Features which are robust to adversarial attacks are also robust to several poisoning attacks

Adrien Chan-Hon-Tong

Symmetric adversarial poisoning against deep learning

Adrien CHAN-HON-TONG
ONERA université Paris Saclay
Palaiseau France
orcid=0000-0002-7333-2765 adrien.chan_hon_tong@onera.fr

***Abstract*—Data poisoning is known as the goal of finding small modifications of training data which make them not suitable anymore for training a targeted model.**

Recently, an efficient symmetric poisoning attack targeting frozen deep features plus support vector machine has been found. However, new experiments presented in this paper shows that this attack is not symmetric anymore on unfrozen/real deep networks.

Then, several extensions of this attack are considered on CIFAR10/CIFAR100 with both VGG and ResNet backbone leading to a symmetric attack. On VGG/CIFAR10 setting, this extended attack makes performances moving by -60%,+5% from native accuracy using perturbations invisible to human eyes. Code is available at github.com/achanhon/AdversarialModel.

***Index Terms*—data poisoning, adversarial examples, deep learning**

I. Introduction

A. Adversarial examples

Deep learning (**DL**) which appears in computer vision with [1] (see [2] for a review) is now a mature technology for many digital application e.g. [3]. But, current DL can be hacked. This could forbid application of DL for critical applications including autonomous driving [4], health care [5], or security (e.g. [6]). The most salient example of this fault is adversarial examples [7]–[11] (which may exist with other machine learning algorithms but which is a real issue for DL). At test time, it is possible to design a specific invisible perturbation such as a targeted network eventually predicts different outputs on original and disturbed input. Computer vision is especially concerned with accuracy of unprotected network dropping close to 0% under state of the art attack [12] but other fields are concerned (e.g. [13] highlights this issue in cyber security context with performance of a malware detector dropping from 87% to 66% on adversarial malwares). Worse, producing adversarial examples does not require to have access to the internal structure of the network [14], [15] and can have physical implementation [16].

Mathematically, producing adversarial example is classically considered as the task of maximizing the cross-entropy (**CE**) of a target network f with weights w on a data x thank to a perturbation δ constrained to be small (typically a L_1 norm bounded by ε):

$$\max_{\delta \ / \ ||\delta||_1 \leq \varepsilon} CE(f, w, x+\delta) \tag{1}$$

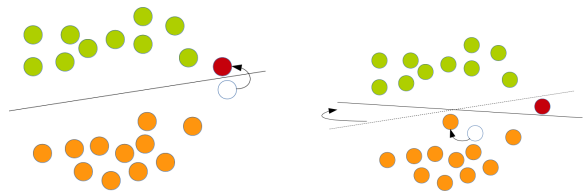

Fig. 1. Illustrations of classical adversarial attack vs poisoning attack: goal of the hacker is to have the red point classified as green and not orange, black line is the targeted classifier.

B. poisoning

A smaller but non negligible issue is poisoning [17], [18]. Data poisoning (which also works [19] on support vector machine **SVM** [20]) is known as the goal of finding small modification of training data (testing data being unchanged) changing the model behaviour on test (for example, changing the testing accuracy). In other words, data poisoning hacks training data (using or not knowledge on testing data and/or model) while adversarial attack hacks testing ones (using or not knowledge on the model): see Fig.1.

Mathematically if the targeted learning pipeline f is trained with stochastic gradient descent [21] (**SGD**) or incremental versions (e.g. [22]), the goal of poisoning is to solve:

$$\begin{array}{ll} \underset{\delta \ / \ ||\delta|| \leq \varepsilon}{\text{minimize}} : & \mathbf{E}_\theta \left[\text{Accuracy}\left(f, w, Test\right)\right] \\ st : & w \sim \text{SGD}_\theta(f, Train + \delta) \end{array} \tag{2}$$

where expectation is required as SGD relies on random variable θ, and, $||.||$ represents constraints on δ the poison. Just for highlighting the hardness of this problem, comparing Eq.1 and Eq.2, one can see that adversarial attack is just optimization of δ through a network while poisoning is optimization through the training of a network.

