



HAL
open science

Adversarial poisoning and inverse poisoning against deep learning

Adrien Chan-Hon-Tong

► **To cite this version:**

Adrien Chan-Hon-Tong. Adversarial poisoning and inverse poisoning against deep learning. 2019. hal-02139074v1

HAL Id: hal-02139074

<https://hal.science/hal-02139074v1>

Preprint submitted on 24 May 2019 (v1), last revised 3 Nov 2021 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Adversarial poisoning and inverse poisoning against deep learning

Adrien CHAN-HON-TONG

May 24, 2019

Abstract

Efficient attacks for both adversarial poisoning and adversarial inverse poisoning have recently been found on frozen deep feature plus support vector machine. But, new experiments show that such attacks only works for poisoning, but, not inverse poisoning when targeting deep networks.

Investigating this observation, this paper found that adversarial poisoning attacks based on energetic landscape modification outperforms previous ones based on energetic minimum modification. This way, this paper shows that stochastic training is not sufficient defence against inverse poisoning.

1 Introduction

1.1 Adversarial examples

Deep learning (**DL**) which appears in computer vision with [12] (see [14] for a review) is now a mature technology for many web application e.g. [27]. But, current DL can be hacked which is problematic for critical applications including autonomous driving [3], health care [7], or security (e.g. [10]).

The most salient example of this fault is adversarial examples [17, 29, 22, 26, 20]. 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 but [8] highlights this issue in cyber security context. In [8], a deep learning system is designed to detect malwares from large app repository (like android app). This network reaches state of the art detection performance of 87%, but, detection performances drop to only 66% on adversarial malwares. Let us stress that producing adversarial examples does not require to have access to the internal structure of the network [2, 19] and can have physical implementation [13].

From theoretical point of view, adversarial examples could exist for other machine learning algorithms, but, as previous algorithms were both much more robust and much less accurate, adversarial example was not a threat like it is for DL.

1.2 poisoning

A smaller but non negligible issue is poisoning [18, 1]. Contrary to adversarial attacks, poisoning does not become critical with DL: other machine learning algorithm like support vector machine [28] (**SVM**) were known to be sensible to such attack [16]. Basically, to evaluate a learning pipeline f , a judge should train the pipeline f on training data $Train$, and, compute the accuracy of the resulting model on a testing data $Test$. Now, poisoning scenario is that an hacker tries to pervert the evaluation using the fact that he can both read and change training data, and, he can read, but not change, testing data and f . For example, an hacker may want to make an antivirus f ineffective on some known but not owned testing data by modifying some of his malwares/softwares.

Let us assume that the targeted learning pipeline f is trained with stochastic gradient descent [25] (**SGD**) or incremental versions [23]. Then, the goal of the hacker is to solve:

$$\begin{aligned} \text{minimize :} \quad & \mathbf{E}_\theta [\text{Accuracy}(f, w, Test)] \\ \text{st :} \quad & w \sim \text{SGD}(\text{loss}, f, Train + \delta, \theta) \\ & \|\delta\| \leq \varepsilon \end{aligned} \quad (1)$$

where expectation is required as SGD relies on random variable θ , and, $\|\cdot\|$ represents constraint on δ .

A more generic formulation may replace SGD by a generic training routine. This can even allow to remove the expectation if training is deterministic. Also, as it will be a central in this paper, let stress that an other version of this problem is inverse poisoning where the hacker wants to increase the accuracy (i.e. maximize instead of minimizing in eq.1).

Classically, poisoning focus on modifying only few training samples, this assumption is both realistic (no hacker can own all training data), and, in addition, it forces the attack to hardly detectable for the judge. So, in classical attack, $\|\cdot\|$ of eq.1 is a 0-norm (number of non null components). In these cases, eq.1 can be approximated by considering candidates for modification. Typically, if changing labels is possible, candidates may be best classified samples (which suddenly should be oppositely classified) [18]. If labels can not be changed, candidates may be samples which most impact decision boundary (support vectors in SVM framework) [18, 16].

