Decoupling Direction and Norm for Efficient Gradient-based L2 Adversarial Attacks

Jérôme Rony[*] **Luiz G. Hafemann**[*] **Robert Sabourin** **Eric Granger**
LIVIA, École de technologie supérieure
Montréal, Canada
`jerome.rony@gmail.com` `luiz.gh@mailbox.org`
`{robert.sabourin, eric.granger}@etsmtl.ca`

Abstract

Research on adversarial examples in computer vision tasks has shown that small changes to an image (usually imperceptible to a human observer) can induce misclassifications, which has security implications for a wide range of image processing systems. When considering distortions using the L_2 norm, the Carlini and Wagner (C&W) attack is presently the most effective white-box attack in the literature, being able to obtain adversarial examples with very small distortions. However, this method is slow, since it requires a line-search in one of the optimization terms, and often requires thousands of iterations. In this paper, a new approach is proposed for generating attacks with low L_2 norm, by decoupling the direction and the norm of the adversarial noise that is added to the image. We obtain results comparable to the C&W attack even with as few as 100 iterations, which allows its usage for adversarial training. Experiments conducted on MNIST an CIFAR-10 show that our attack achieves comparable results to the state-of-the-art (in terms of L_2-norm) in much fewer iterations.[2]

1 Introduction

Several machine learning models, most notably neural networks, are susceptible to adversarial examples, in which adding small perturbations to an input causes a misclassification [1, 2]. While defenses have been proposed to address this issue [3, 4, 5], developing robust models is still an open research problem [6, 7].

Many attacks for neural networks have been proposed in the literature to achieve different objectives, such as obtaining the lowest amount of noise that induces misclassification [1, 8], or being fast enough to be incorporated in the training procedure [3, 4]. In the case of obtaining adversarial examples with lowest noise (measured by its L_2 norm), the state-of-the-art is the attack proposed by Carlini et al. [8]. While this attacks obtains good results, it also requires a high number of iterations, which makes it impractical to be used for training. On the other hand, one-step attacks are fast to generate, but using them for training do not increase model robustness on white-box scenarios (full knowledge of the model under attack) [4]. Developing an attack that finds adversarial examples with low noise in few iterations would enable adversarial training with such examples, which we hypothesize can increase model robustness against white-box attacks.

In this paper, we propose a new gradient-based attack called *DDN* (Decoupled Direction and Norm), that achieves similar performance to C&W L_2, while requiring fewer iterations, being amenable to

[*] Equal contribution.

[2] Preliminary work. We refer the reader to the paper at `https://arxiv.org/abs/1811.09600`. Code at `https://github.com/jeromerony/fast_adversarial`

Preprint. Work in progress.

be used during training. We first review the C&W L_2 attack, exploring the reason why this attack requires many iterations, and then present our attack that address this issue. Our attack obtains comparable results to the state-off-the-art on MNIST and CIFAR-10 (in terms of success rate and L_2 norm) while requiring much fewer iterations (~100 times).

Having an efficient attack allowed us to adversarily train a classification model for the NIPS 2018 Adversarial Vision Challenge [9], achieving 3rd place in the Robust Model track on the Tiny ImageNet dataset (64×64 natural images).

2 Related work

In this work, we consider attacks that are generated by a gradient-based optimization procedure to obtain a minimum distortion when considering the L_2-norm. For this case, the state-of-the-art is the attack proposed by Carlini *et al.* [8].

Carlini and Wagner L_2 Attack. Carlini and Wagner designed a L_2 attack [8] that optimizes two criteria at the same time: producing a perturbation that makes the sample adversarial (e.g. incorrectly classified by the model), and minimizing the L_2-norm of the perturbation. By making a change of variable using the $\tanh$ function, we can formulate a continuous optimization problem that constrains the L_2 norm of the distortion to find an adversarial perturbation δ that will make the original sample $x \in [-1,1]^n$ be assigned to class t by a model Z which outputs pre-softmax activations:

$$\begin{aligned} \min_{\delta} \Big[\|\tilde{x} - x\|_2^2 + C \cdot f(\tilde{x}))\Big] \quad \text{where} \quad & f(\tilde{x}) = \max(\max_{i \neq t}\{Z(\tilde{x})_i - Z(\tilde{x})_t\}, -\kappa) \\ \text{and} \quad & \tilde{x} = \frac{1}{2}(\tanh(\operatorname{arctanh}(x) + \delta) + 1) \end{aligned} \tag{1}$$

