathbf{x}_0 - \epsilon \cdot \text{sign}(\nabla J(\mathbf{x}_0, t)), \tag{1}$$

where ϵ specifies the L_∞ distortion between $\mathbf{x}$ and $\mathbf{x}_0$, and $\text{sign}(\nabla J)$ takes the sign of the gradient. Untargeted attacks can be implemented in a similar fashion. In [12], an iterative version of FGSM was proposed (I-FGSM), where FGSM is used iteratively with a finer distortion, followed by an ϵ-ball clipping. In [16], PGD is introduced, where I-FGSM is modified to incorporate random starts.

C&W & EAD

Instead of leveraging the training loss, Carlini and Wagner designed an L_2-regularized loss function based on the logit layer representation in DNNs for crafting adversarial examples [3]. Its formulation is as follows:

$$\begin{aligned} &\text{minimize}_{\mathbf{x}} \;\; c \cdot f(\mathbf{x}, t) + \|\mathbf{x} - \mathbf{x}_0\|_2^2 \\ &\text{subject to} \;\; \mathbf{x} \in [0,1]^p, \end{aligned} \tag{2}$$

where $f(x, t)$ is the logit layer loss function. By increasing κ, one increases the necessary margin between the predicted probability of the target class and that of the rest, generating stronger adversarial examples with increased distortion. The untargeted attack formulation is similar. EAD generalizes the C&W attack by incorporating L_1 minimization via performing elastic-net regularization [4], and has been shown to generate more robust, transferable adversarial examples [19, 20, 15].

2.2 Black-box attacks

In the literature, the *black-box* attack setting has been referred to as the case where an attacker has free access to the input and output of a targeted DNN but is unable to perform back propagation on the network. Proposed approaches have relied on transferability and gradient estimation, and are summarized below.

Substitute Networks

Early approaches to black-box attacks made use of the power of free query to train a substitute model, a representative substitute of the targeted DNN [18]. The substitute DNN can then be attacked using any white-box technique, and the generated adversarial examples are used to attack the target DNN. As the substitute model is trained to be representative of a targeted DNN in terms of its classification rules, adversarial attacks to a substitute model are expected to be similar to attacking the corresponding targeted DNN. This approach however relies on the transferability property rather than directly attacking the target DNN, which is imperfect and thus limits the strength of the adversary. Furthermore, training a substitute model is computationally expensive and hardly feasible when attacking large models, such as Inception-v3 [22].

ZOO

Due to the unfavorable properties of existing approaches, ZOO was proposed [5]. ZOO builds on the C&W attack, due to its state-of-the-art performance, and modifies the loss function such that it only depends on the output of the DNN, as opposed to depending on the logit layer representation. Furthermore, an approximate gradient is computed using the finite difference method on the targeted DNN, and the optimization problem is solved via zeroth order optimization.

For each coordinate, 2 function evaluations are required to estimate the gradient. Performing this computation for all coordinates quickly becomes too expensive in practice. To resolve this issue, stochastic coordinate descent is used, which only requires 2 function evaluations for each step. Still, when attacking large black-box networks, such as Inception-v3, computation is quite slow and thus a dimension reduction transformation on the perturbation is applied. With these optimizations, unlike the substitute model approach, attacking Inception-v3 becomes computationally tractable. However, as we demonstrate in our experimental results, the attack is still quite query-inefficient and thus limited in power and impractical for attacking real-world systems. The authors in [10] note this as well, and target a similar contribution. We treat their contribution as parallel work.

2.3 Defending against adversarial attacks

Adversarial Training

Adversarial training is typically implemented by augmenting the original training dataset with the label-corrected adversarial examples to retrain the network. In [16], a high capacity network is trained against L_∞-constrained PGD, I-FGSM with random starts, which is deemed to be the strongest attack utilizing the local first-order information about the network. It has been shown that the defense is less robust to attacks optimized on other distortion metrics, namely L_1 [19]. In [24], training data is augmented with perturbations transferred from other models, and was demonstrated to have strong robustness to transferred adversarial examples. We demonstrate in our experimental results that its less robust to query-efficient black-box attacks, such as `GenAttack`.