C. Symmetric adversarial poisoning

Poisoning is a real issue typically when δ is constrained in L_0 pseudo norm [17]: an hacker may modify few samples of a training database without being detected.

Now, poisoning is also a way to improve our understanding of what and how deep networks learn: this paper focuses on L_1-norm **symmetric** poisoning problem (**SAP**)

- constraint on δ is a L_1-norm (taking advantage of adversarial sensibility of the targeted network)
- the *same* attack is required to be able both decrease or increase the accuracy (i.e. minimize or maximize in Eq.(2))

This paper follows [18] which introduces a symmetric adversarial poisoning attack (**SAP**) based on energetic landscape hacking (see github.com/achanhon/AdversarialModel).

Precisely, [18] presents a SAP attacks on classical computer vision benchmarks targeting a frozen DL + SVM pipeline (f is a deep network but only last layer weights are updated during training resulting in a pipeline sensible to small perturbation but with convex training). This attack called *energetic level attack* is based on the *idea* that the more a model w has high/low training cross entropy $CE(f, w, Train)$, the less/high is the probability that w will be returned by SGD when trained on $Train$. In practice, [18] offers to use a proxy $w_{desired}$ (by training on the test $w_{desired} = \mathrm{SGD}(f, Test)$) and to optimize δ to increase/decrease $CE(f, w_{desired}, Train + \delta)$ *in the hope* on decrease/increasing the probability of $w_{desired}$ to be returned.

Thus, [18] transforms Eq.2 in Eq.1 thank to the use of a proxy, eventually leads to produce training adversarial examples. Mathematically, this *energetic level attack* is the combination of the two following equations:

$$\underset{\delta \ / \ ||\delta||_1 \leq \varepsilon}{\text{minimize}} : E_\theta \left[CE(f, w_{desired}, Train + \delta) \right] \tag{3}$$

$$w_{desired} \sim \mathrm{SGD}_\theta(f, Test) \quad or \quad -\mathrm{SGD}_\theta(f, Test) \tag{4}$$

with $w_{desired} \sim \mathrm{SGD}_\theta(f, Test)$ when goal is to minimize accuracy and $-\mathrm{SGD}_\theta(f, Test)$ when goal is to maximize accuracy. Importantly, Eq.3 is almost exactly Eq.1 but the critical difference is the use of specific weights $w_{desired}$: using other weights does not lead to a SAP. Typically, using the weights resulting from a standard training is just adversarial retraining.

D. Contribution

The starting point of this paper is a set of experiments described in section 2 which shows that this attack is **not** symmetric anymore when targeting unfrozen/real deep network. This intriguingly failure is interesting as it is due to the difference between deep network and SVM training.

Then, the main contribution is to offer a modification of the original attack which allows a SAP attack for real deep network (i.e. extending [18] to real deep network). This is an important improvement as using frozen DL + SVM is clearly a deprecated practice for image classification. As a teasing of section 3, attacks offered in this paper make accuracy changes from 86% to 27% (minimizing) or 93% (maximizing) for VGG on CIFAR10 (a classical computer vision model/dataset).

In section 4, experiments are presented to highlight that offered modification is not trivial especially by comparing it with two other related attacks. Then, conclusion is presented in section 5.

proxy used Eq.3	testing accuracy	desired
$\mathrm{SGD}_\theta(f, Test)$	27%	$\ll 87\%$ (-31% in [18])
$\mathrm{SGD}_\theta(f, Train)$	34%	$\approx 87\%$ (0% in [18])
$-w_{imagenet}$	64%	$\approx 87\%$
$w_{imagenet}$	58%	$\approx 87\%$
$-\mathrm{SGD}_\theta(f, Train)$	73%	$\approx 87\%$ (-1% in [18])
$-\mathrm{SGD}_\theta(f, Test)$	77%	$\gg 87\%$ (+7% in [18])
Original accuracy	87%	-

TABLE I
TESTING ACCURACY OF VGG ON CIFAR10 UNDER DIFFERENT POISONING CORRESPONDING TO EQ.3 WITH DIFFERENT PROXY: RESULTING ACCURACY MATCHES EXPECTATION ONLY IN FIRST ROW.