By increasing the confidence parameter κ, the adversarial sample will be misclassified with higher confidence. To use this attack in the untargeted setting, the definition of f is modified to $f(\tilde{x}) = \max(\max_{i \neq y}\{Z(\tilde{x})_y - Z(\tilde{x})_i\}, -\kappa)$ where y is the original label. The difficulty in this optimization is to find the best value for C; a C too small will lead to a small distortion stuck in a local minimum that may not be adversarial and a C too large will lead to a large norm of the distortion.

3 Decoupled Direction and Norm attack

Algorithm 1 Decoupled Direction and Norm Attack

```
Input: $x$: image to be attacked
Input: $y$: true label (untargeted) or target label (targeted)
Input: $\alpha$: step size
Input: $\gamma$: factor to increase/decrease the norm in each iteration
Output: $\tilde{x}$: adversarial example
Initialize $\delta_0 \leftarrow \mathbf{0}$, $\tilde{x}_0 \leftarrow x$, $\epsilon_0 \leftarrow 1$
If targeted attack: $m \leftarrow -1$ else $m \leftarrow +1$
for $k \leftarrow 1$ to $K$ do
    $g \leftarrow m\nabla_{\tilde{x}_{k-1}} L(\tilde{x}_{k-1}, y, \theta)$
    $g \leftarrow \alpha \frac{g}{\|g\|_2}$                    ▷ Step of size $\alpha$ in the gradient direction
    $\delta_k \leftarrow \delta_{k-1} + g$
    if $\tilde{x}_{k-1}$ is adversarial then          ▷ Reduce or increase the distortion size
        $\epsilon_k \leftarrow (1-\gamma)\epsilon_{k-1}$
    else
        $\epsilon_k \leftarrow (1+\gamma)\epsilon_{k-1}$
    end if
    $\tilde{x}_k \leftarrow x + \epsilon_k \frac{\delta_k}{\|\delta_k\|_2}$        ▷ Project example to an $\epsilon_k$-ball around x
    $\tilde{x}_k \leftarrow$ clip$(\tilde{x}_k, 0, 1)$                ▷ Ensure the image is within bounds
end for
Return $x_k$ that has lowest norm $\|x - x_k\|_2$ and is adversarial
```

As mentioned above, optimizing the classification loss (either cross-entropy or Carlini's formulation) and the L_2-norm at the same time can be tricky, as it requires an almost optimal hyper-parameter to find an adversarial with a small norm (compared to other attacks). Therefore, we propose not imposing a penalty on the L_2-norm in the optimization, but rather constraining it with a projection. Modifying the L_2-norm is, from there, a binary decision: if the sample is not adversarial at step k, the norm is multiplied by a factor $1 + \gamma$, otherwise, it is multiplied by $1 - \gamma$.

The full procedure is described in Algorithm 1. We start from the original sample x, and iteratively refine the noise δ_k. In iteration k, if the current point $\tilde{x}_k = x + \delta_k$ is still not adversarial, we consider a larger norm (ϵ_{k+1}) for the for the next iteration. Otherwise, if the sample is adversarial, we consider a smaller ϵ_{k+1}. In both cases, we take a step $g = \alpha \frac{\nabla_{\tilde{x}_k} L(\tilde{x}_k, y, \theta)}{\left\| \nabla_{\tilde{x}_k} L(\tilde{x}_k, y, \theta) \right\|_2}$ from the point $\tilde{x}_k$, and project it back on an ϵ_{k+1}-ball around x, obtaining $\tilde{x}_{k+1}$. Lastly, we project the sample into the feasible region of the input space $\mathcal{X}$. In the case of images normalized to $[0, 1]$, we simply clip the value of each pixel to be inside this range. Besides this step, we can also consider quantizing the image in each step, to make sure the attack is a valid image.

4 Experimental methodology

As a preliminary evaluation for our attack, we compare it to Carlini and Wagner L_2 attack [8]. We use the same model architectures with identical hyper-parameters as in [10, 8] to train relatively small models on MNIST and CIFAR-10, which obtain 99.44% and 85.51% accuracy on the test sets respectively. We evaluate our attack on the first 1 000 images of the MNIST and CIFAR-10 test sets, in the untargeted setting, as in [8]. We also report the number of gradient computations required for the attacks and the runtime on a NVIDIA GTX 1080 Ti.