Input Transformations

In recent work, attempts have been made to remove adversarial perturbations from the input. In [9], transformations based on image cropping and rescaling, bit-depth reduction, JPEG compression, total variance minimization, and image quilting were explored. These defenses were demonstrated to be surprisingly effective against existing attacks, and the strongest defenses were found to be total variance minimization and image quilting, due to their non-differentiable nature and inherent randomness. We demonstrate in our experimental results that these defenses are less robust to `GenAttack`, due to its gradient-free nature.

3 GenAttack Algorithm

`GenAttack` relies on genetic algorithms, which are population-based gradient-free optimization strategies. Genetic algorithms are inspired by the process of natural selection, iteratively evolving a population of candidate solutions towards better solutions. The population in each iteration is a called a *generation*. In each generation, the quality of population members is evaluated using a *fitness* function. "Fitter" solutions are more likely to be selected for breeding the next generation. The next generation is generated through a combination of *crossover* and *mutation*. Crossover is the process of taking more than one parent solution and producing a child solution from them; it is analogous to reproduction and biological crossover. In addition, at each iteration, a small random mutation to the population members occurs during evolution according to a small user-defined mutation probability. This is done in order to increase the diversity of population members and provide better exploration of the search space.

Algorithm 1 describes the operation of `GenAttack`. The input for the algorithm is the original image $\mathbf{x}_{orig}$ and the target classification label t chosen by the attacker. The algorithm computes an adversarial image $\mathbf{x}_{adv}$ such that the model classifies $\mathbf{x}_{adv}$ as t and $||\mathbf{x}_{orig} - \mathbf{x}_{adv}||_\infty \leq \delta_{max}$. We define the population size to be N, the mutation probability to be ρ, and the step-size to be α.

`GenAttack` initializes a population of examples around the given input example $\mathbf{x}_{orig}$ by applying independent and uniformly distributed random noise in the range $(-\alpha\,\delta_{max}, \alpha\,\delta_{max})$ to each dimension of the input vector $\mathbf{x}_{orig}$ with probability ρ. Then repeatedly, until a successful example is found, each population members' fitness is evaluated, parents are selected, and `crossover` & `mutation` are performed to form the next generation.

The subroutine `ComputeFitness` evaluates the fitness, i.e. quality, of each population member. As the fitness function should reflect the optimization objective, a reasonable choice would be to use the output score given to the target class label directly. However, we find it more efficient to also jointly motivate the decrease in the probability of other classes. We also find that the use of $\log$ proves to be helpful in avoiding numeric instability issues [5]. Therefore, we pick the following function:

$$ComputeFitness(\mathbf{x}) = \log f(\mathbf{x})_t - \log \max_{c \neq t} f(\mathbf{x})_c$$

Population members at each iteration are ranked according to their fitness value. Members with higher fitness are more likely to be a part of the next generation while members with smaller `fitness` values are more likely to be replaced. We compute the probability of selection for each population member by normalizing the fitness values into a probability distribution. Then, we stochastically and independently select random parent pairs among the population members according to that

Algorithm 1 `GenAttack` (Targeted case)