This way, [18] recently proves that DL is sensible to poisoning. In [18] error jumps from 1.3% to 2% and 4% just by manipulating 3% and 6% of the training dataset on MNIST with a classical deep network.

1.3 Adversarial poisoning

Independently, [1] introduces adversarial poisoning (**AP**) and adversarial inverse poisoning (**AIP**): goal is to produce poisoning with only small modification of training data instead of heavy modification of few training samples. Names of these use case highlights that such attacks may take advantage of adversarial example behaviour of the pipeline to be efficient. So, in adversarial poisoning, $\|\cdot\|$ of eq.1 is typically a L_2 -norm, and, f should be sensible to small perturbation (namely, f is a deep network). Typical scenario of AIP may be a deep learning dieselgate.

Typically, [1] presents attacks on classical computer vision benchmarks, for AP and AIP, 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 still convex).

As, optimizing δ through $\text{SGD}(\text{loss}, f, \text{Train} + \delta)$ (see eq.1) is intractable, [1] offers to use a proxy w_{desired} to approximate this problem. The underlying idea is that the more $\text{loss}(f, w_{\text{desired}}, \text{Data})$ is high (relatively to others w), the less the probability that w_{desired} will be returned by SGD when trained on Data , which is the goal of the hacker in AP. So, instead of looking for optimizing δ through SGD, algorithm just proceed in two steps. It computes the proxy w_{desired} for example $w_{\text{desired}} \sim \text{SGD}(\text{loss}, f, \text{Test})$ (this just corresponds to do a training on testing data). Then, it optimize δ to increase $\text{loss}(f, w_{\text{desired}}, \text{Train} + \delta)$ (respectively decrease for AIP) which can be done with adversarial example tools and which eventually leads to produce training adversarial examples:

$$\begin{aligned} \text{maximize : } & \mathbf{E}_{\theta}[\text{loss}(f, w_{\text{desired}}, \text{Train} + \delta)] \\ \text{st : } & w_{\text{desired}} \sim \text{SGD}(\text{loss}, f, \text{Test}, \theta) \\ & \|\delta\| \leq \varepsilon \end{aligned} \quad (2)$$

1.4 Problematic

Some new experiments presented in this paper show that this algorithm eq.2 can be also applied against DL (unfrozen DL, all weights of all layers are updated during training). But, one key point of this paper is to observe that this algorithm is completely ineffective for AIP against DL.

One explanation of this failure is that using adversarial example tools to modify relative energetic level of $\text{loss}(f, w_{\text{desired}}, \text{Train} + \delta)$ could completely changes the energetic landscape when targeting DL. Such phenomenon could not arise in [1] which work on frozen DL + SVM because in this case energetic landscape is convex. But, for DL, it highly modifies SGD path and tends to strongly degrade generalisation. This way, the approximation eq.(2) has a dramatic impact for AP as both energetic shape modification, and, energetic level modification works together (see following experiments accuracy drops from 86% to 27% on CIFAR10). But, it is useless for AIP as energetic shape modification has a stronger impact than energetic level modification.

The contribution of this paper is to extend [1] to DL for both AP and AIP. The effortlessly extension of AP against DL and related experiments are presented in section 2 with the overall data used in all the paper. The hard extension of AIP against DL is presented in section 3, before conclusion of section 4.

2 Adversarial poisoning targeting deep learning

2.1 Experimental setting

To provide results comparable to [1], the experimental setting is kept unchanged. Like in [1], this paper focus on computer vision datasets: precisely, on CIFAR10 and CIFAR100 datasets [11] (first introduce in 2009 in a technical report *learning-features-2009-TR.pdf* available from www.cs.toronto.edu/~kriz/).

Targeted network is first layers of a classical network named VGG [24] up to conv43 (stopping at conv43 is needed due to small size of CIFAR images) without batch normalization. Weight are initialized from a classical model trained on IMAGENET [5] (which can be found for example here: github.com/jcjohnson/pytorch-vgg).