II. ENERGETIC LEVEL ATTACK IS ASYMMETRIC ON REAL DEEP NETWORKS

In this section, experiments show that energetic level attack introduced in [18] is not symmetric anymore on real deep network.

A. Experimental setting

The experimental setting is exactly the same as in [18]: same data, same network, same pretraining, attack amplitude. Precisely, data are CIFAR datasets [23]. All attacks are designed to produce a poisoning with average L_1 pixelwise distance bounded by 3. Networks considered are VGG [24] and ResNet [25] (cut when spatial dimension is less than convolution kernel). In most experiments, weights are initialized from IMAGENET [26] pretraining.

Only difference (which is significant) with [18] is that all layers of deep network are updated during training, instead of just the last one. By learning all layers, native (healthy) performance are much higher. Typically, without poisoning, accuracy of unfrozen pipeline is 87% against only 75% for frozen one on CIFAR10 [18]. This level of performance of 87% is standard [27] for a VGG without batch normalization contrary to the 75% with frozen network. As optimizations are not convex anymore (multiple runs lead to different results) with real deep network, all accuracy measures reported in this paper are averaged over several runs (typically 8 runs).

B. Results

Naive application of [18] algorithm corresponding to Eq.(3-4) targeting an unfrozen deep network decreases performance even when the attack is setup to increase accuracy. Worse, Table.I shows that virtually any proxy leads to an accuracy drop while using unrelated proxy should not impact the resulting accuracy.

Level of accuracy dropping is impressive: from 87% to 27%. But, the main interesting point is that the attack offered in [18] is not symmetric anymore while data, network, pretraining are all the same: the only change is that all the layers of the networks are trained (against only the last one in [18]).

III. ENERGETIC DIFFERENCE IS SYMMETRIC

This section describes the new attack designed for symmetric adversarial poisoning on real deep model.

setting vs accuracy	CIFAR10	CIFAR100
VGG no poison	87%	78%
RESNET no poison	81%	75%
VGG poisoned (min)	28%	34%
RESNET poisoned (min)	43%	33%
VGG poisoned (max)	93%	86%
RESNET poisoned (max)	85%	82%

TABLE II
PERFORMANCE OF VGG/RESNET ON CIFAR10/CIFAR100 WITH AND WITHOUT POISONING EQ.(5-4). BOTH MINIMIZATION AND MAXIMIZATION ARE EFFECTIVE MEANING THAT THIS IS A SYMMETRIC ATTACK.

A. Offered attack

The dynamic of cross entropy curves during training on healthy vs poisoned Eq.(3-4) data are very different: convergence is much more fast on poisoned data. By trying to force equivalent dynamic between both curves, it comes that modifying the difference of $CE(f, w_{desired}, Train + \delta)$ and $CE(f, w_{fair}, Train + \delta)$, with w_{fair} being the weights corresponding to a standard poison-free training leads to a SAP Precisely, it requires to average the cross entropy over several $w_{desired}$ (different sampling over $\mathrm{SGD}_\theta(f, Test)$ or $-\mathrm{SGD}_\theta(f, Test)$ depending on the goal of minimizing/maximizing).

Mathematically, the offered attack correspond to:

$$\underset{\delta \ / \ ||\delta||_1 \leq \varepsilon}{\text{minimize}} : E_\theta \left[\begin{array}{c} CE(f, w_{desired}, Train + \delta) \\ -CE(f, w_{fair}, Train + \delta) \end{array} \right] \quad (5)$$

combined with Eq.4.

Importantly, designing this attack was not trivial despite the close similitude with [18]. A discussion on this point is presented in section 4.

B. Results

The experimental setting is the same than in previous section, results are presented in Table.II.

The results show that the offered attack is effective both for minimization or maximization setting with the two backbones/datasets: for VGG on CIFAR10, performance drops to 28% when minimizing but jumps to 93% when maximizing contrary to [18] attack which leads to 27% when minimizing but only 77% when maximizing (see Table.I).

This is the main contribution of this paper: this is the first known SAP targeting a deep network.

IV. DISCUSSION

This section presents a discussion on these results supported by many complementary experiments. This section may also emphasises the contribution which could otherwise seem limited seen the similarity between Eq.(3+4) and Eq.(5).