Input: original example $\mathbf{x}_{orig}$, target label t, maximum L_∞ distance δ_{max}, step-size α, mutation probability ρ, population size N.
for $i = 1, ..., N$ in population **do**
  $\mathcal{P}_i^0 \leftarrow \mathbf{x}_{orig} + Bernoulli(\rho) * \mathcal{U}(-\alpha\,\delta_{max}, \alpha\,\delta_{max})$
end for
for $g = 1, 2...G$ generations **do**
  for $i = 1, ..., N$ in population **do**
    $F_i^{g-1} = ComputeFitness(\mathcal{P}_i^{g-1})$
  end for
  $\mathbf{x}_{adv} = \mathcal{P}^{g-1}_{\arg\max_j F_j^{g-1}}$
  if $\arg\max_c f(\mathbf{x}_{adv})_c == t$ **then**
    Return: $\mathbf{x}_{adv}$ ▷ { Found successful attack}
  end if
  $\mathcal{P}_1^g = \{\mathbf{x}_{adv}\}$
  $probs = Normalize(F^{g-1})$
  for $i = 2, ..., N$ in population **do**
    Sample $parent_1$ from $\mathcal{P}^{g-1}$ according to $probs$
    Sample $parent_2$ from $\mathcal{P}^{g-1}$ according to $probs$
    $child = Crossover(parent_1, parent_2)$
    $child_{mut} = child + Bernoulli(\rho) * \mathcal{U}(-\alpha\,\delta_{max}, \alpha\,\delta_{max})$
    $\mathcal{P}_i^g = \{child_{mut}\}$
  end for
end for

distribution. In addition to that, the *elite* member, the one with highest fitness, is guaranteed to become a member of the next generation.

After selection, parents are mated together to produce members of the next generation. A child is generated by selecting the feature value from either $parent_1$, or $parent_2$ with probabilities $\frac{fitness(parent\_1)}{fitness(parent\_1)+fitness(parent\_2)}$, and $\frac{fitness(parent\_2)}{fitness(parent\_1)+fitness(parent\_2)}$, respectively.

To encourage diversity among the population members and promote exploration of the search space, at the end of each iteration, population members can be subject to mutation, according to probability ρ. Random noise uniformly sampled in the range $(-\alpha\,\delta_{\max}, \alpha\,\delta_{\max})$ is applied to individual features of the chosen population member. Also, clipping is performed to ensure that the pixel values within the permissible L_∞ distance away from benign example $\mathbf{x}_{orig}$.

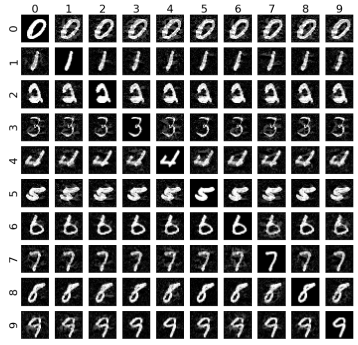

Figure 1: MNIST adversarial examples generated by `GenAttack`. Row label is the true label and column label is the target label.

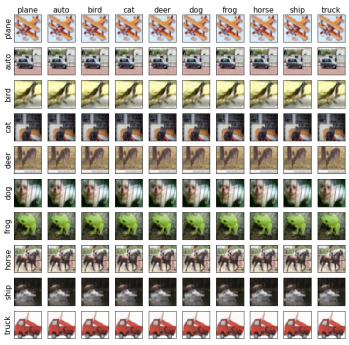

Figure 2: CIFAR-10 adversarial examples generated by `GenAttack`. Row label is the true label and column label is the target label.

4 Results

We evaluate GenAttack by running experiments attacking state-of-art MNIST, CIFAR-10, and ImageNet image classification models. For each dataset, we use the same models as used in the ZOO work [5]. For MNIST and CIFAR-10, the model accuracies are 99.5% and 80%, respectively. The reader can refer to [3] for more details on the architecture of those models. For ImageNet, we use Inception-v3 [22], which achieves 94.4% top-5 accuracy and 78.8% top-1 accuracy. We compare the effectiveness of GenAttack to ZOO on these models in terms of the attack success rate (ASR), the runtime, and the median number of queries necessary for success. The runtime and query count statistics are computed over successful attacks only. A *single* query means an evaluation of the target model output on a *single* input image. Using the authors' code [1], we configure ZOO for each dataset based on the implementations the authors used for generating their experimental results [5]. We also evaluate against the state-of-the-art white-box C&W attack, to give perspective on runtime.