All attacks are designed to produce poisoning with a pixel amplitude bounded by 3: assuming $v, v' \in [0, 255]$ are the value of a pixel before and after the poisoning, then $|v - v'| \leq 4$. This results in a poisoning invisible to human eyes (see [1]).

Main difference with [1] is that all layers of VGG are updated during training, instread of just the last one. This result in a much more powerful pipeline. Typically, without poisoning, accuracy of our pipeline is 87% while it is 75% for [1] on CIFAR10. This level of performance is standard [15] for a VGG without batch normalization. But, mostly, this pipeline is much more realistic as using frozen DL + SVM is clearly a deprecated practice (especially for image classification). Only disadvantage of this new pipeline is that training is not convex, and, so, multiple runs may lead to different results. To mitigate this issue, all accuracy reported in this paper are averaged over several runs (typically 8 runs).

2.2 Breaking testing accuracy

Naive application of algorithm eq.2 from [1] with an unfrozen deep network works. It produces a strong decrease of testing accuracy of the targeted network after training on poisoned data.

Implementation detail important: expectation of eq.2 is approximated by considering 5 different models trained by SGD (in [1], only one model is considered as their problem is convex). This leads to a decrease of accuracy from 87% to 41%.

A better attack consists to sum gradient sign instead of using the sign of summed losses. This attack is described in pseudo code 1 and leads to a dramatic decrease of accuracy from 87% to 27%.

2.3 Discussion

The results from previous subsection is already a contribution: it shows that deep learning is very sensible to adversarial poisoning, typically implemented by equation 2, or, even more by algorithm table1. Yet, it is a very incremental contribution. But, mostly, it should be a false conclusion to state that this drops of accuracy results from the increase of the

```

energeticLevelModification(CNN,Xtrain,Ytrain,Xtest,Ytest)
  // compute proxies
  wDesired = []
  for k from 1 to K:
    wDesired[k] = SGD(CNN,Xtest,Ytest,crossentropy)
  // modify energetic level of wDesired
  X' = []
  for x,y in Xtrain,Ytrain:
    gradient = []
    for k from 1 to K:
      gradient[k] = grad[x](crossentropy(CNN(x,wDesired[k]),y))
    x' = x + sum[k](sign(gradient[k]))
    X'.append(x')
  return X'

```

Table 1: Code of an efficient adversarial poisoning attack against deep learning. By modifying x to increase $\text{crossentropy}(\text{CNN}(x, w_{\text{Desired}}[k]), y)$, hacker can hope that applying SGD on X', Y_{train} will not return good weights like w_{Desired} . Indeed, this attack leads to a dramatic testing accuracy drop on CIFAR10 with VGG network.

energetic level of the proxies. If it was, using unrelated weight as proxies and/or using opposite proxy (like in AIP) should not lead to a decrease in accuracy. Yet, it does. Table 2 shows that virtually any proxy leads to an accuracy decrease.

The results from table 2 highlights the claim presented in introduction. Modifying energetic level of w_{desired} does have side effect, and, in this case, side effect are even stronger than desired ones: even minimizing energetic level still decreases accuracy while it brings an increase in [1]. So, as this side effect does not exist on convex problem, it should be due to the modification of energetic landscape. This is also consistent with the fact that optimization is sensible for DL.

At this point, this paper offers attacks for adversarial poisoning targeting deep learning pipeline. But, this attack despite being efficient is uncontrolled, and, will not work for adversarial inverse poisoning.