A. Comparison with other attacks

Several other attacks have been tested but where found to be asymmetric like energetic level. Before presenting these attacks, let recall the *idea* underlying energetic level attack: minimizing energetic level of $w_{desired}$ may increase the probability for $w_{desired}$ to be returned by SGD, and so the average accuracy to increase/decrease depending on $w_{desired}$.

Of course, this *idea* is false: there is no direct link between energetic level and probability of being returned by SGD. Even if the training is convex, only the lowest points are expected to be returned (so probability of being returned by SGD is a dirac regarding energetic level), and, in not convex optimization it is known that global minimum are rarely reached.

But, one could still have hoped that decreasing energetic level of desired weights may disturb the training toward those weights. It currently works with deep feature + SVM [18] but not with deep network as pointed by Table.I.

Now, there is two other *idea* which could lead to an algorithm: first, that SGD tends to follow energetic valley, and, then that SGD tends to return critical point.

1) Path based attack: If SGD tends to follow energetic valley, then, one could be able to make the optimization to reach $w_{desired}$ by decreasing the energetic level of a complete path in weights space from initial weights to desired ones instead of just the energetic level of the desired ones. This attack can be implemented as:

$$\begin{array}{l} \underset{\delta}{\text{minimize}} : \mathbf{E}_\theta[CE(f, w_{barycentre}, Train + \delta)] \\ w_{desired} \sim \mathrm{SGD}(loss, f, Test, \theta) \\ w_{barycentre} = \alpha w_{desired} + (1 - \alpha) w_{imagenet} \\ \alpha \sim \mathcal{U}(0, 1) \\ ||\delta||_1 \leq \varepsilon \end{array} \quad (6)$$

$\mathcal{U}(0, 1)$ is a uniform sampling on [0,1], thus, equation 6 offers to decreases the line (in weight space) between starting weights and final ones.

2) Gradient based attack: If SGD tends to return critical point, then, one can increase the probability of $w_{desired}$ to be returned by forcing gradient (relatively to weight) to be null at $w_{desired}$ (in addition to force energetic level to be low). This leads to the following implementation (with $\mu \ll 1$):

$$\underset{\delta \ / \ ||\delta||_1 \leq \varepsilon}{\min} : \mathbf{E}_\theta \left[\begin{array}{c} \mu CE(f, w_{desired}, Train + \delta) + \\ ||\nabla_w CE(f, w_{desired}, Train + \delta)||_2^2 \end{array} \right] \quad (7)$$

combined with Eq.4. It could be seen that this attack require 2nd order derivative hopefully implemented in Pytorch (https://pytorch.org/).

3) Results: All those attacks are compared on VGG/CIFAR10 and results are presented in Table.III. Both path and gradient based attacks are not SAP: only energetic level difference is.

It is important here to distinguish how algorithms really work (which is unfortunately out of the scope of this paper) and why they have been designed as it. All energetic level / path based / gradient based attacks are designed around an idea: the idea that there is a link between energetic value and probability of being returned by SGD, the idea that SGD follows energetic valley, and finally, the idea that SGD is expected to return critical point.

poisoning	max accuracy	intelligible
no poisoning	87%	
Energetic level Eq.(3+4)	77%	yes
Path based attack Eq.6	87%	yes
Gradient based attack Eq.7	80%	yes
Diff based attack Eq.5	**93**%	no
GAN based attack Table.IV	92%	no

TABLE III
TESTING ACCURACY WITH VGG/CIFAR10 FOR DIFFERENT POISONING ATTACK SETUP TO INCREASE ACCURACY: ONLY DIFFERENCE BASED ATTACK WORKS HIGHLIGHTING HARDNESS TO DESIGN SAP FOR DEEP NETWORKS.

Yet, even if both these 3 attacks works on deep feature + SVM (i.e. like in [18]), they does not work on deep network as pointed by Table.III.

Inversely, there is no simple explanation about the success of difference based attack. Yet, this is the only one which is symmetric. This highlights the difficulty to design symmetric attack despite the similarity between them.