In addition, we evaluate the effectiveness of GenAttack against ensemble adversarial training [24], which is considered to be the state-of-art ImageNet defense against black-box attacks. Ensemble adversarial training was proven to be effective at providing robustness against black-box attacks relying on transferability during the NIPS 2017 Competition on Defenses against Adversarial Attacks [24, 13], using models released by the authors at the following link [2]. Finally, we evaluate against recently proposed randomized, non-differentiable input transformation defenses [9]. We demonstrate that GenAttack, unlike gradient-based attacks, can handle such defenses as-is due to its gradient-free nature.

For all of our MNIST and CIFAR-10 experiments, we limit GenAttack to a maximum of 100,000 queries, and use the following hyperparameter values: mutation probability $\rho = 5\text{e}{-2}$, population size $N = 6$, and step-size $\alpha = 1.0$. For all of our ImageNet experiments, as the images are nearly 100x larger than those of CIFAR-10, we use a maximum of 1,000,000 queries and lower ρ to $1\text{e}{-4}$. To match the mean L_∞ distortion computed over successful examples of ZOO, and thereby make the query comparison fair, we set $\delta_{max} = \{0.3, 0.05, 0.05\}$, for our MNIST, CIFAR-10, and ImageNet experiments, respectively. To encourage further research, we are releasing our code as open source at [3].

4.1 Query Comparison

We compare GenAttack and ZOO in terms of queries, and provide C&W white-box results to put the runtime in perspective. For all experiments, we use an AMD Threadripper 1950X CPU with a single NVIDIA GTX 1080 Ti GPU. For MNIST, CIFAR-10, and ImageNet, we use 1000, 1000, and 100 randomly selected correctly classified images from the test sets. For each image, we select a random target label. Table 1 shows the results of our experiment. The results show that both ZOO and GenAttack can succeed on the MNIST and CIFAR-10 datasets, however GenAttack is 2,126 times and 2,568 times more efficient on each. On ImageNet, ZOO is not able to succeed consistently in the targeted case, is still quite query inefficient, and has exceptional computational cost (see runtime)[4].

A randomly selected set of MNIST and CIFAR-10 test images and their associated adversarial examples targeted to each other label are shown in Figure 1 and Figure 2. An ImageNet test image with its associated adversarial example is shown in Figure 3.

4.2 Attacking Ensemble Adversarial Training

Ensemble adversarial training incorporates adversarial inputs generated from other models into the model's training data in order to increase its robustness to adversarial examples [24]. This has proven to be the most effective approach at providing robustness against transfer-based black-box attacks during the NIPS 2017 Competition. We demonstrate that the defense is much less robust against query-efficient black-box attacks, such as GenAttack.

[1] https://github.com/huanzhang12/ZOO-Attack

[2] https://github.com/tensorflow/models/tree/master/research/adv_imagenet_models

[3] https://github.com/nesl/adversarial_genattack.git

[4] Each iteration, ZOO performs 2*128 queries at once. This would not be possible attacking a real-world system, queries would have to be iterative, thus the runtime statistics are artificially low.

	MNIST ($L_\infty = 0.30$)			CIFAR-10 ($L_\infty = 0.05$)			ImageNet ($L_\infty = 0.05$)		
	ASR	Queries	Runtime	ASR	Queries	Runtime	ASR	Queries	Runtime
C&W	100%	–	0.006 hr	100%	–	0.006 hr	100%	–	0.025 hr
ZOO	98%	2,118,222	0.013 hr	93.3%	2,064,798	0.025 hr	18%	2,611,456	2.25 hr
GenAttack	100%	996	0.002 hr	96.5%	804	0.001 hr	100%	97,493	0.51 hr

Table 1: Attack success rate (ASR), median number of queries, and mean runtime for the C&W (white-box) attack, ZOO, and `GenAttack` with equivalent L_∞ distortion. Query and runtime statistics are computed only over successful examples.