3 Adversarial inverse poisoning against deep learning

Seeing the difficulties to produce AIP for DL, this paper offers to benchmark several kind of attacks on AIP. In addition, to highlight potential threat of AIP, this section also provides information about training of a deep network which is a very interesting topic.

| proxy used in algorithm Table1 | resulting testing accuracy |
|---|----------------------------|
| SGD(CNN, Xtest, Ytest, crossentropy) | 27% |
| SGD(CNN, Xtrain, Ytrain, crossentropy) | 34% |
| -Imagenet weights | 64% |
| Imagenet weights | 58% |
| -SGD(CNN, Xtrain, Ytrain, crossentropy) | 73% |
| -SGD(CNN, Xtest, Ytest, crossentropy) | 77% |
| Original accuracy | 87% |

Table 2: Testing accuracy when changing the proxy used in algorithm table 1. Attack presented in table 1 has dramatic impact, but, not for the reason for which it has been designed: if so, only first row should contain an accuracy largely under 87% and last row should be largely above 87%.

3.1 Path based attacks

As seen in previous section in table 2, energetic level modification (table 1) does not work for AIP. Indeed, minimizing energetic level of $w_{desired}$ should increase the probability for $w_{desired}$ to be returned by SGD. So, minimizing energetic level of $w_{desired}$ should increase accuracy. Yet, it decreases it.

An important point is that minimizing energetic level of $w_{desired}$ leads to 77% of accuracy when network is initialized with IMAGENET weights. But it leads to 93% when initialised with $w_{desired}$ and 89% when initialized with $w_{fair} = SGD(CNN, Xtrain, Ytrain, crossentropy)$. Consistently with experiments from section 1, these results show that energetic level is modified as expected, but, that SGD has a problem to handle poisoned images.

This last point invites to modify the algorithm table 1. In [1], as the problem is convex, focusing on $w_{desired}$ is sufficient, but, to produce AIP against DL, it may be important to consider not only $w_{desired}$ but a path in weight space between $w_{imagenet}$ and $w_{desired}$. More formally, this idea can be implemented as:

$$\begin{aligned}
 \text{minimize :} & \quad \mathbf{E}_{\theta}[\text{loss}(f, w, \text{Train} + \delta)] \\
 \text{st :} & \quad w \sim \text{partial}_{\text{S}}\text{GD}(\text{loss}, f, \text{Test}, \theta) \\
 & \quad \|\delta\| \leq \varepsilon
 \end{aligned} \tag{3}$$

or

$$\begin{aligned}
 \text{minimize :} & \quad \mathbf{E}_{\theta}[\text{loss}(f, \alpha w_{desired} + (1 - \alpha)w_{imagenet}, \text{Train} + \delta)] \\
 \text{st :} & \quad w_{desired} \sim \text{partial}_{\text{S}}\text{GD}(\text{loss}, f, \text{Test}, \theta) \\
 & \quad \alpha \sim \mathcal{U}(0, 1) \\
 & \quad \|\delta\| \leq \varepsilon
 \end{aligned} \tag{4}$$

where $\text{partial}_{\text{S}}\text{GD}$ consists to train the network but to stop at different stages. This way, $w \sim \text{partial}_{\text{S}}\text{GD}$ represents a sampling on a path from $w_{imagenet}$ to some $w_{desired}$. Alternatively, $\mathcal{U}(0, 1)$ is a uniform sampling on $[0, 1]$. Thus, eq.4 offers to decrease the line (in weight space) between starting weights and final ones.

From current results, these attacks are not very efficient. Currently, it is not trivial to know SGD does not follow the designed path, or, if modifying the energetic level of a complete path does even more distort the energetic landscape, or, if the problem comes from the difficulties of this type of attacks is to sample a path efficiently. Another option is that there are multiple possible $w_{desired}$. Still, tested implementations do not reach to interesting results.

3.2 Non energetic based attacks

Seeing the difficulties to design energetic level based attacks, one can be interested by other kind of attack.

A good candidate is generative adversarial network (**GAN**) based attacks. There is a tremendous literature for GAN see [6, 9, 21] as examples and [4] 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 (ground truth is image source)
- G is trained to minimize D confidence
- G eventually will produces good images or more precisely, image that D is not able to distinguish from true images.