B. Going deeper into attack failures

Even if the reason why some attacks fails is out of the scope of this paper, one could make hypothesis to explain these failures. Clearly Eq.6 could be a good idea (controlling the behavior of the SGD by creating a valley in the energetic landscape) if there were only one start and one end. But, there are multiples starting weights (random initialisation) and multiples equally good ending weights. This way, it is not clear anymore to understand what is the *valley* that the algorithm tries to create in standard training.

Then, Eq.7 is based on the idea that SGD is expected to return a critical point, but, deep networks are trained with early stopping, thus, returned weights could be not critical point.

Finally, a global possible explanation is that as the energetic landscape is badly modified, interesting points could become unreachable from common initialization, thus, focusing only on the energetic level of the target could create damaging side effect breaking the idea that $CE(f, w, Train)$ is correlated with probability of w to be returned by SGD.

Currently, an experiment is possible to check this last hypothesis that lowering the energetic level of a point (and/or of it surrounding with gradient penalty and/or of the path leading to it) is not sufficient as it could create side effect making then unreachable from normal initialisation. This experiment is to evaluate the accuracy from initial point being closer and closer to the desired end. This way, only the effect of the local energetic landscape modification is considered but not the global ones (which could break the dynamic of SGD).

Typically, training on the normal CIFAR10 from $w_{imagenet}$ leads to w_{fair} with 87% of accuracy. And, training from w_{fair} also leads to 87% of accuracy (weights are exactly equal if w_{fair} is a real critical point, in practice weights are marginally modified but resulting accuracy is not - in average). But, training on poisoned CIFAR10 leads to 93% from w_{fair}, and, only 77% from $w_{imagenet}$. This fits with the idea that SGD does not take advantage of energetic level change around $w_{desired}$ from $w_{imagenet}$, but, that the change are real (as it takes advantage of them from w_{fair}).

```
GANbasedAttack(f,Xtrain,Ytrain,Xtest,Ytest)
  // compute discriminator
  wD = SGD(f, X_train :: X_test, Y_train × {0} :: Y_test × {1} )
  // modify images according to wD
  X' = []
  for x,y in Xtrain,Ytrain:
    gradient = grad[x](CE(f(x,wD),y x {1}))
    x' = x +/- sign(gradient)  depending on the goal
    X'.append(x')
  return X'
```

By modifying `x` such that training images are closer (for D) to testing images, hacker can hope that applying `SGD` on `X',Ytrain` will return weights more adapted to testing set. Indeed, this attack leads to a significant testing accuracy gap on CIFAR10 with VGG: from 87% to 92%.

TABLE IV
GAN BASED SAP ATTACK TARGETING DL.

Unfortunately, this observation makes even harder to understand the energetic level difference mechanism. As w_{fair} and $w_{desired}$ should be close (seeing this last experiment), energetic level difference should put both these point on a slope i.e. building a mountain in the energetic landscape. Yet, despite this mountain, $w_{desired}$ seems still reachable from $w_{imagenet}$ which is not the case with all 3 other differences.

C. About not energetic attack

To add element to the discussion, one can be interested by attack which does not directly hack the energetic landscape. Obviously, such attack has few chance to be intelligible. Yet, previous sections show that intelligible attacks may not be the better ones.

A good candidate is generative adversarial network (**GAN**) based attacks. There is a tremendous literature for GAN see [28]–[30] as examples and [31] as a review. Overall principle of GAN is:

- one network G (generator) produces images
- one network D (discriminator) classifies images between true or generated one
- D is trained with true images and images generated by G (and should predict image source)
- G is trained to minimize D confidence
- G eventually will produces image that D is not able to distinguish from true images.

In context of SAP, a possible implementation is to learn a discriminator between training and testing images, and, to setup the perturbation to be added to the image (δ in previous equations) to minimize/maximize discriminator confidence. Optimizing δ on all training images eventually produces a poisoned dataset. Training on this poisoned dataset may result in a model less/more testing set friendly as poisoned images are expected to be between original images and testing images. Pseudo code is presented in Table.IV.