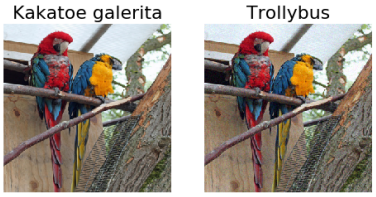

Figure 3: Adversarial example generated by `GenAttack` against the `InceptionV3` model ($L_\infty = 0.05$). Left figure: original, right figure: adversarial example.

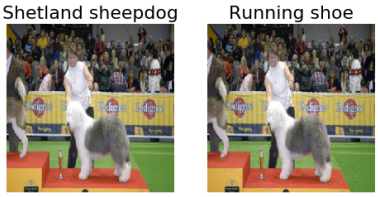

Figure 4: Adversarial example generated by `GenAttack` against the `JPEG` compression defense ($L_\infty = 0.15$). Left figure: original, right figure: adversarial example.

	InceptionV3		Ens4AdvInceptionV3	
	ASR	Queries	ASR	Queries
GenAttack	100%	97,493	93%	163,995

Table 2: `GenAttack` results (w/ $L_\infty = 0.05$) attacking the vanilla and ensemble adversarially trained Inception-v3 models. Median query counts are computed only over successful examples.

We performed an experiment to evaluate the effectiveness of `GenAttack` against ensemble adversarially trained models released by the authors, namely `Ens4AdvInceptionV3` and `EnsAdvInceptionResNetv2`. We use the same 100 randomly sampled test images and targets used in our previous ImageNet experiments. We find that `GenAttack` is able to achieve 93% success and 88% success against both models, respectively. In Table 2, we compare the success rate and median query count between the ensemble adversarially trained and the vanilla Inception-v3 models. Our comparison shows that these positive results are yielded with only a limited increase in query count. We additionally note that the max L_∞ used for evaluation in the NIPS 2017 competition was varied between 4 and 16, which when normalized equals 0.02 and 0.06, respectively. Our δ_{max} (0.05) falls in this range.

	CIFAR-10		ImageNet	
	ASR	Queries	ASR	Queries
Bit depth	93%	2,796	100%	116,739
JPEG	88%	3,541	89%	190,680
TVM	70%	5,888 x 32	–	–

Table 3: Evaluation of `GenAttack` against non-differentiable and randomized input transformation defenses. We use $L_\infty = 0.05$ for bit-depth, and $L_\infty = 0.15$ for JPEG and TVM experiments.

4.3 Attacking Non-Differentiable, Randomized Input Transformations

A set of input transformations have been recently proposed to mitigate adversarial perturbations: image cropping, bit-depth reduction, JPEG compression, total variance minimization, and image quilting [9]. Bit-depth reduction and JPEG compression add robustness by virtue of being non-

differentiable transformations, while total variance minimization and image quilting introduce additional randomization. We demonstrate that `GenAttack` is robust to these input transformations, primarily due to its gradient-free nature. Our results are summarized in Table 3.

Circumventing non-differentiable transformations such as bit-depth reduction and JPEG compression represents an obstacle for gradient-based attack methods (e.g. [3, 4]). However, `GenAttack`, being gradient-free, is largely unaffected by such defenses. In fact, our results show high attack success rates against both bit depth reduction (3 bits of reduction, as in [9]), and JPEG compression (JPEG quality level = 75, as in [9]). A visual example of our results against JPEG compression is shown in Figure 4.