In context of AIP, a possible implementation is to consider the generator as δ from equations 1-4, and, discriminator as a network which should classify between training and testing images. D is trained to distinguish training and testing images. Then, δ is designed to fool D . Optimizing δ on all training images eventually produces a poisoned dataset. Training on this poisoned dataset may result in a model more close to $w_{desired}$ as poisoned images are expected to be closer than testing images. Pseudo code is presented in table 3.

Indeed, GAN based attack leads to 93% of accuracy instead of 86% on CIFAR10.

This result is a contribution by itself (extension of AIP to DL target). Yet, the main disadvantage of this attack is that image modifications do not target accuracy by design.

3.3 Energetic landscape based attacks

Attacks based on energetic level fails due to the sensibility of DL to energetic landscape. Attacks based on energetic path does not mitigate the previous issue (probably due to difficulty to sample wisely the path in weight space). GAN based attack provides accuracy gap but not directly in link with energetic consideration.

Now, in this subsection, I offer an other kind of attack linked to energy consideration. In current experiments, this attack has reached highest accuracy. The basic idea is to find an other way to distort energetic landscape in order that SGD produces $w_{desired}$ which is neither single


```

GANbasedAttack(CNN,Xtrain,Ytrain,Xtest,Ytest)
// compute discriminator
wD = SGD(CNN, Xtrain ∪ Xtest, Ytrain × {0} ∪ Ytest × {1} ,crossentropy)
// modify images according to wD
X' = []
for x,y in Xtrain,Ytrain:
    gradient = grad[x](crossentropy(CNN(x,wD),y x 1))
    x' = x + sign(gradient)
    X'.append(x')
return X'

```

Table 3: GAN based attack for AIP targeting DL.

By modifying x such that training images are closer (for D) than testing images, hacker can hope that applying SGD on X', Y_{train} will return good weights like $w_{Desired}$. This attack leads to a testing accuracy gap on CIFAR10 with VGG network equivalent to the one presented in [1] (but for a real network instead of a frozen one in [1]). Main disadvantage of this algorithm is that image modifications do not target accuracy by design.

level based or path based. One possibility can be based on gradient: instead of forcing $w_{desired}$ to have low energy, it is possible to force it to be a local minimum (currently it can be a saddle point, but, as energy level is also decreased this issue seems mitigated). This leads to the following approximation (with $\mu \ll 1$):

$$\begin{aligned}
 \text{minimize :} & \quad \mathbf{E}_{\theta}[\mu \text{loss}(f, w_{desired}, Train + \delta) \\
 & \quad + \|\nabla_{w_{desired}} \text{loss}(f, w_{desired}, Train + \delta)\|^2] \\
 \text{st :} & \quad w_{desired} \sim \text{SGD}(\text{loss}, f, Test, \theta) \\
 & \quad \|\delta\| \leq \varepsilon
 \end{aligned} \tag{5}$$

This way, the attack tries to modify the landscape around $w_{desired}$ instead of focusing on this single point or focusing on an hypothetical path from IMAGENET weights to this point.

Inconvenient of this attack is obviously to require second order derivative as the minimization of $\|\nabla_w \text{loss}(f, w, Train + \delta)\|^2$ assume to the capacity to compute $\nabla_{\delta} (\delta \rightarrow \|\nabla_w \text{loss}(f, w, Train + \delta)\|^2)$ while this last function $\nabla_w \text{loss}(f, w, Train + \delta)$ is already derivative based. Yet, this can be done with recent PYTORCH version using `grad` function (equivalent function may exist in TENSORFLOW).

In current experiments, this attack have in average a low accuracy about around 80%. Yet, variance is very high with several attack leading to as much as 98% of accuracy on CIFAR10 (to be compared with 87% for fair training and 92% for frozen deep pipeline).