As a result, GAN based attack leads to 92% of accuracy instead of 86% on CIFAR10 (in maximization setting) and 44% (in minimization one). So, it is also a SAP. Now, this attack seems limited to small dataset: assuming data are i.i.d. in train and test dataset, the discriminator is learning a model

on a random labelling. Thus, on larger dataset, it should not be able to learn (i.e. accuracy should be 50%). One could claim that with very very large dataset, there is no reason why $w_{desired}$ should be different than w_{fair}. This is true (as DL has finite dimension VC [32]), but not with the same scale: overfitting exits even with the largest academic datasets like [26], while learning random labelling should quickly be impossible [33].

Yet, this attack (despite not directly designed to modify accuracy) is a second SAP on small datasets like CIFAR10/100.

V. Conclusion

This paper offers symmetric adversarial attacks targeting deep networks, not just deep features plus SVM. Several attack related to the offered ones are showed asymmetric highlighting that producing symmetric attack is not trivial.

Main results is that the best offered attack makes the accuracy going from 87% to 27% / 93% when minimizing/maximizing (VGG on CIFAR10). Future works should assess these attack behaviors on larger datasets.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014.

[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Conference on Computer Vision and Pattern Recognition*, 2016.

[5] H. Greenspan, B. van Ginneken, and R. M. Summers, "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1153–1159, 2016.

[6] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016, pp. 21–26.

[7] S. M. Moosavi Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[8] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[9] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*. IEEE, 2016, pp. 372–387.

[10] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *technical report arxiv:1312.6199*, 2013.

[11] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 427–436.

[12] C. Finlay, A.-A. Pooladian, and A. Oberman, "The logbarrier adversarial attack: Making effective use of decision boundary information," in *The IEEE International Conference on Computer Vision*, October 2019.

[13] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial examples for malware detection," in *European Symposium on Research in Computer Security*. Springer, 2017, pp. 62–79.

[14] M. M. Cisse, Y. Adi, N. Neverova, and J. Keshet, "Houdini: Fooling deep structured visual and speech recognition models with adversarial examples," in *Advances in Neural Information Processing Systems*, 2017, pp. 6977–6987.

[15] N. Narodytska and S. Kasiviswanathan, "Simple black-box adversarial attacks on deep neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, July 2017.

[16] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *International Conference on Learning Representations (ICLR)*, 2017.

[17] L. Muñoz-González, B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu, and F. Roli, "Towards poisoning of deep learning algorithms with back-gradient optimization," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, 2017.

[18] A. CHAN-HON-TONG, "An algorithm for generating invisible data poisoning using adversarial noise that breaks image classification deep learning," *Machine Learning and Knowledge Extraction*, vol. 1, no. 1, p. 192204, Nov 2018. [Online]. Available: http://dx.doi.org/10.3390/make1010011

[19] S. Liu, J. Zhang, Y. Wang, W. Zhou, Y. Xiang, and O. D. Vel., "A data-driven attack against support vectors of svm," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ser. ASIACCS '18. New York, NY, USA: ACM, 2018, pp. 723–734. [Online]. Available: http://doi.acm.org/10.1145/3196494.3196539

[20] V. N. Vapnik and V. Vapnik, *Statistical learning theory*. Wiley New York, 1998, vol. 1.

[21] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, 2013, pp. 1139–1147.

[22] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Advances in Neural Information Processing Systems*, 2016.

[23] A. Krizhevsky and G. E. Hinton, "Using very deep autoencoders for content-based image retrieval." in *ESANN*, 2011.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[26] J. Deng, W. Dong, R. Socher, L.-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*. IEEE, 2009, pp. 248–255.

[27] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Nov 2015.

[28] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014.

[29] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, 2017, pp. 5767–5777.

[30] S. Nowozin, B. Cseke, and R. Tomioka, "f-gan: Training generative neural samplers using variational divergence minimization," in *Advances in neural information processing systems*, 2016, pp. 271–279.

[31] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, Jan 2018.

[32] N. Harvey, C. Liaw, and A. Mehrabian, "Nearly-tight vc-dimension bounds for piecewise linear neural networks," in *Conference on Learning Theory*, 2017, pp. 1064–1068.

[33] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*, 2016.

About this paper vs HAL versions

As requested by reviewer, I confirm that this paper has never been published anywhere. Previous versions could be found in HAL. However, they were mainly technical reports presenting the raw results, while, (I hope) this paper is easy to be read.