Circumventing randomized transformations such as total variation minimization (TVM), is difficult because they are not only hard to differentiate, but also introduce randomization to their outputs. For example, TVM randomly drops many of the pixels (dropout rate of 50%, as in [9]) in the original image and reconstructs the input image from the remaining pixels by solving a denoising optimization problem. Due to randomization, the classifier returns a different score at each run for the same input, confusing the attacker. To circumvent the TVM defense, we generalize the `ComputeFitness` function to be

$$ComputeFitness(\mathbf{x}) = \mathop{\mathbb{E}}_{r}[\log f(\mathbf{x}, r)_t - \log \max_{c \neq t} f(\mathbf{x}, r)_c]$$

where $f(\mathbf{x}, r)$ is the randomization-defended model query function and r is the noise input to the TVM function. The expectation is computed by querying the model t times (we used $t = 32$) for every population member to obtain a robust fitness score at the cost of an increased number of queries. Due to the computational complexity of applying TVM on each query, we performed the TVM experiment only using the CIFAR-10 dataset and achieved 70% success with $L_\infty = 0.15$. Due to the large randomization introduced by TVM, we counted an adversarial example as success only if it is classified as the target label three times in a row. Notably, TVM largely decreases the model accuracy on clean inputs (e.g. in our CIFAR-10 experiments, from 80% to 40%) unless the model is re-trained with transformed examples [9].

5 Conclusion

`GenAttack` is a powerful and efficient black-box attack which uses a gradient-free optimization scheme via adopting a population-based approach using genetic algorithms. We evaluated `GenAttack` against well-trained MNIST, CIFAR-10, and ImageNet models. We found that `GenAttack` is successful at performing targeted black-box attacks against these models with significantly less queries than the current state-of-the-art. Additionally, we demonstrate that `GenAttack` can succeed against ensemble adversarial training, the state-of-the-art ImageNet defense found to be robust to transfer-based black-box attacks in the NIPS 2017 competition, with only a limited increase in queries. Finally, we showed that `GenAttack` can succeed against non-differentiable input transformations, due to its gradient-free nature, and can even succeed against randomized ones by generalizing the fitness function to compute an expectation over the transformation. Our results suggest that population-based optimization opens up a promising area of research into effective gradient-free black-box attacks.

References

[1] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[2] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*, 2017.

[3] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. *arXiv preprint arXiv:1608.04644*, 2017.

[4] P. Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C. Hsieh. Ead: Elastic-net attacks to deep neural networks via adversarial examples. *arXiv preprint arXiv:1709.0414*, 2017.

[5] P. Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C. Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26. ACM, 2017.

[6] J. Deng, W. Dong, R. Socher, J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[7] I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[8] S. Gu and L. Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

[9] C. Guo, M. Rana, and L. van der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

[10] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin. Black-box adversarial attacks with limited queries and information. *arXiv preprint arXiv:1804.08598*, 2018.

[11] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009.

[12] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[13] A. Kurakin, I. Goodfellow, S. Bengio, Y. Dong, F. Liao, M. Liang, T. Pang, J. Zhu, X. Hu, C. Xie, J. Wang, Z. Zhang, Z. Ren, A. Yuille, S. Huang, Y. Zhao, Y. Zhao, Z. Han, J Long, Y. Berdibekov, T. Akiba, S. Tokui, and M. Abe. Adversarial attacks and defences competition. *arXiv preprint arXiv:1804.00097*, 2018.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[15] P. H. Lu, P. Y. Chen, K. C. Chen, and C. M. Yu. On the limitation of magnet defense against l1-based adversarial examples. *arXiv preprint arXiv:1805.00310*, 2018.

[16] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[17] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, number EPFL-CONF-218057, 2016.

[18] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519. ACM, 2017.

[19] Y. Sharma and P. Y. Chen. Attacking the madry defense model with l1-based adversarial examples. *arXiv preprint arXiv:1710.10733*, 2017.

[20] Y. Sharma and P. Y. Chen. Bypassing feature squeezing by increasing adversary strength. *arXiv preprint arXiv:1803.09868*, 2018.

[21] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. Stanley, and J. Clune. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv preprint arXiv:1712.06567*, 2018.

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. CVPR, 2015.

[23] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, and I. Goodfellow. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[24] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.