3.4 Discussion

All experiments are summarize in table 4. Direct extension of [1] is completely inefficient on AIP when targeting DL (algorithm 1 but with minimisation instead of maximization). This failure seems due to the fact

| algorithm | resulting testing accuracy |
|-----------------------------------|----------------------------|
| Energetic level minimization | 77% |
| Energetic level path minimization | 88% |
| Energetic gradient minimization | 80% |
| GAN based attack | 92% |
| Original accuracy | 87% |

Table 4: Testing accuracy for different kind of attack in adversarial inverse poisoning context on CIFAR10

All attacks target an unfrozen VGG. Although the average accuracy of energetic gradient minimization is low, this attack did jump as high as 98% of accuracy in some experiments.

that modifying energetic level of a point may distort the energetic landscape in such way that desired point is even not reachable from SGD. Extending this idea by considering path (eq.3-4) instead of point does not work either. This failure is probably due to an inefficient sampling along the path. GAN based attack are currently the attack with best average performance reaching same accuracy than [1], but, on the harder situation of an unfrozen deep network. Finally, gradient based attack seems promising as some experiments realized under this setting reach very high accuracy. Yet, from current experiments average accuracy is still low.

4 Conclusion

This paper offers attacks targeting deep networks for both adversarial poisoning and adversarial inverse poisoning.

These attacks are evaluated in computer vision context, precisely, on CIFAR10 with a VGG network as target. In adversarial poisoning, offered attack based on energetic level modification makes the accuracy dropping from 87% to 27%. In adversarial inverse poisoning, offered attack on generative adversarial network makes the accuracy going from 87% to 92%. Although these last results should be consolidated, energetic landscape modification offers very promising attack for adversarial inverse poisoning with some experiments having made accuracy jumping as high as 98%.

An other conclusion of this paper is that stochastic training is not a sufficient defense against poisoning. This open the question of defense/attack of deep learning in context of adversarial (inverse) poisoning.

References

- [1] Adrien CHAN-HON-TONG. An algorithm for generating invisible data poisoning using adversarial noise that breaks image classification deep learning. *Machine Learning and Knowledge Extraction*, 1(1):192204, Nov 2018.

- [2] Moustapha M Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. Houdini: Fooling deep structured visual and speech recognition models with adversarial examples. In *Advances in Neural Information Processing Systems*, pages 6977–6987, 2017.
- [3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Conference on Computer Vision and Pattern Recognition*, 2016.
- [4] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, Jan 2018.
- [5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li 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.
- [6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [7] Hayit Greenspan, Bram van Ginneken, and Ronald M Summers. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, 35(5):1153–1159, 2016.
- [8] Kathrin Grosse, Nicolas Papernot, Praveen Manoharan, Michael Backes, and Patrick McDaniel. Adversarial examples for malware detection. In *European Symposium on Research in Computer Security*, pages 62–79. Springer, 2017.
- [9] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.
- [10] Ahmad Javaid, Quamar Niyaz, Weiqing Sun, and Mansoor 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 BIONET-ICS)*, pages 21–26. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016.
- [11] Alex Krizhevsky and Geoffrey E Hinton. Using very deep autoencoders for content-based image retrieval. In *ESANN*, 2011.
- [12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [13] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *International Conference on Learning Representations (ICLR)*, 2017.

- [14] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [15] 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)*, pages 730–734, Nov 2015.
- [16] Shigang Liu, Jun Zhang, Yu Wang, Wanlei Zhou, Yang Xiang, and Olivier De Vel. A data-driven attack against support vectors of svm. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security, ASIACCS '18*, pages 723–734, New York, NY, USA, 2018. ACM.
- [17] Seyed Mohsen Moosavi DeZfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [18] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 27–38. ACM, 2017.
- [19] 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)*, pages 1310–1318, July 2017.
- [20] Anh Nguyen, Jason Yosinski, and Jeff 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*, pages 427–436, 2015.
- [21] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pages 271–279, 2016.
- [22] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 372–387. IEEE, 2016.
- [23] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909, 2016.
- [24] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [25] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

- [26] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *technical report arxiv:1312.6199*, 2013.
- [27] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.
- [28] Vladimir Naumovich Vapnik and Vladimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.
- [29] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.