For MNIST, we use a higher value of γ for small numbers of iterations (100). This is due to the fact that even for a network trained without defense, the adversarial perturbations on MNIST have a high L_2-norm compared to natural images. In all other cases, the value of γ is fixed at 0.05. We observed an improvement in the results when reducing the value of γ through the iterations (e.g. with cosine annealing) but decided not to include it as it was a data- and model-specific tuning. We also did not try to find the optimal initial value of ϵ for each dataset and number of iterations and kept it at 1. The only hyper-parameter that we found that was important to modify throughout the iterations was the step size α, starting at 1 and reaching 0.01 using a cosine annealing.

As discussed in [8], we also perform quantization to have valid values for the adversarial samples. In our attack, quantization can be included in the search of the adversarial as a part of the projection in the valid range (step 13 of Algorithm 1).

5 Results and discussion

Table 1 reports the results of DDN compared to the C&W L_2 attack, on the 1 000 first images of the MNIST and CIFAR-10 test sets. In particular, we report the success rate of the attack (percentage of samples for which an attack was found), the mean L_2 norm of the adversarial noise (for successful attacks) and the median L_2 norm over all attacks while considering unsuccessful attacks as worst-case adversarial (distance to a uniform gray image [9]). We also report the number of gradient computations and the run-time.

In both datasets we obtain results comparable to the state-of-the-art. We obtain slightly worse L_2 norms on the MNIST dataset, however, our attack is able to get within 4% of the norm found by C&W in 100 iterations compared to the 58,015 iterations required for the C&W L_2 attack. On CIFAR-10, our attack requires 500 iterations to reach 100% success rate with a lower norm than the C&W L_2 attack.

6 Conclusion

In this article, we presented a new attack called DDN (Decoupled Direction and Norm attack), that obtains comparable results with the state-of-the-art for L_2-norm adversarial perturbations in fewer iterations. This attack represents an advance in two directions: it allows for faster evaluation of

	Attack	Budget	% Success	Mean L_2	Median L_2	#Grads	Run-time
MNIST	C&W	9 steps, 1 000 iters	100.0	1.4056	1.4214	7 402	118.3
		9 steps, 10 000 iters	100.0	**1.3961**	1.4121	54 007	856.8
	DDN	100 iters	100.0	1.4563	1.4506	100	1.5
		300 iters	100.00	1.4357	1.4386	300	4.5
		1 000 iters	100.00	1.4240	1.4342	1 000	14.9
CIFAR-10	C&W	9 steps, 1 000 iters	100.0	0.1552	0.1456	3 409	171.3
		9 steps, 10 000 iters	100.0	0.1543	0.1453	36 009	1793.2
	DDN	100 iters	100.0	0.1503	0.1333	100	4.7
		300 iters	100.0	0.1487	0.1322	300	14.2
		1 000 iters	100.0	**0.1480**	0.1317	1 000	47.6

Table 1: Comparison of our DDN attack to the C&W L_2 attack on the first 1 000 images of the MNIST and CIFAR-10 test sets. Run-times are in seconds.

the robustness of differentiable models and it makes a new kind of adversarial training conceivable without the need for a lot of resources. Future work is oriented towards using this attack for adversarial training.

References

[1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv:1312.6199 [cs]*, 2013.

[2] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, December 2018.

[3] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *ICLR*, 2015.

[4] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Dan Boneh, and Patrick McDaniel. Ensemble Adversarial Training: Attacks and Defenses. In *ICLR*, 2018.

[5] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. *ICLR*, 2018.

[6] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. *ICML*, 2018.

[7] Anish Athalye and Nicholas Carlini. On the Robustness of the CVPR 2018 White-Box Adversarial Example Defenses. *arXiv:1804.03286 [cs, stat]*, 2018.

[8] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *SP*, 2017.

[9] Wieland Brendel, Jonas Rauber, Alexey Kurakin, Nicolas Papernot, Behar Veliqi, Marcel Salathé, Sharada P Mohanty, and Matthias Bethge. Adversarial vision challenge. *arXiv:1808.01976*, 2018.

[10] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *SP*, 2016.